



# INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

---

## TABLE OF CONTENTS

	Page
<b>Editor's Note</b> .....	1
<i>Frederick C. Harris, Jr., Rui Wu, and Alex Redei</i>	
<b>Cloud Computing and Its Applications: A Comprehensive Survey</b> .....	3
<i>Jalal H. Kiswani, Sergiu M. Dascalu, and Frederick C. Harris, Jr.</i>	
<b>Automatic Detection of Novelty Galaxies in Digital Sky Survey Data</b> .....	25
<i>Venkat Margapuri, Basant Thapa, and Lior Shamir</i>	
<b>Innovation on Digital Platforms: Impacts of Contgrol Portfolios on Novelty</b> .....	34
<i>Allan Hevner and Onkar Malgonde</i>	
<b>Preprocessing Techniques' Effect on Overfitting for VGG16 Fast-RCNN Pistol Detection</b> .....	45
<i>Jiahao Li, Charles Ablan, Rui Wu, Shanyue Guan, and Jason Yao</i>	
<b>Exploiting a Real-time Non-geolocation Data to Classify a Road Type with Different Altitudes for Strengthening Accuracy in Navigation</b> .....	55
<i>Thitivar PatanasakPinyo</i>	
<b>Applications of Virtual Reality Hand Tracking for Self-Defense simulation</b> .....	65
<i>John Apo and Alexander Redi</i>	

\* "International Journal of Computers and Their Applications is Peer Reviewed".

---

# International Journal of Computers and Their Applications

*A publication of the International Society for Computers and Their Applications*

## EDITOR-IN-CHIEF

Dr. Wenying Feng, Professor  
Department of Computer Science  
Department of Mathematics  
Trent University  
Peterborough, Ontario, Canada K9L 0G2  
Email: wfeng@trentu.ca

## ASSOCIATE EDITORS

**Dr. Hisham Al-Mubaid**  
University of Houston  
Clear Lake, USA  
hisham@uhcl.edu

**Dr. Antoine Bossard**  
Advanced Institute of Industrial  
Technology  
Tokyo, Japan  
abossard@aait.ac.jp

**Dr. Mark Burgin**  
University of California,  
Los Angeles, USA  
mburgin@math.ucla.edu

**Dr. Sergiu Dascalu**  
University of Nevada  
Reno, USA  
dascalus@cse.unr.edu

**Dr. Sami Fadali**  
University of Nevada, USA  
fadali@ieee.org

**Dr. Vic Grout**  
Glyndŵr University  
v.grout@glyndwr.ac.uk

**Dr. Yi Maggie Guo**  
University of Michigan,  
Dearborn, USA  
hongpeng@brandeis.edu

**Dr. Wen-Chi Hou**  
Southern Illinois University, USA  
hou@cs.siu.edu

**Dr. Ramesh K. Karne**  
Towson University, USA  
rkarne@towson.edu

**Dr. Bruce M. McMillin**  
Missouri University of Science  
and Technology, USA  
ff@mst.edu

**Dr. Muhanna Muhanna**  
Princess Sumaya University  
for Technology  
Amman, Jordan  
m.muhamna@psut.edu.jo

**Dr. Mehdi O. Owrang**  
The American University, USA  
owrang@american.edu

**Dr. Xing Qiu**  
University of Rochester, USA  
xqiu@bst.rochester.edu

**Dr. Juan C. Quiroz**  
Sunway University, Malaysia  
juanq@sunway.edu.my

**Dr. Abdelmounaam Rezgui**  
New Mexico Tech, USA  
rezgui@cs.nmt.edu

**Dr. James E. Smith**  
West Virginia University, USA  
James.Smith@mail.wvu.edu

**Dr. Shamik Sural**  
Indian Institute of Technology  
Kharagpur, India  
shamik@cse.iitkgp.ernet.in

**Dr. Ramalingam Sridhar**  
The State University of New York  
at Buffalo, USA  
rsridhar@buffalo.edu

**Dr. Junping Sun**  
Nova Southeastern University,  
USA  
jps@nsu.nova.edu

**Dr. Jianwu Wang**  
University of California,  
San Diego, USA  
jianwu@sdsc.edu

**Dr. Yiu-Kwong Wong**  
Hong Kong Polytechnic University,  
Hong Kong  
eeykwong@polyu.edu.hk

**Dr. Rong Zhao**  
The State University of New York  
at Stony Brook, USA  
rong.zhao@stonybrook.edu

ISCA Headquarters.....278 Mankato Ave, #220, Winona, MN 55987.....Phone: (507) 458-4517  
E-mail: isca@ipass.net • URL: <http://www.isca-hq.org>

Copyright © 2021 by the International Society for Computers and Their Applications (ISCA)  
All rights reserved. Reproduction in any form without the written consent of ISCA is prohibited.

## **Guest Editorial: Special Issue from ISCA Fall--2020 SEDE Conference**

This special issue of the International Journal of Computers and their Applications (IJCA) is a collection of six refereed papers selected from SEDE 2020: the 29th International Conference on Software Engineering on Data Engineering, held October 19-20, 2020. This conference was supposed to be in Las Vegas, NV, USA but due to the pandemic was held virtually.

Each paper submitted to the conference was reviewed by at least two members of the international program committee, as well as by additional reviewers, judging for originality, technical contribution, significance and quality of presentation. The proceedings for this conference can be found online at [https://easychair.org/publications/volume/SEDE\\_2020](https://easychair.org/publications/volume/SEDE_2020). After the conference, the six best papers were recommended by the program committee members to be considered for publication in this special issue of IJCA. The authors were invited to submit a revised version of their papers. After extensive revisions and a second round of review, these papers were accepted for publication in this issue of the journal.

The papers in this special issue cover a broad range of research interests in the community of computers and their applications. The topics and main contributions of the papers are briefly summarized below.

JALAL H. KISWANI, SERGIU M. DASCALU, and FREDERICK C. HARRIS, JR. of the University of Nevada, Reno present their paper “Cloud Computing and its Applications: A Comprehensive Survey” where they provide an overview of cloud computing from IaaS to PaaS and SaaS. They present a brief history, talk about deployment and service models. They then spend time talking about benefits as well as challenges. Then they shift to cloud applications and go over design and architecture, DevOps and the development process as well as benefits and challenges. They end with a user study to analyze the adoption of cloud computing and cloud applications across a variety of organizations.

VENKAT MARGAPURI, BASANT THAPA and LIOR SHAMIR of Kansas State University present their paper “Automatic Detection of Novelty Galaxies in Digital Sky Survey Data.” In this work, they discuss the petabytes of data coming from robotic telescopes and the task of identifying novel galaxies for further study. They propose two techniques for novelty detection: the first uses the entropy on pre-defined content descriptors and the second uses scale-invariant feature transform. They compare the performance between these and traditional ML algorithms.

ALAN HEVNER of the University of South Florida and ONKAR MALGONDE of Northern Illinois University present their paper “Innovation on Digital Platforms: Impacts of Control Portfolios on Novelty.” In this work, they seek to address “the right balance between project controls while supporting the creative design and implementation of novel features.” They propose a research model for this balance and then select two case studies where they show how two companies handled this issue in their software development process. They then spend time assessing and analyzing these case studies and presenting four conjectures for further study.

JIAHAO LI, CHARLES ABLAN, RUI WU, SHANYUE GUAN, and JASON YAO of East Carolina University present their work “Preprocessing Techniques’ Effect On Overfitting for VGG16 Fast-RCNN Pistol Detection.” They analyzed three techniques for image preprocessing to see what effects they had on overfitting a convolutional neural network (specifically VGG16 F-RCNN). This would allow them to have better results (particularly with a small data set) while not modifying the CNN architecture.

THITIVATR PATANASAKPINYO of Mahidol University, Salaya, Thailand presents their paper “Exploiting a Real-time Non-geolocation Data to Classify a Road Type with Different Altitudes for Strengthening Accuracy in Navigation.” This work presents the problem of just using geolocation data in cities with elevated roads over ground-level roads. With the problem presented, they propose a solution, implement it, and show that the solution works at a high level of accuracy.

JOHN APO and ALEXANDER REDEI present their paper “Applications of Virtual Reality Hand Tracking for Self-Defense Simulation.” In this work, they present an application called KickVR which is designed to help community members learn self-defense. One of the unique features of this application is the hand tracking that is used instead of controllers. This is through the use of a Leap Motion controller and an Oculus Rift CV1 headset. A Unity application environment was selected and the authors describe their design process and contributions.

As guest editors, we would like to express our deepest appreciation to the authors and the program committee members of the conference these papers were selected from.

We hope you will enjoy this special issue of the IJCA and we look forward to seeing you at a future ISCA conference. More information about ISCA society can be found at <http://www.isca-hq.org>.

Guest Editors:

*Frederick C. Harris, Jr*, University of Nevada, Reno, USA, SEDE 2020 Conference Chair

*Rui Wu*, East Carolina University, Greenville, NC, USA, SEDE 2020 Program Chair

*Alex Redei*, Central Michigan University, Mount Pleasant, MI, USA, SEDE 2020 Program Chair

March 2020

# Cloud Computing and Its Applications: A Comprehensive Survey

Jalal H. Kiswani\*, Sergiu M. Dascalu\*, and Frederick C. Harris, Jr.\*  
University of Nevada, Reno, USA. \*

## Abstract

Cloud computing is one of the most significant trends in the information technology evolution, as it has created new opportunities that were never possible before. It is utilized and adopted by individuals and businesses on all scales, from a cloud-storage service such as Google Drive for normal users, to large scale integrated servers for online social media platforms such as Facebook. In cloud computing, services are offered mainly on three levels: infrastructure, platform, and software. In this article, an extensive and detailed literature review about cloud computing and its applications is presented, including history and evolution. Moreover, to measure the adoption of cloud applications in industry and academia, we conducted a user-study survey that included professionals and academics from various levels. The user-study methodology, details, and results are also presented and discussed.

**Key Words:** Cloud Computing; Cloud Applications; Software as a Service, Cloud Adoption, Survey.

## 1 Introduction

Cloud computing has made tremendous changes and improvements in the information technology industry, where using 1,000 servers on the cloud for one hour is cheaper than using one server for 1,000 hours [5]. In fact, without cloud computing, many startup companies would not even exist [80] or would not achieve their current economies of scale [68]. As shown in Table 1, worldwide spending on public cloud services was almost \$210 billion in 2016, and it is expected to reach \$383 billion by 2020 according to Gartner news [72].

Table 1: Total worldwide spending on cloud services forecast

Year	Spending (Millions of dollars)
2016	\$209,244
2017	\$246,841
2018	\$287,820
2019	\$332,723
2020	\$383,355

Since the beginning of the computing industry in the last century, the offering of computation hardware or software was mainly based on the perpetual on-premise approach.

\*Department of Computer Science and Engineering. Email: jalal@nevada.unr.edu, dascalus@cse.unr.edu, fred.harris@cse.unr.edu

Organizations would buy computers, install them locally on their premises, and use them for their computational needs. Even with the rise of personal computers in the 1980's, the same behavior continued, so that organizations and individuals followed the same approach. During those eras, some people and organizations thought about a different approach: why not offer computation hardware and/or software as utility services, same as electricity or telephone landlines? Thus, users could use these services on a subscription-based approach (e.g., such as monthly or yearly), and pay-as-they-use (i.e., usage-based costing) [88].

John McCarthy was the first to present the idea of offering computing as utility in 1961 [25]. Five years later, "The Challenge of the Computer Utility" book written by Parkhill discussed many aspects of computing as a utility [70]. Although it has been more than 50 years, Parkhill's definitions and discussion are still relevant. In particular, he defined utility computing as permitting a number of remotely located users to utilize a group of facilities of large central computers with the same ease and flexibility of using their on-premise computers. Furthermore, he emphasized the need of developing low-cost wide bandwidth transmission. In fact, he argued that having such facilities may eliminate the need of local storage and, more interestingly, it may allow direct memory-to-memory communication between remote computers, which may enable faster growth of distributed computing.

Time-sharing was one of the approaches that were used at that period, to enable the expensive main-frames to be shared between users [22], as shown in Figure 1. However, since these machines were used for internal private use by the owning organizations, they were only useful for users within the same organization, or for external parties with constrained use for the obvious reasons of security and scalability.

With the evolution of the Internet in the 1990's, things started to change, where centralized public servers have become available, and unified client applications to communicate with these servers over a standard protocol were developed. In particular, web browsers that communicate with backend web-servers over the Hypertext Transfer Protocol (HTTP) was created. This evolution, along with the massive growth in the wide communication bandwidth opened the opportunity for a new model of software delivery: Software as a Service.

Meanwhile, grid computing started to gain some traction to utilize the ability of distributing tasks between commodity computers that are available in different geographical locations, and offer computing power on demand [12, 25]. In fact, grid

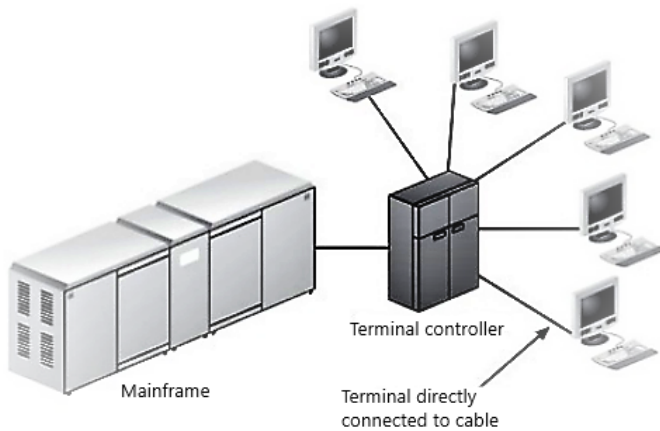


Figure 1: Mainframes time-sharing [69]

computing was later developed into cloud computing with the support of virtualization [10].

A generic definition of Software as a Service (SaaS) is: “applications delivered as a service over the Internet” [5]. Another more specific definition is: delivering software functionalities over the Internet for a large group of customers, based on a multi-tenant single instance software system [38]. However, software is a generic term and can include many categories and definitions [18], therefore, in this article, SaaS is referred to as cloud applications. From the 1990’s era, Yahoo [90] and Hotmail (now Live) [60] are examples of cloud applications, which were targeting both individual consumers (B2C) and small-medium businesses (B2B) [1]. B2C model typically generates revenue by advertisements and subscription for premium services while B2B generates revenue by subscriptions.

On the other hand, new technology companies were founded to provide pure business applications based on the subscription model. Examples of such applications are the Customer Relation Management (CRM) software offered by Salesforce company [78], and Enterprise Resource Planning (ERP) by Oracle NetSuite [67]. With the massive growth and expansion of the Internet, companies started to offer new categories of new online services such as e-commerce, search engines, and social media. As a result, new technology giants started to gain significant traction and market shares, such as Amazon and Google. These companies have built large-scale data-centers to provide their services and products [5].

The availability of such large-scale data-centers, the drop in communication costs, and the decreased cost of electricity and hardware were the most important reasons for the rise of cloud computing and the offering of computation as a utility. Organizations started to offer Infrastructure as Service (IaaS) where computation, network, and storage were provided for customers [59]. In this case, however, customers still had to install the required software needed for running their systems. In addition, organizations started to offer another type

of services where they provide platforms that enable domain-specific programming, such as Google App Engine [35] and Oracle Cloud Platform [66]. This type of service is called Platform as a Service (PaaS) [59].

In the beginning, providing IaaS and PaaS over the Internet was not efficient because infrastructure and platform resources were physically allocated for every customer, which caused many limitations, mainly related to scalability and management. These issues were the motivation for enhancing the so-called virtualization technology [10]. In fact, virtualization has been one of the primary enablers for many technology transformations that occurred in the last decade. In particular, physical resources are treated as a pool of virtual resources that can be shared between users, with the illusion of having dedicated resources for every user. Furthermore, virtualization provides elasticity, where the provisioning of services is automated and more efficient; compared to manual and physical provision, it may reach a near real-time provision.

Data centers that are used to host the three services (IaaS, PaaS, SaaS) based on the utilization of the virtualization technology embody what is called now cloud computing [5]. Figure 2 shows these services. Based on the National Institute of Standards and Technology (NIST), cloud computing is defined as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [59].

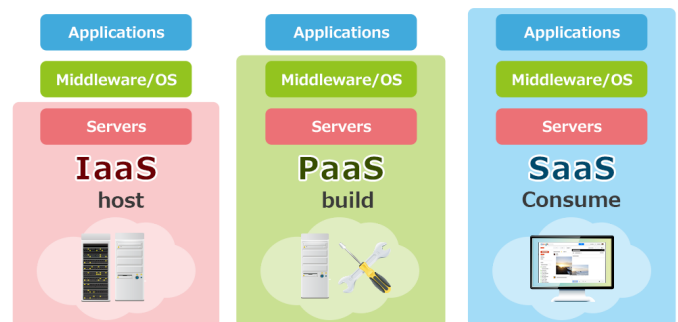


Figure 2: Services provided by cloud computing [15]

Organizations that provide cloud services are called service providers, while customers who use these services are called cloud users. In fact, service providers may also be service users at the same time. For example, an online software service provider (SaaS provider) for a student information system may host their applications on other organization’s cloud infrastructure such as Amazon EC2 (IaaS provider) and be a cloud user (IaaS user) at the same time.

Cloud can be deployed in four categories: (i) private cloud, which is exclusively used and accessed by a specific organization, (ii) community cloud, which can be exclusively accessed and used by a community, such as the large datasets

repository by Amazon that mainly targets data scientists [3], (iii) public cloud, which provides services open for use by the general public, and (iv) hybrid cloud, which consists of two or more deployment models [59]. In 2017, hybrid cloud was the most adopted model by enterprises [74].

As shown in Figure 3, this article presents a detailed survey of cloud computing in general, and cloud applications in particular, aimed at providing researchers and practitioners with comprehensive information on cloud computing history and its evolution.

In addition, we conducted a user study survey that gave us a more clear idea about the actual adoption of cloud computing and its applications by professionals from both industry and academia.

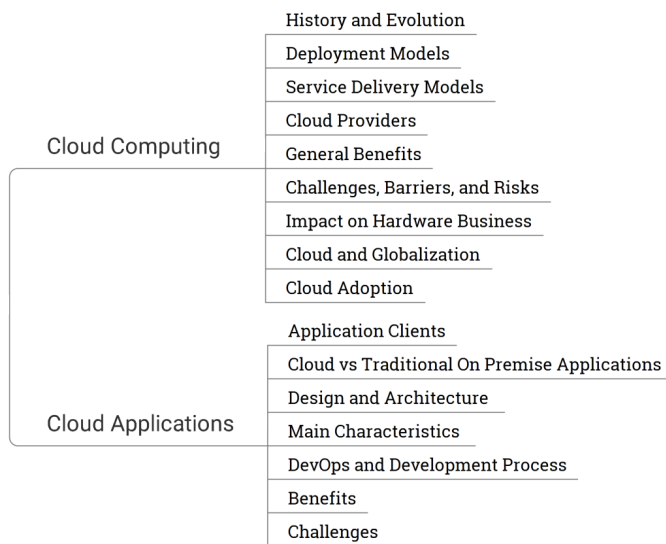


Figure 3: Review Structure of Cloud Computing and Cloud Applications

The rest of this paper is structured as follows: Section 2 discusses cloud computing, and Section 3 presents cloud applications. These sections follow the structure presented in Figure 3. Section 4 describes the user study survey which we conducted, including its methodology, results, and discussion. Finally, Section 5 concludes this article and identifies several directions of future research.

## 2 Cloud Computing

In the information technology industry, there have been many innovations that revolutionized how services are designed and delivered to customers. One of these important innovations is Cloud computing, which has affected almost every industry and discipline. With cloud computing, startup companies do not need to worry about investing in large data centers and hardware anymore. Software developers can start building software applications on top of platforms that can enable rapid-web application development, that can be deployed immediately.

Enterprise organizations do not need to buy expensive software that may become stale with time and need large operational and maintenance costs. In addition, the new categories of Internet hardware clients (i.e. mobile and IoT devices) will be 10 times more than the traditional Internet clients, consisting of over 10 billion devices [62, 8]. Moreover, it will enable saving costs and delegate liabilities [36].

Cloud computing has opened new opportunities, trends, and needs such as mobile interactive applications, parallel application processes, the large growth in data size, the rise of analytics, the need of bringing the data near the applications, real-time decisions, and the Internet of Things.

Even though many definitions introduced in this article are about cloud computing, we find the following definition of cloud computing to be the most comprehensive:

*A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [25].*

Cloud computing has five main characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service [59]. Figure 4 shows the simplified cloud infrastructure.

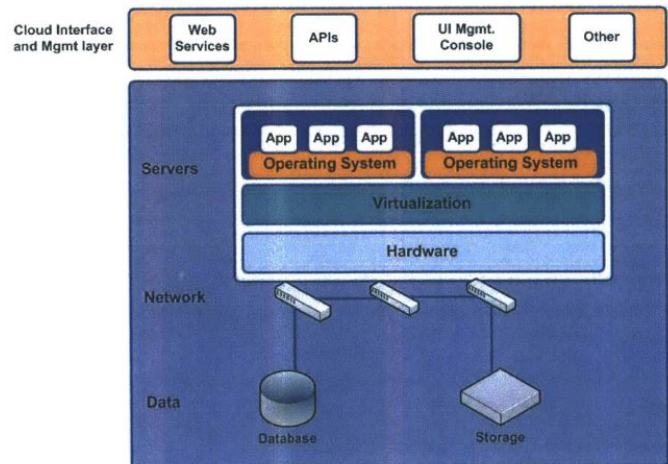


Figure 4: Simplified cloud infrastructure [36]

According to NIST, the main characteristics of cloud computing are: “on-demand self-service, broad network access, resources pooling, rapid elasticity, and measured service” [59].

Even though cloud computing started to get high traction after 2005 [36], the idea was discussed in the middle of the 20th century as Utility Computing. History and evolution are presented in Section 2.1. Furthermore, the section includes the enablers that caused the massive growth of this technology.

Cloud can be deployed on different models, private cloud, community cloud, public cloud, and hybrid cloud. The main

difference between these deployment models is the target audience and public accessibility. More details about cloud deployment models are discussed in Section 2.2. However, for the rest of this article, the focus is on public and community clouds (assuming community is publicly accessible).

With cloud computing, several data centers will be turned into a single pool of computing utilities, which will enable the illusion of infinite resources. Cloud computing can deliver services using three different models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The sum of these services, data centers they operate on, and the software that runs and manages these data centers, is called “The Cloud” [5]. These three service delivery models are discussed in detail in Section 2.3.

As shown in Figure 5, the entities on which providers of cloud services operate are called cloud providers, and the entities in which users consume those services are called cloud users. Dedicated discussion about cloud providers, the advantages of being one, major providers and their classifications are presented in Section 2.4.

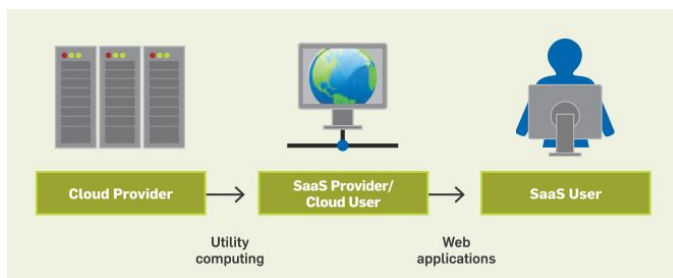


Figure 5: Providers and users of cloud services [4]

Furthermore, and since every technology introduces advantages and benefits on one side, there are disadvantages and challenges on the other side. Benefits of cloud computing are presented in Section 2.5, while challenges, barriers, and risks are discussed in Section 2.6 for both, providers and users.

The impact on the hardware industry is discussed in Section 2.7 and cloud globalization is briefly covered in Section 2.8. Finally, the adoption of cloud computing is presented in Section 2.9.

## 2.1 History and Evolution

Users of utility services such as electricity and telephones employ services based on a model called subscription. In the subscription model, the billing amount is based on service usage. The usage is measured by specific metrics based on the type of service. Usage-based billing is called the pay-as-you-go model. The rationale behind this approach of service delivery is based on resources and services shared among different customers. In particular, this service delivery will reduce the overall cost of resources and services and enable providing services at competitive prices. For example, the sharing of electrical lines connecting a whole neighborhood can reduce the

cost by having only very short distance lines to be dedicated for individuals, which will enable a lower cost of service. On the other hand, having an electricity line for a person who lives in his own large farm may be very expensive, since it may require dedicated resources.

Offering information technology services on the utility-based subscription model is not a new idea. In fact, the first discussion about offering computing as a utility started in 1961 when John McCarthy said that computation might be offered as a utility someday [25]. Later on, in 1966, Parkhill discussed in on how a revolution in distribution and utilization of computer power may enable social changes and opportunities of human development [70]. In particular, he discussed benefits, challenges, and future directions and potential of computing utility. Parkhill said that data transmission, expensive hardware, and limited hardware capabilities were the main barriers. Furthermore, he discussed working to overcome these barriers to open the door for new opportunities of eliminating the need of local storage and allow direct memory-to-memory communication between two remotely located computers, and enable faster growth in distributed computing. In addition, Parkhill considered, this may enable better teaching and information sharing.

Meanwhile, companies like IBM were working on the time-sharing technique [22], as shown in Figure 1. In time-sharing, an expensive computer such as a mainframe was enabled to share computing resources with many users. However, due to the limitations described earlier (e.g., expensive hardware), this was not executed on a large scale and was based on a single organization or community.

The evolution of personal computers in the 1980’s has made computers available for common people. This was one of the motivations to create the next big thing, the Internet. In the 1990’s, a global network of networks was built based on standards protocol such as TCP/IP, SSL, HTTP, FTP, SMTP, and POP. This global network was named the Internet and has affected all aspects of our lives. Meanwhile, the communication technology and hardware were improving as well, which enabled cheaper and faster computing and data transmission.

Since the late 1990s, businesses, academia, governments, and the general public started to offer and use services over the Internet. In the beginning, these services were mainly offering software applications over the Internet. In 2000-2002, Intel failed to offer computing services for organizations and companies, because it required negotiations and long-term contracting, which did not enable scaling their approach [5].

In 2003, Jim Gray, the manager of Microsoft Research Lab in San Francisco, expressed his opinion about the future of distributed computing, where software can be available in different physical locations and communication with other software over standard protocols. He discussed how the free services provided by internet service providers were actually not free and were paid for by advertisements. In addition, he discussed the high cost of ownership, which reached \$1 trillion per year. An exciting point in his report is that in 2002, at Google, only 25 operation staff were managing a two-



petabyte database and distributed it over 10,000 servers using automation tools. In fact, this was the main reason Google was generating profits since they had lower operational costs. This applies also to other giant vendors such as Yahoo and Hotmail. Gray discussed how the future of the Internet would depend on computer-to-computer interaction. However, the advertisement model, which was the main revenue stream at that time, was not sufficient, and companies needed to invent a new business model that leverages the new trend [37].

Later, smart-phones started to be everywhere, which increased the number of Internet users by orders of magnitude. Sensors and smart-device industries were growing, and the Internet of Things (IoT) became a major discipline. For example, a cloud computing approach was proposed to solve the problem of the increasingly growing size of satellite weather data [77]. On the other hand, the economic crisis in 2008 motivated all types of entities to look into cheaper and more efficient ways of doing their businesses. Moreover, space has become a serious issue for corporations, where a quarter of their data centers are out of space [80].

From technical perspectives, the drop-in hardware prices along with the increase in computation power and storage capacity enabled building data centers from commodity computers [25]. In addition, this was helped by the decrease in the cost of utilities, and communication of small to medium data centers [80]. Moreover, the operational cost was reduced due to the availability of automation software tools. Furthermore, wireless device adoption and smartphones removed the dependency of poor infrastructure and enabled even developing countries to be part of the revolution. Moreover, resource pooling using virtualization technology enabled the lower cost and more scalability [5, 36, 41]. Also, adoption of free open-source software, global workforces, and agile software processes played important roles in this evolution [36].

From business and economical perspectives, customer behaviors have changed from buying expensive long-term services and assets to subscription low-price services with low commitments. In particular, they were moving from capital and asset expenses (CapEx) to operational expenses (OpEx). In CapEx, organizations buy computing assets, while in OpEx organizations pay for cloud services on pay-as-you-go option, which can be beneficial from economic perspectives. In addition, the economic crisis in 2008 played a major role where organizations started to look at alternatives to reduce the cost of doing business [4, 41].

## 2.2 Deployment Models

Cloud computing can be deployed in different deployment models, based on the target audience and users. These models are (i) private cloud, (ii) community cloud, (iii) public cloud, (iv) and hybrid cloud.

Private cloud is a data center or set of data centers that are provisioned for and used by only one organization or

corporation. The users of this model are the organization's internal and/or external users. In fact, this model is very similar to a traditional on-premise data-center; however, the utilization of cloud computing technologies (e.g., virtualization), and the possibility to be run and managed by third party organizations are key differences [59]. An example is a virtual data center built by a bank to deploy a core banking system.

A community cloud has a broader audience beyond the same organization. It serves a group of users who share common interests and concerns. For example, satellite image datasets and tools may be deployed on specific data centers managed by NASA or the National Science Foundation in the USA, to enable domain and data scientists to perform big-data analytics on these data sets [59].

In the public cloud, services are open to the general public on the pay-as-you-go model for open use. An example of this is the public hosting providers, which can enable businesses to deploy their Internet applications and support rapid scalability and monitoring [59]. This model is a popular type of cloud deployment model among individuals and small to medium businesses. On the other hand, hybrid cloud is preferred for enterprises [74].

## 2.3 Service Delivery Models

Cloud computing provides services in different service delivery models. Even though there is a unified classification for these services, for now we will go through NIST service models, which are Infrastructure as a Service, Platform as a Service, and Software as a Service [59]. These delivery models are shown in Figure 6.

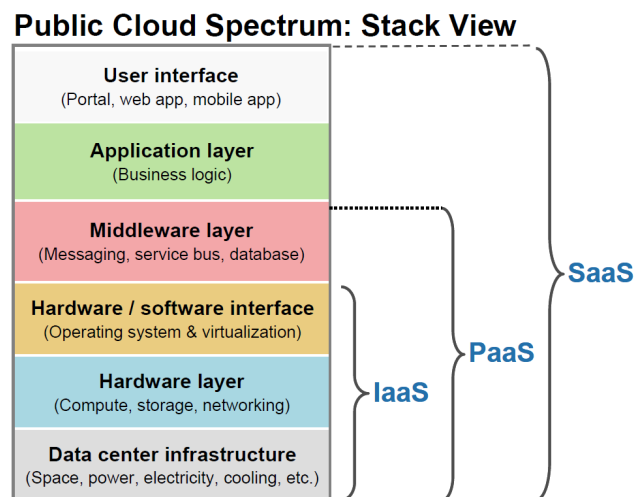


Figure 6: Cloud computing services delivery models [41]

Infrastructure as a Service (IaaS) provides the lowest level of services from cloud providers to cloud user (consumers). IaaS includes services such as computation, storage, networks, and any other fundamental infrastructure. In this model, cloud users will be able to install and deploy arbitrary software including

operating systems and applications. However, users do not have the ability to manage the underlying cloud infrastructure. In particular, it is a collection of hardware and software that enables the special characteristics of cloud computing. IaaS includes virtual machines, servers, storage, load balancer, a network [36], and may include an operating system [74].

On the top of IaaS, Platform as a Service (PaaS) is provided. PaaS is a service that enables its users to build and/or deploy their applications using language, libraries, services and tools provided by PaaS providers on a cloud platform. As we might expect, PaaS users do not have any control over the platform or cloud infrastructure. However, they have control over their application's management [59]. PaaS may include execution runtime, database, web servers, and development tools [36, 74].

Software as a Service (SaaS) is defined as the applications provided over the Internet [5]. Another definition is the capability given by application providers (SaaS providers) to consumers (SaaS users) to use applications running over a cloud infrastructure. In this model, SaaS users cannot control the underlying infrastructure, the platform or the applications; however, a possible configuration for the user's specific instances may be provided [59]. A research group that works for IBM argues that SaaS should be a single instance of a software system, serving a large group of customers over the Internet, built on top of a multi-tenancy platform [38]. Some examples of SaaS are CRM, email, virtual desktops, and computer games [36].

However, many other flavors of services are provided, especially in industry, such as Container as a Service [74], Function as a Service, Testing as a Service, Database as a Service, Security as a Service, and even Metadata as a Service [17]. All these types of services are sometimes grouped as XaaS, where X is anything as a service [5, 76].

## 2.4 Cloud Providers

Low-level service delivery models such as infrastructure as a service require a large investment in properties, hardware, software, and operations. However, many technology giants such as Google, Amazon, and Microsoft already had these infrastructures in their data centers to be able to provide their services to their customers. The availability of these assets gave those companies an edge over other companies, by leveraging their current investments and thus getting a higher market share. In addition, this was one of the main enablers of cloud computing, as further discussed in Section 2.5. The companies that provide low-level services mainly related to hardware are called cloud providers. Figure 7 shows the list of top cloud providers in 2018, by RightScale [75].

Being a cloud provider can bring many benefits. In particular, depending on the economies of scale concept, long-term financial stability and high revenue are possible. In fact, based on the dynamics of innovations theory by Utterback [86], being part of the innovation will give them the chance to dominate the market. On the other hand, new innovations may, directly and

Area	AWS	Azure	Google	IBM
% Adoption	68%	58%	19%	15%
YoY Growth in Adoption	15%	35%	26%	50%
% Adoption in Beginners	47%	49%	18%	14%
% with Footprint >50 VMs	58%	44%	17%	14%
YoY Growth in Footprint > 50 VMs	14%	38%	42%	56%

■ AWS leads  
■ Other vendors lead AWS

*Source: RightScale 2018 State of the Cloud Report*

Figure 7: Top cloud services providers [74]

indirectly, affect other businesses. For example, after offering the cloud infrastructure for a fraction of the price, hardware sales started to drop dramatically. In fact, this has forced hardware manufacturers such as Dell and HP to collaborate with research labs on cloud computing and start investing in this field to catch the wave.

Choosing a data center's location to provide low-level cloud services depends on many factors. For example, choosing locations with low-price property prices, low labor cost, and lower taxes could ensure offering the service at more competitive prices, while leveraging higher profit margins. On the other hand, to deliver the reliability of service, location selection criteria should include the quality of infrastructure and utilities, such as Internet speed and electricity reliability.

However, based on the physical theories, shipping photons over fibers optics is cheaper than shipping electricity [5], and cooling is still a challenge and consumes most of the electricity cost. Many research efforts are focused on this issue, for example, Google has applied for a patent for a water-based data center. Their patent application shows data centers on large ships utilizing the sea motion and water in both electricity generation and cooling [80]. In addition, setting up the data centers in cold areas might be an option; however, infrastructure availability may be a limitation. Moreover, current data center's design for modularity, power, and cooling requires new innovations [39].

## 2.5 General Benefits of Cloud Computing

Cloud computing provides benefits for both service providers and service customers.

Adopting different cloud services can be beneficial for both service providers and service users. For example, SaaS service providers will not need to maintain multiple versions of their software, since most likely they will have only one version running at any point in time, while utilizing the multi-tenant design to achieve higher scalability. On the other hand, SaaS users will be able to get higher-quality service and software that can be accessed from anywhere, with a fraction of the price of traditional software, and with risk delegation to service providers.

A general benefit for service providers is to utilize the economies of scale concept, which means having a large

customer base with recurring revenue. From the IaaS perspective, as discussed in Section 2.4, the providers can gain many benefits such as leverage current investments and defend their franchise. From PaaS perspective, PaaS providers will have higher opportunities of customers lock in, which will reduce the risk of customer turn-over.

A general benefit for service users is the reduction in the overall cost, and delegation of liabilities to service providers [36]. Moving from CapEx to OpEx will get better tax benefits as well as reducing the operational and administration costs. Liability delegation is achieved by transferring risks such as security, availability (e.g. DDoS attack), and legal liabilities to service providers. In fact, the cost of protecting the cloud is less than the cost of attacking it, since the attackers will require huge resources and bandwidth to be able to attack a cloud data center, which results in this not being feasible in most of the cases for the attackers. In addition, it is possible to reduce the risk of procuring a service that does not achieve the business goals by leveraging the test-before-you-buy flexibility, since most of the cloud services are coming with no long-term commitment.

Furthermore, cloud computing can protect against the risk of load mis-estimation. In fact, the elasticity provided by cloud computing enables efficient scalability, which can make the service more reliable since the demand of service may be affected by many uncontrolled events such as news and holidays. Additionally, with cloud computing, there is no need for a large capital investment in hardware or operations, especially for startup companies. Figure 8 shows the difference between automated elasticity in scalability conditions compared to traditional non-automated solutions, and how automated elasticity can almost reach the actual demand without consuming higher resources.

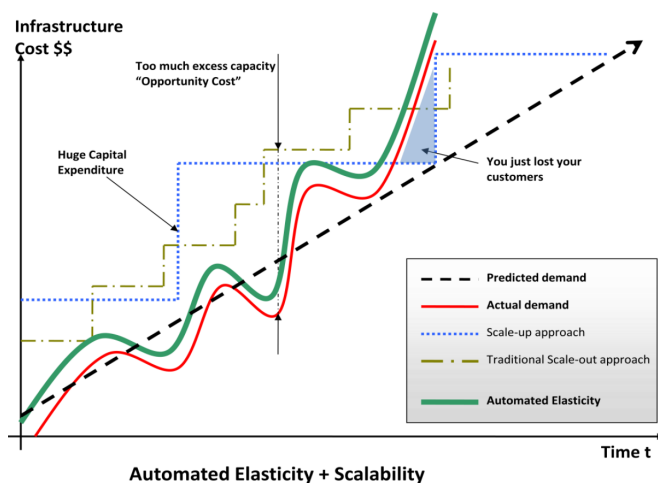


Figure 8: Automated vs traditional applications scalability comparisons [88]

From an operational perspective, cloud computing can benefit service users by enabling them to access the service anytime anywhere. In addition, collaboration and data sharing are

easier. However, the risk of data security exists, as discussed in the challenges covered in Section 2.6. In addition, having more control over servers and installed applications, where 30% of on-premise servers for an enterprise have applications that nobody knows about, and they follow the rule of “Let’s pull the plug and see who calls” [80].

From a business perspective, since the largest cost for enterprises is people, with about 1 employee for 100 servers [39], automated elasticity can also contribute to reducing the operational labor cost.

## 2.6 Challenges, Barriers and Risks

However, the advantages of cloud computing come with challenges, costs, and risks. Major concerns include security [14], data-privacy and data lock-in, bandwidth limitation, and availability. Also, the Internet neutrality [84], political conflicts and governmental regulations may provide a variety of constraints for both service providers and service users.

One of the disadvantages is that it is a single point of failure, where most of the cloud providers use the same infrastructure and the same software, in which any introduced bugs or issues can affect all service users. Obviously, bugs in large-scale distributed systems used by cloud providers are hard to debug and fix. Another disadvantage is performance unpredictability, since shared virtualization resources and dynamic elasticity may affect the performance of other users. Moreover, the provision of new services is still not happening in real-time. Finally, technical up-to-date expertise in cloud technology in general, and in cloud services management and administration in particular, is relatively not easy to find [41, 74].

One of the main challenges of cloud computing is data transfer, which is still a bottleneck. For example, moving large datasets is still cheaper and faster with a traditional mail service such as FedEx. Moreover, another challenge is internal organizational and business policies, which enforce the usage of internal data centers.

In addition, many customers have concerns in regards to security [14], data lock in, data confidentiality [32], data auditability, and control loss [36]. In fact, data can be exposed in multiple scenarios, it can be exposed during upload, while in the cloud, as well as during backup and restore [48].

From a business perspective, pricing uncertainty and cost model complexity, may not fit with some organizations’ policies such as government. Fate sharing between service users and service providers is becoming more common, where a failure in the cloud provider’s service may affect the reputation of service users as well [5]. In addition, since cloud services are based on heavy marketing campaigns, this may not be appropriate for some type of businesses such as some scientific disciplines. Furthermore, lowering consultancy revenue is another concern for some service providers.

Also, some broad license agreements are a critical risk, which may enable service providers to terminate the service any time for any reason, without any customer communication or

feedback [32].

Moreover, politics and regulations can affect the evolution of cloud computing. For example, Internet neutrality [84], political and governmental conflicts may cause a suspension of services. In addition, some compliance regulations may put some special constraints, such as the European Union Data Protection act [29], where all European customers data should be saved in data centers located in the European Union [36].

Another barrier may be the rejection from freeware and privacy advocates such as Richard Stallman, founder of the Free Software Foundation [27] and the creator of the computer operating system GNU [28]. In his interview with the Guardian in 2008 [46] he argued that cloud computing is a trap for enforcing people to buy proprietary licenses, and the main reason for its adoption is the marketing campaigns.

In addition, availability is still a challenge, as service issues such as service suspension by Amazon and Google [24, 25, 80] add certain concerns. Furthermore, disasters may destroy infrastructure and interrupt service for days or perhaps weeks. On the other hand, service suspension may be caused by the unavailability of Internet access for some special places like underground halls or airplanes.

Figure 9 shows the barriers ranking for cloud users based on a study done in 2011 by Morgan Stanley financial services firm [41]. Based on this study, data security is the main concern, followed by cost uncertainty, loss of control, regulatory or compliance requirements, reliability, data portability/ownership, software compatibility, performance, and finally lock-in. Even after 7 years from that report, as presented in Figure 10, a report by RightScale published in 2018 shows many similarities in the challenges with the dramatic rise of *lack of the resources and expertise* challenge [75].

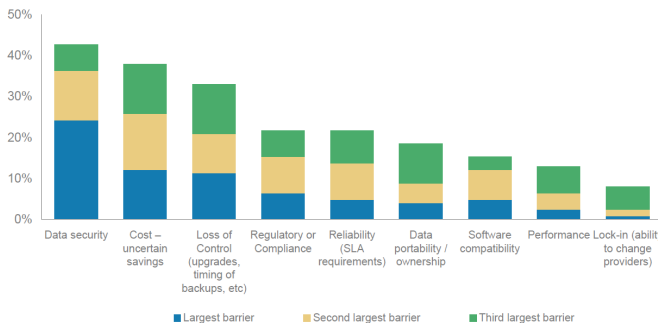
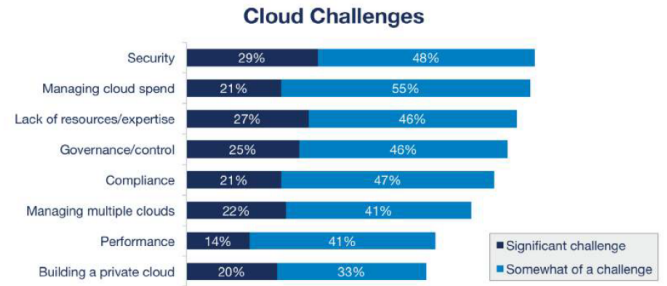


Figure 9: Cloud computing barriers ranking to users [41]

## 2.7 Impact on Hardware Business

Cloud computing has affected many businesses directly and indirectly. The computer hardware industry has directly been affected in many aspects. In particular, the behavior changes of customers toward OpEx instead of CapEx has affected hardware sales [41]. Customers now prefer virtual infrastructure instead of physical hardware for many reasons, including the illusion of



Source: RightScale 2018 State of the Cloud Report

Figure 10: Cloud computing challenges in 2018 [75]

infinite resources available on demand and eliminating upfront investments [5]. On the other hand, cloud providers' orders of hardware have increased and are sold by a scale of containers. For example, about 2,500 servers were delivered by a 13-meter shipping container, then they were installed in a new Microsoft data center in Chicago and the center was up and running in only four days, including electricity and water supply for cooling and network setup [80].

In the hardware labs, power saving features are now one of the leading topics, to reduce the operational cost of data centers by reducing the cost of cooling and utilities. In addition, requirements for higher communication speed in routers and media WANs, and compatibility with virtualization technologies are significant. These factors and others, such as the new behavior of customers to go for short-term payments on pay-as-you-go option, led hardware manufacturers such as Dell, HP and IBM to jump-in and start providing cloud services [53, 5, 41].

## 2.8 Cloud and Globalization

Cloud computing may be the ultimate form of globalization, which may enable new worldwide business opportunities [80] and achieve higher economies of scale. In fact, even in the developing countries, cloud computing has been widely adopted from the first wave of cloud computing between 2006-2010. Countries such as China, India, and Turkey used cloud computing for E-Education, E-Health, and other applications [53].

## 2.9 Cloud Adoption

As discussed in the previous sections, cloud computing has many benefits that can reduce risks and increase profits. Cloud computing utilization is highly recommended over private data centers in many scenarios, in particular when the demand of service varies with time. This allows a benefit from elasticity and ensures efficient utilization of resources. In addition, cloud computing is recommended when the demand of service is not known in advance, such as the growing demand of new services or needs of startup companies. Furthermore, cloud computing

is preferred with batch analysis jobs, which most likely will get results faster. Moreover, running out of space for new data centers may be a strong motivation to adopt the cloud. Finally, resource limitations such as the inability to provide extra utility power for cooling [80] is another motivation to move to cloud computing.

Based on the cloud maturity model of Rightscale [74], cloud users are classified in four categories: (i) cloud watchers, (ii) cloud beginners, (iii) cloud explorers, and (iv) cloud focused. Cloud watchers have not adopted any cloud technologies yet are still in the evaluation phase; however, they are planning a cloud strategy. On the other hand, cloud beginners started to do some experiments for cloud services such as proof of concepts or running these services on a small scale. The third category is cloud explorers, consisting of the users who have adopted cloud computing in serious work for multiple projects, and they have the required expertise to use and manage their cloud services. However, they are still exploring new opportunities to expand their business on the cloud. Finally, there are focused users, where the business is heavily based on cloud computing, and they work on cost and optimization for their cloud infrastructure. Figure 11 shows the cloud adoption percentages as of 2017.

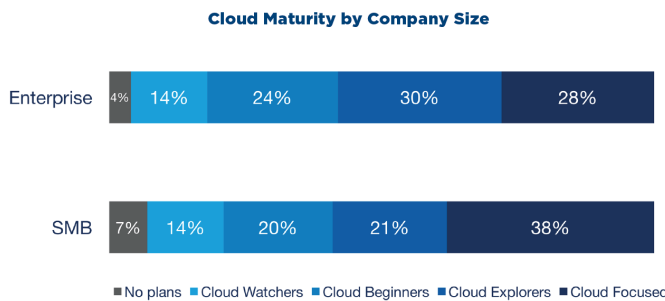
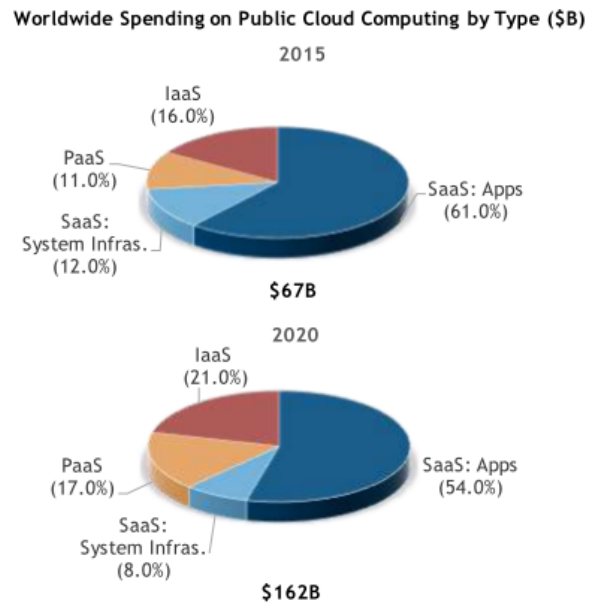


Figure 11: Cloud adoption as of 2017 [74]

### 3 Cloud Applications

Software as a Service (SaaS) is one of the service delivery models of cloud computing. In fact, SaaS is projected to have 54% spending share from cloud computing services by 2020 [30], as shown in Figure 12.

From technical and software engineering perspectives, cloud applications offered by SaaS services are different from traditional software applications. As Bill Gates, the founder of Microsoft, said: “We now live in a world where a subroutine can exist on another computer across the Internet” [80]. Several factors such as scalability, multi-tenant support [38], billing, monitoring, data-locality, and integrability enforced the creation of new terms in technology such as Native Cloud Applications (NCA) and Microservices Architecture (MsA), which enable utilizing the full benefits of cloud computing [7]. In addition, software development processes have improved, for example software operation activities may be assigned to



Source: IDC, 2016

Figure 12: Cloud computing services worldwide spending [30]

software developers in what is called DevOps [9]. From small startup companies to large-scale enterprise software development houses, all are adopting these concepts. For example, Peter Zencke, SAP ERP new version development lead, indicated how exciting it is that any components of the software can be a service provided by other vendors [80].

As shown in Figure 13 software is a very generic concept, which can include operating systems, programming languages, tools, and applications [18]. Hence, these different categories of software can be part of any cloud service delivery model (i.e. IaaS, PaaS, and SaaS). In this paper the term Software as a Service is used interchangeably with Cloud Applications. A discussion about a proposed new taxonomy for service delivery is further presented in Section 5.

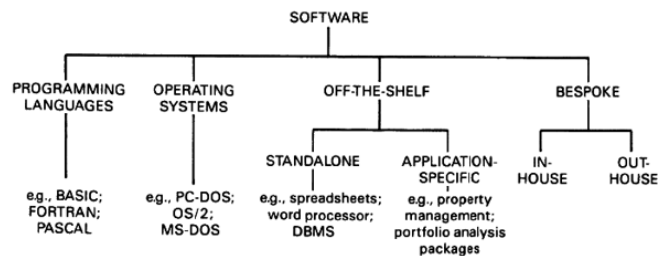


Figure 13: Software taxonomy [18]

Generally, cloud applications are defined as applications delivered over the Internet [5]. Cloud applications are most likely to run over PaaS (discussed in Section 2.3) [83]. A more detailed definition is the functionality provided over the Internet for a large group of clients, based on a multi-tenant

platform, with a single instance of software applications; which also could be provided at the application level [38]. Section 3.1 presents the different types of clients that can access cloud services. Cloud applications are different from traditional on-premise applications in many aspects, and these differences are discussed in Section 3.2. The design and architecture of cloud applications are presented in Section 3.3. New patterns and techniques for developing cloud applications are required and offer advantages over traditional monolithic applications, as discussed in Sections 3.3.3 and 3.3.4. Benefits of cloud applications are discussed in Section 3.6, and related challenges are presented in Section 3.7.

### 3.1 Application Clients

Most cloud applications consist of server-side components (i.e. application backend) and client-side components (e.g. front-end). Since the beginning of cloud evolution, HTML-Internet standard front-end technology- was the main used technology. Internet browsers have been the most widely used HTML clients to communicate with application backends. Desktop widgets became another form of clients [80]. Meanwhile, and with the rise of mobile devices and Internet of Things, new browsers, desktop applications, and sensors applications were added to the client's stack. Consequently, the complexity of building Internet-based applications has increased, and the need for more sophisticated front-end, a new term was introduced: Rich-Internet-Applications (RIA).

RIA is commonly based on utilizing client-side features that depend on JavaScript, HTML5, and CSS3. In particular, the Asynchronous JavaScript And XMLHttpRequest technology (AJAX) is the main enablers for modern interactive web-based applications. Even though it has been the common approach, the user-experience of browser-based applications was not sufficient for scaling cloud applications and could not be used by non-technical people, which forced the way to native applications. In particular, native applications are developed using a programming language to build applications that can utilize native platform components. An example of native applications are applications built using the Java programming language to create Android platform apps. Another example is using Objective-C or Swift to build Apple IOS apps.

Another form of application clients is desktop widgets that communicate with back-end cloud services, such as weather or stock widgets [80].

The new trends of having different front-end clients require new architecture, design patterns, and tactics [40]. Design and Architecture of cloud applications are discussed in Section 3.3.

### 3.2 Cloud vs Traditional On-premise Applications

On-premise traditional applications are the software application instances that are designed to be installed on a client's environment (e.g. local data centers or computers) on the client's premises. The local installation of these applications includes all the application dependent artifacts and

software systems, such as web servers, application servers, and databases. In on-premise applications, dedicated support for clients is provided, with different versions installed in every client's environment. In addition, there is no resource or access sharing with other clients. However, some form of integration with other systems, and external access to the client's services may be provided from that on-premise deployment.

The common licensing model for traditional applications is perpetual-licensing, where clients can use the software without any time limitations, and cost can be a one-time implementation [13]. In this model, the cost can be accurately estimated from the beginning. In addition, in this approach, clients have almost full control over the applications and its data. On the other hand, cloud applications can be licensed on pay as you go or rental licensing models [56, 65, 89].

There are many disadvantages and challenges for on-premise applications. Firstly, clients pay a relatively large amount of money for licensing compared to cloud applications. Secondly, on-premise applications require special upfront consulting and implementation costs [41]. In addition, long implementation time is one of the main risks. In particular, hardware procurement and installation, software environment configurations and setup, application deployment, and on-site implementation are all causing implementation delays. Moreover, most of the time support and upgrades are not included in the initial cost of the system.

Although cloud applications sound tempting, they are not fit for all types of applications [56]. For example, the traditional approach is more appropriate for real-time stock trading which requires microsecond precision [5], since performance is not guaranteed like on-premise deployment because of the sharing nature of cloud applications [41].

Adding to that, the on-premise approach can make perfect sense in organizations working in sensitive domains, such as governmental or financial organizations. In fact, some domain compliance regulations and procedures require only the internal existence of their applications. For example, central banks in some countries enforce the core-banking to be locally installed and managed to ensure availability and data privacy.

On-premise software is normally built in a monolithic fashion. However, cloud applications are designed and built based on self-independent services [80], in what is called Microservice architecture [7]. A comparison between Monolithic and Microservices architectures is presented in Section 3.3.3.

### 3.3 Design and Architecture of Cloud Applications

In all categories of software engineering processes (i.e. waterfall, agile or component-reuse), the design is a significant phase [81]. Architecture is the core part of the design. Software architecture is an abstract, technology-neutral, representation of software systems elements, their relations and how they interact with each other. Moreover, architecture is important to deliver the quality attributes and the non-functional requirements of

software systems [55]. Furthermore, the architecture can be used as an input for development, documenting, and evaluating software [33].

Over the years, various software-architectural tools and techniques have been developed and evolved to enable a more systematic approach of designing software applications, such as architectural styles, patterns, and tactics [40]. These styles include but not limited to the monolithic, Service Oriented Architecture (SAO), and the Microservices-based approach [50].

This section discusses the evolution of cloud applications, from Monolithic applications to SOA, Microservices Architecture, and Cloud Native applications.

**3.3.1 Monolithic Applications Architecture.** Since the beginning of computer software development disciplines, building applications was mainly done using the monolithic approach. In particular, in the monolithic approach all the components of a software application are built as a single unit that should be compiled and deployed as a single instance on the edit-compile-link concept [85], and most likely with the same programming language or technology. Even though this type of software architecture is easier for software developers to understand, develop, deploy and operate, it has many disadvantages, including: full system compilation is required for any change, having all the teamwork on same technology or programming language, and harder horizontal scalability because of the application's heaviness. Another major issue with the monolithic architecture is that the system is a single point of failure, where a single error in the application can take the whole system down.

**3.3.2 Service-Oriented Architecture.** Service-Oriented Architecture (SOA) is an approach used to overcome some of the monolithic application's limitations [85]. In particular, SOA is about decomposing applications into smaller-unit (services) that integrate and are composed at runtime with each other using a standard protocol. Various XML based web-service protocols are used as standard protocols, including SOAP, WSDL, and UDDI [54]. Even though it was an elegant approach based on standard technologies, it didn't get high-traction from small-medium organizations because of its complexity, protocol overhead, and heaviness of the final full system. In addition, having remotely located services was not practical since the units of most SOA applications were communicating with the same data stores (e.g. databases), which caused many performance and reliability issues.

With all the limitations and issues discussed, new factors led to new requirements being needed. These factors include: (i) smart-phones and IoT require new lightweight yet interactive front-end technologies, (ii) entrepreneurship-wave and startup companies require faster time-to-market and lower development cost, (iii) cloud computing requires economies of scale models and scalability with minimal hardware and infrastructure cost. All these factors caused the innovation of many new technologies that have disrupted the software industry. Those new technologies have also created another issue of lack of

human-resources.

On the other hand, there is a conflicted misunderstanding about the relationship between SOA and SaaS. In fact, SOA is a software construction model, while SaaS is a software delivery model [54, 83].

**3.3.3 Microservices Architecture.** To overcome all the constraints, limitations, and disadvantages of the monolithic approach and SOA, and to achieve the requirements enforced by the discussed new trends, Microservices architecture was innovated. As shown in Figure 14 the microservices architecture is a modern way of building cloud-based software applications, in which software applications are decomposed into small light components (services) that communicate with each other over light protocols and light messages exchange. The most commonly used protocol is Representational State Transfer (REST), which is a light-weight text-based solution over the HTTP protocol [23]. JavaScript Object Notation (JSON) is the common message format used by inter-services and service to front communications [16]. In Microservices, every service may be developed using different technology, must access its own data-store, and may not access any other service's data-stores directly, only over the exchange protocol. In fact, directly accessing other service's data-store directly will increase coupling and reduce the service's portability. Moreover, these services are also independently deployable [26]. Also, horizontal scalability of Microservices is light and more efficient than other architectures. In fact, scalability is performed on the service level, where the service which has more load will be replicated on another instance, and there is no need to replicate the whole system. Application containers such as docker are the main enablers for this feature.

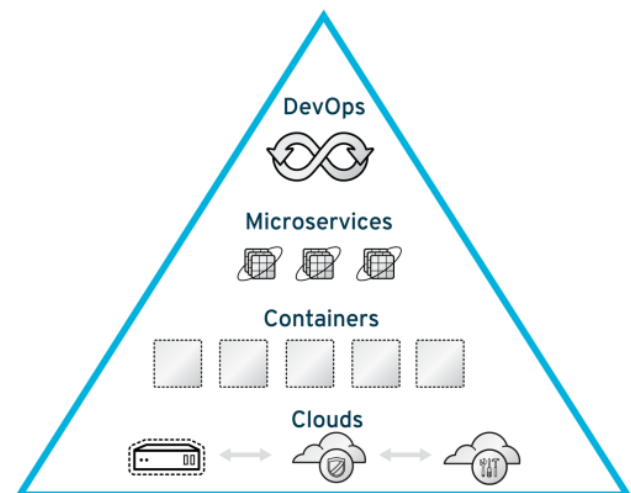


Figure 14: A pyramid of modern cloud native applications [20]

Furthermore, the microservices-based architecture can also come with other “dividends,” such as enabling innovations for developers, since they have full control over the design of their microservice, and hence they can easily replace components and

enable freedom of testing [49].

Even though the microservices style is similar to SOA in many aspects, such as decomposing software into smaller-deployable parts and communicating over a standard protocol [88], the lightness of communication based on REST, messages based on JSON, and separate data stores, might be the main differences.

Even though the microservices architecture solves many issues and problems, and creates potential for new opportunities, it introduces new complexity issues. In particular, special expertise is required to design and architect software solutions based on the microservices architecture. In addition, there is a complexity of integrating the services, testing, and deploying them. Moreover, monitoring and supporting services at runtime by the operations and support team is harder than supporting single processes applications such as the monolithic-based applications.

To reduce some of the risks of microservices architecture, intensive automation is required. In particular, automation can be achieved by applying automation software infrastructures such as Continuous Integration (CI), Continuous Delivery (CD), Test-Driven-Development (TDD), standard projects structure, and other [51].

**3.3.4 Cloud-Native Applications.** The complexity introduced by the microservices architecture, discussed in Section 3.3.3, has led to a new term: Cloud Native Applications (CNA). CNA are portable applications that exploit the full benefits of cloud computing without being dependent on a specific cloud provider or infrastructure [73]. Features such as services scalability, registry, binding, orchestration, and monitoring are supported out of the box. However, a platform is required to act both as middleware and application server for those services.

In addition, as shown in Figure 15, cloud-native applications integrate well with CD, Microservices, DevOps, and Containers.

### 3.4 Main Characteristics of Cloud Applications

Cloud applications have special characteristics that make them different from traditional applications in many aspects [83]. These characteristics are:

1. **Scalability up & down (Elasticity):** In traditional applications, the scalability requirement includes scaling-up, so that a system should be able to handle a larger number of users if required, without modifying the software's code. This was normally achieved by vertical scalability [79]. In particular, vertical scalability is achieved by increasing systems resources, such as memory, storage or computing power [87]. On the other hand, horizontal scalability is widely used by enterprises, by adding extra nodes to the application cluster [25], but it is not common in small-medium organizations and businesses since it is relatively expensive, and not easy to configure and manage.

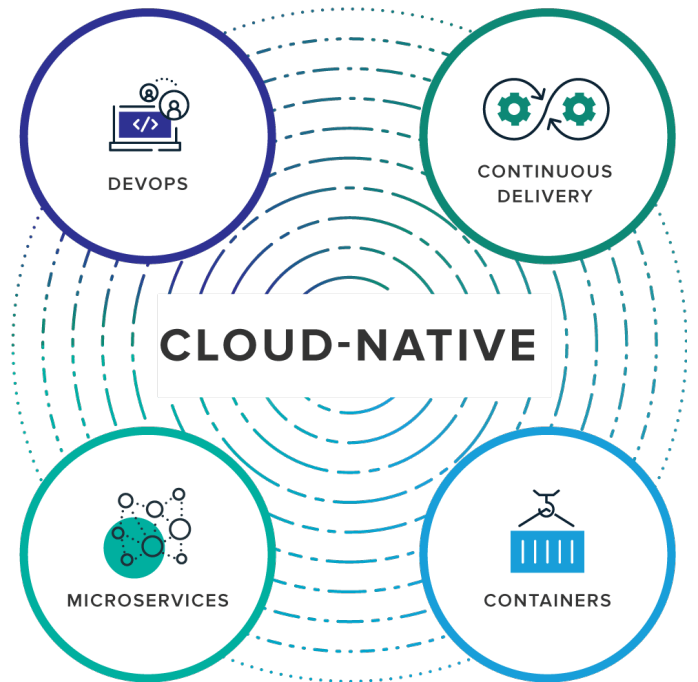


Figure 15: Cloud native applications external environment [73]

Even though scaling-up is important in cloud applications to utilize large data volumes and a vast list of services [33], scaling down is also significant, because it will minimize resource utilization, which reduces the cost for service users [5, 88]. In addition, horizontal scalability is lighter and more cost-effective than vertical-scalability. In fact, application containers such as Docker [19] are the main enablers for lighter and more cost efficient horizontal scalability. These new trends (horizontal scalability and application containerization) led to the innovation of a new architecture, microservices architecture, which was discussed in Section 3.3.3.

Designing an application for exact scalability needs is still a challenge, where having under-utilization, even with a small percentage, increases the cost more than actually needed, and over-utilization makes the services slower, and cause service users to look for alternatives [21]. Furthermore, in-advance testing and benchmarking of the cloud-application scalability can reduce the risk of downtime or load mis-estimation [31, 82].

2. **Support for different front-end technologies:** Currently, trends such as IoT and mobile devices have created the need for supporting widely different types of application clients (e.g. smart-phones, smart-cars, smart-televitions). A special design and architecture should be taken into consideration to support different types of front-ends without the need of modifying the back-end.
3. **Usage of Metrics:** Since most cloud applications are based on subscriptions and pay-as-you-go models, usage metrics should be taken into consideration from



the beginning since they will be the basis for financial billing [5, 80].

4. **Monitoring:** Application monitoring is required to directly ensure that expected quality attributes are being met at runtime, especially in non-normal scalability conditions, such as holidays for e-commerce platforms, or breaking stock market news for real-time trading applications. It may include frequent health checks, heartbeats, and resources visualizations.
5. **Offline support:** Even though Internet services have become more accessible and reliable over the years, customers still have access difficulties to the Internet in many locations and places (e.g. airplanes, underground floors, trains). In addition, with the rise of IoT, scientific devices and sensors may be deployed in some remote locations (e.g. deserts, mountains, oceans), which also may not have an available or reliable Internet connection. Consequently, an offline support feature is important. Having this feature gives service users the ability to use the service while disconnected from the Internet, which can be synchronized once re-connected later with the server. The offline support feature is critical for many applications, such as word editing tools, project management applications, and IoT devices.
6. **Configurability:** In multi-tenant cloud applications, clients should have the illusion of separate application instances, while service providers may maintain single instance to be able to maintain only one version and to achieve economies of scale. Designing the applications to be configurable and parameterized at runtime is important[2]. In fact, having the quality attribute of configurability can reduce support cost, and give more customization and preference features for clients, which can increase customer traction and reduce their turn-over [54]. Furthermore, variability modeling from Software Product Lines (SPL) [47] can also be implemented to achieve the configurability quality attribute [61][63].
7. **Data locality:** The decision on whether to pull or keep data on the cloud requires special attention and balance between performance, data transfer cost, and usability. In particular, data locality is important to improve performance. For example, keeping data on a server may be efficient for server-side processing (e.g. search, filters), however, it might be more efficient to pull data to client-side for visualization applications. Nevertheless, in general, data-locality can achieve better usability and processing performance [37].
8. **Quality of Service:** Finding a way for separating the quality of service for multi-tenant services is important to ensure a reliable service and the containment of the shared-fate issue discussed in Section 2.6.

The dynamic nature of cloud computing and the difference between physical environments and virtualized cloud

environments plays an important role in distinguishing between the architectures of traditional and cloud applications [88].

### 3.5 DevOps and Cloud Applications Development Process

Developing and managing cloud applications has caused a serious issue of mis-coordination between development and operation. In fact, the microservices architecture is the main reason for the increase in this issue, as discussed in Section 3.3.3. To overcome this issue and potential conflicts, a new term was created: DevOps. The main concept behind DevOps is the idea that “you built it, you run it”, where application/service developers are also responsible for supporting and maintaining their applications/services while in production [45] and reducing the friction that appears while in deployment and operation phases [6]. Another approach of DevOps is that the development and operation teams work closely with each other to reduce the gap and taking ownership of the project overall success [42]. In addition, this decreases the time between changing a system and reflecting that change into the live environment [11]. As of 2018, 84% of enterprises are adopting this approach. In fact, 30% of these companies implement this approach on a company-wide policy [74].

Adopting the DevOps approach and culture requires a lot of tooling and software infrastructure to be implemented, such as version control, continuous integration, continuous delivery, and artifacts repository [52].

In reference to the high diversity of roles involved in cloud application development (e.g. security, networks, business), DevOp has four main perspectives: (i) culture of collaboration where all team members from the different project life cycle stages have the required knowledge about the project, (ii) automation, continuous delivery, and deployment pipelines, (iii) high-level and accessible measurement and metrics, (iv) and sharing of knowledge, development, tools, techniques, and other aspects that can enable the required understanding for the system [43, 44]. Moreover, the knowledge, skills, and ability used in developing modern web-based applications were discussed in what is called “grounded theory” [9]. Figure 16 shows how DevOps changed the traditional structure of software development teams.

Moreover, the cloud has changed the role of the System Administrator to a Virtual System Administrator, where there is no need for any cabling/wiring, or server installation required, or any other manual activity, it is all now done through a web console that enables the network and the system to be administrated and managed virtually. This led to software developers and systems administrators being more collaborative and having more interaction [88].

Since DevOps is considered as a culmination of what the agile method started [64], with both encouraging running software over writing documentation [81], the main disadvantages of applying this approach is the risk produced when the developer leaves and not enough documentation is available.

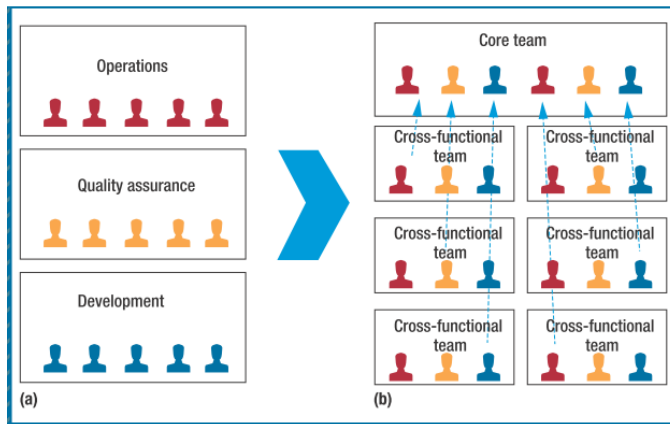


Figure 16: An example of DevOps team structure [11]

### 3.6 Benefits of Cloud Applications

The general benefits of cloud computing were discussed in Section 2.5, which also apply to cloud applications. In addition, in this section benefits of cloud applications are presented from the perspectives of both service providers and service users.

Common advantages are the almost zero upfront investment, just-in-time infrastructure, and reduced time to market [88].

For service providers, running single versions will simplify maintenance, and lower customer support consequently reduces the cost of operations; also, it can reduce the research cost. Furthermore, this will give the ability for small non-risky updates. From an operational perspective, service installation and deployment are easier, especially when utilizing the appropriate software infrastructure [51]. On the other hand, from an economic perspective, since organizations are not willing to pay a large amount of money for software anymore [80], providing applications on the cloud will enable providers to take advantage of this change in customer behavior to make more traction and profit. Moreover, software piracy is impossible, which is another major advantage for service providers [65].

The utilization of cloud applications can bring many advantages to service users. Firstly, low cost may be the most important factor. Secondly, data security may be better than on-premise applications, especially in small businesses, where most likely there is no dedicated support team to operate and support these applications. Furthermore, in general, cloud applications have better quality than on-premise applications and you always have access to the latest stable version of the system [13]. In fact, SaaS providers should invest in building higher quality software to ensure the increase of customers retention [34, 58].

### 3.7 Challenges of Cloud Applications

Cloud applications have many advantages; however, it also introduces many challenges and issues. Debugging a cloud application is not as easy as traditional application debugging. In addition, the support of multi-tenant, and adopting cloud

native applications' properties discussed in Section 3.3.4 introduced an extra complexity for the application development, deployment and management [38].

Furthermore, even though building cloud applications based on economies of scale model sounds tempting, marketing cost is the main challenge for customer acquisition. In 2012, even with 90K customers, and a revenue of \$2.3 billion per quarter, the profit margin for Salesforce was negative because of the high cost of sales and marketing to attract and keep customers [36].

Furthermore, the competition between service providers makes the customers more selective and the decision to switch to another service provider is easier than ever. Increasing the service cancellation cost of customers may be a solution, however, customers will not continue if the service is poor, or may sacrifice that extra cost if they found better quality elsewhere, so working on the application quality is significant to reduce customer turnover [58]. In fact, service providers need 12 months subscriptions on average to cover the expenses of a single customer [34].

Moreover, data integration and interoperability are challenging and include many concerns. These encompass difficulty in large data transmission, from both security and bandwidth perspectives; data integrity and support of transaction across the cloud; expensive data change detection; controlling data quality; and determining the original source of data [57].

## 4 Cloud Applications Adoption User Study

This section includes the details of the user study conducted to analyze the adoption of cloud computing and cloud applications in organizations from different levels.

### 4.1 Methodology

The user study is based on a survey conducted between the period of 24 to 31 December, 2018. It included 36 software technology practitioners and academics with various seniority levels, positions, and education. In addition, the participants were from different regions all over the world.

The user study survey was organized into three sections: Introductory Information, Participant Information, and Cloud Applications.

- **Introductory Information:** In the survey, this section included an introduction about the user study, so that participants could understand the goals. Then, it followed by a consent required by the Institutional Review Boards (IRB) process of the University of Nevada, Reno (UNR). After that, it included the email address of the participant, then (optionally) his/her name. Finally, we asked the participants whether we could contact them for future feedback or questions if needed.
- **Participant Information:** The second section, aimed to get general information about the participants themselves to ensure that we have a representative group. The

information questions included the participant's overall experience in the software development field, highest academic degree, current job title, years of experience, and their day-to-day activities working on software development projects.

- **Cloud Applications:** This section included questions about cloud applications significance, cost, and limitations. Specifically, the first question was whether choosing cloud applications is more beneficial than the traditional approaches for green-field projects. The second question was to verify if there is a shortage in experienced software engineers who can develop high-quality cloud-based applications. The third, fourth, and fifth questions were about the cost of software development, maintenance, and operation of cloud versus traditional applications respectively. The last question in this section was about the risk of uplifting traditional applications (i.e., migrating applications to the cloud).

## 4.2 Participants

As shown in Figure 17, the survey included a wide range of participants with different professional levels in cloud computing. Around 33% were professionals, 36% intermediate, and 27% experts in the field of study.

What is your experience level in the software development field?

36 responses

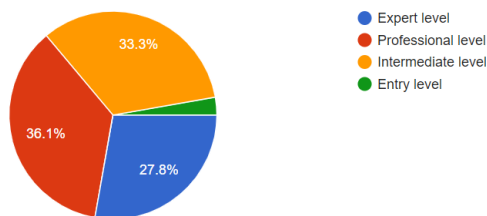


Figure 17: Experience levels of the user study participants

In addition, the participants came from different academic backgrounds, as shown in Figure 18. Around 55% have a bachelor degree, 30% a Masters degree, and 14% a PhD degree.

Moreover, Figure 19 shows the participants' day-to-day involvement in software development project activities, such that we could get a clearer understanding of the different points of view. The day involvement categories included: research, project management, business analysis, software architecture, software development, software implementations, testing, teams management, technical support, and teaching or training. As shown in the figure, the categories included participants with 72% of software development, 44% teams management and software architecture, and 50% software implementation.

Also, we asked a question about the job titles of the participants. The participants titles include: Solution Managers, System Architects, Associate and Assistant Professors, IT

What is your highest academic degree?

36 responses

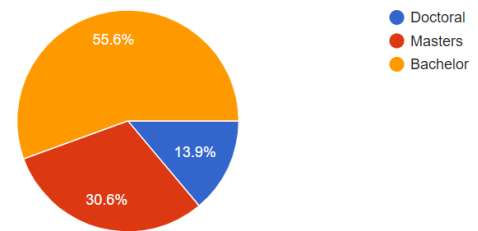


Figure 18: Academic levels of the participants in the user study

Directors, System Analysts, PhD Students, Implementation Managers, Services Manager, Integration Specialist, Technical Team Leader, Head of QA, Software Engineers, Senior Software Engineers, and Software Development Consultants.

Finally, we asked the participants about their total years of job experience in the IT field. This total ranged from 3 to 20 years of experience.

## 4.3 Results and Discussion

The results of the user study are shown in Table 2, which includes the questions and their aggregate responses.

Nowadays, a cloud-based approach of software application development is one of the important current trends in the software engineering industry. However, we believe that there are many challenges surrounding the adoption of this approach in many organizations.

Before trying to identify these problems, we wanted to be sure that building applications following the cloud approach is the preferred way especially in green-field projects over the traditional approach, in organizations in different domains. As shown in Figure 20 more than 82% of the participants strongly agreed or agreed with the statement "For new software applications, choosing a cloud-based approach can be more beneficial than choosing a traditional approach." The average of the responses for this statement was 4.18/5.0, where 5 is Strongly Agree and 1 is Strongly Disagree.

In addition, and as shown in Table 2, over 50% of the participants think that the development, maintainability, and operational cost of cloud applications is lower than traditional approaches. As shown in the table, the average of three related responses were 2.71, 2.72, and 2.80 respectively, where lower values means that the cloud is cheaper. The goal behind designing the survey for these questions in an opposite direction was to ensure that the participants completed the survey with reasoned inputs and did not rush their answers.

The above results shows the significance of cloud computing and cloud applications in reducing the cost of development, maintenance, and operations.

However, since averages of development, maintainability, and operational cost are all close to the median, we think that these

**What is your current day to day work in software development projects?  
(please check all that apply)**

36 responses

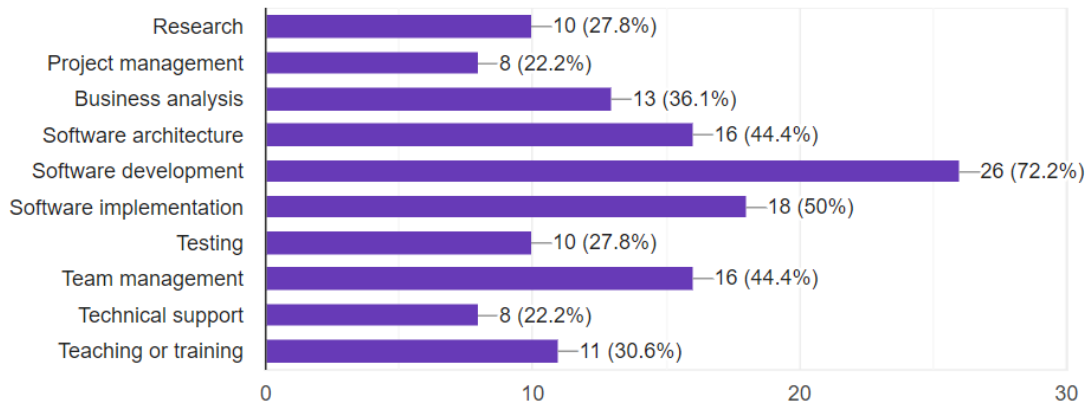


Figure 19: Involvement of the participants in software development activities

Table 2: The results of the user study

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Average
Ranking Value	(1)	(2)	(3)	(4)	(5)	
For new software applications, choosing a cloud-based approach can be more beneficial than choosing traditional approaches	-	1 (2.8%)	3 (8.3%)	19 (52.8%)	11 (30.6%)	4.18
There is a shortage of experienced software engineers who can develop high-quality cloud applications	-	-	10 (27.8%)	16 (44.4%)	10 (27.8%)	4
The cost of software development of cloud based applications is higher than cost of developing traditional software	1 (2.8%)	18 (50%)	6 (16.7%)	10 (27.8%)	-	2.71
The maintainability cost of cloud based applications is higher than that of traditional software applications	4 (11.1%)	19 (52.8%)	4 (11.1%)	6 (16.7%)	2 (5.6%)	2.72
The operational cost of cloud based applications is higher than that of traditional software applications	3 (8.3%)	15 (41.7%)	5 (13.9%)	10 (27.8%)	2 (5.6%)	2.80
Migrating applications developed using traditional approaches to be cloud-based is an expensive and risky process	1 (2.8%)	5 (13.9%)	8 (22.2%)	17 (47.2%)	2 (5.6%)	3.42

questions require more investigation.

On the other hand, all the previous questions were related to new projects in a green field situation. So, we also wanted to get the participants opinion about brown field projects to check if migrating current applications to be cloud-based is considered a risky process for the organizations, and the results showed that more than 40% of the participants think it is risky to uplift on-premises applications (e.g. migrating applications to the cloud), with an average of 3.42/5. However, we also

think that this requires more investigation, since based on the feedback of some the participants privacy, compliance, and security concerns are still dominant.

So the question is, if most of the participants think that cloud applications are more beneficial than the monolithic based, then why are there still many on-premises applications in many organizations? The answer to this question is shown in Figure 21, where more than 72% of the participants believed that there is a shortage in experienced software engineers

For new software applications, choosing a cloud-based approach can be more beneficial than choosing traditional approaches.

36 responses

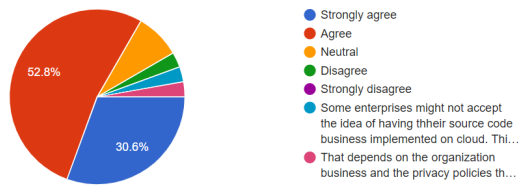


Figure 20: Cloud-based approach versus the traditional approach

who can develop high-quality cloud software applications, with an average of 4.0/5.0 of the participants thinking that there is a shortage in this area. This confirms the results of RightScale [75], discussed in Section 2.6, that the lack of resources and expertise has become a major barrier for adopting cloud computing. While the results in the RightScale report focused on cloud computing in general, our study shows that they could also be applied to cloud applications in particular.

There is a shortage of experienced software engineers who can develop high-quality cloud applications

36 responses

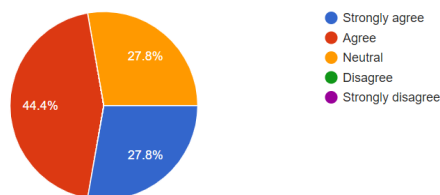


Figure 21: Shortage in experienced cloud application developers

We believe that the above results show there is a significant need for an approach that enables developing cloud-based applications in an efficient and effective way, without requiring particular expertise.

## 5 Conclusion and Future Work

This article has overviewed and discussed cloud computing and cloud applications. History and evolution of cloud computing were presented, and how the expensive hardware and infrastructure, along with the absence of economies of scale, might be the main reasons for delaying its adoption. These were followed by how the Internet, low-cost commodity-computers based data centers, smart-phones, and economic crisis played important roles in moving forward in cloud computing and offering computer services as utilities. The main advantages of cloud computing were presented as well, such as reducing total cost of ownership, time to market, and liabilities delegation.

On the other hand, disadvantages and challenges were also examined, such as security, loss of control, regulations, and political conflicts. Moreover, the effects of cloud computing on startups, economic disciplines, and hardware businesses were also discussed.

In addition, the standard services delivery models (IaaS, PaaS, and SaaS) were presented. However, we think SaaS term is being misused, and service delivery models require a standardized new taxonomy. In particular, software is a generic term that includes operating systems, platforms, applications, and even virtualization technologies such as hypervisors. Consequently, all service delivery models are SaaS in some way. The main issue with this is that future regulations of taxation, billing, and licenses may be based on the categories of the software provided.

Even though we believe the work presented in this article is sufficiently comprehensive to serve as an introductory survey for cloud computing and cloud applications, having more details about the architectural styles and patterns for building cloud applications can also be beneficial for software architects and developers. Additionally, discussion about PaaS platforms will enable them to choose whether to build on one of the available options, develop a platform on top of another one, or even create from scratch a new domain-specific platform.

A primary issue with the current available cloud computing services and technologies is the lack of standardization. This increases the risk of service provider lock. Even though this can be mitigated by creating an abstract layer between the service user and provider, this will raise the development cost and may introduce buggy features, and will not allow full utilization of the services provided. In fact, we think Amazon is leading the de-facto standardization of cloud computing following the dominant design concept [86]. However, this situation is risky because long-term stability is not guaranteed, and increasing the number of proprietary services, technologies, and protocols are more likely to occur.

We also believe that standardization is significant because it could solve many constraints, risks, and challenges, and enable more user traction. Moreover, it may eliminate privacy and data constraint issues, and support interoperability.

On the other hand, since the design and architecture of cloud applications are challenging and require specialized expertise, the utilization of native cloud platforms such a Cloud Foundry, and the work presented in some published works [6][71] may reduce the cost, and enable proper utilization of cloud resources. In fact, worldwide spending on PaaS is expected to increase from 11% in 2015 to 17% in 2020, which may be a sign of emerging need of supporting native cloud applications out of the box [30]. However, the relative novelty of this field and its lack of standardization open the door for future research in designs and methods for building native cloud application platforms.

Finally, a user-study survey to understand the actual adoption of cloud-based approach of developing new software systems in industry and academia and in different domains was presented. The user study included 36 professionals from academia and

industry from different regions of the world, with different expertise and job roles. The results of the study show that cloud-applications were the preferred approach for most of the participants, who also considered that the cost of adopting cloud applications is lower than the cost of traditional approaches. However, the participants indicated that the lack of expertise is in their view the main challenge of adopting cloud-based applications. Because the results on this item were close to the median, we believe that there is a need to further investigate the actual cost of development, maintenance, and operations of cloud-based applications versus the traditional on-premise category. In addition, finding an approach that enables effective and efficient development of cloud-based applications without the need for special expertise will be both useful and significant and will increase the adoption of cloud computing.

### Acknowledgment

This material is based upon work supported by the National Science Foundation under grant number IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The user study was approved by IRB at the University of Nevada, Reno (IRB #1362116-1).

### References

- [1] Allan Afuah and Christopher L Tucci. *Internet Business Models and Strategies*. McGraw-Hill New York, USA, 2001.
- [2] Saiqa Aleem, Faheem Ahmed, Rabia Batool, and Asad Khattak. "Empirical Investigation of Key Factors for SaaS Architecture Dimension". *IEEE Transactions on Cloud Computing*, 2019. <https://ieeexplore.ieee.org/document/8669948> (Early Access).
- [3] Amazon Web Services, Inc. "Registry of Open Data on AWS". <https://aws.amazon.com/public-datasets/> (Date last accessed January 20, 2021).
- [4] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. "A View of Cloud Computing". *Commun. ACM*, 53(4):50–58, April 2010. <https://doi.org/10.1145/1721654.1721672>.
- [5] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy H Katz, Andrew Konwinski, Gunho Lee, David A Patterson, Ariel Rabkin, Ion Stoica, and Mate Zaharia. "Above the Clouds: A Berkeley View of Cloud Computing". Technical Report UCB/Eecs-2009-28, Eecs Department, University of California, Berkeley, February 2009. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/Eecs-2009-28.html> (Date last accessed January 20, 2021).
- [6] Leonardo G Azevedo, Leonardo P Tizzei, Maximilien de Bayser, and Renato Cerqueira. "Installation Service: Supporting Deployment of Scientific Software as a Service". In *the 7th IEEE Latin-American Conference on Communications (LATINCOM)*, pp. 1–6, 2015. <https://ieeexplore.ieee.org/document/7430148>.
- [7] Armin Balalaie, Abbas Heydarnoori, and Pooyan Jamshidi. "Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture". *IEEE Software*, 33(3):42–52, 2016. <https://ieeexplore.ieee.org/document/7436659>.
- [8] Prith Banerjee, Richard Friedrich, Cullen Bash, Patrick Goldsack, Bernardo Huberman, John Manley, Chandrakant Patel, Parthasarathy Ranganathan, and Alistair Veitch. "Everything as a Service: Powering the Sew Information Economy". *Computer*, 44(3):36–43, 2011. <https://ieeexplore.ieee.org/document/5719575>.
- [9] Soon K. Bang, Sam Chung, Young Choh, and Marc Dupuis. "A Grounded Theory Analysis of Modern Web Applications: Knowledge, Skills, and Abilities for DevOps". In *Proceedings of the 2nd Annual Conference on Research in Information Technology*, RIIT '13, New York, NY, USA, pp. 61–62, 2013. Association for Computing Machinery. <https://doi.org/10.1145/2512209.2512229>.
- [10] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. "Xen and the Art of Virtualization". *SIGOPS Oper. Syst. Rev.*, 37(5):164–177, October 2003. <https://doi.org/10.1145/1165389.945462>.
- [11] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, USA, 2015.
- [12] Rajkumar Buyya and Kris Bubendorfer, editors. "Market-Oriented Grid and Utility Computing", volume 75 of *Wiley Series on Parallel and Distributed Computing* (Albert Y. Zomaya, Series Editor). John Wiley & Sons, USA, November 2009.
- [13] Vidyanand Choudhary. "Software as a Service: Implications for Investment in Software Development". In *the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007)*. IEEE, pp. 209a–209a, 2007. <https://ieeexplore.ieee.org/document/4076800>.
- [14] Pushpinder Kaur Chouhan, Feng Yao, and Sakir Sezer. "Software as a Service: Understanding Security Issues". In *Science and Information Conference (SAI)*. IEEE, pp. 162–170, 2015. <https://ieeexplore.ieee.org/abstract/document/7237140>.
- [15] CISCO. "Cloud Computing Services - PBXL - CISCO". <http://pbxl.co.jp/en/saas-paas-iaas/> (Date last accessed April 15, 2019).
- [16] Douglas Crockford. "JSON (JavaScript Object Notation)". <https://www.json.org/> (Date last accessed April 15,

- 2019).
- [17] Akon Dey, Gajanan Chinchwadkar, Alan Fekete, and Krishna Ramachandran. “Metadata-as-a-service”. In *the 31st IEEE International Conference on Data Engineering Workshops (ICDEW)*. IEEE, pp. 6-9, 2015. <https://ieeexplore.ieee.org/document/7129536>.
- [18] Tim Dixon and Stephen Hargitay. *Software Selection for Surveyors: A Guide and Directory for surveyors in General Practice*. Springer, USA, 1989. <https://link.springer.com/book/10.1007/978-1-349-21696-3>.
- [19] Docker Inc. “Empowering App Development for Developers — Docker”, 2019. <https://www.docker.com> (Date last accessed January 21, 2021).
- [20] Markus Eisele. *Modern Java EE Design Patterns: Building Scalable Architecture for Sustainable Enterprise Development*. O’Reilly Media, USA, 2016. <https://www.oreilly.com/library/view/modern-java-ee/9781492042266/>.
- [21] Javier Espadas, Arturo Molina, Guillermo Jiménez, Martín Molina, Raúl Ramírez, and David Concha. “A Tenant-Based Resource Allocation Model for Scaling Software-as-a-Service Applications Over Cloud Computing Infrastructures”. *Future Generation Computer Systems*, 29(1):273–286, 2013. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures <https://doi.org/10.1016/j.future.2011.10.013>.
- [22] Robert M Fano and Fernando J Corbató. Time-Sharing on Computers. *Scientific American*, 215(3):128–140, 1966. <https://doi.org/10.1038/SCIENTIFICAMERICAN0966-128>.
- [23] Xinyang Feng, Jianjing Shen, and Ying Fan. “REST: An alternative to RPC for Web Services Architecture”. In *the First International Conference on Future Information Networks (ICFIN 2009)*. IEEE, pp. 7–10, 2009. <https://ieeexplore.ieee.org/document/5339611>.
- [24] Fortune.com. “Here’s Why Amazon’s Cloud Suffered a Meltdown This Week”, 2017. <http://fortune.com/2017/03/02/amazon-cloud-outage/> (Date last accessed April 15, 2019).
- [25] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. “Cloud Computing and Grid Computing 360-Degree Compared”. In *2008 Grid Computing Environments Workshop*. IEEE, pp. 1–10, 2008. <https://ieeexplore.ieee.org/document/4738445>.
- [26] Martin Fowler and James Lewis. “Microservices: a Definition of this New Architectural Term”, March 2014. <https://martinfowler.com/articles/microservices.html> (Date last accessed April 15, 2019).
- [27] Free Software Foundation, Inc. “Front Page - Free Software Foundation - Working Together for Free Software”. <https://www.fsf.org/> (Date last accessed April 15, 2019).
- [28] Free Software Foundation, Inc. “The GNU Operating System and the Free Software Movement”. <https://www.gnu.org> (Date last accessed April 15, 2019).
- [29] Julia M Fromholz. “The European Union Data Privacy Directive”. *Berkeley Technology Law Journal*, 15:461, 2000. <https://lawcat.berkeley.edu/record/1117206?ln=en>.
- [30] John F. Gantz and Pam Miller. “The Salesforce Economy: Enabling 1.9 Million New Jobs and \$389 Billion in New Revenue Over the Next Five Years”. Technical Report #US41691316, IDC, September 2016. [https://www.salesforce.com/content/dam/web/en\\_us/www/academic-alliance/datasheets/IDC-salesforce-economy-study-2016.pdf](https://www.salesforce.com/content/dam/web/en_us/www/academic-alliance/datasheets/IDC-salesforce-economy-study-2016.pdf).
- [31] Jerry Gao, Pushkala Pattabhiraman, Xiaoying Bai, and Wei-Tek Tsai. “SaaS Performance and Scalability Evaluation in Clouds”. In *IEEE 6th International Symposium on Service Oriented System Engineering (SOSE)*. IEEE, pp. 61–71, 2011. doi: 10.1109/SOSE.2011.6139093 <https://ieeexplore.ieee.org/abstract/document/6139093>.
- [32] Simson L. Garfinkel. “An Evaluation of Amazon’s Grid Computing Services: EC2, S3, and SQS”. Technical Report TR-08-07, Harvard Computer Science Group, 2007. <http://nrs.harvard.edu/urn-3:HUL.InstRepos:24829568>.
- [33] David Garlan. “Software Architecture: A Travelogue”. In *Proceedings of the International Conference on Future of Software Engineering*, pp. 29–39, FOSE 2014, New York, NY, USA, 2014. Association for Computing Machinery. <https://doi.org/10.1145/2593882.2593886>.
- [34] Yizhe Ge, Shan He, Jingyue Xiong, and Donald E Brown. “Customer Churn Analysis for a Software-as-a-service Company”. In *Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, pp. 106-111, 2017. <https://ieeexplore.ieee.org/document/7937698>.
- [35] Google Inc. “App Engine Application Platform — Google Cloud”. <https://cloud.google.com/appengine/> (Date last accessed April 15, 2019).
- [36] Eugene Gorelik. “Cloud Computing Models”. Master’s Thesis, Massachusetts Institute of Technology, 2013. <https://dspace.mit.edu/handle/1721.1/79811> (Date last accessed (01/21/2021)).
- [37] Jim Gray. “Distributed Computing Economics”. *Queue*, 6(3):pp. 63–68, May 2008. <https://dl.acm.org/doi/10.1145/1394127.1394131>.
- [38] Chang Jie Guo, Wei Sun, Ying Huang, Zhi Hu Wang, and Bo Gao. “A Framework for Native Multi-Tenancy Application Development and Management”. In *The 9th IEEE International Conference on e-commerce Technology and the 4th IEEE International Conference on Enterprise Computing (CEC/EEE)*. IEEE, pp. 551–558, 2007. <https://ieeexplore.ieee.org/document/4285271>.
- [39] James Hamilton. “Internet-Scale Service Efficiency”,

- September 2008. Keynote Presentation at Large-Scale Distributed Systems and Middleware (LADIS) Workshop (Program: <http://www.cs.cornell.edu/projects/ladis2008/program.html>) (Slides: <https://perspectives.mvdirona.com/2008/09/internet-scale-service-efficiency/>).
- [40] Neil B. Harrison and Paris Avgeriou. “How Do Architecture Patterns and Tactics Interact? A Model and Annotation”. *J. Syst. Softw.*, 83(10):1735–1758, October 2010. <https://doi.org/10.1016/j.jss.2010.04.067>.
- [41] Adam Holt, Simon Flannery, Sanjay Devgan, Atif Malik, Nathan Rozof, CFA1 Adam Wood, Patrick Standaert, Francois Meunier, Jasmine Lu, Grace Chen, et al. “Cloud Computing Takes Off”. *Morgan Stanley Blue Paper*, 2011. [http://www.dabcc.com/documentlibrary/file/cloud\\_computing.pdf](http://www.dabcc.com/documentlibrary/file/cloud_computing.pdf) (Date last accessed: 01/21/2021).
- [42] Rick Kazman Humberto Cervantes. *Designing Software Architectures: A Practical Approach*. SEI Series in Software Engineering. Addison-Wesley Professional, USA, 1st edition, 2016.
- [43] Jez Humble and David Farley. *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Pearson Education, USA, 2010.
- [44] Jez Humble and Joanne Molesky. “Why Enterprises Must Adopt Devops to Enable Continuous Delivery”. *Cutter IT Journal*, 24(8):6–12, August 2011. <https://www.cutter.com/sites/default/files/itjournal/fulltext/2011/08/itj1108.pdf>.
- [45] Michael Hüttermann. “*DevOps for Developers*”. Apress, USA, 2012. <https://www.apress.com/gp/book/9781430245698>.
- [46] Bobbie Johnson. “Cloud Computing is a Trap, Warns GNU Founder Richard Stallman”. *The Guradian*, September 2008. <https://www.theguardian.com/technology/2008/sep/29/cloud.computing.richard.stallman> (Date last accessed January 21, 2021).
- [47] Timo Käköla and Juan Carlos Duenas, editors. “*Software Product Lines: Research Issues in Engineering and Management*”. Springer, USA, 2006. <https://www.springer.com/gp/book/9783540332527>.
- [48] Lori M Kaufman. “*Data Security in the World of Cloud Computing*”. *IEEE Security & Privacy*, 7(4):61 – 64, 2009. <https://ieeexplore.ieee.org/document/5189563>.
- [49] Tom Killalea. “The Hidden Dividends of Microservices”. *Commun. ACM*, 59(8):42—45, July 2016. doi: 10.1145/2948985.
- [50] Jalal Kiswani, Sergiu M Dascalu, and Frederick C Harris Jr. “Cloud-RA: A Reference Architecture for Cloud Based Information Systems”. In *ICSOFT*, pp. 883–888, 2018.
- [51] Jalal Kiswani, Muhanna Muhanna, Sergiu Dascalu, and Frederick Harris. “Software Infrastructure to Reduce the Cost and Time of Building Enterprise Software Applications: Practices and Case Studies”. In *the proceedings of ISCA 26th International Conference on Software Engineering and Data Engineering (SEDE 2017)*. pp. 93-98, ISCA, 2017. [https://www.researchgate.net/publication/322267534\\_Software\\_Infrastructure\\_to\\_Reduce\\_the\\_Cost\\_and\\_Time\\_of\\_Building\\_Enterprise\\_Software\\_Applications\\_Practices\\_and\\_Case\\_Studies](https://www.researchgate.net/publication/322267534_Software_Infrastructure_to_Reduce_the_Cost_and_Time_of_Building_Enterprise_Software_Applications_Practices_and_Case_Studies).
- [52] Jalal Kiswani, Muhanna Muhanna, Sergiu Dascalu, and Frederick Harris. “Software Infrastructure to Reduce the Cost and Time of Building Enterprise Software Applications: Practices and Case Studies”. In *Proceedings of ISCA 26th International Conference on Software Engineering and Data Engineering (SEDE 2017)*. ISCA, 2017.
- [53] Nir Kshetri. “Cloud Computing in Developing Economies”. *Computer*, 43(10):47–55, 2010. <https://ieeexplore.ieee.org/document/5530325>.
- [54] Phillip A Laplante, Jia Zhang, and Jeffrey Voas. “What’s in a Name? Distinguishing between SaaS and SOA”. *IT Professional*, 10(3), 2008. <https://ieeexplore.ieee.org/document/4525542>.
- [55] Rick Kazman LenBass, Paul Clements. *Software Architecture in Practice*. SEI Series in Software Engineering. Addison-Wesley Professional, USA, 3rd Edition, 2012.
- [56] Björn Link and Andrea Back. “Classifying Systemic Differences Between Software as a Service and On Premise Enterprise Resource Planning”. *Journal of Enterprise Information Management*, 28(6):808–837, 2015. <https://doi.org/10.1108/JEIM-07-2014-0069>.
- [57] Feng Liu, Weiping Guo, Zhi Qiang Zhao, and Wu Chou. “SaaS Integration for Software Cloud”. In *3rd International Conference on Cloud Computing (CLOUD)*. pp. 402–409, IEEE, 2010. <https://ieeexplore.ieee.org/document/5557968>.
- [58] Dan Ma and Robert J Kauffman. “Competition Between Software-as-a-service Vendors”. *IEEE Transactions on Engineering Management*, 61(4):717–729, 2014. <https://ieeexplore.ieee.org/document/6857369>.
- [59] Peter Mell and Tim Grance. “The NIST Definition of Cloud Computing”. Technical Report SP 800-145, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory, Gaithersburg, MD, September 2011. <https://csrc.nist.gov/publications/detail/sp/800-145/final>.
- [60] Microsoft Corp. “Outlook – Free Personal Email and Calendar from Microsoft”, 1996. <https://outlook.live.com/owa/> (Date last accessed January 21, 2021).
- [61] Ralph Mietzner, Andreas Metzger, Frank Leymann, and Klaus Pohl. “Variability Modeling to Support



- Customization and Deployment of Multi-tenant-aware Software as a Service Applications”. In *Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems*. IEEE, pp. 18–25, 2009. <https://ieeexplore.ieee.org/document/5068815>.
- [62] Morgan Stanley Research. “The Mobile Internet Report: Ramping Faster than Desktop Internet, the Mobile Internet Will Be Bigger than Most Think”. Technical report, Morgan Stanley & Co. Incorporated, December 2009.
- [63] Taewoo Nam and Keunhyuk Yeom. “Ontology Model to Support Multi-tenancy in Software as a Service Environment”. In *the International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, pp. 146–151, 2014. <https://ieeexplore.ieee.org/document/6984188>.
- [64] Linda Northrop. “Trends and New Directions in Software Architecture”, October 2014. Keynote Talk: Grace Hopper Celebration of Women in Computing: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=438673> (Video, Transcript, Slides).
- [65] Arto Ojala. “Software-as-a-Service Revenue Models”. *IT Professional*, 15(03):pp. 54–59, may 2013. <https://doi.ieeecomputersociety.org/10.1109/MITP.2012.73>.
- [66] Oracle Inc. “Cloud Infrastructure — Oracle”. <https://www.oracle.com/cloud/platform.html> (Date last accessed January 21, 2021).
- [67] Oracle NetSuite. “Business Software, Business Management Software – NetSuite”, 1998. <https://www.netsuite.com/portal/home.shtml?noredirect=T> (Date last accessed January 21, 2021).
- [68] Arthur O’Sullivan and Steven M. Sheffrin. *Economics: Principles in Action*. Pearson Prentice Hall, USA, 2003.
- [69] Michael Palmer and Michael Walters. *Guide to Operating Systems*. Cengage Learning, USA, 4th edition, 2012. <https://www.amazon.com/Guide-Operating-Systems-Michael-Palmer/dp/1111306362>.
- [70] Douglas Parkhill. *The Challenge of the Computer Utility*. Addison-Wesley Educational Publishers Inc. US, USA, 1966.
- [71] “Petcu, Dana and Macariu, Georgiana and Panica, Silviu and Crciun, Ciprian”. Portable cloud applications- from theory to practice. *Future Gener. Comput. Syst.*, 29(6):417—1430, August 2013. <https://doi.org/10.1016/j.future.2012.01.009>.
- [72] C Pettey. “Gartner Says Worldwide Public Cloud Services Market to Grow 18 Percent in 2017”. *Gartner, Press Release*, 2017. <https://www.gartner.com/en/newsroom/press-releases/2017-02-22-gartner-says-worldwide-public-cloud-services-market-to-grow-18-percent-in-2017#:~:text=The%20worldwide%20public%20cloud%20services,%20according%20to%20Gartner%2C%20Inc.> (Date last accessed January 20, 2021).
- [73] Inc Pivotal Software. “Spring Cloud-Native”. <https://pivotal.io/cloud-native> (Date last accessed April 15, 2019).
- [74] RightScale. “State of the Cloud Report”. Technical report, RightScale, 2017. Slides:(<https://www.slideshare.net/rightscale/rightscale-2017-state-of-the-cloud>).
- [75] RightScale. “State of the Cloud Report, Date to Navigate your Multi Cloud Strategy”. Technical report, RightScale, 2018.
- [76] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. “A Taxonomy and Survey of Cloud Computing Systems”. In *Fifth International Joint Conference on INC, IMS and IDC (NCM’09)*. IEEE, pp. 44–51, 2009. <https://ieeexplore.ieee.org/document/5331755>.
- [77] Remi Sahl, Paco Dupont, Christophe Messenger, Marc Honnorat, and Tran Vu La. “High-Resolution Ocean Winds: Hybrid-Cloud Infrastructure for Satellite Imagery Processing”. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE, 2018. <https://ieeexplore.ieee.org/document/8457895>.
- [78] Salesforce.com. “CRM Software from Salesforce.com - Customer Relationship Management - Salesforce.com”, 2021. <https://www.salesforce.com/crm/> (Date last accessed January, 21, 2021).
- [79] Theo Schlossnagle. “*Scalable Internet Architectures*”. Pearson Education, USA, 2006.
- [80] Ludwig Siegele. “Let it Rise: a Special Report on Corporate IT”. Special report, The Economist, 2008. <https://www.economist.com/special-report/2008/10/23/let-it-rise> (Date last accessed January 20, 2021).
- [81] Sommerville. “*Software Engineering*”. Addison Wesley, USA, 10th Edition, 2016.
- [82] Wei-Tek Tsai, Yu Huang, and Qihong Shao. “Testing the Scalability of SaaS Applications”. In *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE, pp. 1–4, 2011. <https://ieeexplore.ieee.org/document/6166245>.
- [83] WeiTek Tsai, XiaoYing Bai, and Yu Huang. “Software-as-a-service (SaaS): Perspectives and Challenges”. *Science China Information Sciences*, 57(5):1–15, 2014. <https://doi.org/10.1007/s11432-013-5050-z>.
- [84] Matteo Turilli, Antonino Vaccaro, and Mariarosaria Taddeo. “Internet Neutrality: Ethical Issues in the Internet Environment”. *Philosophy & Technology*, 25(2):133–151, 2012. <https://doi.org/10.1007/s13347-011-0039-2>.
- [85] Turner, Mark and Budgen, David and Brereton, Pearl. “Turning Software into a Service”. *Computer*, 36(10):38–44, 2003. <https://ieeexplore.ieee.org/document/1236470>.
- [86] James Utterback. *The Dynamics of Innovation*.

EDUCAUSE, Boulder, CO, USA, 1994.

- [87] Vaquero, Luis M. and Rodero-Merino, Luis and Buyya, Rajkumar. "Dynamically Scaling Applications in the Cloud". *SIGCOMM Comput. Commun. Rev.*, 41(1):45–52, January 2011. <https://doi.org/10.1145/1925861.1925869>.
- [88] Jinesh Varia. "Architecting for the Cloud: Best Practices". *Amazon Web Services*, 1:1–21, 2011. (Slides <https://www.slideshare.net/AmazonWebServices/best-practices-in-architecting-for-the-cloud-webinar-jinesh-varia>).
- [89] Shiliang Wu, Hans Wortmann, and Chee-Wee Tan. "A Pricing Framework for Software-as-a-service". In *the Fourth International Conference on Innovative Computing Technology (INTECH)*. IEEE, pp. 152-157, 2014. <https://ieeexplore.ieee.org/document/6927738>.
- [90] Yahoo Inc. "yahoo", 1996. <https://mail.yahoo.com> (Date last accessed April 15, 2019).



**Jalal H. Kiswani** is an academic, digital transformation advisor, and entrepreneur with more than 20 years of experience in industry and academia. In 2019 he received a PhD degree in Computer Science and Engineering from the University of Nevada, Reno, USA. He also holds a Master's degree in Enterprise Systems Engineering received in 2016 from the Princess Sumaya University and

the German-Jordanian University, Amman, Jordan, as well as a Bachelor's degree in Computer Science earned in 2002 from Mu'tah University, Karak, Jordan. He is a certified expert by Oracle and Sun Microsystems in the Java technology; in particular, he is a certified Java Programmer, Java Developer, Web Components Developer, Business Components Developer, and Java Server Faces developer. He is the founder of Cloud-Wizard low-code no-code platform, Solid-Soft for information technology solutions, and Final Solutions for training and consulting. Currently, he is an Assistant Professor in the Computing and Informatics School at Al-Hussein Technical University (HTU), a technical digital transformation advisor at Arab Bank, and a cloud platform architect at cloud-wizard.com.



**Sergiu M. Dascalu** is a Professor in the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR), which he joined in July 2002. He received his PhD degree in Computer Science (2001) from Dalhousie University, Canada and a Master's degree in Automatic Control and Computers (1982) from the Polytechnic of Bucharest, Romania. At UNR he is also the Director of the Software Engineering Laboratory (SOELA) and the Co-Director of the Cyberinfrastructure Lab (CIL). Since joining UNR, he has worked on research projects funded by federal agencies (NSF, NASA, DoD-ONR) as well as the industry. He has advised 11 PhD and over 50 Master students. He received several awards, including the 2009 Nevada Center for Entrepreneurship Faculty Advisor Award, the 2011 UNR Outstanding Undergraduate Research Faculty Mentor Award, the 2011 UNR Donald Tibbitts Distinguished Teacher of the Year Award, the 2014 CoEN Faculty Excellence Award, and the 2019 UNR Vada Trimble Outstanding Graduate Mentor Award. He is a Senior Member of the ACM.



**Frederick C. Harris, Jr.** received his BS and MS degrees in Mathematics and Educational Administration from Bob Jones University, Greenville, SC, USA in 1986 and 1988 respectively. He then went on and received his MS and PhD degrees in Computer Science from Clemson University, Clemson, SC, USA in 1991 and 1994 respectively. He is currently a Professor in the Department of Computer Science and Engineering and the Director of the High Performance Computation and Visualization Lab at the University of Nevada, Reno, USA. He is also the Nevada State EPSCoR Director and the Project Director for Nevada NSF EPSCoR. He has published more than 250 peer-reviewed journal and conference papers along with several book chapters, and has been co-editor of 13 books. He has had 14 PhD students and 79 MS Thesis students finish under his supervision. His research interests are in parallel computation, simulation, computer graphics, and virtual reality. He is also a Senior Member of the ACM, and a Senior Member of the International Society for Computers and their Applications (ISCA).

# Automatic Detection of Novelty Galaxies in Digital Sky Survey Data

Venkat Margapuri\*, Basant Thapa\*, and Lior Shamir\*  
Kansas State University, Manhattan, KS, USA

## Abstract

Galaxy images of the order of multi-PB are collected as part of modern digital sky surveys using robotic telescopes. While there is a plethora of imaging data available, the majority of the images that are captured resemble galaxies that are “regular”, i.e., galaxy types that are already known and probed. However, “novelty” galaxy types, i.e., little-known galaxy types are encountered on occasion. The astronomy community shows paramount interest in the novelty galaxy types since they contain the potential for scientific discovery. However, since these galaxies are rare, the identification of such novelty galaxies is not trivial and requires automation techniques. Since these novelty galaxies are by definition, not known, supervised machine learning models cannot be trained to detect them. In this paper, an unsupervised machine learning method for automatic detection of novelty galaxies in large databases is proposed. The method uses a large set of image features weighted by their entropy. To handle the impact of self-similar novelty galaxies, the most similar galaxies are ranked-ordered. In addition, Bag of Visual Words (BOVW) is assimilated to the problem of detecting novelty galaxies. Each image in the dataset is represented as a set of features made up of key-points and descriptors. A histogram of the features is constructed and is leveraged to identify the neighbors of each of the images. Experimental results using data from the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) show that the performance of the methods in detecting novelty galaxies is superior to other shallow learning methods such as one-class SVM, Local Outlier Factor, and K-Means, and also newer deep learning-based methods such as auto-encoders. The dataset used to evaluate the method is publicly available and can be used as a benchmark to test future algorithms for automatic detection of peculiar galaxies.

**Key Words:** Entropy based algorithms, bag of visual words, Pan-STARRS, novelty detection, feature extraction.

## 1 Introduction

In the past two decades, Earth-based astronomical instruments have largely shifted from manually controlled

telescopes to robotic telescopes that survey and image the entire sky [3], making their data available to the astronomy community through virtual observatories [6]. The astronomer community relies on the data from the observatories to aid in the furthering galactic scientific discovery. These powerful imaging instruments generate some of the world’s largest databases, contain billions of astronomical objects, and lead to numerous scientific discoveries that were not possible in the pre-information era. Sloan Digital Sky Survey (SDSS) alone has produced data leading to more than  $3 \cdot 10^4$  peer-reviewed papers, and it is very reasonable to assume that more discoveries of paramount scientific interest are hidden inside these databases. Any attempt to examine the abundance of information produced by the observatories is unrealistic and requires automation techniques to turn them into knowledge and scientific discoveries. One of the effective scientific tasks enabled by digital sky surveys is the identification of the databases. Most extra-galactic objects belong in the galaxy classification scheme, known as the “Hubble sequence” [13]. However, some galaxies do not fit any stage on the Hubble sequence and are considered “peculiar” galaxies [9]. Although these galaxies are rare, they are of high scientific interest as they carry important information about the past, present, or future universe. The Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) is an array of two robotic telescopes synchronized to observe the same part of the sky simultaneously to increase the cost-effectiveness of its imaging power. Launched in 2008, Pan-STARRS used its wide  $3^\circ$  field of view and 1.4 Gigapixel digital camera to image over  $3.5 \cdot 10^9$  astronomical objects and generated the world’s largest astronomical database of  $\sim 1.6$ PB.

In this paper the task of identifying novelty astronomical objects automatically is investigated. Deep-learning based auto-encoders technique is compared to statistical methods based on “shallow learning”. The paper proposes two techniques for novelty detection - a detection algorithm that uses the concept of entropy of a set of pre-defined numerical image content descriptors and Bag of Visual Words technique that represents an image as a set features using Scale-Invariant Feature Transform (SIFT). The performance of the proposed techniques is compared against the performance of common “traditional” unsupervised machine learning algorithms such as One-Class Support Vector Machines (OCSVM), K-Means Clustering, Local Outlier Factor (LOF), and K-Nearest Neighbors algorithm which falls in the realm of supervised

\* Department of Computer Science. Email: marven@ksuJ.edu, thapa@ksu.edu and lshamir@ksu.edu

learning. In addition, the deep learning technique of auto-encoders are applied and investigated.

## 2 Related Work

Relevant research in the area of study, while not abundant, is existent and studied to help pave a segue for the current work. The first attempt to identify peculiar galaxies on data from the Sloan Digital Sky Survey (SDSS), a sky survey with data analysis, faces challenges that largely overlap with the data analysis challenges of Pan-STARRS. It was done by using a large number of “citizen scientists” who observed the images manually over several years and determined whether the astronomical object is peculiar [4]. That initiative allowed the compilation of a large catalog of rare ring galaxies [18]. However, statistical analysis using ring galaxies detected automatically showed that many more ring galaxies were hidden inside [8]. Additionally, after several years of work involving over  $10^5$  volunteers, less than  $10^6$  objects were observed [23]. Applying the same method for the analysis of all objects imaged so far by Pan-STARRS will require over  $\sim 10^4$  years to complete. The size of the data of digital sky surveys reinforces the use of automation.

An example of automatic outlier detection applied to datasets of astronomical objects is the application of outlier detection to SDSS galaxy data to identify galaxies with unusual spectroscopic profile [2]. The method is based on unsupervised Random Forest [24], and was applied on the spectroscopic data of the galaxies rather than their images.

Substantial research has been done for general outlier detection. Among numerous approaches, the concept of entropy of features was used to mine outliers in databases [21]. Among more recent approaches, deep neural networks were used for automatic detection of outliers in data, including image data [7, 15]. While deep artificial neural networks, and in particular deep convolutional neural networks, have shown excellent performance in supervised learning of image data, the use of auto-encoders [7, 15] allows using the power of deep neural networks also for unsupervised machine learning.

## 3 Data

In the absence of a benchmark with ground truth for novelty galaxy detection, a controlled benchmark dataset of galaxy images from the Pan-STARRS sky survey is compiled. Each image is a 120 x 120 image in the JPG image format. The benchmark includes three datasets, such that each dataset contains 200 celestial objects. The first contains spiral galaxies, the second contains lenticular galaxies, and the third contains stars. The reason for using stars is that data analysis pipelines of digital sky surveys such as Pan-STARRS often struggle to classify between stars and galaxies, and therefore more objects identified as galaxies are in fact stars. Therefore, a practical algorithm for novelty galaxy detection needs to handle the existence of stars identified incorrectly as galaxies. The datasets are used such that in each run 200 galaxies from one dataset are combined with 10 galaxies from another dataset to create a

dataset in which the majority of the galaxies are “regular” galaxies, but a small number of galaxies which are different from the majority of the galaxies are also included. That allows to develop and test methods for identifying galaxies that are different from most other galaxies. For instance, in a late-type universe that contains only spiral galaxies, a lenticular galaxy would be considered a rare novelty galaxy. Similarly, in a universe of just stars, a lenticular galaxy is considered peculiar. Therefore, it can be reasonably assumed that an unsupervised machine learning algorithm that is not trained on spiral galaxies yet automatically detects a small number of spiral galaxies among a large number of lenticular galaxies, is an algorithm that will also be able to identify other novelty galaxies without training. Figure 1 shows examples of the celestial objects as imaged by Pan-STARRS.



Figure 1: Example image of star (left), lenticular galaxy (center) and spiral galaxy (right) imaged by Pan-STARRS

The dataset is freely available at [PanSTARRSData](#) and can be used as a benchmark dataset for developing future algorithms for automatic detection of novelty galaxies.

## 4 Method

### 4.1 Entropy Based Algorithm

According to shallow supervised learning of image data, each image in the dataset is first converted to a set of numerical image content descriptors that reflect its visual content through numerical values. The set of numerical image content descriptors used in this study is WND-CHARM [19], that was proven effective to machine analysis of galaxy images [14, 17, 20, 22]. In summary, the WND-CHARM library computes a comprehensive set of 2883 numerical image content descriptors that reflect numerous aspects of the visual content such as the shape, color, edges, textures (e.g., Gabor, Haralick, Tamura), fractals, polynomial decomposition of the image (e.g., Chebyshev polynomials, Zernike), and statistics of the distribution of the pixel values (e.g., Radon features, multi-scale histograms, first four moments). That feature set is described in detail in [16, 19, 26], and is applied successfully to the task of galaxy image analysis [11, 25].

The feature extraction process computes 2883 numerical image content descriptors for each galaxy image. That large set is sufficiently comprehensive to reflect numerous aspects of the galaxy morphology [14, 17, 20, 22]. However, it can also be assumed that many of these descriptors are not informative for unsupervised detection of novelty galaxies, and possibly add noise to the system. In order to select the most informative features and avoid noise to better detect novel objects in the

dataset, a process of feature selection is required. Since the learning is unsupervised, many “traditional” feature selection algorithms are not suitable. Therefore, in this study, the concept of Entropy is used as a technique to perform unsupervised feature selection on datasets with a large number of features. The entropy of a system  $S$  with  $N$  possible outcomes is computed as  $-\sum_{i=1}^N p_i \cdot \log(p_i)$ , where  $p_i$  is the frequency of outcome  $i$  in  $S$ . To compute entropy on the numerical image content descriptors, the value of each numerical content descriptor is convolved into a histogram of  $N$  bins, and  $p_i$  is the frequency of the values in the histogram bin  $i$ , such that  $i \in \{1..N\}$ . The intuition behind this method of feature selection is that informative features tend to have their values distributed in some non-random clusters of values, while non-informative features have their values randomly distributed.

Identification of novelty galaxies is unique in the sense that due to the enormous size of the datasets of galaxy images, a single one-of-a-kind peculiar galaxy is unlikely to exist. For instance, the future Vera Rubin observatory is expected to collect  $\sim 10^{10}$  galaxies, and therefore even an extremely rare one-in-a-million object is expected to appear in the dataset about  $10^4$  times. Therefore, an effective novelty galaxy detection algorithm is required to be sensitive to the number of galaxies in the dataset, and assume that many of the novelty galaxies are self-similar to each other.

To handle the self-similarity of novelty galaxies, the intuition of the algorithm is that, given a set of galaxies, the farthest  $K^{\text{th}}$  neighbor amongst the  $K^{\text{th}}$  nearest neighbors of all the galaxies is a novelty galaxy. This allows the user of the algorithm to specify a minimum number of self-similar novelty galaxies. For example, consider a dataset of 100 galaxies with a  $K$  value of 10. The distance of each galaxy in the dataset is determined by its  $10^{\text{th}}$  nearest neighbor. Therefore, if a galaxy has nine similar neighbors but is different from the remaining 90 galaxies, it will be assigned a high distance that reflects its dissimilarity from most of the galaxies. This simple mechanism might be inferior to other algorithms for the general case of novelty detection, but it is suitable for the detection of novelty galaxies as it provides the user with clear control over the number of self-similar novelty galaxies. This number changes with the type of galaxies considered, and therefore, the user is required to adjust the number based on the size of the dataset and the estimated frequency of different types of novelty galaxies.

The algorithm is described as follows:

1. Normalize the values in the dataset using Min-Max normalization.
2. Compute the entropy of each of the features of the dataset.
3. Choose a value between 0 and the greatest entropy of the features as the entropy threshold.
4. Apply the entropy threshold to the entropies of the features and set all entropies greater than the threshold to 0.
5. Pick a  $K$ , the order of the neighbor to be considered as the nearest neighbor. For instance, if the value of  $K$  is set to 5, the distance to the  $5^{\text{th}}$  closest neighbor of each of the galaxies is used as the dissimilarity measure of that galaxy.
6. Compute the distance to the  $K^{\text{th}}$  neighbor of each of the

galaxies using Minkowski distance i.e., weighted Euclidean Distance where the weights of the features are the entropy values obtained in Step 4.

7. Sort the galaxies by their distance to their  $K^{\text{th}}$  neighbor. Greater the distance, higher the likelihood that the galaxy is a novelty.

The algorithm depends on two parameters that control its performance:

1. **The order of the closest neighbor (K):** If the value of  $K$  is lower than the number of novelty galaxies of a specific type, it is possible that the distance between a certain galaxy and its  $K^{\text{th}}$  neighbor is not larger than other non-novelty galaxies. Therefore, the user is required to select a value that is higher than the number of novelty galaxies of a certain type that are expected to exist in the dataset. The number depends on the size of the entire dataset and also not necessarily known to the user. In that case the user will need to attempt several  $K$  values and inspect the results to see if the detected novelty galaxies are indeed not “regular” galaxies.
2. **The value of the entropy threshold (Step 3 in the algorithm above):** A high entropy threshold might lead to the rejection of features that carry information about the morphology of the galaxy. On the other hand, a low threshold might lead to the inclusion of noisy features.

The source code of the algorithm can be found at [PanSTARRSNoveltyDetectionAlgorithm](#).

## 4.2 Bag of Visual Words

Bag of Visual Words (BOVW) [1, 10, 12] is a technique assimilated to image classification from the popular Bag of Words (BOW) technique used in information retrieval and natural language processing. The idea is to represent an image as a set of features. Each feature consists of keypoints and descriptors. Keypoints refer to the important defining points in an image that remain unaltered even upon the application of operations such as rotation, compression and expansion. Descriptors are the entities that describe the keypoints. The combination of keypoints and descriptors are used to construct vocabularies. Each image is represented as a frequency histogram of features present in the image. The histogram is leveraged to identify the similarity of one image to another.

The detection of keypoints on the images is made using Scale-Invariant Feature Transform (SIFT) [5, 10, 12]. The procedure is defined as follows:

1. **Construction of Scale Space:** The idea behind the construction of a scale space is to ensure that the detected features are not scale dependent. In some cases, an image can appear differently at different scales. However, the detection of similarity between images is required to be performed agnostic of the scale of the images. Gaussian blur is applied on the image to reduce the noise on the

image. The original is reduced in half resulting in a scaled image. Varying degrees of Gaussian blur are applied to original and scaled images resulting in images of varying scale space.

2. **Difference of Gaussian:** The procedure is to subtract one blurred version of an original image from another, less blurred version of the original image. The intuition is that the features of the images are enhanced providing images of better quality since they are put through a blurring effect in Step 1.
3. **Keypoint Localization:** The feature selection aspect of the algorithm lies in this step. Initially, the local minima and maxima of the images are identified by comparing each pixel in the image with every other pixel in its neighborhood. Later, the keypoints that provide the most information are kept and the low contrast keypoints are discarded. The prominence of the keypoints is identified using the second-degree Taylor expansion. Only the keypoints that result in a magnitude of 0.03 are kept and the others are discarded.
4. **Orientation Assignment:** This stage of the process assigns an orientation to each of the keypoints identified in Step 3 to make them invariant to rotation. Firstly, the magnitude and orientation for each of the pixels is computed where the former represents the intensity and latter represents the orientation of the pixel. The computation of magnitude and orientation warrants that the gradients in X and Y directions be computed. Assuming that the gradient in the X direction is  $G_x$  and Y direction is  $G_y$ , the magnitude is given by  $\sqrt{G_x^2 + G_y^2}$  and orientation by  $\text{atan}(G_y/G_x)$ . The obtained magnitude and orientation are plotted as a histogram with orientation on the X axis and magnitude on the Y axis, where each bin represents a  $10^\circ$  orientation yielding in 36 bins. The peak of the histogram is considered the orientation for the keypoint.
5. **Generation of Keypoint Descriptors:** The final step is obtaining the keypoint descriptors for each of the keypoints obtained in Step 4. The descriptors for a keypoint are identified by taking a  $16 \times 16$  neighborhood around the keypoint. The neighborhood is then split into four  $4 \times 4$ -pixel neighborhoods. Similar to Step 4, a histogram is plotted between magnitude and orientation. However, the histogram is made up of only eight bins with each bin representing a  $45^\circ$  orientation. Overall, 128 bins indicating magnitude and orientation for each keypoint are obtained.

The implementation of BOVW technique for novelty detection is as follows:

1. Extract the set of features from each of the images in the data set using Scale-Invariant Feature Transform (SIFT).
2. Convert the extracted features into visual words by using the K-Means Clustering algorithm. The centers identified by the algorithm form the vocabulary of visual words.

3. Compare the features of each of the images against the vocabulary and create histograms for each of the images in both the training and testing data sets.
4. Select a K, the order of the neighbor to be considered as the nearest neighbor and compute the Euclidean distance from each galaxy to its  $K^{\text{th}}$  neighbor using the data from the histogram.
5. Sort the galaxies by the distance to their  $K^{\text{th}}$  neighbor. Greater the distance, higher the likelihood that the galaxy is a novelty.

The source code of the algorithm can be found at PanSTARRSVisualBOWAlgorithm.

## 5 Method

The concept of 'rank' is used to express the performance of the proposed techniques. Rank  $r$  is the number of query galaxies determined by the algorithm as the most likely to be novelty galaxies. If a novelty galaxy is among these  $r$  galaxies, the attempt is considered a hit, and otherwise a miss. Since candidates of novelty galaxies are inspected manually, a method that returns false positives is acceptable as long as the novelty galaxies are among a set that is small enough for manual analysis. Note that the problem of novelty galaxy detection does not require identifying all novelty galaxies, as novelty galaxies of the same type are expected to be present multiple times in galaxy datasets acquired by robotic telescopes.

### 5.1 Entropy Based Algorithm

Figure 2 shows the performance of the Entropy based algorithm stated in Section 4.1 when the K parameter is set to 5, 10, and 20. The results show that the performance of the algorithm when identifying spiral galaxies among lenticular galaxies is better than the performance of the algorithm when identifying stars among lenticular galaxies. This is partly explained by the fact that lenticular galaxies and stars are more similar in morphology to each other compared to lenticular and spiral galaxies.

### 5.2 Bag of Visual Words

The performance of the Bag of Visual Words technique described in Section 4.2 is shown in Figure 3 when the value of K is set to 5, 10 and 20. From the results, it is inferred that the performance of the technique while identifying stars is far superior compared to the performance of the technique while identifying spiral and lenticular galaxies. It is perhaps due to the similarities observed between the images of spiral and lenticular galaxies. While the galaxies are structurally different, both lenticular and spiral galaxies contain a sea of nebulous matter around them. The error rate for stars is significantly lower because the images of stars contain no nebulous matter around them and are structurally circular. This characteristic of stars aids the technique in being identified better.

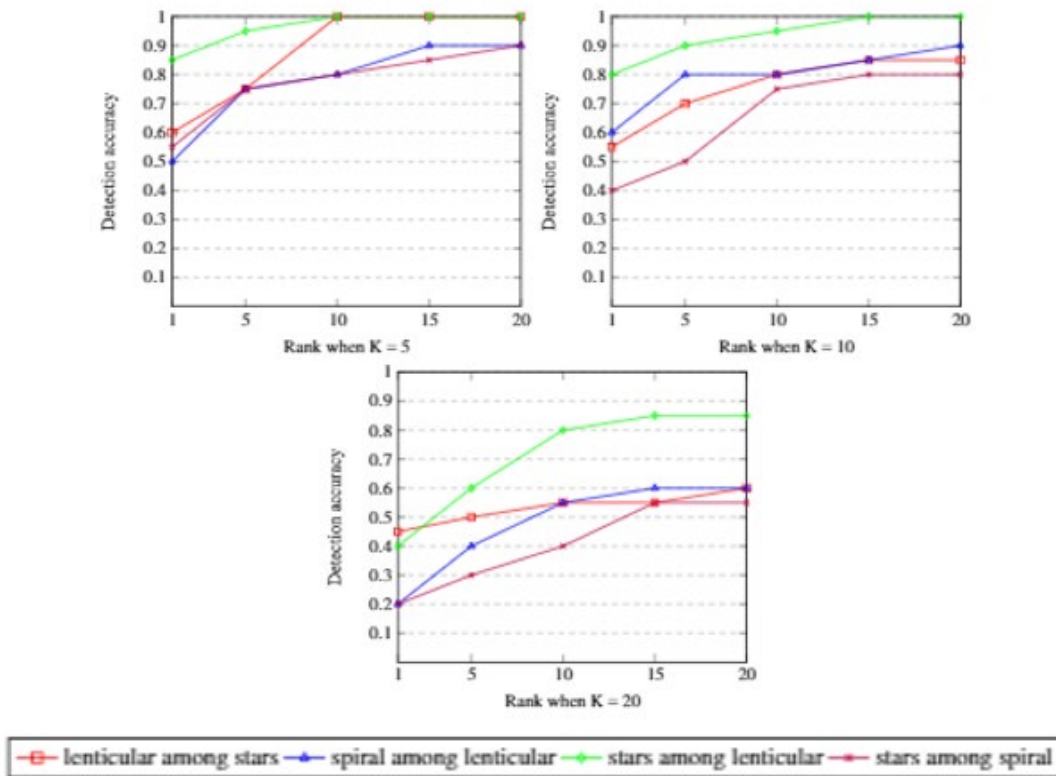


Figure 2: Detection accuracy when using different datasets and ranks using entropy-based algorithm

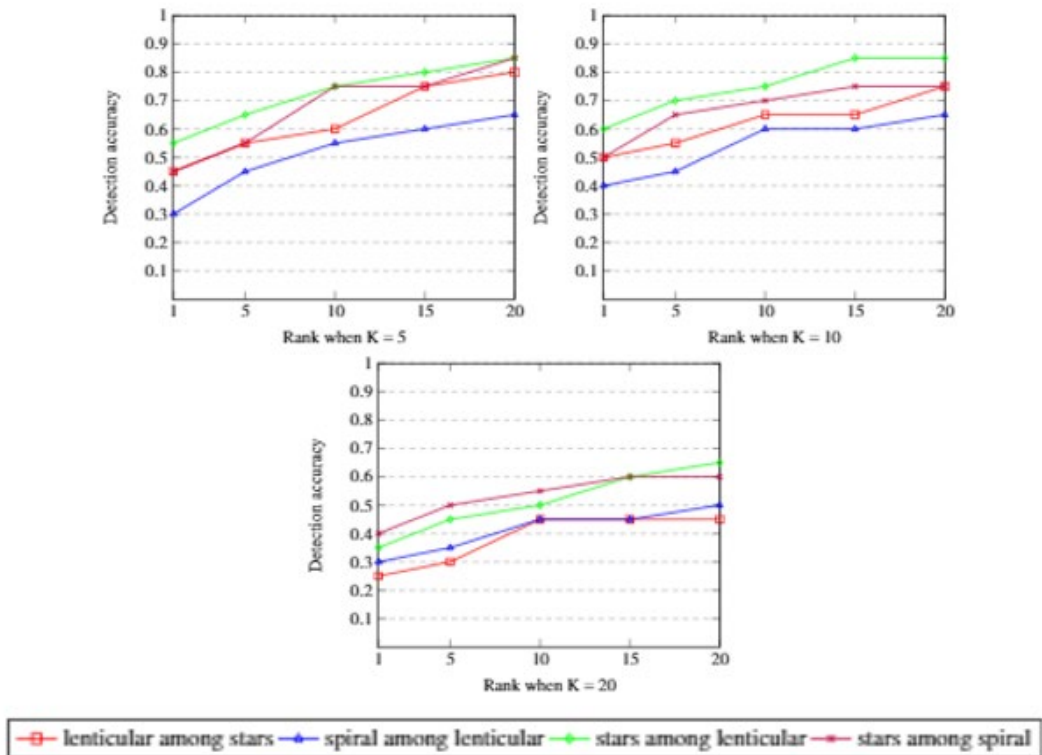


Figure 3: Detection accuracy when using different datasets and ranks using bag of visual words

## 6 Comparison to Novelty Detection Algorithms

Since the problem of automatic novelty galaxy detection is relatively new, not many proposed novelty detection algorithms for galaxies are available in the existing literature. Hence, the performance of the proposed algorithm is compared against “traditional” novelty detection algorithms such as one-class SVM, K-Means, and Local Outlier Factor (LOF), as well as the deep learning-based auto-encoders.

### 6.1 Comparison to Deep Learning with Auto-Encoders

Auto-encoders [7] are a class of unsupervised machine learning using artificial neural networks (ANN). A typical artificial neural network consists of an input layer, which inputs the data to the layers of the neural network, several hidden layers, and an output layer, which outputs the outcome. Each of the hidden layers in the network performs computations on the weighted inputs and transfers the computed result to the next layer. An auto-encoder can be conceptualized as a specific type of neural network that copies the input values to the output without requiring a target variable. Since target variables are not required, it is a good fit for unsupervised learning [15].

For this experiment, a deep auto-encoder is used. The auto-encoder architecture comprises of ReLU activation function in the encoding layers and sigmoid activation function in the decoding layers. The loss function used is binary cross-entropy and the optimizer used is RMSProp. The size of the input of 120 x 120. Auto-encoders with three different

architectures are developed. Architecture#1, with hidden layers of sizes 128, 64, 32, 64, 128, architecture#2, with hidden layers of sizes 1024, 512, 256, 512, 1024 and architecture#3 with hidden layers of sizes 2048, 1024, 512, 1024, 2048. In each of the datasets, the “regular” galaxy images are split into two groups, one containing 180 images to train the auto-encoder, and another of 20 images to test on the auto-encoder to obtain the reconstruction losses. Then, the “novelty” galaxy images are tested on the auto-encoder, and the loss of the “novelty” galaxies is compared to the loss of the “regular” galaxies. For evaluation, the 30<sup>th</sup> to the 90<sup>th</sup> percentile of reconstruction loss values on “regular” galaxies are used as thresholds, and the percentage of “novelty” galaxies identified from amongst 200 images of “novelty” galaxies is computed as shown in Figure 4.

### 6.2 One-Class Support Vector Machines (OCSVM)

The OCSVM algorithm is applied to each of the four datasets using the scikit-learn library. The performance of the algorithm is measured as the number of actual “novelty” galaxies identified by the algorithm divided by the total number of “novelty” galaxies attempted. Ideally, only the ten “novelty” galaxies are identified as “novelty” galaxies by the algorithm, in which case the detection rate would be 100%. However, the observation on all four datasets is that the algorithm identifies a large portion of “regular” galaxies also identified as “novelty” galaxies while also misidentifying some “novelty” galaxies as regular galaxies. So, the performance of the algorithm is similar to that of novelty

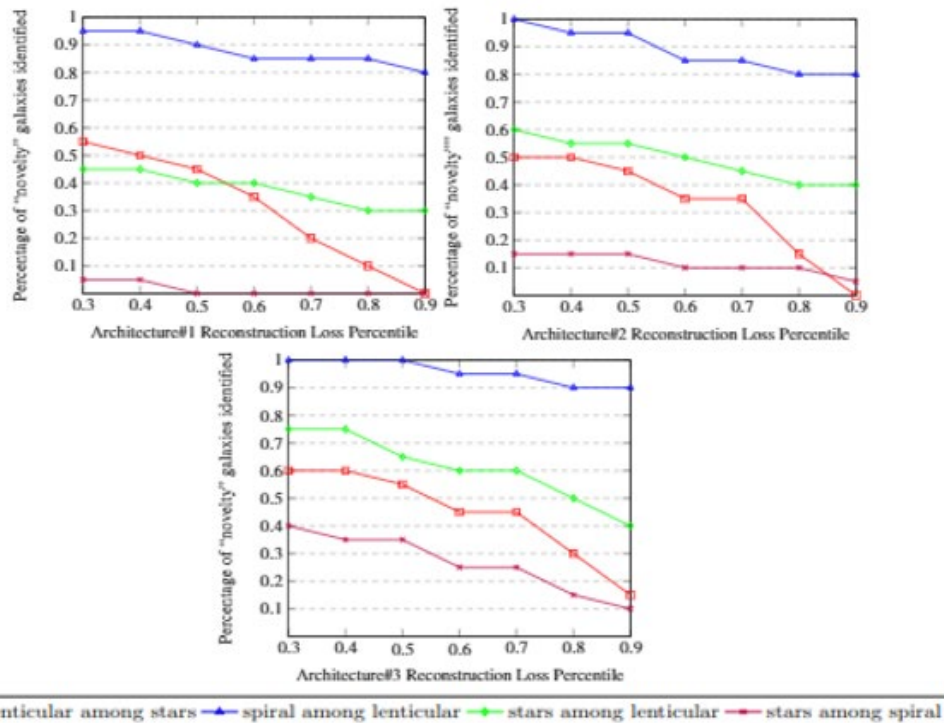


Figure 4: Detection accuracy when using different datasets and ranks using auto-encoders



galaxy detection by random chance. The outcomes of the algorithm are shown in Figure 5.

### 6.3 Local Outlier Factor (LOF) Algorithm

The Local Outlier Factor (LOF) algorithm produces a score that provides an insight into the likelihood of a data point being an outlier in a given dataset. The scikit-learn LOF library is used to apply the algorithm to each of the datasets. Since the algorithm is unsupervised, no alteration is made to the datasets. A score close to 1 means that the sample is an inlier, while outliers have a larger LOF score. The results show that for each of the datasets, all of the values obtained for the LOF scores are 1, indicating that the algorithm considers all of the images, including the outliers, as the same class as the inliers. As a result, the accuracy obtained using the algorithm is 0 % on all four of the datasets.

galaxies are the most frequent. The results are as shown in Figure 6.

### 7 Comparison to Novelty Detection Algorithms

Automation techniques in the field of astronomical discovery and analysis are the need of the hour considering the enormous amount of information recorded by modern sky surveys using robotic telescopes. The infrequent occurrence of novelty galaxies makes the problem of novelty galaxy detection complex since conventional machine learning classifiers don't always perform well owing to lack of enough training data.

The proposed unsupervised novelty detection algorithm uses a comprehensive set of numerical image content descriptors, and therefore depends on feature selection. Entropy is shown as a useful way to select features for the

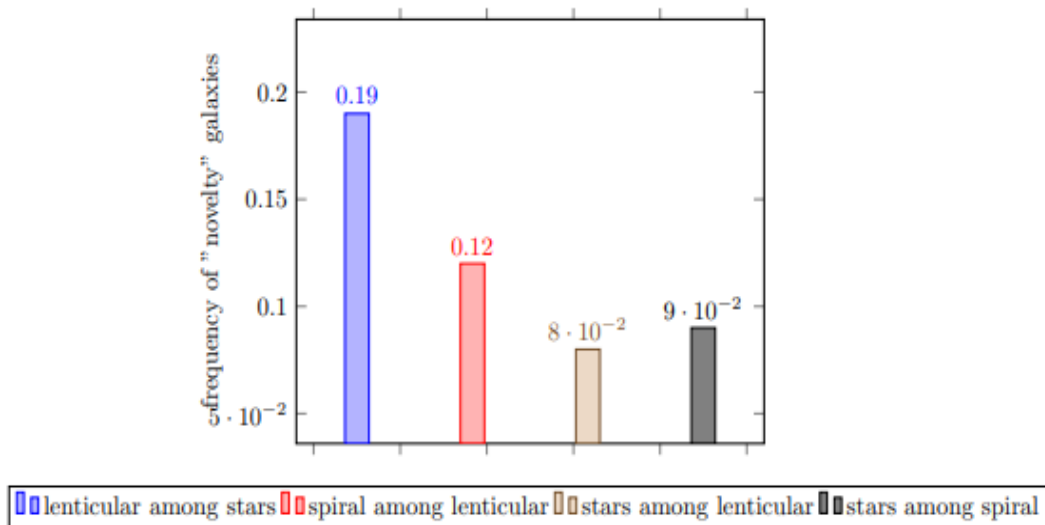


Figure 5: Detection accuracy when using different datasets and ranks using OCSVM

### 6.4 K-Means Clustering Algorithm

K-Means is a simple and established unsupervised learning algorithm which works by choosing a centroid value for each randomly chosen cluster, and iteratively assigning each data point to a cluster that best fits based on the Euclidean distance between the data point and the centroids of the clusters. K-Means is typically used for automatic clustering. However, in some cases it can be used for novelty detection by identifying small clusters. If a small cluster is identified, the cluster may contain a small number of self-similar samples that are different from the other samples in the dataset. Therefore, K-Means is an algorithm that could be possibly used for novelty detection in the current scenario. The algorithm is tested with two through 10 clusters. The performance is measured as the number of novelty galaxies among regular galaxies in the cluster in which novelty

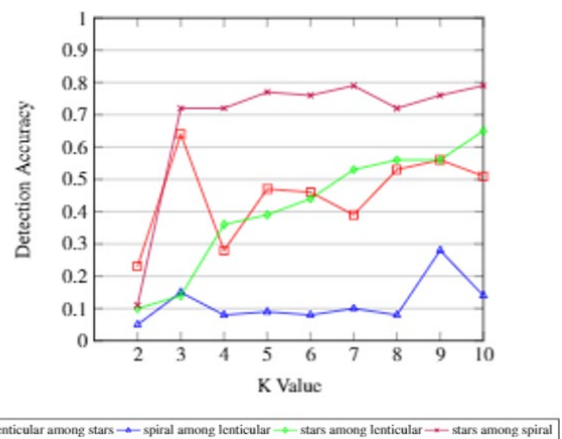


Figure 6: Detection accuracy using different datasets and using K-Means Clustering

problem of unsupervised detection of novelty galaxies.

The Visual Bag of Words technique assimilated to novelty detection identifies the distinctive features of galaxies to help identify novelty galaxies. The technique can scale to images of different dimensions and orientations since it is built to be scale and orientation invariant.

The methods proposed in the paper outperform “traditional” methods such as one-class SVM, K-Means, and newer methods based on deep neural networks such as auto-encoders. It should be noted, however, that the relatively low number of annotated samples does not allow efficient training of an autoencoder, that normally requires a high number of samples. The dataset used for the experiments is publicly available and can be used for the development and testing of new algorithms for novelty galaxy detection in large astronomical databases.

The downside of the evaluation is that it is performed on a relatively small and controlled dataset, far smaller than the huge datasets generated by modern digital sky surveys. The efficacy of the method will be tested in the future by applying it to extremely large image databases and evaluating its ability to identify real novelty galaxies hidden among millions of celestial objects that have not been inspected yet.

#### Acknowledgement

The research was funded in part by NSF grant number AST-1903823.

#### References

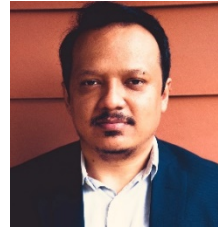
- [1] Singh Aishwarya, “A Detailed Guide to the Powerful Sift Technique for Image Matching,” *Medium*, 2019.
- [2] T. Amarbayasgalan, B. Jargalsaikhan, and K. H. Ryu, “Unsupervised Novelty Detection using Deep Autoencoders with Density Based Clustering,” *Applied Sciences*, 8(9):1468, 2018.
- [3] K. Borne, “Virtual Observatories, Data Mining, and Astroinformatics,” *Planets, Stars and Stellar Systems*, 2:403-443, 2013.
- [4] R. J. Buta, “Galactic Rings Revisited—I. CVRHS Classifications of 3962 Ringed Galaxies from the Galaxy Zoo 2 Database,” *Monthly Notices of the Royal Astronomical Society*, 471(4):4027-4046, 2017.
- [5] Tyagi Deepanshu, “Introduction to SIFT,” *Medium*, 2019.
- [6] S. G. Djorgovski, A. A. Mahabal, A. J. Drake, M. J. Graham, and C. Donalek, “Sky Surveys,” *Planets, Stars, and Stellar Systems*, 2:223-281, 2013.
- [7] E. Kuminski, J. George, J. Wallin, and L. Shamir, “Combining Human and Machine Learning for Morphological Analysis of Galaxy Images,” *Publications of the Astronomical Society of the Pacific*, 126(944):959, 2014.
- [8] C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. C. Nichol, M. J. Raddick, and A. Szalay, “Galaxy Zoo 1: Data Release of Morphological Classifications for Nearly 900,000 Galaxies,” *Monthly Notices of the Royal Astronomical Society*, 410(1):166-1768, 2011.
- [9] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, and P. Murray, “Galaxy Zoo: Morphologies Derived from Visual Inspection of Galaxies from the Sloan Digital Sky Survey,” *Monthly Notices of the Royal Astronomical Society*, 389(3):1179-1189, 2008.
- [10] G. Lowe, “Sift-the Scale Invariant Feature Transform,” *Int. J.*, 2(91-110):2, 2004.
- [11] Z. Lu, L. Wang, and J. R. Wen, “Image Classification by Visual Bag-of-Words Refinement and Reduction,” *Neurocomputing*, 173:373-384, 2016.
- [12] Ning Minghao, “Sift (Scale-Invariant Feature Transform),” *Medium*, 2019.
- [13] W. W. Morgan and N. U. Mayall, 1957. “A Spectral Classification of Galaxies,” *Arp and Halton, Atlas of Peculiar Galaxies, Publications of the Astronomical Society of the Pacific*, 69(409):291-303, 1966.
- [14] L. Shamir, “Evaluation of Face Datasets as Tools for Assessing the Performance of Face Recognition Methods,” *International Journal of Computer Vision*, 79(3):225-230, 2008.
- [15] L. Shamir, “Automatic Morphological Classification of Galaxy Images,” *Monthly Notices of the Royal Astronomical Society*, 399(3):1367-1372, 2009.
- [16] L. Shamir, “Morphology-Based Query for Galaxy Image Databases,” *Publications of the Astronomical Society of the Pacific*, 129(972):024003, 2016.
- [17] A. Schutter and L. Shamir, “Galaxy Morphology—An Unsupervised Machine Learning Approach,” *Astronomy and Computing*, 12:60-66, 2015.
- [18] L. Shamir, “Automatic Detection of Full Ring Galaxy Candidates in SDSS,” *Monthly Notices of the Royal Astronomical Society*, 491(3):3767-3777, 2020.
- [19] L. Shamir, A. Holincheck, and J. Wallin, “Automatic Quantitative Morphological Analysis of Interacting Galaxies,” *Astronomy and Computing*, 2:67-73, 2013.
- [20] L. Shamir, T. Macura, N. Orlov, D. M. Eckley, and I. G. Goldberg, “Impressionism, Expressionism, Surrealism: Automated Recognition of Painters and Schools of Art,” *ACM Transactions on Applied Perception (TAP)*, 7(2):1-17, 2010.
- [21] L. Shamir, N. Orlov, D. M. Eckley, T. Macura, J. Johnston, and I. G. Goldberg, “Wndchrm—an Open-Source Utility for Biological Image Analysis,” *Source Code for Biology and Medicine*, 3(1):1-13, 2008.
- [22] L. Shamir and J. Wallin, “Automatic Detection and Quantitative Assessment of Peculiar Galaxy Pairs in Sloan Digital Sky Survey,” *Monthly Notices of the Royal Astronomical Society*, 443(4):3528-3537, 2014.
- [23] T. Shi and S. Horvath, “Unsupervised Learning with Random Forest Predictors,” *Journal of Computational and Graphical Statistics*, 15(1):118-138, 2006. Ming-Jian Zhou and Jun-Cai Tao. “An Outlier Mining

Algorithm Based on Attribute Entropy,” 2011.

- [24] K. Sun, J. Zhang, C. Zhang, and J. Hu, “Generalized Extreme Learning Machine Autoencoder and a New Deep Neural Network,” *Neurocomputing*, 230:374-381, 2017.
- [25] J. Yang, Y. G. Jiang, A. G. Hauptmann, and C. W. Ngo, “Evaluating Bag-of-Visual-Words Representations in Scene Classification,” *Proceedings of the International Workshop on Multimedia Information Retrieval*, pp. 197-206, September 2007.
- [26] Y. Zhang, R. Jin, and Z. H. Zhou, “Understanding Bag-of-Words Model: A Statistical Framework,” *International Journal of Machine Learning and Cybernetics*, 1(1-4):43-52, 2010.



**Venkat Margapuri** is a researcher working on his PhD. in Computer Science at Kansas State University. His interests are in the areas of machine learning, scientific computing, robotics and data science.



learning.

**Basant Thapa** is currently pursuing a Master's in Software Engineering from Kansas State University. He graduated from Wichita State University with a Bachelor's in Computer Science and a minor in Mathematics. He is interested in advancing his study in the field of big data solutions and machine



**Lior Shamir** is an Associate Professor of computer science at Kansas State University. He received his Ph.D from Michigan Technological University in 2006, and postdoc at the National Institutes of Health (NIH).

# Innovation on Digital Platforms: Impacts of Control Portfolios on Novelty

Alan Hevner\*

University of South Florida, Tampa, FL USA

Onkar Malgonde†

Northern Illinois University, DeKalb, IL USA

## Abstract

The development of software applications on digital platforms requires both agility and restraint to meet rapidly changing user requirements while adding novel features to a platform-based application domain. User value creation focuses on exploring the solution space to innovate and attract new customers while retaining existing customers. In this pilot study, we analyze the essential tensions between software project controls and the development activities to achieve novelty in the software product. Drawing from cognitive theories of creativity and reasoning, we posit the need for both informal controls that enhance creativity and formal controls that enhance reasoning in a balanced portfolio of project controls. Two case studies provide preliminary evidence that a well-balanced portfolio of controls can result in the effective design and implementation of novel product features. We position the case studies in the context of digital platforms to bound our definitions of control mechanisms and novelty. We conclude with implications for software development on digital platforms and future research directions.

**Key Words:** Control, novelty, creativity, reasoning, digital platforms.

## 1 Introduction

Recent advancements in software systems and information technologies are driving digital transformation initiatives within organizations and renewed organizational focus on innovation [14, 24, 26]. To support such endeavors, we are witnessing significant changes in business practices such as partner networks, subscription-based usage, and open innovation. With a renewed focus on innovation in digital era, software development projects are increasingly identifying and incorporating enabling technologies and tools such as platform-based application development, low-code development platforms, enterprise application packages, and prototyping tools [24]. Many organizations focus on achieving value creation opportunities in the context of digital platforms which

represent a mainstream channel for development and deployment of software development projects. Software application development on digital platforms requires project teams to achieve application-platform match, realize application-market match, exceed core value proposition of the platform, and provide novelty of the application. This is in addition to traditional outcomes of project success such as project efficiency, quality, and adaptiveness [11, 19, 23].

A key research question is how to achieve the right balance between project controls while supporting the creative design and implementation of novel features. Prior research has found contradictory results. Several studies focusing on the control of software projects identify a negative effect of control mechanisms on innovation outcomes [4, 6] suggesting a “stifling of creativity and limiting of adaptability” [6, p. 225]. However, many software projects with portfolios of existing control mechanisms do effectively release novel products on different platforms and in many application domains.

We explore the activities needed to produce a novel application. Drawing from research on human cognition, novelty is achieved via synergy between creative, divergent thinking and reasoning, convergent thinking [12, 21]. Divergent thinking engenders imagination, provocation, unstructured syntheses, serendipitous discovery, and answers that break with conformity. This mode of cognition focuses on the synthetic generation of multiple disparate answers to a given problem [1]. Convergent thinking refers to the mode of human cognition that strives for the generation of a single, concrete, accurate, and effective solution. Thus, divergent cognition (creativity) produces many possible new and interesting solutions; while convergent cognition (reasoning) assesses the feasibility of these solutions and identifies the best solution for implementation. Thus, we explore the following research question:

*How does a portfolio of formal and informal controls relate to the creative and reasoning activities required for the development of novel features in software projects on digital platforms?*

To address this question, we propose a research model and assess it on two real-world case studies. Via qualitative interviews with members of the two software development projects, we identify the control mechanisms applied and the

\* School of Information Systems and Management. Muma College of Business. Email: ahevner@usf.edu.

† Operations Management and Information Systems. College of Business. Email: omalgonde@niu.edu.

development activities performed to achieve novel application features. We organize this paper as follows. In sections 2 and 3, respectively, brief literature surveys present the grounding theories of control and novelty. Section 4 presents a proposed research model relating controls and novelty. In Section 5, we discuss our research methodology, data collection, and analysis approach. We present the criteria for selecting two case studies. Section 6 discusses the results of each case study. We conclude in section 7 with the implications of this research for building control portfolios to support the design and implementation of novel features on digital platforms. Future research directions are presented.

## 2 Software Development Controls

Controls in software projects have received considerable attention in research and practice [19]. However, several surveys identify a lack of rigorous study on the effect of control mechanisms on novelty of software application features [15, 19]. Control theory as applied to software projects recognizes two categories of controls – formal and informal [6].

### 2.1 Formal Controls

Formal control types are classified as input control, behavior control, outcome control, and emergent outcome control. The key distinction for formal control is the presence of an identifiable controller and contree. Each type of formal control is briefly described.

- **Input Control:** The controller assigns resources (inputs) to the development project team (contree) that are sufficient to successfully complete a desired result. Input controls are closely related to the Theory of Effectuation in the field of entrepreneurship [22]. An entrepreneurial software team is provided with means to achieve project aspirations [15]. The team decides how best to affect the desired results. There has been limited discussion on the use of input control for software projects [23]. However, in the context of R&D projects in the pharmaceutical industry, Cardinal [4] finds empirical evidence in support of input controls leading to incremental and radical innovation. Software project managers can alter resources such as team composition, technical environments, tools, and knowledge resources, among others, for the project team to facilitate identification and assimilation of novel features for the focal application.
- **Behavior Control:** The controller uses processes and rules to direct contrees towards accomplishment of organizational goals. In software project control, behavior control is exercised using mandated routines such as meetings and development methodologies that signal use of specific methods in the project. Prior research attributes use of behavior control to outcomes of project adaptiveness, efficiency, and quality [23]. In the context of R&D projects in the pharmaceutical industry, Cardinal [4] finds a negative effect of behavior controls on novelty. Some studies posit

a negative effect of behavior control due to rigidity and lack of experimentation which stems from overly specifying contree behaviors [6, 11].

- **Outcome Control:** The controller specifies an outcome and evaluates the project based on the contree achieving the outcome. For outcome control to be effective, the controller should be able to specify the outcome a priori and measure the achieved outcome. Typically, software requirement/specifications and fixed timelines form popular control mechanisms for outcome control. An important challenge with the use of outcome control is the ability of the controller to define a novel product at the start of a project. In the digital platform environment, novelty of an application will change over time, as platform and competitors update their offerings. Also, measuring novelty of an application is challenging [15].
- **Emergent Outcome Controls:** Instead of novel outcomes being predictable at the beginning of a project, novel outcomes often emerge during the course of a software development project. The controller sets defined project milestones to assess the trajectory of the emerging novel product. The use of scope boundaries and ongoing feedback are important forms of emergent outcome controls that allow the project team to revise outcomes that are difficult to identify a priori by facilitating feedback [13]. Scope boundaries channel the team's efforts while allowing autonomy within the boundaries. For development teams developing applications for digital platforms, emergent outcome controls provide mechanisms which can enable the team to explore the technological space provided by the platform and seek feedback from within and outside the development team.

### 2.2 Informal Controls

Informal controls rely on a software team's shared values and vision of the application. There is no strict hierarchy of team structure between controller and contree. Two forms of informal controls are clan and self.

- **Clan Control:** Shared values and goals among team members motivate the project to a successful result. Chua et al. [5] find that clan controls need to be developed over time with careful maneuvering to be effective. Experienced teams demonstrate higher levels of clan control. Although difficult to implement [9], clan controls demand minimal monitoring once implemented. Empirical evidence suggests a positive effect of clan control on project's success [5, 23]. In digital platform environment, clan control can play an important role in channeling the team's efforts towards developing a novel application [15].
- **Self-Control:** Team members have internal motivations to self-direct their actions to achieve project goals. Prior research suggests a positive effect of self-control on project's success [14]. In the dynamic environment of digital platforms, it is important to enable individual autonomy in order to identify and design novel features that

will set apart the focal application. The project manager may identify appropriate control mechanisms to enable team members to exercise self-controls to experiment with features and technological advancements to develop novel features for the application.

### 3 Software Application Novelty

Traditionally, organizations develop innovative product lines through a linear value chain [20]; products are designed, developed, and marketed by a single firm. However, with pervasive digital innovations and technology, the locus of organizational innovation has shifted to digital software platforms which rely on external entities to develop innovative solutions. Thus, current conceptualizations of novelty in an application refer to the features and extensions offered by the application relative to the platform and other competing applications.

For our research, we extend this definition of novelty, as the dependent variable of our study, to include content provided by the application, data sources and their designs, user interfaces, alerts/messages, and platform's ecosystem that distinguishes the focal application from its competition (competing applications that may or may not be on the same platform). Consequently, novelty of the application is not limited to its features. Novelty for a focal application may arise from its choice of platform since the application's user may not differentiate<sup>1</sup> between the application and its platform.

To achieve application novelty on digital platforms, the software development team must effectively iterate between two cognitive modes – *creative* activities that generate new ideas and *reasoning* activities that analyze the feasibility of the new ideas to determine how best to implement the novel application features. The following subsections briefly survey and distinguish these two essential mindsets in software development.

#### 3.1 Creativity in Software Development

The literature on the cognitive bases of creativity is fragmented with little consensus about the neural mechanisms underlying creativity. This is true for the literature on creativity as a whole and for the sub-domains of divergent thinking, aesthetics (e.g., style, art, music), and insight. Creativity is viewed as a complex computational model of activities 'in' many areas of the brain [10]. Conceptualizing and treating creativity as if it is a single entity fails to accommodate its complexity and infers that it comprises a limited number of fundamental processes and brain structures underlying it. Dietrich and Kanso [8] point out that this is likely to be a fallacy, and that "it is hard to believe that creative behavior in all its manifestations – from carrying out exquisitely choreographed dance moves, to scientific discovery, constructing poems and coming up with ingenious ideas of what

to do with a brick - engages a common set of brain areas or depends on a limited set of mental processes" (p. 845).

While neuroscience provides no definitive answers on the origin of creativity, the software engineering community has applied several development processes that aim to generate novel artifacts. Creative processes incorporate the dynamics of the (socio-cognitive) activities underlying an artifact's complexity, creation, composition, and later use and evolution. High levels of creativity are fostered by radical, out-of-the-box thinking and non-conventional approaches for the development of new ideas. Organization policies that foster creativity are key; particularly those that provide the entrepreneurial team time to think and try out their own ideas. Specific techniques that could be used include *genius grants* and *bootlegging* [7, 17]. Other examples are *tinkering time* and *hack-a-thons*.

Seminal research on creative teams by Amabile and Pillemer [2] identifies the following four components as integral to the creative process:

- Domain-relevant skills include intelligence, expertise, knowledge, technical skills, and talent in the particular domain in which the team is working;
- Creativity-relevant processes are enabled by personality and cognitive characteristics that lend themselves to taking new perspectives on problems, such as independence, risk taking, self-discipline in generating ideas, and a tolerance for ambiguity.
- Intrinsic task motivation is seen as a central tenet. People are most creative when they feel motivated primarily by the interest, enjoyment, satisfaction and challenge of the work itself – and not by extrinsic motivators.
- The social environment, the only external component, addresses the working conditions that support creative activity. Negative organizational settings harshly criticize new ideas, emphasize political problems, stress the status quo, impose excessive time pressures, and support low-risk attitudes of top management. While positive organizational settings stimulate creativity with clear and compelling management visions, work teams with diverse skills working collaboratively, freedom to investigate ideas, and mechanisms for developing new ideas and norms of sharing ideas.

It is important to note that Amabile's work is based on two important assumptions. First, there is a continuum from relatively low, everyday levels of adaptive creativity to the higher levels of creativity found in significant inventions and scientific discoveries. Second, there are degrees of creativity exhibited in the work of any single individual at different points of time and circumstances [2].

#### 3.2 Reasoning in Software Development

A student once asked Linus Pauling, "Dr. Pauling, how does one go about having good ideas?" He replied, "You have lots of ideas and throw away the bad ones." [2, p. 116]. Effective innovation requires more than just the generation of many

<sup>1</sup> In enterprise grade applications, users are often unaware about the digital platform and its offerings when using the application.

creative ideas. Many creative individuals waste time, energy, and resources chasing infeasible and unprofitable hunches into blind alleys. Successful innovation also requires the intellectual control to refine creative thinking into practical solutions. Such control is dependent on the cognitive skills of reasoning and judgment.

Human cognitive reasoning reflects thinking in which plans are made, hypotheses are formed, and conclusions are drawn on the basis of evidence in the form of data, past experience, or knowledge. While creativity often calls for divergent thinking to break out of mindsets; reasoning calls for convergent thinking to refine ideas into practical artifacts and actions. Moving design ideas from ‘blue sky’ to artifact instantiations requires goal setting and a plan to answer the following types of systems development questions:

- *Is the design feasible?* - Can the proposed design be implemented and does the proposed design meet the requirements of the stakeholders and the platform?
- *Does the design have value?* - Does the design offer benefits unmatched by competing candidate designs? Here the objective becomes to establish an ordinal valuation that can be used to rank candidate designs.
- *How can the design be most effectively represented?* – How can we best communicate the intricacies of the design to collaborators, implementers (e.g., architects, programmers), and other stakeholders?
- *How best to construct the actual use artifacts?* How do we guide the construction of the use artifact? As examples - a

blueprint is a construction artifact that serves to guide the physical construction of a house; source code is a construction artifact that serves to generate the programs that are distributed to users.

Closely related to reason is the human cognitive facility to judge, or evaluate, ideas at various design stages of the development process. The goal of judgment is to predict the future; to predict which candidate designs will be better than others. Without the ability to narrow the field (i.e., design space) it would be impossible to refine many good ideas down to one ‘satisfactory’ design artifact. This is a very tricky area of human cognition since it involves self-criticism, self-esteem, and motivation. However, studies have shown that humans are capable of making effective and rapid judgments based on first impressions (e.g. [3]). Beyond first impressions, measurements and evaluations are based on the rigorous definition of utility functions that estimate the values of candidate designs in order to facilitate the ranking of alternatives.

#### 4 Research Model

While the topics of software development controls and software application novelty have received considerable attention in the research literature individually [19], there exists few formal studies of how these topics are related. Thus, grounded by the previous two sections, we propose the following research model (Figure 1) for our study.

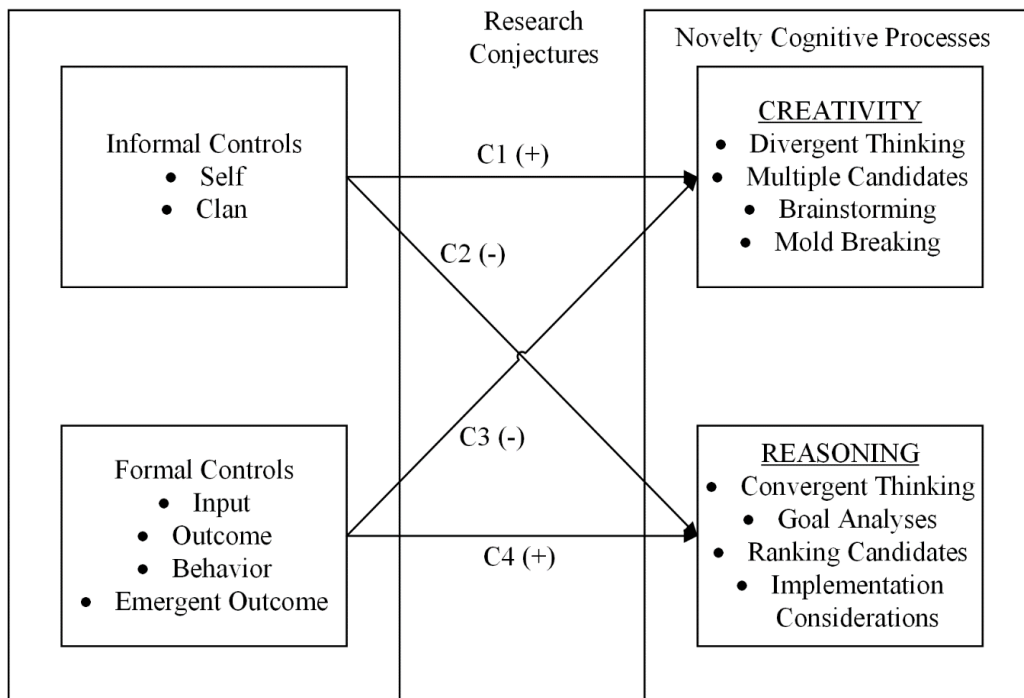


Figure 1: The relationship of controls and novelty cognitive processes

Four research conjectures are illustrated in the model:

**Conjecture 1:** Informal controls increase support for creativity in the design and development of novel application features on digital platforms.

**Conjecture 2:** Informal controls decrease support for reasoning in the design and development of novel application features on digital platforms.

**Conjecture 3:** Formal controls decrease support for creativity in the design and development of novel application features on digital platforms.

**Conjecture 4:** Formal controls increase support for reasoning in the design and development of novel application features on digital platforms.

Guided by these research conjectures, we perform a study to investigate the relationships of software controls and application novelty in software development projects on digital networks.

## 5 Research Methodology

This research studies the relationship of control mechanisms and the use of cognitive processes to produce novelty in platform-based applications. We conduct case studies to answer our research question. A case study methodology is appropriate when ‘how’ questions are posed in the research [25]. Case studies also allow us to extract a nuanced understanding of the control mechanisms identified by software development teams that contribute to novelty of the application. The unit of analysis is the project team that is developing the application on digital platform. To provide empirical grounding, we conduct two case studies.

### 5.1 Case Selection and Site Description

Selection of projects is driven by the following criteria: (a) the application is built on a digital platform, (b) stakeholder roles

(controllers and controlees) can be identified, (c) competing applications for the focal application exist, (d) novel features in the focal application are identifiable, and (e) ownership of application and platform are not held by same organization(s). Table 1 provides details about the case study sites, number of interviews at each site, informant roles in the interviews, and brief descriptions of the projects at the case study sites.

### 5.2 Data Collections

To test the efficacy of our selection criteria, interview protocols, and theoretical understandings, we performed two pilots [16]. Our first pilot location is an IT-department of a large public university in the Southeastern United States. The application under consideration allows universities to complete their reporting obligations for state-mandated requirements. The second pilot location is a Fortune-500 organization with a large development team of analysts, architects, and a project manager. This team is developing an application that supports online subscription of enterprise software. The application connects to multiple external platforms, increasing the complexity of the project. The pilots refined our protocol and data analysis methods.

The questioning protocol forms the basis for each interview with occasional deviations to accommodate any contemporary issues such as reordering questions based on an interviewee’s response or dropping certain questions that are not consistent with an interviewee’s role in the project. Follow up questions (not included in the protocol) may be included to seek clarification and/or reconfirmation. Finally, questions exploring interviewee’s role in the development project may be included to better understand the controls adopted by the team. Table 1 provides a summary of the two case study sites, named AT and TB. In case of AT, the majority of interviews took place on-site whereas a small fraction are individual online sessions. In case of TB, all the 7 interviews are individual online sessions.

Table 1: Summary of case study sites

Organization	Interviews	Informant Roles	Project Description
AT is a software consulting firm specializing in development, maintenance, and deployment of software applications across different industries.	9	<ul style="list-style-type: none"> <li>• Delivery Lead</li> <li>• Team Lead</li> <li>• Senior Developers</li> <li>• UI Designer</li> <li>• UI Developer</li> <li>• Technical Architect</li> </ul>	The client (a non-profit organization) wants to develop a mobile application (iOS-based) that would allow healthcare professionals to stream on-demand educational content, videos, support dynamic notetaking, and resume playback.
TB develops IT solutions to challenges in different domains such as CRM, Healthcare, and Operations	7	<ul style="list-style-type: none"> <li>• Product Owner</li> <li>• Product Manager</li> <li>• Practice Manager</li> <li>• Sales Consultant</li> <li>• Technical Architect</li> <li>• Solution Architect</li> <li>• Functional Consultant</li> </ul>	The product is a cloud-based Healthcare management application and competes with other offerings on Microsoft Azure platform. TB partners with select customers (hospitals) to develop features which are incorporated into the product – streamline patient care with CRM platform and consolidate patient care.



### 5.3 Data Analysis

To answer our research questions, we analyzed the qualitative data in two phases [18]. First, open coding systematically labeled our data to identify events, actions, and interactions. Second, axial coding related categories and subcategories around the data. We used independent coders with experience in software development projects to code our data. Coders were trained using the pilot interviews' transcripts. We discussed and clarified any ambiguity in conceptual understanding and operational definitions. Coders were blind to the research question. We performed within case analysis, followed by cross-case analysis. Our analysis focused on identifying key findings on the relationships between control mechanisms and the novel features of the software system under development.

## 6 Case Study Results

The two case studies supplied rich detail for qualitative analyses of the software development projects in the two organizations. We present the results of interviews with a focus on the control processes employed and the novel features produced in each software development project.

### 6.1 Case Study 1: AT

**6.1.1 Control Mechanisms.** We find use of different technical environments to facilitate experimentation with new ideas. Specifically, teams use *sandbox* environments to trial new ideas, *demo* environments to integrate new ideas with existing application, and *quality assurance* environments to test new ideas with existing application features; in addition to the *production* environment that hosts the actual application. Setting, maintaining, and transferring artifacts from such environments utilizes resources (time and cost). In case of AT, we find controls in the form of changes in team composition to facilitate development of novel features. Specifically, the video playback feature with positional saving and scrolling for AT's application is not supported by the platform's native capabilities and requires technical expertise. We find use of *collaborative sessions* aided by interactive mockups and designs to be a key behavioral control mechanism to identify and refine novel features for the application. We find different labels for these sessions: backlog refining/grooming, or brainstorming sessions (including a session with the project manager of a Fortune 500 organization).

We find support in the use of ongoing feedback during development of the application. The traditional conceptualization of ongoing feedback has a directionality from clients/users to the development team. However, we find that feedback may be bidirectional during and/or after iterations.

Table 2 summarizes formal control mechanisms in AT that contribute to the novelty of the application.

Regarding informal controls, we find evidence for the use of clan control in the AT project. Discussions with the client's liaisons pertaining to requirements are typically viewed as meetings that facilitate the project's understanding of the

domain. However, as part of these discussions, the team discusses alternative ways to either improve known requirements or recommend new requirements based on the team's prior experience. In case of AT, we see evidence of a clan mentality where entrepreneurial thinking [15] is encouraged so that the client is successful. However, we also see the negative influence of a "consultant" mentality. Specifically, AT's consultants realize that their role is limited to a module. Similarly, AT's leadership team acknowledges that decisions on features are made by the client based on time, cost, and desirability of recommended feature. Such client-vendor relationship may hinder identification and assimilation of novel features as AT may be wary to discuss potential features. The presence of experienced software developers on the AT project provides some evidence for individual self-controls.

Table 3 summarizes informal control mechanisms for AT project.

**6.1.2 Novel Features.** Novel features in AT project are threefold. First, the application allows its users to seamlessly stream content such as text, pictures, and videos on mobile devices. This requires dynamic adjustments to the content. The content is stored on the client's servers. Previously the native video playback feature from the platform lacked finesse. Second, the application enables users to make and retrieve notes while they are watching videos. Finally, the user can resume video playback from the last viewed location. These features set the focal application apart from competing applications on the platform and off-platform alternatives including a client's website. Some of these features exist in other domains. For example, resuming video playback from the last view position has been staple for video streaming application on the same platform. However, the ability to stream video content and extend platform's native capability is novel in the client's competition space. Table 4 lists the novel features of the AT application.

### 6.2 Case Study 2: TB

**6.2.1 Control Mechanisms.** At TB, the management team identified *springboard* clients ("early adopters" as noted by TB's Product Manager) that provided market needs and differentiators in exchange for access to the application. We classify partnerships with such clients as input control due to the emphasis by controller to leverage access to springboard clients and temporary association with such clients. One of TB's team members (title—Solution Consultant) is a registered nurse (the target users of TB's application) who participates in product demonstrations and identification and vetting of potential features. Inclusion of a team member who can provide users' perspective is another input control mechanism. TB teams use mock-up screens and designs so that all stakeholders can visualize potential features, alter designs to visualize focal feature, and identify approaches to incorporate potential features in the application. In addition to discussions based on interactive mockups.

TB's team also performs configurational changes to the

Table 2: Formal control mechanisms in AT project

Control Mode	Control Mechanisms
Input Controls	<ul style="list-style-type: none"> <li>• Change team composition (add and drop skills/personnel) to identify and develop novel features</li> <li>• Setup different technical environments to execute proof of concepts and integrate ideas in the application</li> </ul>
Behavior Controls	<ul style="list-style-type: none"> <li>• Facilitate workshops with users/clients at intervals</li> <li>• Interactions/feedback loops during iteration demo</li> <li>• Collaborative discussions/workshops between team and users with user interface mockups</li> <li>• Use technical capabilities to identify new features</li> <li>• Discuss technical approaches to achieve new features</li> </ul>
Output Controls	No evidence found
Emergent Outcome Controls	<ul style="list-style-type: none"> <li>• Feedback during and after each iteration</li> <li>• Ongoing feedback from application's usage data</li> <li>• Content and how it is served by the application</li> </ul>

Table 3: Informal control mechanisms in AT project

Control Mode	Control Mechanisms
Clan Controls	<ul style="list-style-type: none"> <li>• Shared understanding on success criteria</li> <li>• Prefer physical presence for client meetings whenever possible</li> </ul>
Self-Controls	<ul style="list-style-type: none"> <li>• Experienced software developers</li> </ul>

Table 4: Novel features of AT application

Novel Features
<ul style="list-style-type: none"> <li>• Allow users to seamlessly stream content such as text, images, and videos, on mobile devices. This requires dynamic adjustments to the content. The content is stored on the client's servers. The native video playback feature from the platform lacked finesse.</li> </ul>
<ul style="list-style-type: none"> <li>• Application enables users to make and retrieve notes while they are watching videos.</li> </ul>
<ul style="list-style-type: none"> <li>• User can resume video playback from the last viewed location.</li> </ul>

existing application and platform, where possible, to visualize new ideas for the application. We find identification of technology and tools to be made upfront which determines the scope boundaries for technical exploration. In case of TB, there exist organizational restrictions on the use of technology and tools provided by a vendor. We also find that choice of the digital platform introduces a major scope boundary for the team as technical capabilities and tools are bounded. The project team must identify alternatives that can be supported by resources within the scope boundary. We also find use of feedback mechanisms, part of emergent outcome control mode, to facilitate an individual's exploration of specific ideas. Table 5 summarizes formal control mechanisms in TB project.

In the case of TB, we find a shared understanding and importance of novelty across team members. The shared success criteria encourage the team to identify and vet alternatives. Also, we find that team members appreciate and

recognize the value contributed by other team members. For example, analysts recognize the possibilities and limitations faced by the technical team in implementing proposed features. To address limitations, analysts identify feasible alternatives and discuss with the technical team. Further, we find common consensus on the importance of certain processes and events. For example, meetings and discussions with potential customers is an important opportunity to verify and gather new feature ideas. For self-controls, proactive team members may experiment to identify novel features for the application. After implementation, typically as a proof-of-concept, other team members with closer market knowledge may adapt and integrate the novel feature. Given the shared understanding and importance on identification of novel features, the team is willing to discuss and improve any ideas put forth. Table 6 summarizes informal control mechanisms in TB project.

**6.2.2 Novel Features.** Novel features for TB's application are threefold. First, data management in the application is patient-centric whereas competitors use an event-centric approach. With a patient-centric approach, application's users can view all records for a patient on the dashboard. Second, user experience is highly rated. This includes the application's ease of use and performance. Third, easy integration with Microsoft's productivity suite that may be already functional at client's location. Another area of novelty for TB's team is the choice of platform. Microsoft's Azure platform integrates with Microsoft's productivity suite such as calendar, business intelligence reports, emails, and so on, allowing TB's application to differentiate itself from its competitors that use different platform.

Table 5: Formal control mechanisms in TB project

Control Mode	Control Mechanisms
Input Controls	<ul style="list-style-type: none"> <li>• Identify and partner with springboard clients</li> <li>• Choice of focal platform—platform’s features, ecosystem, and maturity—to distinguish the application from competition</li> <li>• Application’s user as part of the team</li> <li>• Technical members attend seminars and conferences hosted by the platform</li> <li>• Setup different technical environments to execute proof of concepts and integrate ideas in the application</li> </ul>
Behavior Controls	<ul style="list-style-type: none"> <li>• Configurational changes to platform before discussing features</li> <li>• Interactions/feedback loops during experimentation, testing, and documentation (within team)</li> <li>• Interactions/feedback loops with springboard clients</li> <li>• Collaborative discussions/workshops between team and users with user interface mockups</li> <li>• Use technical capabilities to identify new features</li> <li>• Discuss technical approaches to achieve new features</li> </ul>
Output Controls	No evidence found
Emergent Outcome Controls	<ul style="list-style-type: none"> <li>• Application is always ready for demo and feedback</li> <li>• Ongoing feedback from application’s usage data</li> <li>• Content and how it is served by the application</li> </ul>

Table 6: Informal control mechanisms in TB project

Control Mode	Control Mechanisms
Clan Control	<ul style="list-style-type: none"> <li>• Shared understanding on the importance of processes (for example, team visit to springboard client’s site) to identify novel features</li> <li>• Challenge team members to extend existing feature set</li> </ul>
Self-Control	<ul style="list-style-type: none"> <li>• Proactive team members try new ideas and discuss alternatives</li> </ul>

Table 7: Novel features of TB application

Novel Features
<ul style="list-style-type: none"> <li>• Data management in the application is patient-centric whereas competitors use an event-centric approach. With a patient-centric approach, application’s users can view all records for a patient on the dashboard.</li> </ul>
<ul style="list-style-type: none"> <li>• User experience is highly rated. This includes the application’s ease of use and performance.</li> </ul>
<ul style="list-style-type: none"> <li>• Easy integration with Microsoft’s productivity suite that may be already functional at client’s location.</li> </ul>

### 6.3 AT and TB case study findings

Summarizing AT and TB case study findings, we find compelling use of both formal and informal control mechanisms that lead to identification and assimilation of novel features which exceed the platform’s core proposition and/or differentiate the application from its competition. Specifically, we find use of a mixed control portfolio consisting of formal and informal controls. We do not find support for traditional outcome control mechanisms which we attribute to challenges in specifying novel outcomes a priori in the dynamic digital platform environment. Finally, we find positive influence of a

more long-term orientation of the team (TB case) in comparison to a short-term focus (AT case) which can be attributed to the perpetual mode of application under development in platform environments.

We identified well-defined novel features in the platform-based applications for each case study. In our interviews with the development team, we elicited the paths of divergent and convergent thinking that led to the novel features. Based on how these paths were influenced by formal and informal controls, we interpret these qualitative data and examine each of the four research conjectures in our research model.

**6.3.1 Informal Controls and Creativity.** We find convincing evidence for role of informal controls in increasing support for creativity in design and development of novel application features. For example, in case of AT, the novel feature of resuming video playback is introduced via individual creativity via self-controls. Specifically, challenges with platform’s technology did not lend itself to develop this feature. A senior programmer in AT’s team ran an experiment. This experimentation involves identifying multiple approaches (brainstorming) that can address the technological challenge (divergent thinking). In case of AT and TB, development teams create interactable mock-ups to facilitate discussions related to application’s features. Discussions based on these mock-ups help the team to identify new features and/or alter existing features (brainstorm) and identify new process flows (mold-breaking).

Clan control mechanisms such as shared understanding of the importance of novel features for the application, acknowledgement of challenges involved in identifying novel features, and a collective mindset to facilitate identification of novel features. In case of TB, proactive team members identify potential features that can extend existing features for the application. Together, informal control mechanisms help to identify multiple candidates (novel features) for the application.

To identify multiple candidates, informal controls facilitate a culture which emphasizes divergent thinking and builds tolerance for radical ideas.

**6.3.2 Informal Control and Reasoning.** We find evidence for the role of informal controls in decreasing support for reasoning in design and development of novel application features. For example, in case of TB, novel features related to user experience do not focus on implementation considerations (reasoning) at the onset. Instead, the team focuses on identifying multiple designs (creativity) without considerations of feasibility and implementation details, such as what can be supported by the platform's capability or the need to build new modules. In case of AT's novel feature of dynamic content display on different devices, AT's team considers divergent thinking (creativity) rather than convergent thinking (reasoning) as the team seeks to identify possible usage scenarios on devices of different size, capability, and software environment.

Following our earlier discussion on information controls and creativity, informal controls do not lend themselves to rank candidates, converge multiple ideas, or implementation considerations. Rather, informal controls seek to extend existing thinking and ideas to identify multiple candidates without considerations to implementation, feasibility, and priority. Reasoning-based processes may be time dependent in comparison to creativity promoted by informal controls. Whereas creative processes seek to identify novel features without any constraints of implementation, reasoning processes may be limited by time. For example, technological limitations may hinder a novel idea. In summary, informal controls increasingly support creativity whereas decrease support for reasoning.

**6.3.3 Formal Controls and Creativity.** We find evidence for the role of formal controls in decreasing support for creativity in design and development of novel application features. This conjecture reflects the long-standing concerns that inflexible software development processes constrain creativity. For example, in case of AT, a new team member was added (input control) to implement the novel video playback feature. This new role was specifically required to implement a given requirement within predefined implementation boundaries. In case of TB, discussion (behavior control) on application's features focuses on identifying implementable features (reasoning) rather than deriving a list of new options (creativity).

Formal control mechanisms focus on the means to accomplish the ideas identified by the team. Formal controls do not facilitate identification of multiple candidates. Although, some behavioral control mechanisms may facilitate brainstorming sessions for the team, these are often limited to discussions on feasibility based on cost and time. Formal control mechanisms aim to converge the development process such that specified deliverables are accomplished. Such a focus does not lend to divergent thinking and identification of multiple candidates.

**6.3.4 Formal Controls and Reasoning.** We find evidence

for the role of formal controls in increasing support for reasoning in design and development of novel application features. In case of TB, teams use feedback from customers and application usage data (emergent outcome control) to prioritize (reasoning) novel user interface features in the application. In case of AT, a time constraint introduced by an external source such as an application demo in a conference, invokes use of an emergent outcome control and behavior control to rank and converge (reasoning) novel feature alternatives, so that they can be demoed in the conference.

Formal control mechanisms facilitate prioritization of novel features identified in earlier iterations. Prioritization is often based on criteria such as deliverables, technical feasibility, impacts on existing application, tool support, available time, and costs. These processes are goal-driven that require evidence-based ranking of alternatives. For example, AT and TB teams use emergent outcome control mechanisms to specify technological boundaries and seek feedback on intermediate outcomes. These mechanisms are focusing on converging the broader novel feature that can be experienced in the application.

## 7 Discussion and Future Research Directions

In this research, we perform two rigorous case studies on platform-based application development projects in order to identify control mechanisms and novel application features on digital platforms. Based on analyses of the case studies, we find that informal and formal controls both contribute to platform innovations; however, in different cognitive ways. Further, we contribute to theory by presenting four conjectures for further study: (a) informal controls support increasing creativity, (b) informal controls support decreasing reasoning, (c) formal controls support decreasing creativity, and (d) formal controls support increasing reasoning.

In our case study research, we find empirical evidence in support of platform's role to enable both formal and informal controls. The role of a project manager to build a mixed portfolio of controls to maximize team members' contributions to application novelty is a key finding of this research. Platforms are a common point of reference during discussions, decision-making, and collaborative sessions. For example, as teams identify new features, a major focus is how to implement the potential features using the platform's current capabilities. The digital platform also facilitates high control communication and evaluation congruence. The platform plays the role of an anchor that is referenced to alter the control portfolio as the project evolves and to evaluate the current appropriateness of the portfolio. As new and/or updated platform offerings are visible to all, antecedents to control portfolio changes are visible and more likely to be accepted by the team.

In an organizational context, other control modes such as structure, market, and culture, have been considered in related domains [4]. However, software systems project control has been theorized to focus on "temporary organizations" that require different control activities [23] than the larger

counterparts of organizational control. In our analysis, we find evidence that challenges this notion of controlling a temporary organization. Applications developed on digital platforms may be perpetually in the state of development due to changes in market and platform.

Our study has two major implications for software project research. First, our findings align with related domains where control's effectiveness to introduce innovation has been established [4]. This finding calls for deeper studies of how controls and novelty relate in software development projects [6]. Second, this research addresses the recent call to investigate project controls in the digital era [24]. Specifically, we isolate the relationships of controls and novelty and perform a preliminary, qualitative study. Such a purpose-oriented focus allows us to investigate the required balance between value-appropriation concerns and value-creation requirements in the digital era.

We present four conjectures on control modes and novelty cognitive processes of creativity and reasoning. While additional empirical research is required to formally hypothesize and test these conjectures, we believe this research sets the stage for an increased understanding of the importance of incorporating cognitive processes in software engineering literature. Consideration of novelty cognitive processes is particularly important for software engineering because of the increasing importance of digital innovation [14]. Also, cognitive processes seek to focus individuals and teams to control portfolios that are more effective in software development projects. This research can complement future studies which incorporate organizational and team innovation literature in software engineering.

There are several important limitations and future research directions in our study. First, we consider only two small development projects with in-house applications. Future research can consider other project settings such as offshoring, large project teams, different application domains, and so on. Second, our findings are limited to projects where novelty is incremental. As we move in the digital era, one of the major challenges for future research on the impacts of control to novelty is to study projects that focus on radical innovation. Third, we did not explore the effect of platform's type on the project control—AT's platform caters to consumers whereas TB's platform caters to enterprises. Consequently, AT's platform has tight integration with products and services offered by the digital platform whereas TB's platform has tight integration with other platform-based services provided by the owners. The platform's coupling has consequences for the project as platforms feature updates, releases, and technology that are dependent on other services.

## References

- [1] T. Amabile, *How to Kill Creativity*, Harvard Business School Publishing, Vol. 87, 1998.
- [2] T. M. Amabile. and J. Pillemer, "Perspectives on the Social Psychology of Creativity," *The Journal of Creative Behavior*, 46(1):3-15, 2012.
- [3] N. Ambady and R. Rosenthal, "Half a Minute: Predicting Teacher Evaluations from Thin Slices of Nonverbal Behavior and Physical Attractiveness," *Journal of Personality and Social Psychology*, 64(3):431, 1993.
- [4] L. B. Cardinal, "Technological Innovation in the Pharmaceutical Industry: The Use of Organizational Control in Managing Research and Development," *Organization Science*, 12(1):19-36, 2001.
- [5] C. E. H. Chua, W. K. Lim, C. Soh, and S. K. Sia, "Enacting Clan Control in Complex IT Projects: A Social Capital Perspective," *MIS Quarterly*, 36(2):577-600, 2012.
- [6] W. A. Cram, K. Brohman, and R. B. Gallupe, "Information Systems Control: A Review and Framework for Emerging Information Systems Processes," *Journal of Association for Information Systems*, 17(4):216-266, 2016.
- [7] P. Criscuolo, A. Salter, and A. L. Ter Wal, "Going Underground: Bootlegging and Individual Innovative Performance," *Organization Science*, 25(5):1287-1305, 2014.
- [8] A. Dietrich and R. Kalso, "A Review of EEG, ERP, and Neuroimaging Studies of Creativity and Insight," *Psychological Bulletin*, 136(5):822, 2010.
- [9] K. M. Eisenhardt, "Control: Organizational and Economic Approaches," *Management Science*, 31(2):134-149, 1985.
- [10] A. Fink, M. Benedek, R. H. Grabner, B. Staudt, and A. C. Neubauer, "Creativity Meets Neuroscience: Experimental Tasks for the Neuroscientific Study of Creative Thinking," *Methods*, 42(1):68-76, 2007.
- [11] B. Fitzgerald, "Formalized Systems Development Methodologies: A Critical Perspective," *Information Systems Journal*, 6(1):3-23, 1996.
- [12] J. P. Guilford, "Creativity: Yesterday, Today and Tomorrow," *The Journal of Creative Behavior*, 1(1):3-14, 1967.
- [13] M. L. Harris, R. W. Collins, and A. R. Hevner, "Control of Flexible Software Development Under Uncertainty," *Information Systems Research*, 20(3):400-419, 2009.
- [14] A. Hevner, and S. Gregor, "Envisioning Entrepreneurship and Digital Innovation through a Design Science Research Lens: A Matrix Approach," *Information & Management*, 2020.
- [15] A. Hevner and O. Malgonde, "Effectual Application Development on Digital Platform," *Electronic Markets*, 29(3):407-421, 2019.
- [16] O. Malgonde, *An Effectual Approach for the Development of Novel Applications on Digital Platforms*, University of South Florida, 2018.
- [17] Y. Masoudnia and M. Szwajczewski, "Bootlegging in the R&D Departments of High-Technology Firms," *Research-Technology Management*, 55(5):35-42, 2012.
- [18] M. Miles, A. M. Huberman, and J. Saldana, *Qualitative Data Analysis: A Methods Sourcebook*, SAGE Publications, 2013.
- [19] W. J. Orlikowski, "Integrated information environment or matrix of control? The contradictory implications of

information technology,” *Accounting, Management and Information Technologies*, 1(1):9-42, 1991.

- [20] G. G. Parker, M. W. Van Alstyne, and S. P. Choudary, *Platform Revolution*, W. W. Norton & Company, 2016.
- [21] M. A. Runco, *Creativity: Theories and Themes: Research, Development, and Practice*, Elsevier 2014.
- [22] S. Sarasvathy, “Causation and Effectuation: Toward a Theoretical Shift from Economic Inevitability to Entrepreneurial Contingency,” *The Academy of Management Review*, 26(2):243-263, 2001.
- [23] M. Wiener, M. Mähning, U. Remus, and C. Saunders, “Control Configuration and Control Enactment in Information Systems Projects: Review and Expanded Theoretical Framework,” *MIS Quarterly*, 40(3):741-774, 2016.
- [24] M. Wiener, M. Mähning, U. Remus, C. Saunders, and W. A. Cram, “Moving IS Project Control Research into the Digital Era: The 'Why' of Control and the Concept of Control Purpose,” *Information Systems Research*, 30(4):1387-1401, 2019.
- [25] R. Yin, *Case Study Research: Design and Methods*, SAGE Publications, 2008.
- [26] Y. Yoo, O. Henfridsson, and K. Lyytinen, “The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research,” *Information Systems Research*, 21(4):724-735, 2010.



**Onkar S. Malgonde** is an Assistant Professor at the Operations Management & Information Systems, College of Business, Northern Illinois University. He received a B. Engg in Information Technology in 2010 from University of Pune and a PhD in MIS from the University of South Florida in 2018. His research interests include software engineering, machine learning, and digital platforms. His research has been published in *MIS Quarterly*, *Empirical Software Engineering*, *Electronic Markets*, *Workshop on Information Systems and Technology (WITS)*, *International Conference on Design Science Research in Information Systems and Technology (DESIRST)*, and *Americas Conference of Information Systems (AMCIS)*.



**Alan R. Hevner** is a Distinguished University Professor and Eminent Scholar in the Information Systems and Decision Sciences Department in the Muma College of Business at the University of South Florida. He holds the Citigroup/Hidden River Chair of

Distributed Technology. Dr. Hevner's areas of research interest includes design science research, digital innovation, information systems development, software engineering, distributed database systems, and healthcare systems. He has published over 250 research papers on these topics and has consulted for a number of Fortune 500 companies. Dr. Hevner received a Ph.D. in Computer Science from Purdue University. He has held faculty positions at the University of Maryland and the University of Minnesota. Dr. Hevner is a Fellow of the American Association for the Advancement of Science (AAAS), a Fellow of the Association for Information Systems (AIS), and a Fellow of IEEE. He is a member of ACM and INFORMS. Additional honors include selection as a Parnas Fellow at Lero, the Irish software research center, a Schoeller Senior Fellow at Friedrich Alexander University in Germany, and the 2018 Distinguished Alumnus award from the Purdue University Computer Science Department. From 2006 to 2009, he served as a program manager at the U.S. National Science Foundation (NSF) in the Computer and Information Science and Engineering (CISE) Directorate.

# Preprocessing Techniques' Effect On Overfitting for VGG16 Fast-RCNN Pistol Detection

Jiahao Li\* Charles Ablan\* Rui Wu† Shanyue Guan\* Jason Yao\*  
East Carolina University, Greenville, NC 27858 USA

## Abstract

Within the past two decades, gun detection became an increasingly popular research topic as gun violence continued to threaten public safety. Of all machine learning algorithms employed to identify weapons, Convolution Neural Networks (CNN) stood out as the most robust method for identifying guns in images. Although CNN had outstanding image classification performances, it is not without limitations. A CNN without large quantities of data suffers from overfitting. While complex architectures reduce overfitting, it also results in slower detection speed and increased memory usage. This study analyzed three image preprocessing techniques' effect on reducing overfitting in VGG16 Fast Regional Convolutional Neural Network (F-RCNN) without modifying network architectures. The base VGG16 was trained with transfer learning in MATLAB on a dataset of 1500 selected images to artificially induce overfitting. The average testing precision of the base VGG16 detector was then compared with the results of other VGG16 detectors supplemented with image processing techniques. The three image processing techniques used are color contrast enhancement, principle component analysis (PCA), and combined preprocessing methods. The study concluded that color contrast enhancement had the greatest impact on reducing the effects of overfitting. It was found that with proper levels of color contrast enhancements, the average testing precision went up noticeably. The PCA supplemented model failed to reduce the number of irrelevant features and did not retain the important features. The PCA method proved to be ineffective in reducing overfitting and resulted in an overall loss of average precision. The combined preprocessing methods combined the images of both PCA and color contrast enhancements into two different training datasets. The first dataset combined PCA with color enhancements and the second only combined color enhancement results. Both combined preprocessing methods did not increase the average precision potentially due to conflicting features.

**Key Words:** VGG16; convolution neural networks; image preprocessing techniques; gun detection.

## 1 Introduction

Past breakthroughs in Convolutional Neural Networks (CNN) using VGG16 and VGG19 architectures achieved an incredible 90 percent accuracy in image classification [3, 23]. As the search for better neural networks continues, increasing complexity becomes unavoidable [3]. A CNN's classification accuracy often increases with layer complexity, however, increasing complexity also causes slower detection speeds and increased memory usage [6]. The new 1000+ layered Inception V networks are strong evidence that CNN is getting increasingly complex at the expense of memory and speed [26]. Maintaining the simplicity of the network architecture while achieving high identification precision becomes a growing concern in recent years [24]. Of the existing neural networks, VGG16 remained one of the simplest yet robust neural networks ever created. This study applied a Fast-Regional Convolutional Neural Network (F-RCNN) model that modified a VGG16 net into an F-RCNN object detector for pistol detection [7]. A base VGG16 net was trained using transfer learning with an original blend of 1500 ground truth images. The base VGG16 was trained within MATLAB to achieve an overall true positive percentage of over 98 percent on pistol detection for the training dataset. Such high levels of training accuracy are often a sign of overfitting [19]. By running the precision versus recall test [8], it was revealed that the base detector had a low average precision. This study focused on determining viable image preprocessing techniques that can address the overfitting problem and increase network performance without modifying the internal architecture.

This experiment utilized MATLAB to both train the F-RCNN object detector and implement various image preprocessing techniques. MATLAB was used because the Deep Learning toolbox is easy to implement with robust support for various neural networks [11, 25]. The built-in conversion function of images to matrix form made image processing easy and efficient due to various image processing techniques requiring matrix transform and matrix transpose [16, 17]. Three image pre-processing trials were applied to the training dataset for the VGG16 F-RCNN pistol detector in hopes of raising average testing precision. A neural network performs best if trained with great variance in the pose and lighting of an object [29]. The image preprocessing techniques applied in this study operated under the assumption that pose and lighting are key features that greatly affect the

\*Department of Engineering. Email: [lij17@students.ecu.edu](mailto:lij17@students.ecu.edu), [ablanc17@students.ecu.edu](mailto:ablanc17@students.ecu.edu), [guans18@ecu.edu](mailto:guans18@ecu.edu), [yaoj@ecu.edu](mailto:yaoj@ecu.edu)

†Department of Computer Science. Email: [wur18@ecu.edu](mailto:wur18@ecu.edu)

precision of any object recognition system [29]. The neural network's tendency of relying on color and lighting makes it challenging to distinguish dark simple objects from lowlighting backgrounds [18]. To resolve the problem of low color contrast with the background, the first trial used color enhancement techniques at set intervals to widen the gap between dark and light-colored regions. The second trial used PCA analysis to reduce the background while retaining most of the pistol's features [10, 22]. After the results of the first two methods were obtained, a fusion of the two modified image datasets along with the original dataset was used to train two combined VGG16 F-RCNN detectors.

In the rest of this paper, Section 2 describes the related works in the area of pistol detection and image preprocessing. Section 3 describes the dataset used in the study as well as the source images used. Section 4 describes the performance of the VGG16 F-RCNN detector on the base dataset. Section 5 describes the effects of various color enhancement trials on detector precision. Section 6 describes the effect of PCA analysis on detector performance. Section 7 describes the effect of combining the different image preprocessing techniques into one dataset. Section 8 discusses the results of the previous experimental trials. The conclusion is given in Section 9.

## 2 Related Works

Pistol detection has been a widely researched topic ever since Neural Networks first gained popularity with Alex-net in 2012 [12]. Many studies in the past have tackled the problem of handgun detection using Regional Proposal Networks and transfer learning from established neural networks such as VGG16 [2, 18]. Two methods are generally used to improve the performance of specific object detectors. The first method is modifying the neural network architectures to become deeper and more robust [26], and the second method is applying image preprocessing techniques to help detectors better separate the important features from background noise [21]. This paper tackles the problem of using image preprocessing techniques to increase the performance of a VGG16 F-RCNN pistol detector.

### 2.1 Pistol Detection

Pistol detection differs from pistol classification in that object detectors must identify the part of the image with the weapon [5, 13-14, 18]. Various pistol detectors have been trained in the past using both neural networks and traditional machine learning techniques such as SVM(Support Vector Machine) and Histogram of Oriented Gradients (HOG) [18, 27]. However, the best performances came from using Regional Proposal Networks(RCNN) and its variants the FAST-RCNN and Faster-RCNN [5, 18].

Akçay et al. [2] used a variety of neural networks with different types of object detectors to detect pistols in x-ray images. Five different object detection models were used in the study including Sliding Windows Convolutional Neural

Network(SW-CNN), Regional Convolutional Neural Network (RCNN), Faster Regional Convolutional Neural Network (Faster-RCNN), Region-based Fully Convolutional Networks (R-FCN) and You Only Look Once object detectors (YOLOv2). The object detection models were trained with Alexnet, VGG16, VGG19, and residual neural networks (ResNets). The resultant data shows that both RCNN and Faster-RCNN outperformed both traditional handcrafted Bag of visual words(BoVW) features and fellow SW-CNN detectors. Akçay et al. [2] have proved with their study that it is possible to achieve high object detection precision with R-CNN models and their variants.

Olmos, Tabik, and Herra [18] analyzed a VGG16 based Faster- RCNN detector for video pistol detection with limited success. The Faster-RCNN's large false positives rates drastically lowered the overall detector precision. The researchers contributed the high false positives ratio to low contrast and luminosity of certain video frames. The pistols that are not clearly distinguished from the backgrounds are often missed and other objects are falsely identified.

### 2.2 Neural Network Image Preprocessing

Within the field of image analysis, image preprocessing techniques are frequently used in combination with a variety of image classification algorithms [5, 13, 28, 30]. Many image preprocessing techniques aimed to either reduce noise or enhance desired features. Rehman et al [21] discussed a variety of image preprocessing methods for character recognition using neural networks. Within the various preprocessing techniques discussed, the researchers highlighted the importance of thresholding in image processing. Thresholding sets a boundary on the original color scheme from which the image is converted to binary or grayscale. Threshold simplifies the image and highlights the desired characters making it an important preliminary image processing technique. The other image preprocessing techniques included the elimination of unwanted features and extraction of key features. The importance of image preprocessing techniques was highlighted by Rehman et al [21] and its effect in increasing neural network performances cannot be overlooked.

## 3 Dataset

This study utilized 2,000 open source images from various image datasets under an academic license as well as original images taken by the research team. Out of a training dataset of 2,000 images, 1,500 were randomly selected for training and 500 were used for testing. The first dataset used was the IMFDdb online firearm dataset, a free gun image dataset that contains a variety of pistols, rifles, shotguns, and other firearms [9]. The second dataset used was the Sci2s weapons dataset from the University of Granada [5, 18]. For this experiment, only pistols were selected from a variety of movie images. The selection criteria for the combined image dataset



included a variety of background and contextual information at various backgrounds, angles, and distances for a balanced selection of various pistol images. The combined image dataset was biased towards images with a variety of background and contextual information for accurate object recognition [20]. The limited number of training images meant a larger chance for the network to overfit, while a diverse spread of gun sample images helped the detector to maintain an acceptable testing precision.

#### 4 Base VGG16 F-RCNN Performance

This study was conducted on the basis that large false positives are generated often due to the overfitting of neural networks [19]. Where a model trained to recognize features in the training dataset might fail to generalize features onto the testing dataset resulting in low testing identification rates. To fully express the influences of various color enhancement methods on overfitting, this research focused on simple VGG-16 nets trained with limited images. This was meant to introduce overfitting in the model and compare the effect of various preprocessing methods on reducing overfitting. Within MATLAB, a pre-trained VGG16 net was trained with the 1,500 pistol images. The VGG16 object detector reached a training accuracy of 98.8 percent after 5 epochs. The trained detector was then tested on the 500 testing images with an average precision of 0.2138. The low precision likely resulted from boxing errors where either a part of the gun or too much of the background was selected. However, upon physical examination, it was found that although the detection boxes did not precisely match the ground truth labeling, over 90 percent of the highest confidence detection box correctly identified most of the weapons and did not mistake other objects for pistols. Thus, the overall accuracy of the base F-RCNN object detector was found to be in an acceptable range. Figure 1 shows the MATLAB’s precision evaluation of the VGG16 during testing. Figure 1 shows the performance graph plotted for precision versus recall, where recall and precision are ratios of true positive instances to the sum of true positives and false negatives in the detector, based on the ground truth [2, 18]. By running through each image and visually judging the accuracy of the neural net, three different category metrics were used to determine true the performance of a neural net performance. i) True Positive means that the network has correctly identified the weapon with an appropriate bounding box. ii) False Positive means that the network has failed to select most of the weapon or has mistaken something else for the pistol. iii) Negative means the Neural net has failed to recognize guns in the image with high confidence. The results of the visual analysis will change slightly from person to person and trial to trial, but five successive repetitions on the testing dataset prove the general effectiveness of the trained F-RCNN object detector. Table 1 shows the classification criteria for accuracy evaluation and the average performance of the VGG16 detector over the five repetitions.

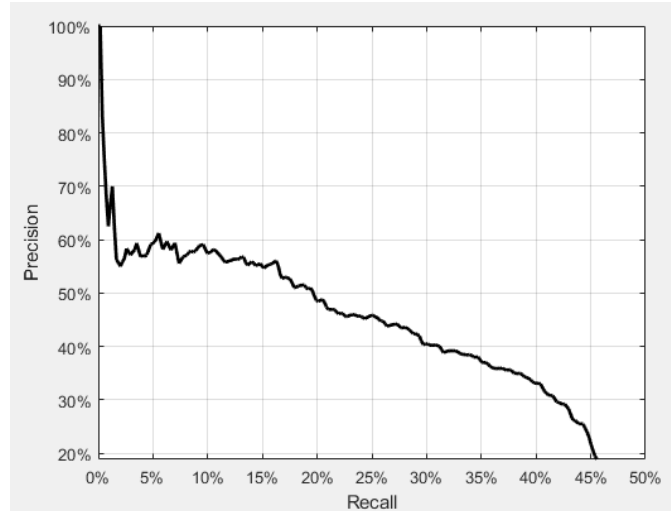




Figure 1: VGG16 precision vs recall on 500 images

Table 1: VGG16 judgment criteria and percentage number

True Positive	False Positive
	
453/500	47/500



#### 5 Fixed Ratio Color Enhancement Trial

The color enhancement preprocessing techniques operate under the assumption that neural nets rely heavily on color and texture for target identification [1, 18]. Since most pistols have a darker color as opposed to their surroundings, by increasing the color contrast the neural network should be able to better

distinguish dark pistols from light backgrounds to avoid false positives. A flowchart of the color enhancement process is shown in Figure 2.

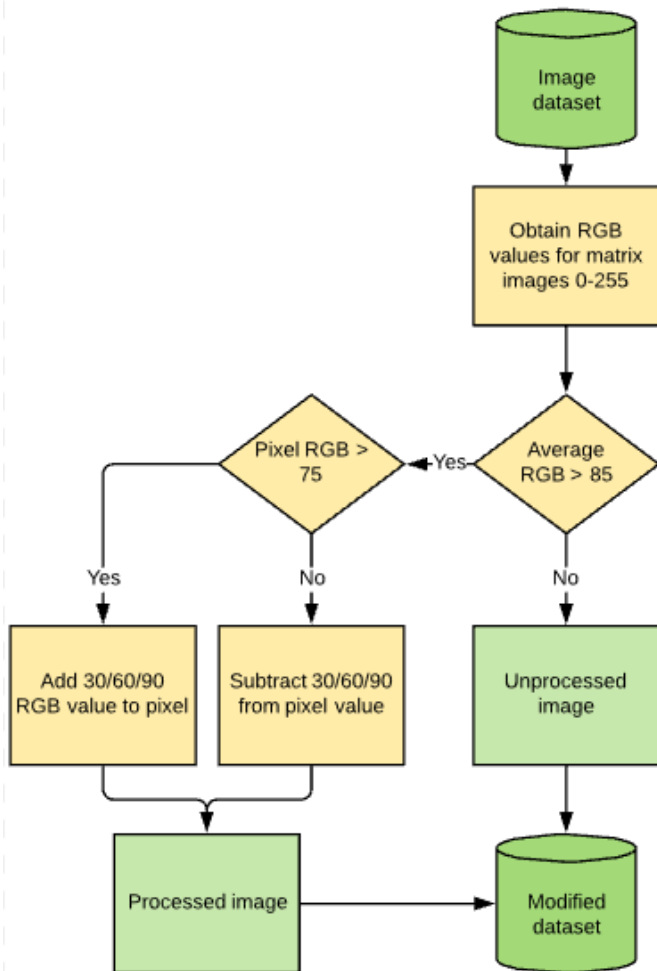


Figure 2: Color enhancement trial flowchart

The first preprocessing method separated light and dark colors at 75 out of 255 for all three RGB color schemes. The 75 separation value is selected because it separates the darker gun color scheme from the lighter background. The parts of the image with an RGB value less than 75 are darkened by -30 and the parts with a number larger than 75 are lightened by +30. The 30 enhancement value was chosen because initial trials indicated that any value less than 30 produced had no significant impact on the testing precision. If a particular pixel has an RGB value of less than 30 it will simplify to zero after preprocessing and the same concept applies to values greater than 225. The images with a mean RGB value of 85 or below were deemed too dark for the pistol to be separated from the background and thus the color enhancement was not performed. Only the training dataset underwent this preprocessing technique and the testing results on the original testing dataset are shown in Figure 3. In the rest of this paper, the “x” of “enhanced-x” refers to the specific RGB value that is modified in the image.

From Figure 3 it can be observed that the enhanced-30 dataset did not result in significant improvements in MATLAB recorded average precision from 0.2138 to 0.2208. Since initial trials with the color enhanced-30 showed only slight improvements in precision, the second trial doubled the modification ratio to +60 and -60. From Figure 3 it is observed that the enhanced-60 trial noticeably increased the average precision from 0.2138 to 0.2944 for a 38 percent increase in precision. Based on the success of the enhanced-60 trial, the enhanced-90 trial further increased the contrast to +90 and -90 in an attempt to achieve higher precision. As it can be observed from Figure 3, overly increasing the color enhancement ratio to +90 and -90 negatively impacted detector performance decreasing the average precision to only 0.0759.

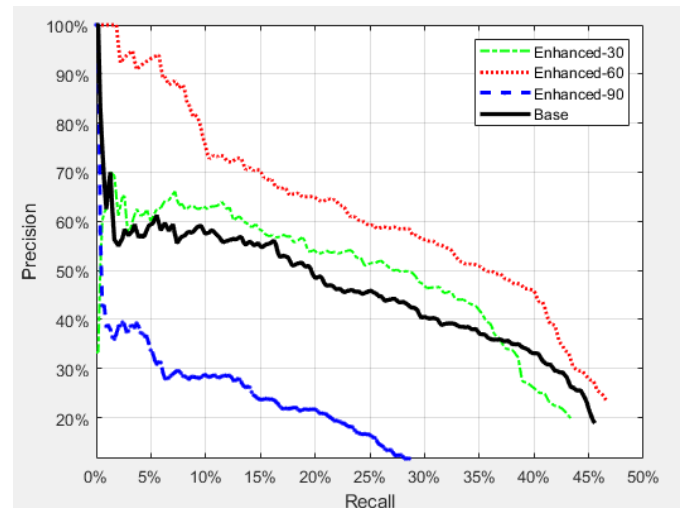


Figure 3: Color enhancement trial performance

Table 2 shows the full comparison of the effect of different color enhancement ratios on training images. It can be seen from the color-enhanced images that the pistol has a drastically higher color contrast as opposed to the background. By using color enhancement techniques, the background noise is also whitewashed resulting in an overall simpler image.

### 5.1 Varied Ratio Color Enhancement Trial

After initial promising results with the enhanced-60 dataset, new color enhancement trials were conducted. The new color enhancement trials focused on analyzing the effect of changing threshold values and enhancement ratios had on average precision. The result of the extended tests is shown in Table 3. FR stands for Fast R-CNN model trained and the number after the “-” symbol stands for various color enhancement ratios used with a threshold of 75. For FR-60/50, FR-60/100, and FR-60/125 the number after the “/” symbol indicates the threshold used to separate light and dark regions. Pixel values below the low threshold are considered light regions. Pixel values above the high threshold are considered dark regions. Low ratios are applied to light regions and high ratios are applied to dark

Table 2: Original and color-enhanced images

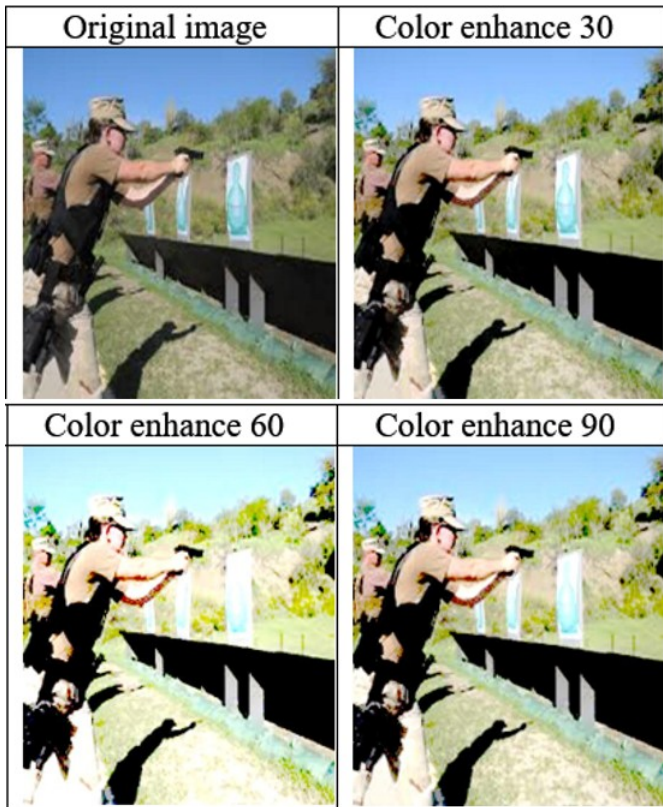


Table 3: Color-enhancement Trials.

Trials	Low Threshold	High Threshold	Low Ratio	High Ratio	Average Precision
Base	N/A	N/A	N/A	N/A	0.2138
FR-30	75	75	30	30	0.2208
FR-60	75	75	60	60	0.2944
FR-90	75	75	90	90	0.0759
FR-60/50	50	50	60	60	0.1158
FR-60/100	100	100	60	60	0.1569
FR-60/125	125	125	60	60	0.0395
FR-0.1	75	N/A	0.1	N/A	0.2852
FR-0.5	75	N/A	0.5	N/A	0.2386
FR-1.5	N/A	180	N/A	1.5	0.1299
FR-1.9	N/A	180	N/A	1.9	0.1725
FR-0.1-1.9	75	180	0.1	1.9	0.0015
FR-0.3-1.7	75	180	0.3	1.7	0.1077
FR-0.5-1.5	75	180	0.5	1.5	0.1503
FR-0.7-1.3	75	180	0.7	1.3	0.1755
FR-0.9-1.1	75	180	0.9	1.1	0.1822

regions; where integer values indicate subtraction and decimal values indicate multiplication. The average precision shows the model performance.

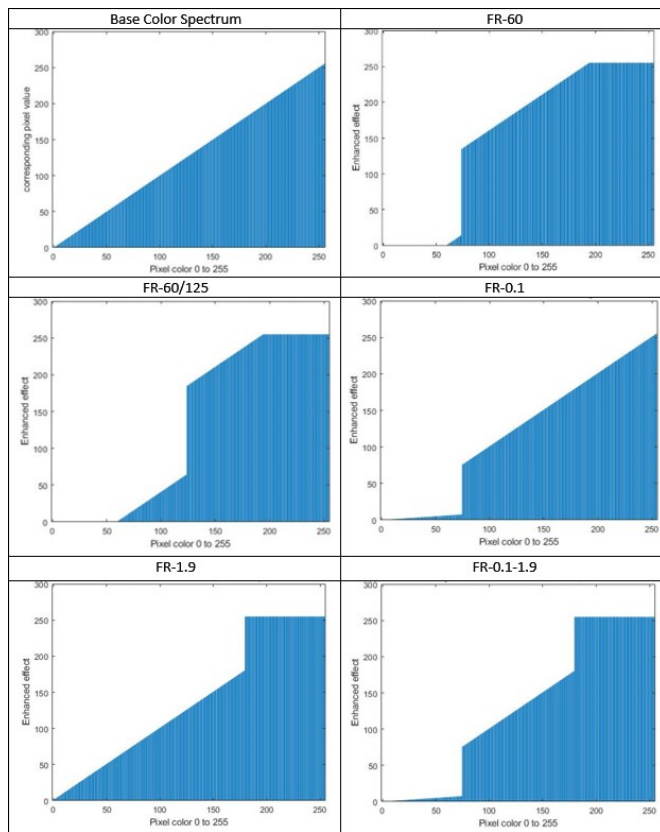
The previous enhanced-60 model used a fixed threshold of 75 to differentiate between light and dark colors. To test for the effect on changing the threshold, new tests were conducted for thresholds at 50, 100, and 125 with the same enhancement ratio of +60 and -60. From Table 3, it can be observed that both decreasing and increasing the threshold has a detrimental effect on model performance. However, by comparing the three new models, FR-60/125 had the most detrimental effect on accuracy with a -0.1743 or 81 percent decrease in average precision. Overall, the new trials indicate that changing the thresholds too far up or down can have a detrimental effect on performance.

The second variable that changed was the enhancement ratio. While enhancement-30, 60, and 90 had fixed changes to pixel values, new tests incorporated varying changes based on the old pixel value. The new trials multiplied the original pixel values with a ratio between 0.1 - 0.9 for value reduction, and 1.1 to 1.9 for value increase. Out of the eight new models, FR-0.1 and FR-0.5 tested for the effect of only enhancing the light portions of the image. FR-1.5 and FR-1.9 tested for the effect of enhancing the darker sections of the image. Trials FR-0.1-1.9 to FR-0.9-1.1 tested for the effect of increasing both the dark and light contrasts. Out of the eight new trials, FR-0.1 had the

highest increase in precision of 34 percent to 0.2852 followed by an 11 percent increase for FR-0.5 at 0.2386. In contrast, FR-1.5 and FR-1.9 both had detrimental effects on precision at 0.1299 and 0.1725. From the single ratio trials, it can be judged that that enhancing the lighter part of the image is more beneficial than enhancing the darker parts. The last four rows of 3 show the result of the multiple ratio trials. The FR-0.9-1.1 had the highest precision and FR-0.1-1.9 had the lowest precision. The average precision of the "FR-X-X" trials followed a distinct trend, where larger enhancement ratios are related to decreases in performance. The combined result of new color enhancement ratio trials suggested that lighter region enhancements have the best result, where dark value enhancements likely resulted in a loss of important pistol features.

Table 4 shows the effect of various color enhancement trials on pixel values. The X-axis is the original pixel values from 0 to 255. The Y-axis is the modified pixel values. The base color spectrum shows the unmodified pixel graph where the X and Y-axis follows a linear one-to-one correlation. Both FR-60 and FR-0.1 both outperformed the base model, and from Table 4 it can be observed that both had a noticeable drop in values less than 75. FR-60/125 had the most dramatic effect

Table 4: Color spectrum comparison chart



on the color spectrum, which could correlate to loss of important features resulting in poor performance. Both FR-1.9 and FR-0.1-1.9 underperformed against the base model and both had a noticeable rise in darker color regions. Overall, it can be inferred from Table 4 that lighter region enhancements without sharp rises in darker regions offer the best enhancement results.

## 6 PCA Feature Reduction

Each layer of a Convolutional Neural Network retains a specific feature of the image [6, 26]. Because of the curse of dimensionality, some researchers showed that fewer features can be less misleading for a machine learning model. PCA feature reduction aimed to reduce the number of features the neural net was exposed to. PCA compresses the data and retains principle features in the original image [10, 22]. The PCA works substantially better for binary images that have only have two dimensions as opposed to the three dimensions of RGB images [15]. While there are ways to perform PCA for RGB images using multi-linear subspace learning algorithms [15], performing PCA on binary images often produces cleaner results. However, binary PCA reduced images cannot take on color as the principle components are analyzed as 1 and 0 inputs. To restore the original color to the PCA image, MATLAB was used to fuse the original image with the PCA binary reduction image. The resultant color scheme differs from the original

image because of the merging process. However, most of the color features are retained through this method.

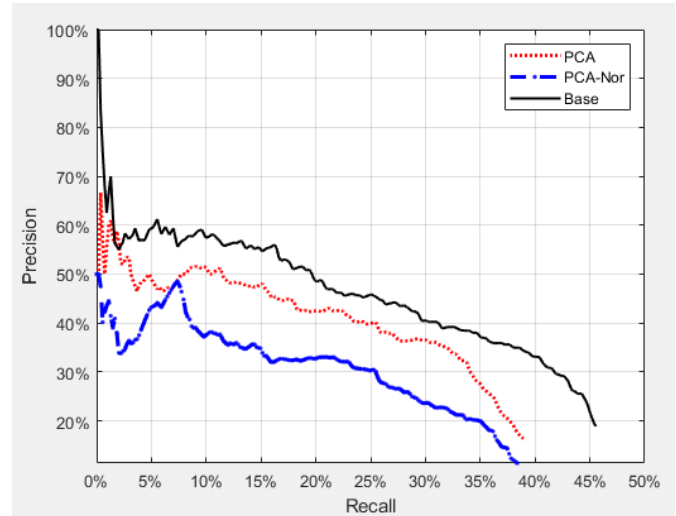
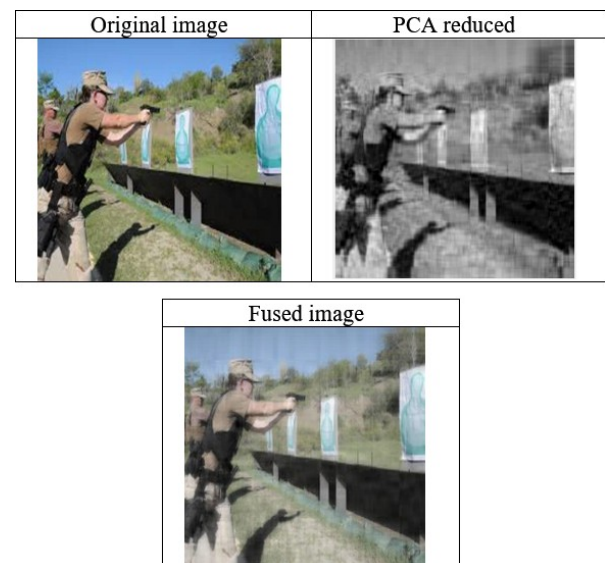


Figure 4: PCA detector performance

Table 5 shows the comparison between the original image, PCA reduced image, and fused image used for analysis. Figure 4 shows the performance of the VGG16 net trained with PCA reduced images on unmodified test images. Figure 4 also shows the performance of the PCA trained net on PCA reduced test images. From Figure 4, it can be observed that the PCA reduced detector tested with the original testing dataset resulted in a significant decrease in average precision from 0.2138 to 0.1223. In comparison, the PCA reduced detector tested with PCA reduced testing dataset resulted in a less significant decrease in average precision from 0.2138 to 0.1636. This disparity between the test results can be

Table 5: PCA sample images



attributed to the loss of features between original and PCA datasets.

## 7 Combined Dataset

For a neural network to retain information about an object from various angles and lighting conditions, a great variance in the training dataset is required [4]. While individual approaches can yield no conclusive results towards increasing accuracy, combining the various approaches into a single dataset can increase data variations without requiring additional images. The first combined dataset used a combination of the color enhance-30, color enhance-60, PCA analysis, and original images to form the 1500 training dataset. The second combined dataset used a combination of the three-color enhance trials and original images. Since color enhance-60 yielded the best results, the combined dataset uses 750 images from the color enhance-60 dataset, 250 images each from color enhance-90, color enhance-30, and the original images. Figure 5 shows the combined F-RCNN performance.

From Figure 5 it can be observed that combining PCA reduced images with color-enhanced images had no statistically significant changes in performance as the average precision raised from 0.2138 to 0.2398. From Figure 5 it can also be observed that mixing different color enhancement ratios significantly decreases the average precision to 0.1524.

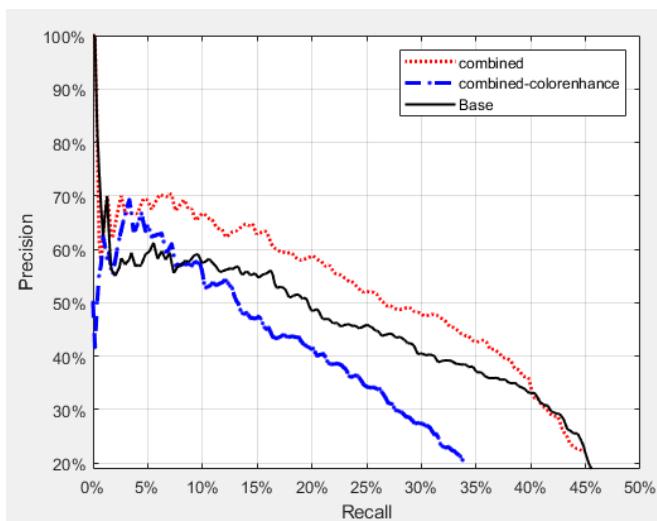


Figure 5: Combined detector performance

## 8 Discussion

Of all the common image preprocessing techniques used in machine vision, three methods were chosen to address the overfitting problem of the base detector. Color enhancement was chosen to address target and background separation. Its flexible nature with many parameters makes it easily transferable from detecting pistols to other objects of interest. Secondly, PCA was introduced to reduce the number of

overlapping features thus lowering the chance for overfitting. Finally, the combined approach focused on merging the two techniques to produce high feature variations.

Three different fixed color enhancement ratios were used at intervals of 30, 60, and 90 with a 75 threshold. Figure 3 shows color enhancement-30 slightly increased the average precision by 0.007 from 0.2138 to 0.2208, which was not statistically significant. The second trial increased the enhancement ratio to 60 which resulted in a noticeable rise in average precision by 0.0806, from 0.2138 to 0.2944. Although this result is not a significant change in precision, it shows that basic color enhancement can mitigate the background noises associated with low lighting environments. The third trial increased the enhancement ratio to 90 and resulted in a significant drop in average precision of -0.1379 from 0.2138 to 0.0759.

To further study the extensive effect of color enhancement, new trials were conducted using various thresholds and enhancement ratios. From the results of Table 3, it can be seen that the varying ratio trials indicated a strong correlation between color contrast and network performance. Since the color enhancement-60 trials yielded the greatest increase in precision, the new trials first aimed at adjusting the 75 percent threshold. FR-60/50 lowered the threshold to 50 and saw a 46 percent loss of precision which is surprising due to the success of the color enhancement-60 trials. FR-60/100 and FR-60/125 then had a more detrimental effect on average precision with FR-60/125 having an 82 percent loss of precision. The failure to raise average precision despite using a tested enhancement ratio suggests that there is an optimal threshold that separates light and dark regions for each database, and deviations from the threshold can lead to detrimental results. FR-60/50 enhanced made 25 pixels in the light region darker when compared to the original color enhancement-60 trials, and this difference contributed to the loss of information. By adjusting the threshold to 100 and 125, more regions are classified as light regions and had their values decreased, also leading to a loss of information. At this point it is unclear why 75 presumably works for this dataset, however, it can be hypothesized that only regions with color values less than 75 are lighter regions that require light enhancement to reduce background noise. Therest of the varying ratio trials will continue to adopt 75 as the low threshold since it provided the best results. The high threshold was chosen at 180 because it was 75-pixel values lower than 255. The varied ratio trials multiplied each pixel value by a ratio instead of adjusting every pixel by the same amount. The difference between the fixed and varying ratio adjustments can be observed in Table 4.

Of all the varied ratio trials, FR-0.1 showed a 34 percent increase in average precision to 0.2852. A follow-up experiment with FR-0.5 showed a less significant increase of 11 percent in average precision to 0.2386. The two trails suggest that large lighter region enhancements can have positive effects on testing precision. FR-1.5 and FR-1.9 were then tested for the effect of enhancements on darker regions. However, FR-1.5 resulted in a 39 percent decrease in average precision and FR-1.9 resulted

in a 19 percent decrease in average precision. The two dark enhancement results suggest that enhancement of dark regions correlates with a small loss of precision. Trials FR-0.1-1.9 to FR-0.9-1.1 tested the combined effect of both enhancing the lighter and darker regions. Although all four trials resulted in a loss of average precision, there is a clear trend linking the level of enhancement with the amount of precision loss. FR- 0.1-1.9 had the most loss in precision of 99 percent. It can then be observed that as the enhancement ratios went down to FR-0.5-1.5, the precision loss also went down to 30 percent. While the lowest enhancement ratio of FR-0.9-1.1 only had a 15 percent loss in precision. The exponential loss of precision following larger enhancement ratios can be attributed to the loss of critical information during the double enhancement process. By enhancing the color spectrum from 0 to 75 and 180 to 255, 150 different pixel values were enhanced. Enhancing over 58 percent of the original color scheme could lead to conflicting features and loss of useful features.

From the color enhancement trials, it can be determined that the ratios of enhancement greatly affect the detector performance; too little color enhancement results in no significant changes but a huge enhancement ratio can have negative impacts. The rise in precision with the color enhancement-60 trial can be attributed to correctly separating key features of the pistol from the background while retaining most of the context. The color enhancement-90 trial likely overly enhanced the image so that key features of the pistol might be lost while the background context became too monotoned. Although the current color enhancement trials do not show significant increases in precision, the preliminary results obtained using minimalist preprocessing algorithms open the way for more advanced enhancement techniques. For future works, potential extensions of this algorithm are directly amplifying a range of pistol color spectrum with machine learning algorithms.

Applying PCA to the training dataset did not result in any increases in average precision. PCA reduced the number of features by half in the training data images, which resulted in a loss of distinguishing features for the VGG16 to use. Figure 4 showed that the PCA trained detector performed poorly when tested on the original testing dataset as the average precision dropped from 0.2138 to 0.1223. In comparison, the detector performed noticeably better when tested on the PCA reduced testing dataset with the average precision only dropping from 0.2138 to 0.1636. This change in performance can be attributed to the VGG16 only retaining features pertinent to the PCA images. While too many features can confuse the neural network, too little feature will cause the learned PCA features to not transfer to original images.

The first combined dataset with an evenly distributed color enhanced, PCA and original images only raised the average precision from 0.2138 to 0.2398. The insignificant change in average precision indicates that blind variations in the training dataset do not necessarily improve performance due to conflicting features. The PCA reduced features likely interfered with the positive effects of the color-enhanced trials. The second dataset focused entirely on the effect of varying color

enhancements. The second combined dataset hoped to retain the positive effects from the color enhance-60 trials while adding beneficial contrast variations. Contradicting initial predictions, the second detector trained with the three color-enhanced image datasets performed drastically worse than the original detector. The combined color-enhanced detector lowered the average precision from 0.2138 to 0.1524. The significant drop in average precision can be attributed to the color enhance-30 and 90 dataset offering conflicting features that overruled the positive effects of the color enhance-60 dataset.

## 9 Conclusions

In conclusion, this paper analyzed three different image preprocessing techniques that can be applied to reduce overfitting in a VGG16 F-RCNN detector. The color enhancement method proved to be most impactful in increasing the average testing precision. A positive correlation was found between increasing the lighter region enhancement ratios and increases in average testing precision. While enhancing darker regions were observed to have detrimental impacts on precision. It was found that on average a small amount of color enhancement is unlikely to result in noticeable changes in detector performance. However, over enhancement can also have negative impacts on performance. The optimal color enhancement ratio and threshold will depend on the target objects and their relative backgrounds. It is recommended that different object databases go through extensive trials to find the correct enhancement ratios and thresholds to optimize performance.

Applying PCA to reduce the number of features in the image proved to be an ineffective method to increase average testing precision for pistol detection. The VGG16 F-RCNN detector trained with PCA reduced images had an overall detrimental effect on testing precision. The loss of precision was largely attributed to a loss of critical information during the feature reduction phase. The first combined dataset of PCA and color enhancement images resulted in no significant increases in performance. The second dataset of only color enhancement trials resulted in a detriment to performance. Overall, the combined methods had no significant impact on performance. The underwhelming performance of the combined method can be contributed to conflicting features. Too large of a color variation can result in conflicting features and had a negative impact on performance.

## Acknowledgments

The following people of ECU's Innovation Design Lab contributed greatly in funding and supporting the project: Dr. Todd Fraley, Dr. Ted Moris, Director Wayne Godwin, lab assistants Marco Agostini and Elliot Paul. This material is based upon work supported by the National Science Foundation under grant number IUSE/PFE: RED award #1730568. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the

authors and do not necessarily reflect the views of the National Science Foundation.

### References

- [1] A. Akbarinia and K. R. Gegenfurtner, "How is Contrast Encoded in Deep Neural Networks?", *arXiv preprint arXiv:1809.01438*, 2018. <https://arxiv.org/abs/1809.01438>, Last Accessed 17 Feb 2021.
- [2] S. Akcay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, "Using Deep Convolutional Neural Network Architectures for Object Classification and Detection Within X-ray Baggage Security Imagery," *IEEE Transactions on Information Forensics and Security*, 13(9):2203–2215, 2018.
- [3] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, "The History Began from Alexnet: A Comprehensive Survey on Deep Learning Approaches," *arXiv preprint arXiv:1803.01164*, 2018.
- [4] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [5] "Brightness Guided Preprocessing for Automatic Cold Steel Weapon Detection in Surveillance Videos with Deep Learning," *Neurocomputing*, 330:151–161, 2019.
- [6] A. Dhingra, "Model Complexity-Accuracy Trade-off for a Convolutional Neural Network," *arXiv preprint arXiv:1705.03338*, 2017. <https://arxiv.org/abs/1705.03338>, Last Accessed 17 Feb 2021.
- [7] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference On Computer Vision*, pp. 1440–1448, 2015.
- [8] J. Grau, I. Grosse, and J. Keilwagen, "PRROC: Computing and Visualizing Precision-Recall and Receiver Operating Characteristic Curves in R," *Bioinformatics*, 31(15):2595–2597, 2015.
- [9] N. A., "IMFDB: Internet Movie Firearms Database," 2020. [http://www.imfdb.org/wiki/Main\\_Page](http://www.imfdb.org/wiki/Main_Page) [Accessed on 5 May 2020].
- [10] I. T. Jolliffe and J. Cadima, "Principal Component Analysis: a Review and Recent Developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.
- [11] P. Kim, "Matlab Deep Learning," *With Machine Learning, Neural Networks and Artificial Intelligence*, vol. 130, 2017.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
- [13] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-Aware Fast R-CNN For Pedestrian Detection," *IEEE Transactions on Multimedia*, 20(4):985–996, 2017.
- [14] M. Lokanath, K. Sai Kumar, and E. Sanath Keerthi, "Accurate Object Classification and Detection by Faster-RCNN," *Materials Science and Engineering Conference Series*, vol. 263, 2017.
- [15] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A Survey of Multilinear Subspace Learning for Tensor Data," *Pattern Recognition*, 44(7):1540–1551, 2011.
- [16] A. McAndrew, "An Introduction to Digital Image Processing with Matlab Notes for SCM2511 Image Processing," *School of Computer Science and Mathematics, Victoria University of Technology*, 264(1): 1–264, 2004.
- [17] J. G. Nagy, K. Palmer, and L. Perrone, "Iterative Methods for Image Deblurring: a Matlab Object-Oriented Approach," *Numerical Algorithms*, 36(1):73–93, 2004.
- [18] R. Olmos, S. Tabik, and F. Herrera, "Automatic Handgun Detection Alarm in Videos Using Deep Learning," *Neurocomputing*, 275:66–72, 2018.
- [19] A. P. Piotrowski and J. J. Napiorkowski, "A Comparison of Methods to Avoid Overfitting in Neural Networks Training in the Case of Catchment Runoff Modelling," *Journal of Hydrology*, 476:97–111, 2013.
- [20] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazechnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, *et al.*, "Dataset Issues in Object Recognition," *Toward Category-Level Object Recognition*, pp. 29–48, 2006.
- [21] A. Rehman and T. Saba, "Neural Networks for Document Image Preprocessing: State of The Art," *Artificial Intelligence Review*, 42(2):253–273, 2014.
- [22] G. Shakhnarovich and B. Moghaddam, "Face Recognition in Subspaces," in *Handbook of Face Recognition*, pp. 141–168, 2005.
- [23] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014. <https://arxiv.org/abs/1409.1556> Last Accessed 17 Feb 2021.
- [24] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," *arXiv preprint arXiv:1412.6806*, 2014. <https://arxiv.org/abs/1412.6806> Last Accessed 17 Feb 2021.
- [25] D. P. Strik, A. M. Domnanovich, L. Zani, R. Braun, and P. Holubar, "Prediction of Trace Compounds in Biogas from Anaerobic Digestion using the MATLAB Neural Network Toolbox," *Environmental Modelling & Software*, 20(6):803–810, 2005.
- [26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
- [27] G. K. Verma and A. Dhillon, "A Handheld Gun Detection using Faster R-CNN Deep Learning," *Proceedings of the 7th International Conference on Computer and Communication Technology*, pp. 84–88,

2017.

- [28] Y. Xu, Z. Zhang, G. Lu, and J. Yang, "Approximately Symmetrical Face Images for Image Preprocessing in Face Recognition and Sparse Representation based Classification," *Pattern Recognition*, 54:68–82, 2016.
- [29] L. Yann, H. Fu, and B. Leon, "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting: Computer Vision and Pattern Recognition." CVPR 2004, *Proceedings of the 2004 IEEE Computer Society Conference*, vol. 2, 2004.
- [30] Y. Zhu, H. Yuan, C. Zhang, and C. Lee, "Image-Preprocessing Method for Near-Wall Particle Image Velocimetry (PIV) Image Interrogation with Very Large In-Plane Displacement," *Measurement Science and Technology*, 24(12):125302, 2013.



**Jiahao Li** is a machine learning researcher at East Carolina University. His primary research interest includes the main areas of machine learning from deep neural networks to evolutionary computing systems. His recent research is aimed at developing intelligent threat response systems with

robust auto-regressive models. Outside of technical research pursuits, Li is also pursuing sensible A.I. ethics and studying the impact of automation on society.



**Charles Ablan** is an undergraduate researcher at East Carolina University pursuing his BS in Engineering. His main areas of research interest include machine learning and engineering technological innovation. He is a member of the ECU underground water level research team developing autoregressive models.



**Wu Rui** received a bachelor's degree in Computer Science and Technology from Jilin University, China in 2013. He received his Master and Ph.D. degrees in Computer Science and Engineering from the University of Nevada, Reno in 2015 and 2018, respectively. Dr. Wu is now working as an assistant professor in the Department of Computer Science at East Carolina University and

collaborates with geological and hydrological scientists to protect the ecological system. His main research interests are data imputation, machine learning, and data visualization using AR/VR devices.



**Guan Shanyue** received his B.S. degree in civil engineering from Tongji University, Shanghai, China, in 2011, and the Ph.D. degree in civil engineering from the University of Florida, Gainesville, FL, USA, in 2017. From 2017 to 2018, he was a Post-Doctoral Fellow with the University of Florida. He joined East Carolina University, Greenville, NC, USA, in 2018, where he is currently an Assistant Professor of Engineering.

His research interests include smart and resilient infrastructure, wireless sensor networks, UAV-based monitoring, image processing, and data analysis.



**Jason Yao** has research interests in the areas of wireless/wearable medical sensors, sensor networks for home environments, telemedicine, and industrial process monitoring and control. Dr. Yao received his Ph.D. degree in electrical engineering from Kansas State University. He is a senior member and an active volunteer of IEEE.



# Exploiting a Real-time Non-geolocation Data to Classify a Road Type with Different Altitudes for Strengthening Accuracy in Navigation

Thitivatr PatanasakPinyo\*  
Mahidol University, Salaya, 73170, THAILAND

## Abstract

Most location-based applications for navigation purposes use geolocation data, i.e., a pair of a latitude and a longitude, to determine a real-time location of a handheld device (e.g., smartphones or tablets) that runs the applications. This can be implemented basically by requesting a pair of a latitude and a longitude from the device's sensor that receives geolocation data from satellites. However, telling a device's location by GPS sensor is sometimes impractical, especially when the device is in a vehicle on a road that shares exactly the same geolocation with other roads. Particularly, this is a scenario that there is a ground-level road along with another elevated road (e.g., a turnpike) which is very common in cities like Bangkok, Singapore, or Hong Kong. The geolocation data yield no clue whether or not a vehicle is running on a ground-level road. Since a pair of a latitude and a longitude can no longer be used in such scenario, we proposed a methodology to identify the correct location of both a device and a vehicle without any involvement of geolocation data by using a Random Forest classifier and real-time traffic data that are able to be captured by a handheld device as training features to train a classification model. A completed experiment and results after testing the model were reported in this article.

**Key Words:** Location-based applications; navigation; geolocation; altitude; random forest.

## 1 Introduction

A location-based application has become a primary tool for road traffic navigation. Most users run the application on a handheld device such as a smartphone or a tablet. Furthermore, some automobile manufacturers provide a simple user interface to connect a smart device with a car's screen via USB or Bluetooth. This highly supports a navigation task since a map can be displayed on a wide screen on a car's console, which is way more comfortable for a driver. Navigating a route via location-based applications relies on a device's location sensor, which is a part that tells an exact location of the device by a pair of a latitude and a longitude. The sensor retrieves a

signal from the satellites and computes both a latitude and a longitude. When the device has both values, it displays the device's location on the map. This whole process of how a location-based application works fine and invulnerable. However, there is a scenario that causes a device to misinterpret a location [7, 9]. In a crowded city like Bangkok, Singapore City, or Hong Kong, it is very common to see a ground-level road along with an elevated road above it. Since both roads are exactly located on the same location, every pair of a latitude and a longitude that belongs to the ground-level road also belongs to the elevated one as well. This confuses a device to distinguish what road a vehicle is on and leads to an incorrect navigation that can result in a major detour. Figures 1 and 2 show examples of roads that are under this condition. Figure 1 shows Borommaratchachonnani (Bor-Rom-Ma-Raj-Cha-Chon-Na-Ni) Elevated Road which is a 15-meter elevated from the ground while Figure 2 shows Borommaratchachonnani Frontage Road. Both roads are located in the city of Bangkok, Thailand. These east-west roads are 16-kilometer long that have an east end in the downtown of Bangkok and a west end at the west border of Bangkok (Figure 3). We would like to see how an application navigated when we drove on the frontage road. Our starting point was at the west end of the frontage road. Before we started, we set the destination on an application to be a shopping mall located at the north of Bangkok, particularly, Central Plaza Westgate. The location-based application that we used was Google Maps for Android. The device that we used was Samsung Galaxy Note 5 (2015). It turned out to be that the application understood that we drove on the elevated road, so it navigated us to take the closest exit to leave the elevated road then make a u-turn. However, the navigation was a major detour (Figure 4). We could have made a left turn at the intersection to head right to the north but the application did not think we can do this because it thought that we were not on the ground-level road. We did the same experiment on another day and found out that the application still navigated incorrectly. It still understood that we were on the elevated road (Figure 5). For this example, if we followed a navigation from the application, the detour would cost us significantly around 10 kilometer.

One question could be asked if there is a way to tell an altitude of a device. An altitude is a metric that informs a y-distance from the sea level. Clearly, knowing an altitude technically

\*Faculty of Information and Communication Technology. Email: thitivatr.pat@mahidol.edu.



Figure 1: Borommaratchachonnani Elevated Road (image source: [www.dailynews.co.th](http://www.dailynews.co.th)).



Figure 2: Borommaratchachonnani Frontage Road (image source: *Google Maps Street View*).

could do the trick. A GPS sensor in a smart device (plus a barometer sensor) is capable of retrieving an altitude from the satellites as well although the device itself does not have a built-in altimeter. However, an issue about an accuracy might exist in some conditions. In general, we can expect around  $\pm 23$  meter for vertical error (altitude), which is around 1.5 times greater than horizontal error (latitude and longitude) [6, 5]. Since a vertical difference of most elevated and ground-level roads are usually less than 20 meter, an altitude calculated via this method is not dependable. For instance, the vertical difference between Borommaratchachonnani Elevated Road (Figure 1) and Borommaratchachonnani Frontage Road (Figure 2) is about 15 meter. It is also quite impractical for smart device manufacturers to equip a device with an actual altimeter since a demand to know an altitude for most users is uncommon.

With the problem being raised, we proposed a geolocation-free solution to determine whether a vehicle/device was on an elevated road or a ground-level road by using a set of real-time traffic data to train a classification model. By the term “real-time traffic data”, we considered every possible metric that an average smart device was able to capture by its built-in sensors/features plus some aggregated data from further computation process. This set of data is detailed in Section 3.

We ran the preliminary experiment by choosing these Borommaratchachonnani Elevated Road and Borommaratchachonnani Frontage Road to collect data [11]. This data gathering process was conducted by driving on each road multiple trips and enabling an Android device to run our own application that invoked all built-in sensors to collect real-time data corresponding to the sensor. We set up a driving schedule in a way that it covered most days (days of

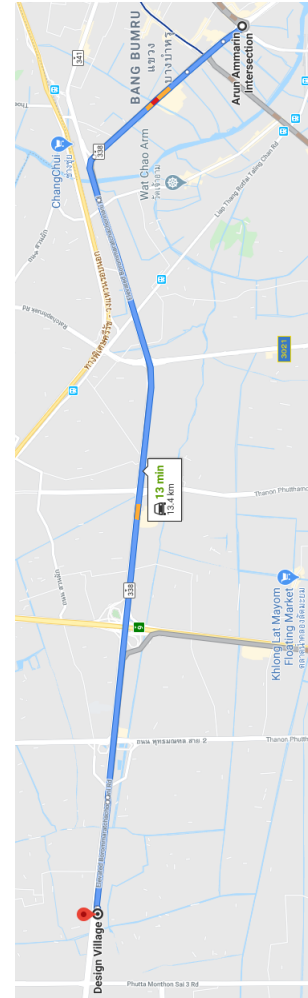


Figure 3: A layout of both the roads illustrated by Google Maps.

week), time (time of day), and direction bounds (eastbound and westbound) to prevent a bias that might occur from imbalanced or inadequate feature values. The application automatically logged a new instance of data every two seconds. We trained two classification models using Random Forest and Bagging (with REPTree as a base classifier), respectively, as a classifier. The first model resulted in 99.7563% accuracy. The second model resulted in 93.4315% accuracy.

The structure of this paper is as follows. The first section introduces the problem scope with the proposed solution plus the results from the preliminary work. The second section refers to related work. Sections 3 and 4 discuss about a methodology and results obtained. The last section concludes this study and sheds some light to a possible future adaptation.

## 2 Related Work

It is quite true to say that a location-based application is necessary for most users of smart devices. A location-based application is capable of serving several purposes. Several

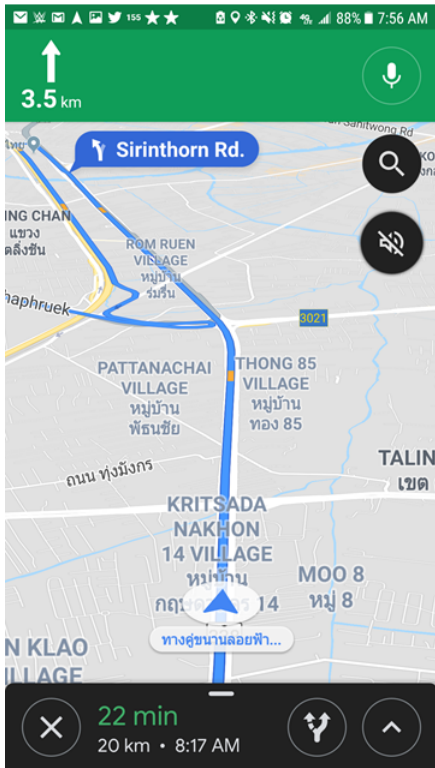


Figure 4: Navigated Routed from Google Maps on January 17, 2019.

studies integrated a location-based application with a census activity. Batinov et al. [1] proposed a detection pattern that can distinguish if a participant's spatial visualization (VZ) [8, 17] was low or high by analyzing their sequences of taps they made on the tablet screen while using it to perform address verification activity. Similarly, PatanasakPinyo et al. [10, 12, 14, 15, 16] did three studies that invited participants to verify addresses in the neighborhood using a location-based application on a tablet and found that there were some metrics that significantly could be used to identify a participant's spatial visualization such as a total number of pans, a total number of zooms, etc. PatanasakPhyo et al. [13] proposed a concept of empowering an indexing ability to a traditional raster map widely used by location-based applications. Sulaiman [19] verified that such metrics were still reliable even though an environment of an address verification task was changed from an actual neighborhood to a virtual reality. Whitney [20] enhanced an address verification task by combining a concept of a location-based application with a virtual reality. One popular activity that involved with location-based application is navigation. Most map applications such as Google Maps are designed and developed for the task of navigation with highly acceptable accuracy, which depended on a location sensor of a smart device (GPS) [3, 4, 18]. Lin et al. reported issues that were found when relying on GPS for navigation purposes [2]. Misinterpreting an exact location because of a difference in altitude is one difficulty

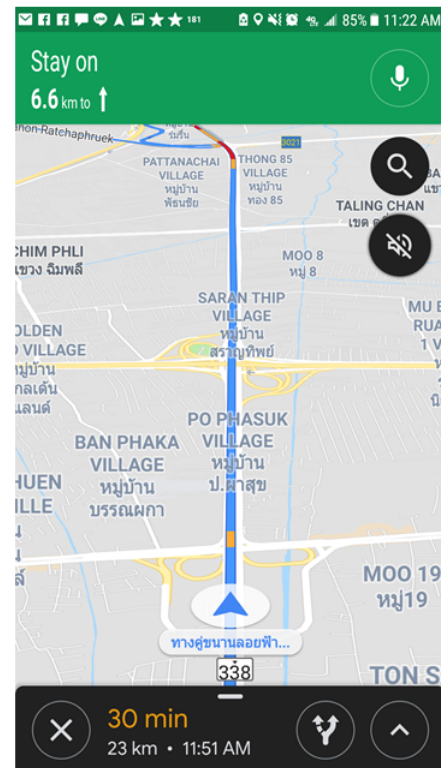


Figure 5: Navigated Routed from Google Maps on January 25, 2019.

users have to deal with when asking the application to navigate in cities with limited area [7, 9].

### 3 Methodology

We replicated the methodology previously implemented by PatanasakPinyo [11] by collecting all possible real-time traffic data that a device's sensor was able to retrieve while targeted driving on an actual road. The roads targeted for this study were Bangna - Chon Buri Expressway and Bangna - Trat Frontage Road. Both elevated and ground-level roads are east-west roads that located in Thailand (Figure 6). The methodology of this study consists of Study Design, Data Gathering, and Model Implementation.

#### 3.1 Study Design

Since our objective was to develop a classification model that can correctly classify whether a vehicle is running on an elevated road or a ground-level road without involvement of any geolocation data (latitude, longitude, altitude), we have to include metrics that a standard smart device can retrieve using its built-in sensors as many as possible. After exploring a device that we would use to collect data in this study (2019 Samsung Galaxy Tab A), we came up with a list of variables that the device was capable to collect as follows:



Figure 6: Bangna - Chon Buri Expressway located above Bangna – Trat Frontage Road (image source: [www.dailynews.co.th](http://www.dailynews.co.th)).

1. **Distance (*distance*):** A distance representing how far a vehicle moved within a 2-second interval (we preset the interval). A distance was obtained by computing a difference between two points (start and end) at a certain time tick.
2. **Speed (*speed*):** A speed of a vehicle at a certain point of time. A speed was obtained by a fraction of distance to time (2 second).
3. **Direction (*direction*):** A direction that a car was heading to. It was computed by evaluating an angle  $\alpha$  between a line segment  $\langle p_s, p_e \rangle$  and a line  $y = 0$  where  $p_s$  and  $p_e$  are a start point and an end point of a certain time interval, respectively.
4. **Light Intensity (*lux*):** A light intensity that can be retrieved by a light sensor which always comes with most smartphone devices.
5. **Time:** A timestamp that consisted of hour (*hour*) and minute (*minute*).
6. **Day-of-Month:** A day of month.
7. **Day-of-Week (*day*):** A day of week.
8. **Bound (*bound*):** A direction bound that informs which side of the road that a vehicle was running on, which can be either eastbound or westbound.
9. **Road Type (*road\_type*):** A class variable indicating whether a vehicle is on an elevated road or a ground-level road.

After we had a set of variables ready and stable, we then developed an Android application that retrieved those variables and recorded as a log file (with CSV extension for easily compatible with most statistics and data science software tools such as R or Python). Note that when we finished the data gathering, we decided not to include Day-of-Month in the set of features because we did not have data of every day (of month) equally, which might lead to an imbalance problem that would insignificantly contribute to the model.

### 3.2 Data Gathering

For the process of data gathering, we selected Bangna - Chon Buri Expressway and Bangna – Trat Frontage Road as an elevated road and a ground-level road, respectively. Both roads lie east-west and link Bangkok, a capital city of Thailand, with Chon Buri, a famous tourist cities in Thailand. The two roads are located on the north of Thai Gulf. Figure 7 shows both roads on Google Maps. To collect data, we drove along the road while enabling the application to automatically get values of all variables from the device's sensors and log them. The application was set to refresh the logging task every two second as previously mentioned.

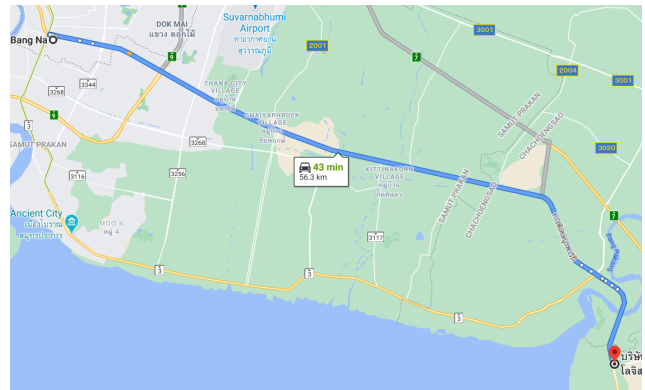


Figure 7: Layout of Bangna - Chon Buri Expressway and Bangna – Trat Frontage Road on Google Maps.

To have data balance in most variables as much as possible, we managed to have a similar number of trips of elevated road & ground-level road, eastbound drive & westbound drive, daytime drive & nighttime drive, and balanced day of week (Sunday to Saturday) since a day of week affects traffic, which might contribute to a prediction model. After data gathering processing, we logged 24362 instances, which can be divided into 11095 instances of the elevated road and 13267 instances of the ground-level road.

Table 1 shows examples of instances of data in the log file comparing with the old one from PatanasakPinyo [11] (Table 2).

### 3.3 Model Implementation

We inherited a concept from PatanasakPinyo [11] to train a classification model to classify the road type using a Random Forest as a classifier to observe whether there existed any differences when training data were collected from different roads, particularly, Bangna - Chon Buri Expressway and Bangna – Trat Frontage Road, rather than Borommaratchachonnani Elevated Road and Borommaratchachonnani Frontage Road. We decided to use R as a tool to filter and handle pre-processing the raw data. We partitioned the data set into a training set (70%) and a test set

Table 1: Instances of data collected from data gathering process.

year	month	date	day	hour	minute	second	lat	ing	altitude	distance	speed	direction	lux	temp	bound	road_type
2020	5	7	4	12	0	50	13.66709596	100.63885018	-34	3.340153002	6012.275403	356.8788	993	0	E	elevated
2020	5	7	4	12	0	52	13.66698917	100.6389127	-17	0.005308390	9.555102	240.6810	989	0	E	elevated
2020	5	7	4	12	0	54	13.66690162	100.6392806	-3	0.004070347	7.326625	241.4041	979	0	E	elevated
2020	5	7	4	12	0	56	13.66686327	100.6395299	-8	0.005221977	9.399559	251.2549	963	0	E	elevated
2020	5	7	4	12	0	58	13.66670674	100.640143	-10	0.004934684	8.882431	245.6843	981	0	E	elevated
2020	5	7	4	12	1	0	13.66661957	100.6406102	-12	0.005106850	9.192331	246.4602	970	0	E	elevated
2020	5	7	4	12	1	2	13.66652733	100.6411005	-14	0.005265021	9.477038	249.0224	959	0	E	elevated
2020	5	7	4	12	1	4	13.66643043	100.6418091	-13	0.005284162	9.511492	248.0496	948	0	E	elevated
2020	5	7	4	12	1	6	13.66632566	100.6421179	-13	0.005281114	9.506005	244.6641	965	0	E	elevated
2020	5	7	4	12	1	8	13.66621879	100.642636	-12	0.005382215	9.687987	246.2216	944	0	E	elevated
2020	5	7	4	12	1	10	13.66611976	100.6431517	-13	0.005352376	9.634277	248.1123	930	0	E	elevated
2020	5	7	4	12	1	12	13.66602714	100.6436797	-12	0.005392556	9.706601	248.7074	936	0	E	elevated
2020	5	7	4	12	1	14	13.66592526	100.6442047	-13	0.005386668	9.663602	246.5293	941	0	E	elevated
2020	5	7	4	12	1	16	13.66581964	100.6447341	-13	0.005407020	9.732636	246.2482	940	0	E	elevated

(30%). The variables that were fed as the model features were day-of-week, hour & minute, distance, speed, direction, light intensity, and direction bound. A class attribute was a type of the road (elevated/ground). The final model was trained in R using Caret Library to handle Random Forest. For train control, we chose to implement a 10-fold cross validation. Since other important training parameters such as *mtry* (i.e., a number of variables to be randomly sampled as candidates at each split in a tree) were not predefined, we trained the model multiple times to obtain optimal values of those training parameters as well.

Table 2: Instances of data in the preliminary study [11].

year	month	date	day	hour	minute	second	lat	ing	altitude	distance	speed	direction	lux
2020	1	2	4	15	15	14	13.78085367847234	100.4300842715502	-8.0	0.005088622	9.159520	255.885606	366
2020	1	2	4	15	15	16	13.780807997100055	100.4305356927216	-9.0	0.005106809	9.192256	254.043083	362
2020	1	2	4	15	15	18	13.780763824470341	100.43102058552008	-11.0	0.005130625	9.235126	255.199469	235
2020	1	2	4	15	15	20	13.780721286311746	100.4315016232431	-11.0	0.005102782	9.168507	256.565564	327
2020	1	2	4	15	15	22	13.780666459504068	100.43196966871619	-11.0	0.005040076	9.072137	257.220647	399
2020	1	2	4	15	15	24	13.780636419542134	100.4324477272731	-11.0	0.005116951	9.210512	251.996177	448
2020	1	2	4	15	15	26	13.78058788832277	100.43291942216456	-10.0	0.005070300	9.129654	254.004095	406
2020	1	2	4	15	15	28	13.78051907289732	100.43360053561628	-10.0	0.004938547	8.889385	253.396242	423
2020	1	2	4	15	15	30	13.7804801389575	100.4340636357665	-11.0	0.005048775	9.087795	256.795132	422
2020	1	2	4	15	15	32	13.78043420612812	100.4345269873383	-12.0	0.004994445	8.990002	254.453015	555

#### 4 Results and Discussion

To come up with the most optimal classification model, we needed to assign training parameters to be a certain value that we had not known until we did experiments. Hence, the first half of this section is to report results from experiments to select the best values of each training parameter. We then reported results of training and testing of the classification model along with important statistics.

The first training parameter was *mtry*. An *mtry* is a number of variables to be randomly sampled as candidates at each split in a tree. We trained a model by using a value from one to eight. After training, we found that *mtry* = 3 is the most optimal.

Next, we trained a model to find the optimal value of *maxnodes*. A *maxnodes* is a maximum number of nodes in a tree. We tried values from five to fifteen. After training, the best value was *maxnodes* = 14, which yielded an accuracy of 0.941349 (Table 3).

Table 3: Accuracy Rates of Different Values of *maxnodes*.

Accuracy	Min.	1st Qu.	Median	Mean
5	0.8533724	0.8617911	0.8660206	0.8661228
6	0.8539589	0.8659824	0.8736440	0.8749770
7	0.8668622	0.8809553	0.8870968	0.8864722
8	0.8826979	0.8912182	0.8947522	0.8943880
9	0.8862170	0.8951613	0.8965115	0.8979657
10	0.8774194	0.8918043	0.8970972	0.8977304
11	0.8938416	0.8996047	0.9047201	0.9036533
12	0.8985932	0.9036657	0.9070660	0.9085798
13	0.9021102	0.9131965	0.9138084	0.9152059
14	0.9043988	0.9083578	0.9167644	0.9176682
15	0.9102639	0.9214076	0.9255357	0.9242368

	3rd Qu.	Max.	NA's
5	0.8743402	0.8751465	0
6	0.8854045	0.8961877	0
7	0.8951776	0.8991202	0
8	0.9006015	0.9043988	0
9	0.9049998	0.9079179	0
10	0.9064657	0.9126100	0
11	0.9086646	0.9114370	0
12	0.9108896	0.9284457	0
13	0.9197947	0.9255132	0
14	0.9236183	0.9413490	0
15	0.9291895	0.9366569	0

The last one was the number of trees in the forest (*ntrree*). We tried values from 100 to 900. After training, the best value was *ntrree* = 100, which yielded an accuracy of 0.9431085 (Table 4).

Table 4: Accuracy Rates of Different Values of *ntrree*.

Accuracy	Min.	1st Qu.	Median	Mean
100	0.9032258	0.9125020	0.9161534	0.9187237
200	0.9043988	0.9083578	0.9167644	0.9176682
300	0.9008798	0.9127566	0.9152738	0.9171405
400	0.9096774	0.9162879	0.9190858	0.9201316
500	0.9090909	0.9167277	0.9199651	0.9200728
600	0.9108504	0.9180713	0.9205512	0.9214799
700	0.9108504	0.9186694	0.9225806	0.9219492
800	0.9120235	0.9171674	0.9211374	0.9220077
900	0.9108504	0.9177538	0.9214310	0.9227111

	3rd Qu.	Max.	NA's
100	0.9231672	0.9431085	0
200	0.9236183	0.9413490	0
300	0.9196952	0.9390029	0
400	0.9217352	0.9384164	0
500	0.9218475	0.9378299	0
600	0.9222874	0.9372434	0
700	0.9225806	0.9378299	0
800	0.9258065	0.9378299	0
900	0.9250733	0.9372434	0

After we had optimal values of training parameters (*mtry*, *maxnodes*, *ntrree*), we assign those values in the model configuration and trained the final model. The final model yielded an accuracy of 0.9245 with (0.9182, 0.9304) as 95% confident interval when we tested the model with the test set. The test set had 7309 instances of data. The 3086 instances of elevated road were correctly classified while 287 were incorrectly classified as a ground-level road. The 3671 instances ground-level road were correctly classified while 265 were incorrectly classified as an elevated road. The first five important variables sorted by variable importance were *lux*, *hour*, *speed*, *day*, and *distance*. Figure 8 shows the completed test result (confusion matrix and statistics) generated by Caret.

Reference		Prediction	
elevated	frontage	elevated	frontage
		3086	265
		287	3671
Accuracy : 0.9245			
95% CI : (0.9182, 0.9304)			
No Information Rate : 0.5385			
P-Value [Acc > NIR] : <2e-16			
Kappa : 0.848			
McNemar's Test P-Value : 0.3714			
Sensitivity : 0.9149			
Specificity : 0.9327			
Pos Pred Value : 0.9209			
Neg Pred Value : 0.9275			
Prevalence : 0.4615			
Detection Rate : 0.4222			
Detection Prevalence : 0.4585			
Balanced Accuracy : 0.9238			
'Positive' Class : elevated			

Figure 8: Confusion Matrix and Statistics after Testing the Model.

We also trained one more model to see how much better a classification model would be if we included an altitude in the feature set regardless of the fact that the accuracy level of an altitude retrieved by a GPS sensor in an average-grade smartphone was not reliable if a difference in height was less than a specific threshold. With this model, we replicated what we did previously. We used the same training and test data sets as well as all presets to ensure that every factor was properly preserved except that the feature set, particularly, we added altitude to it.

With the classification model that included an altitude, we found that *mtry* = 2 was optimal for this case. Next, we looped through various values of *maxnodes* from five to fifteen. It was

showed up that *maxnodes* = 15 is optimum. Table 5 shows an accuracy rate for each value of *maxnodes*.

Table 5: Accuracy Rates of Different Values of *maxnodes*.

Accuracy	Min.	1st Qu.	Median	Mean
5	0.9407625	0.9585287	0.9683284	0.9640532
6	0.9513482	0.9662757	0.9683284	0.9669282
7	0.9659824	0.9708211	0.9730363	0.9736120
8	0.9700880	0.9725930	0.9756670	0.9758992
9	0.9712610	0.9755167	0.9771395	0.9774234
10	0.9747801	0.9755240	0.9774326	0.9778340
11	0.9712610	0.9750844	0.9765463	0.9775411
12	0.9730205	0.9766965	0.9788923	0.9787721
13	0.9724340	0.9768361	0.9785986	0.9787136
14	0.9736070	0.9771395	0.9788920	0.9795347
15	0.9765533	0.9786019	0.9800645	0.9808836

	3rd Qu.	Max.	NA's
5	0.9712610	0.9759531	0
6	0.9709721	0.9771261	0
7	0.9769795	0.9800587	0
8	0.9788856	0.9818182	0
9	0.9791789	0.9835777	0
10	0.9800587	0.9812317	0
11	0.9809384	0.9841642	0
12	0.9810850	0.9835777	0
13	0.9810850	0.9829912	0
14	0.9818182	0.9853372	0
15	0.9818182	0.9894428	0

Similarly, we lopped through various values of *n tree* and found that the best case was when *n tree* = 100. Table 6 shows an accuracy rate for each value of *n tree* that we tried.

We trained the final classification model that included an altitude one last time using optimal values of training parameters that we just retrieved. After we got the model, we fed the test data set to it to observe a performance. The model yielded an accuracy rate of 0.9793 with (0.9758, 0.9825) as 95% CI. In 3304 instances of elevated road were classified correctly while 69 instances were incorrectly classified. On the other hand, 3854 instances of frontage road were correctly classified while 82 were incorrectly classified. Figure 9 shows test results and related statistics.

The model reported the first five important variables as *altitude*, *lux*, *hour*, *distance*, and *speed*. All of them were overlapped with a set of important variables of the first model (the model without an altitude) except *altitude*. Figures 10, 11, 12, and 13 show box plots of distribution of instances of data (only the training data set) group by each important variable, particularly, *lux*, *hour*, *speed*, and *distance* so the reader can view a behavior of training data. Note that we omitted *day* because it contained factor data.

After comparing both results from the two models (with and without an altitude), we found that having an altitude did not

Table 6: Accuracy Rates of Different Values of *n tree*.

Accuracy	Min.	1st Qu.	Median	Mean
100	0.9753810	0.9790445	0.9812317	0.9815875
200	0.9765533	0.9786019	0.9800645	0.9808836
300	0.9765396	0.9772860	0.9794780	0.9804730
400	0.9765396	0.9768464	0.9800643	0.9801798
500	0.9759672	0.9783023	0.9797712	0.9802385
600	0.9759672	0.9778690	0.9809439	0.9803557
700	0.9765396	0.9774326	0.9800643	0.9801797
800	0.9753666	0.9778722	0.9800642	0.9801796
900	0.9747801	0.9784584	0.9803576	0.9802969

	3rd Qu.	Max.	NA's
100	0.9843109	0.9900293	0
200	0.9818182	0.9894428	0
300	0.9824047	0.9900293	0
400	0.9822581	0.9870968	0
500	0.9824047	0.9859238	0
600	0.9828446	0.9847507	0
700	0.9828446	0.9847507	0
800	0.9826979	0.9853372	0
900	0.9828446	0.9853372	0

Reference		
Prediction	elevated	frontage
elevated	3304	82
frontage	69	3854

Accuracy : 0.9793  
 95% CI : (0.9758, 0.9825)  
 No Information Rate : 0.5385  
 P-Value [Acc > NIR] : <2e-16

Kappa : 0.9584

Mcnemar's Test P-Value : 0.3288

Sensitivity : 0.9795  
 Specificity : 0.9792  
 Pos Pred Value : 0.9758  
 Neg Pred Value : 0.9824  
 Prevalence : 0.4615  
 Detection Rate : 0.4520  
 Detection Prevalence : 0.4633  
 Balanced Accuracy : 0.9794

'Positive' Class : elevated

Figure 9: Confusion Matrix and Statistics after Testing the Model.

significantly help improving the model's performance.

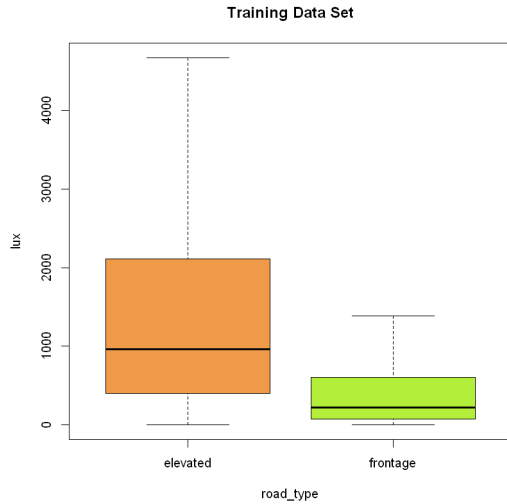


Figure 10: Box plots illustrate a distribution of *lux* of both classes.

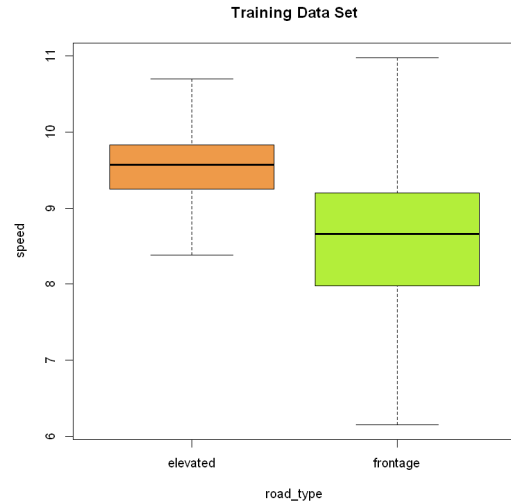


Figure 12: Box plots illustrate a distribution of *speed* of both classes.

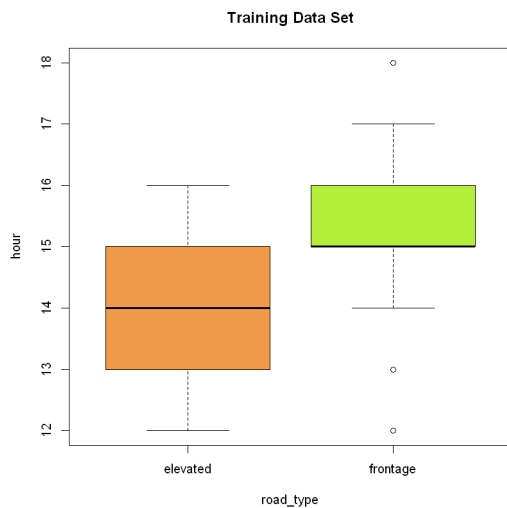


Figure 11: Box plots illustrate a distribution of *hour* of both classes.

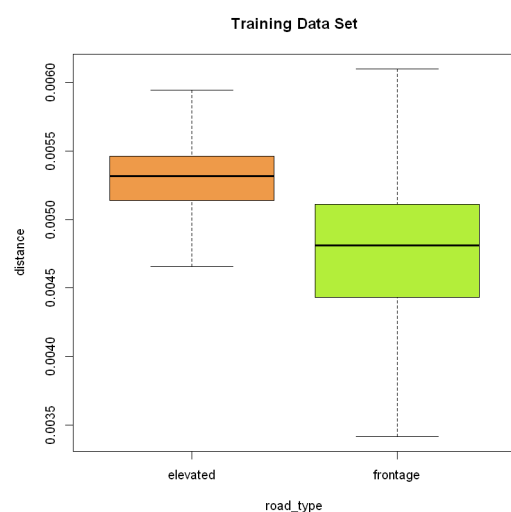


Figure 13: Box plots illustrate a distribution of *distance* of both classes.

## 5 Conclusions

In this study, we developed a classification model that was capable of classifying the road type that a vehicle was running on if it was an elevated road or a ground-level road where a traditional location-based application cannot do using geolocation data since both roads share exact pairs of a latitude and a longitude all along the way. We inherited a methodology from PatanasakPinyo [11] to train a model using Random Forest as a classifier with 10-fold cross validation as a train control. The training parameters were examined to ensure that they were the most optimal. The data that were used in this study were collected from driving on Bangna - Chon Buri Expressway

and Bangna – Trat Frontage Road. Both roads are located in Thailand. The result of testing the model showed an accuracy of 0.9245 with a 95% CI of (0.9182, 0.9304). For future extension of this study, we are going to do a cross-test by testing our classification model on data set used in [11].

## Acknowledgments

This research project is supported by both Mahidol University and The Faculty of Information and Communication Technology, Mahidol University.



## References

- [1] Georgi Batinov, Michelle Rusch, Tianyu Meng, Kofi Whitney, Thitivatr Patanasakpinyo, Les Miller, and Sarah Nusser. "Understanding Map Operations in Location-based Surveys". In *Eighth International Conference on Advances in Computer-Human Interactions (ACHI 2015)*, Lisbon, Portugal. International Academy, Research, and Industry Association (IARIA) pp. 144-149, 2015.
- [2] Allen Yilun Lin, Kate Kuehl, Johannes Schöning, and Brent Hecht. "Understanding 'Death by GPS' A Systematic Study of Catastrophic Incidents Associated with Personal Navigation Technologies". In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1154–1166 2017.
- [3] Paul W McBurney and Arthur N Woo. "Infrastructure-aiding for Satellite Navigation Receiver and Method". US Patent 6,473,030 October 29, 2002.
- [4] Paul W McBurney and Arthur N Woo. "Satellite Navigation Receiver and Method". US Patent 6,437,734 August 20, 2002.
- [5] Joe Mehaffey. "Error Measures". <http://gpsinformation.net/main/errors.htm>.
- [6] Joe Mehaffey. "GPS Altitude Readout How Accurate?". Retrieved, 12(12):2016, 2001.
- [7] Marko Modsching, Ronny Kramer, and Klaus ten Hagen. "Field Trial on GPS Accuracy in a Medium Size City: The Influence of Built-up". In *3rd workshop on positioning, navigation and communication*, 2006:209-218, 2006.
- [8] Kent L Norman. "Spatial Visualization—A gateway to computer-based technology". *Journal of Special Education Technology*, 12(3):195–206, 1994.
- [9] Eunil Park and Ki Joon Kim. "Driver Acceptance of Car Navigation Systems: Integration of locational accuracy, processing speed, and service and display quality with technology acceptance model". *Personal and ubiquitous computing*, 18(3):503–513, 2014.
- [10] Thitivatr PatanasakPinyo. "Flattening Methods for Adaptive Location-based Software to User Abilities". Graduate Theses and Dissertations, Iowa State University, 2017.
- [11] Thitivatr Patanasakpinyo. "Ameliorating Accuracy of a Map Navigation When Dealing with Different Altitude Traffics that Share Exact Geolocation". In Alex Redei, Rui Wu, and Frederick Harris, editors, *SEDE 2020. 29th International Conference on Software Engineering and Data Engineering*, EPiC Series in Computing. EasyChair, 76:95–104, 2021.
- [12] Thitivatr PatanasakPinyo, Georgi Batinov, Kofi Whitney, and Les Miller. "Methods That Flatten the User Space for Individual Differences in Location-based Surveys on Portable Devices". In *31st International Conference on Computers and Their Applications (CATA 2016)*. International Society for Computers and Their Applications (ISCA), Las Vegas, Nevada, pp. 65-70, 2016.
- [13] Thitivatr Patanasakpinyo, Georgi Batinov, Kofi Whitney, Adel Sulaiman, and Les Miller. "Object-Indexing: A Solution to Grant Accessibility to a Traditional Raster Map in Location-Based Application to Accomplish a Location-Based Task". *International Journal of Computing, Communication and Instrumentation Engineering (IJCCIE)*, 5(1):1–5, 2018.
- [14] Thitivatr Patanasakpinyo, Georgi Batinov, Kofi Whitney, Adel Sulaiman, and Les Miller. "Enhanced Prediction Models for Predicting Spatial Visualization (VZ) in Address Verification Task". *Proceedings of 34th International Conference on Computers and Their Applications*, 58:247-256, 2019.
- [15] Thitivatr PatanasakPinyo, Georgi Batinov, Kofi Whitney, Adel Sulaiman, Les Miller, and Stephen Gilbert. "Extracting Useful Features for Users with Different Levels of Spatial Visualization". In *33rd International Conference on Computers and Their Applications (CATA 2018)*. International Society for Computers and their Applications (ISCA), Las Vegas, Nevada, pp. 86-91, 2018.
- [16] Thitivatr Patanasakpinyo and Les Miller. "UI Error Reduction for High Spatial Visualization Users when Using Adaptive Software to Verify Addresses". In Gordon Lee and Ying Jin, editors, *Proceedings of 35th International Conference on Computers and Their Applications*, EPiC Series in Computing, Easy Chair, 69:22-31, 2020.
- [17] Timothy A Salthouse, Renee L Babcock, Debora RD Mitchell, Roni Palmon, and Eric Skovronek. "Sources of Individual Differences in Spatial Visualization Ability". *Intelligence*, 14(2):187–230, 1990.
- [18] Pierluigi Silvestrin, Peter Daly, David Walsh, and Eric Aardoom. "Receiver for a Navigation System, in Particular a Satellite Navigation System", May 30 2000. US Patent 6,069,583.
- [19] Adel Sulaiman. "Training and Evaluation in a Large-scale Virtual Environment for a Location-based Mobile Application", volume 17573. Graduate Theses and Dissertations, Iowa State University, 2019.
- [20] Kofi Whitney. "Taking the Lab on the Road and Bringing the Road to the Lab: On using mixed-methods and virtual reality to study a location-based task", volume 17123. Graduate Theses and Dissertations, Iowa State University, 2019.



**Thitivatr PatanasakPinyo** is a Lecturer at The Faculty of Information and Communication Technology, Mahidol University, Thailand. He received his PhD with Graduate Teaching Excellence Award from Iowa State University in 2017 under a supervision of Professor Dr. Les Miller and Prof. Dr. Wallapak Tavanapong. His area of interest consists of interface

designs for individual difference, location-based systems, and computational theory.

# Applications of Virtual Reality Hand Tracking for Self-Defense Simulation

John Apo\* and Alexander Redei\*  
Central Michigan University, Mount Pleasant, MI, USA

## Abstract

KickVR is a self-defense training simulation designed to help users learn how to better defend themselves as a last resort. In collaboration with the Mount Pleasant Police Department, KickVR empowers users in the community to learn self-defense even if they do not have the time to attend in-person classes or none are accessible in their region. Virtual reality provides convenient access to self-defense training for users in their own homes. The immersion is enhanced through the use of hand-tracking. In place of hand controllers, the user's physical hands are tracked and rendered for the self-defense simulation. Our contributions include: a virtual environment appropriate for encounters in a small town, integration of the Leap Motion Controller to an Oculus Rift CV1 headset, a Unity-driven experience, interviewing members of law enforcement for locations of violent crime, and a framework for investigating external evaluations of the impact of virtual reality simulations in future works.

**Key Words:** Human-computer interaction; virtual reality; simulation; training; self-defense simulation.

## 1 Introduction

Central Michigan University made national headlines in February 2018. During spring break move-out, a 19-year-old student shot and killed his parents in his dorm room causing the campus to go on lock-down [20]. More recently, in February 2020, a 19-year-old freshman student at Central Michigan University went to Wayside Central, the local night club, on a Saturday night and stabbed three attendees during an altercation [13]. These events seemed significant at the time, but crime has become increasingly more common and more likely to occur in this area. NeighborhoodScout ranks Mount Pleasant, Michigan as a 29 out 100 on safety, meaning Mount Pleasant is safer than only 29 percent of cities in the United States with roughly 20 crimes happening per 1,000 people per year. With nearly 15,000 students taking on-campus classes in Fall 2019, that equates to nearly 300 students becoming victims to some sort of crime during the academic year [4]. Although property crimes are lower than the national average, a Mount Pleasant resident is

twice as likely to be murdered, four times as likely to be raped, and seven times as likely to be robbed than the average United States citizen [10].

While crime rates have substantially decreased over the past two decades a study found many victims are not reporting crimes. According to the Pew Research Center study, as many as two-thirds of crimes may be going unreported [6]. To continue to lower crime rates and empower victims of crime who do not feel confident reporting crimes to police, a self-defense class in virtual reality, dubbed KickVR, was created to simulate scenarios where self-defense might be necessary. When conflict resolution skills are not sufficient in de-escalating a conflict, successfully employing self-defense techniques is necessary to protect vulnerable populations like young women. One study at the University of Oregon found positive benefits for female college students who completed a self-defense class [15]. In fact, a national self-defense program which has had over 60,000 people complete their class reported that 97 percent of their graduates were able to fight off their attacker [9].

Using these programs as a model for the ideal self-defense class, we partnered with the Mount Pleasant Police Department to identify aspects of self-defense training that would contribute to successful skill development. Police Officer Justin Nau recommended teaching Krav Maga fighting techniques as they are the techniques that are used in many police and military organizations and are currently taught to sororities that opt-in to training offered by the Mount Pleasant Police Department. He also identified high-risk areas for crime in Mount Pleasant that we could virtually recreate including the local bar called O'Kelly's Sports Bar & Grill, Main Street which hosts a large majority of Central Michigan University's fraternities and sororities, and walking back to the car at night from the library or grocery store. These scenarios were generalized for a wider audience to use for skill development nationwide in hopes that anybody could download KickVR and learn skill development to defend themselves or assist others from becoming victims of violence.

## 2 System Design

KickVR is a virtual reality simulation, which is "a simulated experience that can be similar to or completely different from the real world" [14]. In this situation, the virtual reality

\*Department of Computer Science, Pearce Hall 410. Email: [apo1j,redei1a]@cmich.edu

simulation is meant to be as real as possible to master a difficult, potentially life-threatening task in a safe environment. During the creation of KickVR, it was easy to access resources necessary to develop the virtual reality simulation, but it was difficult to access the technology required to run and test the simulation. The Oculus Rift CV1 requires the user to have the following minimum system requirements: Intel i3-6100 / AMD Ryzen 3 1200, FX4350 or greater processor, NVIDIA GTX 1050 Ti / AMD Radeon RX 470 or greater graphic card, and 8GB+ RAM. However, to maximize the graphics and processing speed of the Oculus Rift CV1, the user must have the recommended system requirements: Intel i5-4590 / AMD Ryzen 5 1500X or greater processor, NVIDIA GTX 1060 / AMD Radeon RX 480 or greater graphics card, and 8GB+ RAM [5].

By default, the first generation of Oculus Rift headsets comes with two hand controllers. While they are easily tracked within the play-space the user sets up through the Oculus' Guardian System, it reduces the immersion and realism of the simulation. In a real-world situation where self-defense may be necessary, the user will not have hand controllers. The absence of hand controllers in combination with quality, supported hardware systems will allow for deeper immersion and a more quality experience. Instead, KickVR utilizes a Leap Motion Controller mounted on the Oculus headset which tracks the movement of the user's hands in real life to project in the simulation. One limitation of the Leap Motion Controller is that tracking is very limited to only in front of the user's headset. This was an acceptable limitation for this simulation since most self-defense moves are meant to be performed head-on and only used enough to incapacitate or escape the conflict. However, the Oculus Quest 2 that was released to the public in October 2020 solved many of these limitations. The Oculus Quest and Quest 2 has built-in hand tracking and an SDK that allows developers to use hand-tracking in their software. Both Quest headsets have four cameras on the four corners of the headset which is much more accurate than the one forward-facing sensor in the Leap Motion sensor. While there are only three default gestures currently supported by v12 Quest software, combining hand tracking and head mounted virtual reality display should reduce latency and thus improve realism [11].

KickVR also uses applicable, real-life scenarios with instructions from certified Krav Maga experts to guide the users to successfully learn self-defense techniques. The user interface is simple but informative as to not interfere with immersion. Additionally, there are several modules of varying difficulty levels and skill development objectives. The easier modules were developed first with more difficult modules being added incrementally. The module will be comprised of a training portion, an application scenario, and an evaluation of skill development upon completion of the training module with short statistics on that scenario in American society to further enrich and motivate skill development as seen in Figure 1.

The following technical requirements added or being added to KickVR is shown in Table 1 & Table 2. These requirements

were prioritized on a three tier system. The first, level 1 requirements, denoted requirements that were implemented by May 2020. The next level, level 2 requirements, denoted requirements that were implemented by December 2020. The final, level 3 requirements, denotes requirements we are either currently working on or planning for future work.

### 3 User Experience

Hand tracking and hand gestures triggering logic in the simulation are one of the most impressive features of KickVR. Using a Leap Motion Controller mounted to the front of an Oculus Rift CV1 headset, key points like joints and the palm of the hand are found in real life and those positions are replicated in the virtual world. A hand mesh is then projected on those points to make the hand appear in the simulation as seen in Figure 2. Latency between hand movements in real life versus the virtual world were a concern while developing KickVR because small delays in visualization of movements can cause disorientation and motion sickness. Many factors play into response time of hand tracking including hardware, software, and graphics card constraints [1]. The Leap Motion Controller is configured to track a user's hands using three different settings: High-Speed Mode, Balanced Mode, and Precision Mode. High-speed mode focuses on maximizing frames per second while minimizing resolution which is good for games like Drunkn Bar Fight where graphics are not necessary for quality gameplay. On the other hand, precision mode focuses on precise hand movements at the expense of frame rate which would be important for simulations that may require the user to interact with objects in the environment. Balanced mode is the middle ground between the two modes and is the default scanning mode.

At the hardware level, there could be as much as a 15 millisecond time difference between using a USB 2.0 versus USB 3.0 port for data transferring [1]. At the software level, the Leap Motion client on the user's PC handles all processing of data transferred from the USB port and cannot be sped up from the user's perspective. For visualization, latency can be affected by hardware components such as operating system, graphics card, monitor resolution, and physical circuitry of the monitor [1]. The average total processing time for the Leap Motion Controller alone is around 60-70 milliseconds since the Leap Motion Controller only uses USB 2.0 speeds [2] which is satisfactory to prevent motion sickness. More settings can be changed to optimize latency including disabling vertical synchronization in graphics card settings and using a higher refresh rate monitor [2].

Complete system latency is a much larger issue when combining two separate hardware systems to achieve the same goal. While the Leap Motion Controller is used for hand-tracking, the Oculus Rift CV1 is in charge of updating what the user sees in the virtual space based on their real world

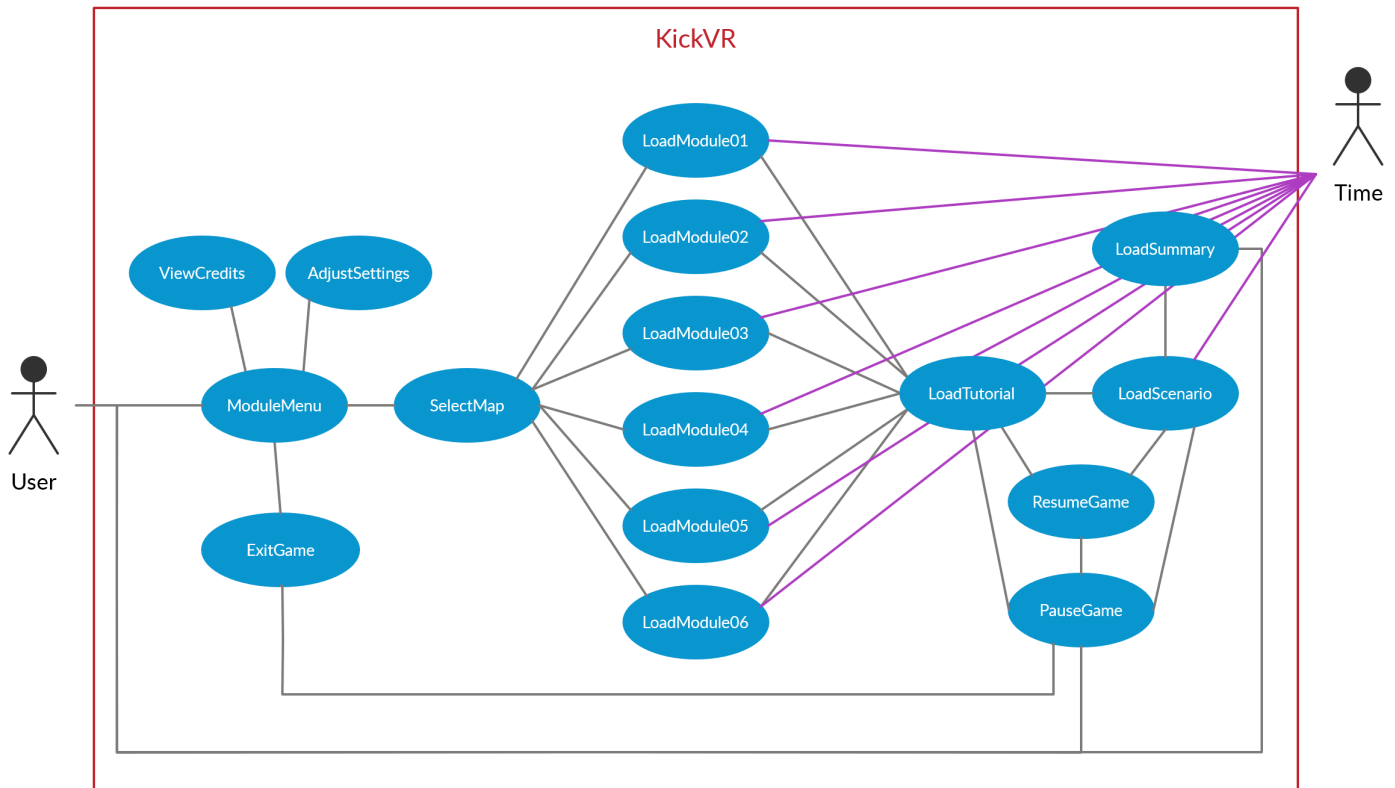


Figure 1: Use case diagram for kickvr

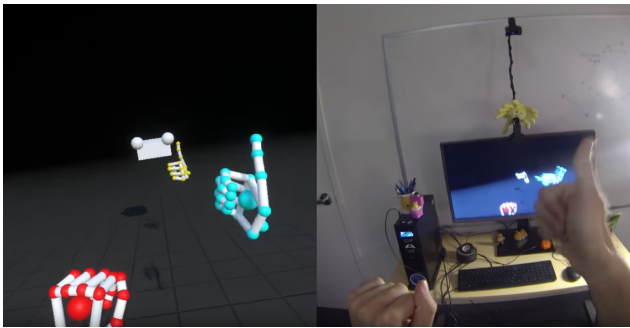


Figure 2: A screenshot of leap motion developers using hand tracking from leap motion's unity core assets[8]

movements. Figure 3 refers to how the user and their Oculus Rift CV1 interact with the KickVR simulation. When actions are made to update the model, view, and controller in KickVR, latency is a factor. The latency between the controller and model via an update request, such as seeing a button appear in the virtual space and the user's hands reacting by moving their hand to click the button, is unavoidable. A joint research project between Microsoft Research and University College London found that the average reaction time when performing a task on a computer was 335 milliseconds whereas the average reaction

time on the Oculus Rift S, the more updated version of the first generation Oculus Rift used for KickVR, was 422 milliseconds [7]. The reaction time of 85 milliseconds was attributed to the average total processing time for the Oculus Rift S alone.

When combined with the latency of the Leap Motion Controller, there could be nearly two-tenths of a second delay between movements the user makes in the real world and movements in the virtual world which increases risk of motion sickness and decreases immersion and realism. As a result, future virtual reality simulations would benefit from an all-in-one virtual reality and hand tracking head mounted display to reduce latency. In December 2019, hand tracking was added to the Oculus Quest and was enabled upon released for the Oculus Quest 2 in October 2020. Cross-platform development for the Oculus Quest and Oculus Quest 2 platforms should be a high priority for KickVR and other virtual reality simulations that use hand tracking to reduce latency and ultimately enhance the user experience.

The user must configure two pieces of hardware for the system. First, the user configures the Leap Motion Controller. Second, the Oculus Rift CV1 "guardian system" is configured, independently from the Leap Motion Controller. Finally, the user executes the simulation, and loads into the virtual hub which acts as a three-dimensional main menu. In this scene, the user can choose which training module and thus which self-

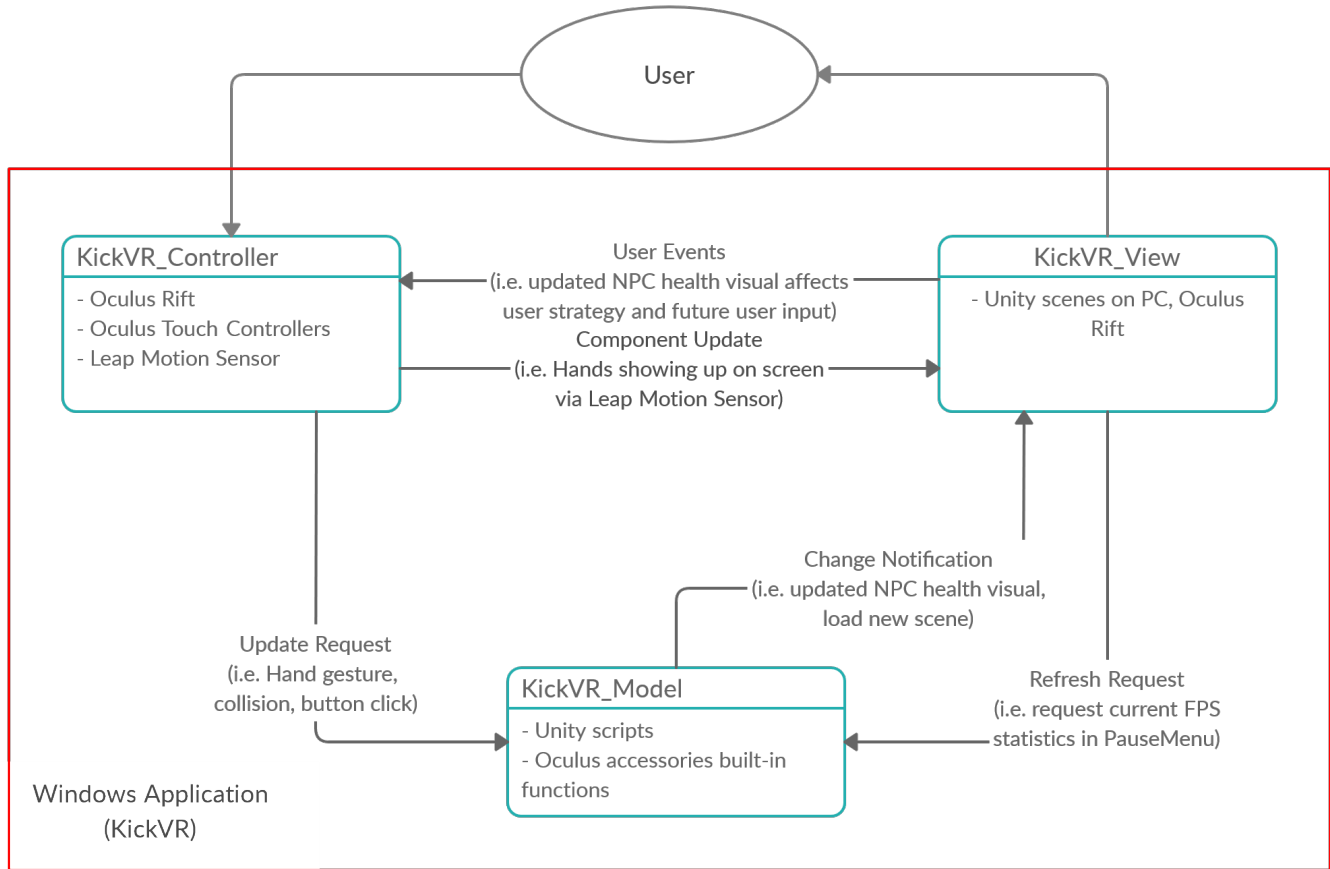


Figure 3: Model-view-controller diagram for kickvr

defense technique they would be interested in learning and can select an environment they would like to apply their skills in as seen in Figure 4.

In each training module, the user completes two segments: a skill development and a skill application segment. The skill development portion is completed first which can be seen in Figure 5. The user will learn that module's skill from a certified Krav Maga expert via video and be able to endlessly practice those skills in a safe gym environment. When they are ready to move on, they use the Leap Motion's hand gestures which take them to the skill application environment they chose in the virtual hub.

There are three environments as seen in Figures 5 and 6: a street scenario at night, a store parking lot scenario at night, and an indoor bar scenario. Additional environments could be added in the future for other common experiences, like being attacked inside a house by an intruder, for user's to have experience with hand-to-hand combat in a wide variety of scenarios. Upon loading into the scenario, the NPC loops through fighting animations such as punching, guarding, and elbowing. The user applies their fighting skills from that training module and previously completed scenarios. The user

is meant to guard themselves from incoming attacks and also attack the NPC when they are vulnerable to a counter attack. The skill application segment ends when the user or NPC loses all of their health. From there, the user is taken to a summary scene which is seen in Figure 7. In this scene, the user can understand all hits they dealt and received via a scoreboard. The point tally is meant to show efficiency of skill application and skill mastery. Higher scores indicate that the user more efficiently utilized their self-defense skills.

The efficiency score is computed via the trigger areas and colliders on the NPC and user models as seen in Figure 8. Trigger areas activate a script when an object comes close to its interaction zone. Colliders, which are slightly smaller than trigger areas, are used as a barrier so models, like the user's hands, do not visually clip into other models, like the NPC's body. Each collider has one trigger. The user has one body and two hand colliders. The NPC model has several colliders including two hand, two arm, one abdomen, one chest, one groin, and one head collider with respective trigger areas. Unless blocked by the user's hand colliders, the NPC deals 50 damage per hit to the user. When a collider on the NPC is struck by the user's hand, a collider script determines which on

Table 1: Functional requirements

Requirement	Priority	Description
FR01	1	The simulation must initially load up to a start screen or main menu.
FR02	1	The user must be able to exit the simulation.
FR03	1	The simulation must have a menu from which the user can select training modules.
FR04	1	The user must be able to control a character in each training module.
FR05	1	The user must be able to receive directions in each training module via signposting and colored visual cues.
FR06	1	The simulation must provide a tutorial and an application in each training module.
FR07	1	The user must receive additional information on that module's self-defense technique or a statistic related to self-defense.
FR08	1	The user must be able to see their physical hands projected virtually in the simulation.
FR09	1	The simulation must have a tutorial training module.
FR10	1	The simulation must have an open-hand strike or punch training module.
FR11	1	The simulation must have an acknowledgment list to give credit to its authors, advisors, and resources.
FR12	2	The simulation should have sound effects and haptic feedback in peripherals to enhance training.
FR13	2	The user should be able to pause and resume their training modules.
FR14	2	The simulation should have a closed-hand strike training module.
FR15	2	The simulation should have an elbow strike training module.
FR16	2	The simulation should grade the user on the effectiveness of their self-defense techniques at the end of each training module.
FR17	3	The user may be able to select the environment they would like to test their skills in during the application portion of their training module.
FR18	3	The simulation may have a grappling training module.
FR19	3	The simulation may have a knee strike training module.
FR20	3	The simulation may have background music samples.
FR21	3	The simulation may have accessibility options to adjust the volume.
FR22	3	The simulation may feature an endless survival mode.
FR23	3	The simulation may feature interactive, physics-driven objects.

Table 2: Non-functional requirements

Requirement	Priority	Description
NFR01	1	The simulation must be playable on Windows platforms.
NFR02	1	The simulation must be developed in Unity.
NFR03	1	The simulation must be compatible with the Oculus Rift/Quest virtual reality system and peripherals.
NFR04	1	The simulation must be compatible with the Leap Motion Controller.
NFR05	1	The simulation must have a functional, aesthetic user interface.
NFR06	2	The latency for loading scenes should not exceed 30 seconds.
NFR07	2	The simulation should have a frame rate of at least 30 frames per second to prevent virtual reality sickness.
NFR08	3	The simulation may be used and enhanced with a haptic vest.
NFR09	3	The simulation may be playable on MacOS platforms.



Figure 4: Module menu (top) and map selection (bottom) in the virtual hub in kickvr

the NPC was struck and increments the respective counter to deduct NPC health. For example, if the user strikes the NPC in the head, it will result in an 80-point decrement of NPC health. Since the head is the most vulnerable part of the human body, which the user learns in the tutorial training module, 80 damage



Figure 5: Training environment in the tutorial module (top) and street scenario (bottom) in kickvr

is the most damage the user can deal with damage scaling down for each body part with the minimum damage dealt being 10 if the user hits the NPC's hands.

To improve the user experience and gauge effectiveness of the virtual reality simulation on skill development, a study was created and approved by the Central Michigan University Institutional Review Board in February 2020. In the study, a user would be given an informed consent form and a pre-test of quantitative survey questions to complete as seen in Table 3. Then, the user would be instructed on how to use the virtual reality equipment and how to exit the simulation if they were to experience motion sickness. The user would start the simulation while investigators observe and ensure they do not exit their play space. Upon completion of the simulation, the user would be given a post-test of both quantitative and qualitative survey questions as seen in Tables 3 & 4. However, due to COVID-19 restrictions at Central Michigan University, this study has been postponed until Central Michigan University deems it safe to study human subjects in an enclosed space.

#### 4 Background

While there are many self-defense resources available over the Internet, none combine the educational, realism, and entertainment components of KickVR. There are plenty of services that offer self-defense training such as free videos on social media platforms like YouTube while other mediums come



Figure 6: Store parking lot scenario (top) and bar scenario (bottom) in kickvr



Figure 7: Skill development summary scene in kickvr

at a price such as physical classes. KickVR combines the benefits of both types of services. With a virtual reality self-defense class, consumers can train through self-paced realistic simulations in the comfort of their own home. Training through the simulation is available 24/7 and users get better, more interactive practice than what passive videos provides. KickVR brings the defense class into living rooms nationwide and help train users to better protect themselves while keeping them in their comfort zone. A survey of our peers found that interest is high in a product like KickVR since self-defense videos garner millions of views and there are many self-defense facilities located around the world. Classes can also be very expensive whereas this product might have a one-time smaller fee similar to a video game or could be a free product.



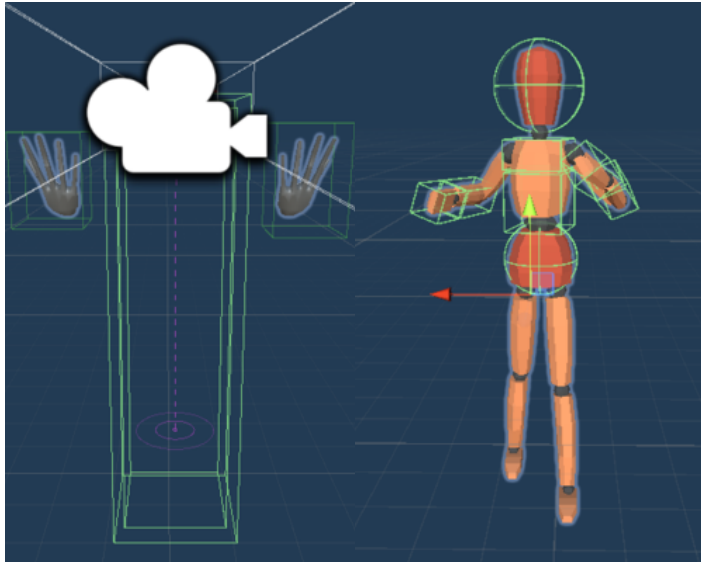


Figure 8: User colliders (left) and npc colliders (right)

Table 3: Quantitative survey questions

Item	Statement
1	I learned something about self defense.
2	I was able to complete the module.
3	I enjoyed the game.
4	I felt challenged by the game.
5	I felt motion sick.
6	I enjoyed the first module.
7	I enjoyed the second module.
8	I found the tutorial to be helpful.
9	I felt the hand tracking was accurate.
10	I enjoyed the combat part of the game.
11	I felt the scoring system was fair.

Table 4: Qualitative survey questions

Item	Question
1	What did you enjoy about the game?
2	Do you see value in virtual reality as a method of training?
3	Do you feel more confident in your ability to defend yourself?
4	What changes would you make to the game to increase your enjoyment?
5	What changes would you make to the game to increase your skill development?
6	Is there any other feedback you would like to add?

Video game marketplaces like Steam feature many self-defense games, but the only similar self defense class that combines gaming and education would be Self-Defense Training Camp, a Microsoft Kinect title that puts players in specific situations and has them mimic the actions on screen to progress through the game. Other video games like Creed:

Rise to Glory and Drunkn Bar Fight seen in Figure 9 bridge the education and video gaming gap, but miss key elements. Creed features realistic boxing combat and boxing technique training, but the goal of the game is to fight for sport and competition. In a real scenario, there are no regulations and someone who is vulnerable needs to be able to protect themselves and counter attacks with every resource available when deescalation is not possible. Drunkn Bar Fight features realistic combat through ragdoll physics and grabbing items in the environment to fight with, but does not teach the user how to properly employ self-defense techniques, solely focusing on video game entertainment. The primary competition are the real physical classes such as Spartan Krav Maga [16] that teach self-defense, and videos that teach self-defense techniques over the Internet. Virtual reality games could also act as secondary competition since this product doubles as a game that is enjoyable to play while informative about self-defense.

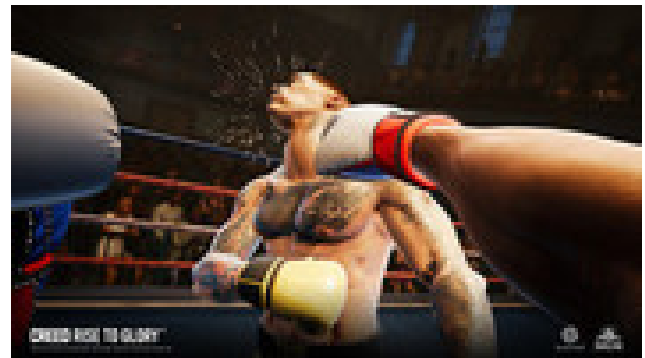


Figure 9: Screenshots of creed: rise to glory [17] (top) and drunkn bar fight [18] (bottom)

On the other hand, self-defense videos are easy to access and can be found online for free. However, they lack real practice or simulation. Viewers might practice the moves shown on screen, but without a scrimmage partner or accountability from an instructor simulating the technique in a realistic scenario, the technique is not developed properly. Physical classes are great for this reason as you can practice the moves taught to participants on other students or the instructor. It can show the user how these self-defense techniques actually work in the real world and let the participant truly practice them. However,

classes can be expensive, require participants to form a schedule around them, show up in person which is difficult in the wake of COVID-19, and can be intimidating for some people, especially those who rarely step foot in a gym or dojo. Our virtual reality self-defense class will let the user choose when to train, and from the comfort of your own home, just like a video. Then, similar to a traditional class, users can practice real-world scenarios through the immersive virtual world. In the end, the virtual reality self-defense simulation doubles as a fun video game and a potentially life-saving course.

Unity and Epic Games' Unreal Engine make virtual reality simulation development possible for most developers. Since Central Michigan University teaches Unity development and has resources and faculty members trained for Unity, KickVR was created in Unity. Unity offers many advantages for VR development including a high definition render pipeline for realistic graphics, spatial audio for positional sounds in the virtual space, and most importantly the ability to build the application for multiple platforms including Android, iOS, Windows, HTC Vive and Oculus Rift [19].

## 5 Future Work and Conclusion

In this paper, we discussed the motivation for creating KickVR, the development process, our findings, our contributions, and its strengths and limitations. In addition, we reviewed the hand-tracking capabilities of the software and offered some insight: providing a mixed-methods interview rubric approach for how virtual reality simulations can be used for training. We strongly believe that self-defense skill development should be prioritized in society. In addition to the physical benefits of self-defense training, it also provides many psychological benefits. One preliminary study found that self-defense training produced increases in self-esteem, perceived control, and confidence while decreasing anxiety, fear, and avoidance behaviors in women who took a self-defense class [3]. KickVR can satisfy both self-defense competency needs and psychological safety needs.

For future work, we are interested in adding features such as new training modules and environments, like the inside of a house. In addition, building an Oculus Quest-compatible simulation is a major priority for better hand-tracking. Hand-tracking was the most difficult part of developing KickVR because of the need to integrate communications between two hardware devices. The Leap Motion Controller was also prone to visual glitches or lack of accuracy due to smudging on the sensor or competitive calls to the CPU. In December 2019, Oculus announced that they have released in-house hand-tracking capabilities for the Oculus Quest through a software update [12]. This reduces overhead price of running KickVR for consumers since there would be no need to purchase a Leap Motion Controller and it would reduce overhead processing power necessary to run both the Oculus Rift CV1 and Leap Motion accessories simultaneously. This could be implemented by updating the Oculus Integration asset package in Unity and

replacing the Leap Motion hand-tracking GameObjects and script logic with the respective Oculus Integration hand-tracking GameObjects.

To increase realism, we are also interested in implementing inverse kinematics (IK) through Unity's physics engine. IK enables the NPC to be effected by a force applied by the user's hand via a punch. For example, a punch to the left side of the NPC's head would result in the NPC's head moving right while not effecting any script logic. We hope to eventually publish KickVR on Steam for widespread public availability.

## References

- [1] Raffi Bedikian. "Understanding Latency: Part 1". *Leap Motion Blog*, Jul 2013. <http://blog.leapmotion.com/understanding-latency-part-1/>, Last Accessed 01/06/21.
- [2] Raffi Bedikian. "Understanding Latency: Part 2". *Leap Motion Blog*, Jul 2013. <http://blog.leapmotion.com/understanding-latency-part-2/>, Last Accessed 01/06/21.
- [3] Leanne R. Brecklin. "Evaluation Outcomes of Self-Defense Training for Women: A Review". *Aggression and Violent Behavior*, 13(1):60–76, Jan 2008. <https://doi.org/10.1016/j.avb.2007.10.001>, Last Accessed 10/08/20.
- [4] Central Michigan University. *On-Campus Student Profile Fall 2019*. Technical report, Central Michigan University, Mount Pleasant, MI, USA, Aug 2019. [https://www.cmich.edu/office\\_provost/academic\\_administration/APA/Reports/Documents/FallStudentEnrollmentProfiles/On-campus/on-campus.student\\_fall\\_profile.2019.pdf](https://www.cmich.edu/office_provost/academic_administration/APA/Reports/Documents/FallStudentEnrollmentProfiles/On-campus/on-campus.student_fall_profile.2019.pdf), Last Accessed 09/13/20.
- [5] Facebook Technologies, LLC. "Oculus Rift and Rift S Minimum Requirements and System Specifications", n.d. <https://support.oculus.com/248749509016567/>, Last Accessed 09/29/20.
- [6] John Gramlich. "5 Facts About Crime in the U.S.". *Pew Research Center*, Oct 2019. <https://www.pewresearch.org/fact-tank/2019/10/17/facts-about-crime-in-the-u-s/>, Last Accessed 09/07/20.
- [7] Robert Gruen, Eyal Ofek, Anthony Steed, Ran Gal, Mike Sinclair, and Mar Gonzalez-Franco. "Measuring System Visual Latency Through Cognitive Latency on Video See-Through AR Devices". In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 791–799, May 2020. <https://doi.org/10.1109/VR46266.2020.00103>, Last Accessed 01/14/20.
- [8] Leap Motion. "Leap Motion Blocks for Oculus Rift Playthrough". YouTube, Feb 2016. [https://www.youtube.com/watch?v=oZ\\_53T2jBGg&feature=emb\\_title](https://www.youtube.com/watch?v=oZ_53T2jBGg&feature=emb_title), Last Accessed 08/16/20.
- [9] Model Mugging Self Defense. "Success Rate of Graduates Fighting Back", n.d. <http://modelmugging.org/suc>

cess-rate-of-graduates-fighting-back/, Last Accessed 09/22/20.

- [10] NeighborhoodScout. "Mount Pleasant, MI Crime Rates", n.d. <https://www.neighborhoodscout.com/mi/mount-pleasant/crime>, Last Accessed 08/23/20.
- [11] Oculus. "Hand Tracking SDK for Oculus Quest Available with v12 Release". *Oculus Developer Blog*, Dec 2019. <https://developer.oculus.com/blog/hand-tracking-sdk-for-oculus-quest-available/>, Last Accessed 01/03/21.
- [12] Oculus. "Thumbs Up: Hand Tracking Available on Oculus Quest This Week", Dec 2019. <https://www.oculus.com/blog/thumbs-up-hand-tracking-now-available-on-oculus-quest/>, Last Accessed 10/03/20.
- [13] Bisma Parvez. "19-Year-Old Freshman Charged After 3 Stabbed at Bar Near Central Michigan University". *Detroit Free Press*, Feb 2020. <https://www.freep.com/story/news/local/michigan/2020/02/25/central-michigan-university-freshman-arraigned-stabbing-wayside-bar/4865968002/>, Last Accessed 08/21/20.
- [14] Bart Pursel. "Chapter 10: Design Methodologies". In *Information, People, and Technology*. Pennsylvania State University, 2005. <https://psu.pb.unizin.org/ist110/chapter/6-3-virtual-reality/>, Last Accessed 11/07/20.
- [15] Lisa Raleigh. "Are Women Safer when They Learn Self-Defense?". *Cascade Magazine*, 2013. <https://cascade.uoregon.edu/spring2013/social-sciences/are-women-safer-when-they-learn-self-defense/>, Last Accessed 10/07/20.
- [16] Spartan Krav Maga. "Store — Spartan Krav Maga", Oct 2016. <https://www.spartankravmaga.com/store/>, Last Accessed 09/17/20.
- [17] Survios. "Creed: Rise to Glory™ on Steam". Steam, Sep 2018. [https://store.steampowered.com/app/804490/Creed\\_Rise\\_to\\_Glory/](https://store.steampowered.com/app/804490/Creed_Rise_to_Glory/), Last Accessed 08/09/20.
- [18] The Munky. "Drunkn Bar Fight on Steam". Steam, Nov 2016. [https://store.steampowered.com/app/528550/Drunkn\\_Bar\\_Fight/](https://store.steampowered.com/app/528550/Drunkn_Bar_Fight/), Last Accessed 08/09/20.
- [19] Unity Technologies. "Virtual Reality Development", n.d. <https://unity.com/unity/features/vr>, Last Accessed 08/18/20.
- [20] Laurel Wamsley, Vanessa Romo, and Parth Shah. "Suspect in Custody After Deadly Shooting at Central Michigan University". *NPR*, Mar 2018. <https://www.npr.org/sections/thetwo-way/2018/03/02/590239044/central-michigan-university-on-lockdown-after-shooting-at-dorm-kills-2>, Last Accessed 08/17/20.



**John Apo** is an undergraduate senior at Central Michigan University where he is pursuing a bachelor's degree in Computer Science. His main research interests include virtual reality simulations and educational Unity applications. He has been working with Dr. Alexander Redei for two years and is currently working with him to install virtual reality capabilities in a full-motion flight simulator.



**Alexander Redei** is an Assistant Professor at Central Michigan University working in the Department of Computer Science. He received his MS and PhD degrees in Computer Science and Engineering in 2013 and 2019, respectively. His interests include flight simulation, software engineering, and human-centered design. His research focuses on using flight simulators to experiment with new techniques for improving pilot training.

# Journal Submission

The International Journal of Computers and Their Applications is published four times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

---

## A. Procedure for Submission of a Technical Paper for Consideration

1. Email your manuscript to the Editor-in-Chief, Dr. Wenying Feng. Email: wfeng@trentu.ca.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.
4. **Note:** Papers shorter than 10 pages long will be returned.

## B. Manuscript Style:

1. **WORD DOCUMENT:** The text should be **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages. Or it can be single spaced double column.  
**LaTeX DOCUMENT:** The text is to be a double column (10 point font) in pdf format.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month and year.
5. The figures are to be integrated in the text after referenced in the text.

## C. Submission of Accepted Manuscripts

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-Chief. If one wished to use LaTeX, please see the corresponding LaTeX template.
2. The submission may be on a CD/DVD or as an email attachment(s). **The following electronic files should be included:**
  - Paper text (required).
  - Bios (required for each author).
  - Author Photos are to be integrated into the text.
  - Figures, Tables, and Illustrations. These should be integrated into the paper text file.
3. **Reminder:** The authors photos and short bios should be integrated into the text at the end of the paper. All figures, tables, and illustrations should be integrated into the text after being mentioned in the text.
4. The final paper should be submitted in (a) pdf AND (b) either Word or LaTeX. For those authors using LaTeX, please follow the guidelines and template.
5. Authors are asked to sign an ISCA copyright form (<http://www.isca-hq.org/j-copyright.htm>), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

## Publication Charges

After a manuscript has been accepted for publication, the contact author will be invoiced a publication charge of **\$500.00 USD** to cover part of the cost of publication. For ISCA members, publication charges are **\$400.00 USD** publication charges are required.

