DEPARTMENT OF COMPUTER SCIENCE

Real Time Image Processing Using Cloud Technologies

Tristan Warren

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Saturday 10th October, 2020

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of MEng in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Tristan Warren, Compiled Saturday 10th October, 2020

Contribution Summary

- In depth research was conducted into Android application design and real time media streaming systems that could facilitate the very low latency needed for a cloud based, real time image processing system. It is expected that future developments in this can use this research as a starting point.

- A cloud based image processing system was developed that allows an Android mobile phone to stream a video feed from its camera to a server where various image processing processes can be applied and then streamed back to the device to be displayed. The system was developed from scratch with no previous work having been conducted into real time, cloud based image processing systems. This proof of concept shows the viability of the paradigm shift and includes the first stages of testing.

- An implementation of the OpenPose library was used to demonstrate the abilities of the system by tracking facial keypoints in real time. This involved compiling the OpenPose project from source and integrating with the cloud based system to complete computations on each image frame.

- A novel approach to testing latency using an augmented reality game was developed to test users acceptance of the system and how latency may affect their use of it. This was also used to test users reactions to artificially added latency.

- Testing of the system was conducted to measure power consumption and how different parameters of the implementation affect the effectiveness of such a system. Data transfer rates were also investigated. To this end a novel profiler was developed to monitor devices wirelessly.

- A research investigation from primary sources was conducted into the impacts that such a system could have on the environment, society, economy and mobile application development as well as potential challenges of such a system and current related technologies. Such research had not been conducted before and this contributed to to new understandings of this topic.

Supporting Technologies

- The WebRTC project was used to allow real time communications through an API. It was used to facilitate image streaming between components of the cloud image processing system [1].

- Parts of the Aiortc library were modified for use as a Python implementation of the WebRTC standard [2].

- An Android WebRTC Implementation by Amr Yousef was modified to enable video streaming to a remote server [3].

- The signalling server used to connect two clients together so that a WebRTC connection could be made was an implementation by Amr Yousef [4].

- The OpenPose deep neural network library was used to demonstrate the capabilities of the system by locating facial keypoints [5].

- The Android OS and Android Studio were used to create the various mobile applications used in this project as well as the Android Debug Bridge to develop a device profiler.

- The OpenCV library was used for benchmarking and testing purposes including the development of an augmented reality task [6].

Mitigation

Due to issues relating to the COVID-19 pandemic not all of the research conducted was completed to the high standard expected. Specific issues are detailed below:

Issue	**Impact on Research**	**Actions to Reduce Impact**	**Remaining Impact**
Lack of available participants for experiments into the effects of latency and user acceptance	Meaningful conclusions are problematic to make with few participants	Using family members as test users	Too few participants and not a representative sample
Loss of time to complete tasks and having to move house	Reduced time to complete development and research including analysis of system	Two week extension to deadline	Still impact from reduction in time
Change of location	Initial design and implementation of system to allow traversal of university's network wasted due to location change	New network navigation system designed	Time lost

Acknowledgements

Firstly I would like to thank Sion Hannuna for supervising this project and helping me to complete this thesis during these challenging times. I am also grateful to Will Hawkins for helping to proof read this thesis and for all the work we have done together over the past four years. Finally I would like to express my thanks to my friends and family for their assistance in this project and for supporting me through my time at university.

Contents

Abstract

Cloud computing has the potential to drastically reduce the energy used in computation whilst at the same time giving the devices that utilise it the ability to perform tasks that would otherwise not be possible. Mobile computing, currently a field that suffers from the lack of performance that the hardware it uses can offer due to size, power and heating constraints, can utilise cloud infrastructure to alleviate these restrictions. Whilst cloud computing has been used in many different applications it is rarely used in tasks that require real time responsiveness from a mobile platform. The only systems that currently take advantage of similar architectures are gaming streaming services. Here a system is proposed to offload computationally expensive image processing tasks from a mobile device and on to a cloud platform. This system is capable of completing processor intensive tasks such as facial keypoint detection using deep neural networks seamlessly on a low powered mobile device. This approach is shown to achieve latency levels that are low enough to be used in augmented reality applications. The proposed shift in paradigm is not limited to portable devices, this methodology can be applied to any low powered device with an internet connection.

Chapter 1

Introduction

Cloud computing is a field that has come to dominance over the past fifteen years with most people using it in some form within every day life and, in a lot of cases, without realising it. Another growing trend is the use of deep neural networks to complete image processing tasks. This method requires a lot of computational power and often necessitates the use of a dedicated graphics processor which limits the ability of most portable or low powered devices from performing these high intensity functions. By taking advantage of the resources made available by a cloud computing system, users can provision their complex computational image processing work off to a remote server. This means that a mobile device with a strong internet connection, using a cloud system, can now complete a task that would normally be impossible.

Mobile devices will always be less powerful than their desktop counterparts due to the electrical power and size constraints that they have, with the lack of computational resources being reported as one of the main restrictions that affects users and developers of mobile applications [7]. With recent advancements in wireless network technologies such as 5G and WiFi 6, the ability of mobile cloud computing to surpass these limitations has begun to be realised.

Developers utilising mobile cloud technologies open up opportunities for new applications and potential environmental benefits: the reduction of electronic waste, reduced energy usage and an increase in device longevity. Here a system is developed that uses cloud based architecture to complete the image processing task of detecting facial keypoints using OpenPose on a mobile device. This report will focus on the performance of this system and what the benefits and drawbacks of using such a system are, including any environmental and economic factors.

In this introductory chapter the motivation of such a mobile cloud computing system is fully investigated along and any notable challenges and related works are reviewed.

1.1 Motivation

1.1.1 Environmental

Such a mobile cloud system as described in this thesis could have many positive environmental and social benefits:

- *Device Longevity:*
  By using this system, a device that may have become obsolete as technology progresses could continue to be used. For example, a museum using hand held devices to add interactivity to exhibits would have to replace all of their devices if they wanted to add features that required a higher level of computational power, such as augmented reality. Looking at the percentages of the total carbon dioxide emitted by the production and user usage of the iPhone XR (76% production and 19% user use) [8], the Google Pixel 3a (74% prod. and 19% use) [9] and the iPad Air (81% prod. and 13% use) [10] it can be seen that approximately 75-80% of emissions arise from the manufacturing of a device and only about 15-20% from its use. This is corroborated by Apple's 2018 environmental [11] report where 77% of the companies CO_2 came from manufacturing and 17% from product

use. Güvendik [12] also reports that the majority of the impact in various environmental categories such as marine ecotoxicity and land occupation occurs from the production stage. Therefore it can be seen that the largest impact in reducing the negative environmental effects from the life cycle of a device can be found by either reducing the effects of the production stage of the device or by extending that devices' lifespan. It must be noted however that these environmental impact reports come from the manufacturers themselves which, whilst they may be accurate in reporting the emissions from manufacturing and user usage, they mostly assume that each device is recycled correctly and not disposed of in another, less environmentally friendly way such as landfill. This suggests that the environmental impact of the disposal of a device as being higher than detailed in these reports. If this is the case then having a system that extends the life span of a device and thus reduces the number of devices being disposed will have a larger environmental impact than expected.

The environmental impact of device manufacturing is not limited to the emission of carbon into the atmosphere. Güvendik [12] reports that more than 99% of the total metal depletion occurring during the life time of a smart phone comes from the production stage of the device. Not only does the Earth have a limited supply of the elements that go into creating mobile devices but these elements often come from so called 'conflict areas' and fuel armed violence, forced labour, extortion, child labour and sexual violence [13]. Some companies are beginning to wake up to the impacts that the resources used in their products have and many such as Intel have declared their ambitions to remove the use of 'conflict minerals' [14]. This however is proving to be very difficult to achieve due to the inability to trace the origins of these materials. There are many actors in supply chains and even though some refineries refuse to process materials from regions that are known to produce conflict minerals often these minerals are mixed in somewhere along the supply chain [15]. As there is no easy way yet to trace minerals to their source the only sure way to reduce the impact mobile devices have is to reduce the number that are produced.

Currently the expected life span of a smart phone is only 2.7 years with the majority of users replacing devices because of restricted functionality of the old device or the newer model being perceived as better [16]. By implementing systems similar to the one described in this thesis users may have less incentive to buy a new mobile device so often and which will reduce the environmental and social impacts discussed above.

- *Processing Efficiency:*
  Using a centralised processing system that uses a single architecture enables software to be developed that is far more efficient than would be financially viable for mobile developers to produce and test for multiple devices. Utilising cloud architecture also means that hardware used to complete tasks can be used in an ideal environment, for example at optimum temperatures, to ensure maximum efficiency. Components that are used in cloud facilities have few design requirements other than performance, efficiency and cost meaning that they do not need to sacrifice any of these for the sake of aesthetics, weight or size which are qualities often sought after in consumer electronics.

  Andrae [17] reports that GHG emissions can be reduced by 20 to 25% by using a thin client workstation compared to using an equivalent number of desktop machines and that by using tablets this could be reduced by a further 1 to 9%. In a similar study by Maga et al. [18] it was found that taking into account the GHG emission of the entire life cycle of computer components meant that a cloud based system could save more than 65% compared to a standalone desktop. It must be noted that this study was applied to desktop machines and not mobile devices but it is encouraging to see that there are large gains to be made in terms of reducing GHG emissions by using a cloud based system.

  For a cloud based system to be more efficient than running the task on a mobile device the amount of energy used in processing the task on the mobile device has to be more than the energy used to complete the task on a centralised computer combined with the energy needed to transmit said data [19]. With the adoption of 5G mobile phone networks over the next few years not only will connection speeds and latency be reduced but power efficiency will be greatly improved. This is achieved using techniques such as using Microsleep, Discontinuous Reception and Transmission, and a wake-up receiver which can save 20%, 80% and 90% respectively compared to 4G technology [20]. This implies that as data transfer becomes more efficient that moving towards a cloud based

system becomes more environmentally viable.

- *Batteries and Electrical Power:*
  Mobile devices all rely on some form of battery power to operate. This usually comes in the form of a lithium ion battery. As mentioned above there are significant environmental impacts to device manufacturing and lithium production has some significant costs in terms of waste material and water usage [21]. The majority of the worlds supply of lithium comes from Argentina, Bolivia and Chile; areas where most biodiversity is related to wetlands. Future climate models suggest persistent drying tendencies in these areas and as the global need for Lithium rises this has the potential to create conflicts for the use of water [22].

  It is known that lithium ion batteries have a limited number of charge-discharge cycles before their capacity is greatly reduced [23]. The number of charge cycles that a battery may have before it becomes practically unusable is affected in part by storage deterioration (deterioration based on the charge state a battery is kept at) and by the rate at which they are discharged during each cycle. Having a device battery kept near or at 100% capacity will reduce the total capacity more than if it is kept at lower capacities [23], this behaviour is difficult to control as it is mostly due to user habits and how they charge their devices. Discharge rates have previously been shown to drastically affect the capacity of lithium-cobalt batteries [24] which are in the majority of smartphones. However a recent study by Diao et al. [25] found that the discharge rate had little to no effect on the capacity of a lithium-cobalt battery at lower temperatures (less than 45°C). They suggest that this is due to recent advancements in battery design. Although the effect that discharge rates have on the lifespan of a battery may be almost negligible it is something that can be controlled, unlike storage deterioration, by minimising the work done by a device. This can be achieved by using cloud based systems to offload power intensive computational tasks; enabling the lifespan of the battery to be prolonged along with the device that it is powering.

  It is estimated that over 6% of electrical energy is lost during transmission and distribution in the electric grid [26]. High voltage power lines with low currents are used to transmit electricity across large distances (for example from a power station to a city) to reduce the energy loss from heating. When power lines reach residential areas the voltage has to be stepped down so that the electricity can be distributed safely, this results in the relative energy loss increasing. Having computational work done in a data centre as opposed to a mobile device results in the electrical power being more efficiently delivered and allows the hardware using it to be optimised to use it more efficiently.

  Energy loss can be clearly observed in the heat given off by the devices used to charge mobile phones. Looking at the environmental reports of the same three devices mentioned above (iPhone XR [8], Google Pixel 3a [9] and iPad Air [10]) shows that the average charging efficiency at different voltages, frequencies and outputs is 73.9%, 84.48% and 79.96% for each device respectively and the average power consumption of their respective chargers is 0.013W, 0.02W and 0.042W when no load is applied. This data reveals that there are large inefficiencies when looking at charging mobile devices and that by reducing the number of times a device has to be charged these can be mitigated. It is worth noting that user habits of leaving chargers plugged in when not in use and overcharging has negligible effects on power consumption and so reducing this has little environmental effect. Determining the exact power usage of a mobile device is difficult as there are large variations in power use between different models of mobile device and the the variation of habits that each user exhibits [27]. Compared to a mobile device it is far easier to model a data centres' power usage. Institutions running data centres monitor power consumption closely in an attempt to reduce running costs and use components that are much more homogeneous making energy usage calculations a lot simpler. Therefore it would be beneficial to move as much power consumption for the use in computational tasks to a data centre to both reduce energy use and make it easier to optimise systems.

  The way that a mobile cloud based system is implemented, and what its use cases are, have a large impact on its energy efficiency. If a device is constantly searching for a connection to a WiFi or mobile internet interface then the battery life of said device is greatly reduced [28]. This means that depending on where a system is used will have a big impact on energy use, for example an office

with a good WiFi connection will use a lot less power than someone on a train constantly searching for mobile internet access points. Along with this, user habits and the constant improvement in the efficiency of mobile technologies, it is not immediately clear whether the overall net energy use will be reduced by switching to a mobile cloud system. This supports the need for research, such as in this report, to be conducted.

1.1.2 Economic

A cloud based system for mobile computation can have economical benefits for both the consumer and producer of mobile devices and systems:

- *Producer Benefits:*
  A product-service system (PSS) is defined as "an integrated product and service offering that delivers value in use. A PSS offers the opportunity to decouple economic success from material consumption and hence reduce the environmental impact of economic activity" [29]. In recent years there has been a significant increase in PSSs which offer a way for companies to reduce their environmental impacts whilst still retaining profitability [30]. Intuitively it may be thought that by producing and selling a smaller number of higher quality products a company will reduce their profits. However, it is possible to achieve a high profit margin by switching to a PSS model by combining products with services [31].

  If a company were to increase the quality and durability of a product then they would expect to see their products being replaced less frequently by newer ones meaning that they have a reduction in sales and are making smaller profits. However, in a PSS, increased product longevity can result in a successful business model and lead to higher satisfaction of customers who regard products as capital assets rather than consumables [31]. A product that utilises a PSS needs to have certain characteristics that enable it to carry out its function at a high level of performance such as a fast internet connection (as in the case of the system developed in this thesis) as well as being durable. A higher quality product means that users are more likely to retain said product as well as continue to use the services that come with it. This enables companies to take advantage of the 'lock-on' effect [32] where customers want the company as their sole choice as well as network effects where an increase in users makes the product more appealing (eg. a social network). This can lead to 'lock-in' where users have little choice in what product to use as everyone else is using the same option.

  Using a cloud based system for completing tasks enables a mobile developer to create applications that would otherwise not be possible. In this thesis OpenPose is used to demonstrate the ability of such a system to complete tasks that would not be feasible on even the best mobile devices. Not only does this allow new applications to be developed but also allows software to run on a much wider range of devices that are only limited by the speed of their internet connection and not by computational abilities. This can greatly improve a products market reach and customer base. A users' propensity to launch an application is dependent on battery level [33], this means that users take into account the effect of an application on battery life and if an application is more power intensive then it is less likely that it will be used. A cloud based system can reduce the power consumption of an application on a device and therefore make users more likely to use it.

  If a company is to move towards a PSS business model then they may expect to retain profitability whilst reducing their production and environmental impacts. Such an economy that relies on services is referred to as a functional economy where the aim is to create the highest possible use value, for the longest time and with the smallest amount of material [34]. The system described in this thesis could be used to help develop a PSS and meet the objectives of a functional economy. It has to be noted that a lot of the studies into PSSs are not based around the fast evolving industry of mobile computing and so it is difficult to say whether these theories are applicable to this scenario.

- *User Benefits:*
  Consumer needs are infinite and so companies are beginning to adapt towards using a PSS model

as a way to overcome the limitations of simply providing and selling products (Kang et al. [31]). What this means for customers is that they are able to pay and use for only the services that they want: they do not have to buy an expensive device that is capable of running computationally intense software if they only want to use it irregularly for one particular task. In effect this makes each device highly customisable to each individual. Payment for each service does not have to even come directly from users but can be generated by secondary sources such as advertisements and data gathered about the user from their use of the service.

Emotional satisfaction and non-functional values that consumers place on a product plays a large part in the value that they place on said product [35]. This includes things such as being able to have the best and latest product or having a device that puts form over function. These user requirements are at odds with a PSS or a functional economy where the objective is to limit the production of products by increasing their life span and adding functionality through provided services.

Utilising a cloud based system for completing computational tasks give users the ability to access services and features that may have only been available if they had bought the the most expensive device on the market. There is also potential for cost saving from the reduction in electricity usage although the estimated cost of powering a mobile phone for a year such as the Google Pixel 3a is very low [9] (only €1.20) meaning that any energy savings have a limited economic benefit to users. Any small costs savings that may be made have the potential to be offset by the need for extra internet connectivity which in certain cases can be very expensive for users [36]. The system developed here offers a test bed for these issues to be researched.

1.2 Similar Technologies

1.2.1 Gaming

Over the past two years products such as Google Stadia and Nvidia GeForce Now have begun to emerge on to the gaming market. These services move the task of running computer games from personal consoles onto remote servers and allows players to stream a game to almost any device that has a screen and a good internet connection. Not only has this shown that this kind of technology is financially viable but it also demonstrates the benefit of being able to run computationally expensive software on many devices that would normally not be able to handle it. These services require very low latency otherwise players will experience sub quality game play when compared to their consoles. This reveals that it is possible to achieve very low latency when streaming media data.

1.2.2 AWS lambda

Amazon Web Services, the leading provider of cloud services, offer AWS lambda, a system that allows a user to run small micro processes on their servers. Products such as Amazon Alexa use this extensively to complete small tasks without the need for users to purchase large and expensive hardware. This is analogous to certain use cases of the system described in this thesis as it could be used to run short image processing jobs. This demonstrates that it is possible for a system to have a short use of cloud infrastructure and still be financially viable.

1.2.3 Image processing as a service

There are many services that offer Image Processing as a Service (IPaaS). The majority of these services are aimed at batch processing images, either as mass data entry with simple image processing or trying to optimise a library of images with compression or automatic image enhancement. This shows that there is a need for cloud based image processing where images need to be processed somewhere else and that optimising a system for mobile use could open up the market to new customers.

1.3 Notable Challenges

1.3.1 Latency

Latency and the effects that it has on gaming, performance in virtual environments and telepresence has been shown to have differing effects depending on the application. Even within the category of gaming there are big differences. In First Person Shooter (FPS) games latency has been shown to have a considerable effect on player performance [37] whereas strategy based games have been shown to be affected a lot less by latency [38].

The extreme effects that latency can have on performance can be seen in studies showing how high latency in a virtual reality (VR) context can cause feelings of sickness and lack of stability [39]. Not only is it of special importance in the case of VR to reduce the latency but the jitter (variation in latency) also needs to be controlled to reduce nausea [40]. Applications that are more immune to the effects of latency include teleconferencing. It has been shown that latency of 250 ms has moderate effects but that if it is greater than two seconds then obvious impacts to conversation are noticed although there isn't a complete communication breakdown [41].

Latency has a considerable effect on how well a system can be used and it is important that this is investigated to discover what an acceptable level of latency is. One of the main challenges of this thesis is to have a latency that is low enough for the system to be used and to evaluate what effect different levels of latency can have on user experience.

1.3.2 Frame Rate

Similarly to latency, frame rate has a significant impact on user acceptance and ability to perform tasks. Janzen et al. [37] show that when completing certain tasks at lower Frames Per Second (FPS) there is a larger impact in user performance when increasing the FPS compared to when the FPS is increased by the same amount at a higher starting FPS. This study looked at frame rates of 15 FPS and higher and the impact on the task of moving target selection. However in the system developed here, user perception of the 'watchability' of the video stream is of more importance and it has been found that having a higher frame rate is not necessarily the most important factor influencing this [42].

Higher frame rates have been show to increase the quality and enjoyment that a user experiences but also have shown not to increase significantly the level of user information assimilation [43] so simply aiming for the highest frame rate may not always be best practice. This is supported by the fact that, when using a small screen, people prefer high-resolution to high frame rate [42]. With a constant bit rate, viewers showed a preference for a frame rate of 15 FPS [44] revealing that this frame rate is what is typically accepted as being of watchable quality. Apteker at al. [45] discovered that users find video played at a frame rate of 15 FPS just about acceptable but at rates lower than this it becomes much less so, at 5 FPS video quality becomes unacceptable. However a lot of these studies were undertaken when online media streaming was in its infancy and, in general, people were used to lower quality and reduced frame rates. It is therefore unknown whether people today would still find a frame rate 15 FPS to be as acceptable.

The main challenge of achieving an acceptable frame rate for video streaming is balancing the FPS against the picture quality of each individual frame. Different applications require different frame rates and whilst there has been research into the use of variable frame rates for different types of scene in cinematic presentations [46] the aim of this thesis is to discover a minimal accepted frame rate and then use that with the highest quality image possible.

1.4 Related Work

Maga et al. [18] looked at the energy and material usage of a desktop computer compared to a thin client computer working with a centralised server. They found that there were significant improvements to the Global Warming Potential (GWP) to be made by using a server and thin client based system. This is a comprehensive study that considers many factors when deciding the GWP of each system including material extraction, manufacturing, distribution and the end of life process and also takes into account

different scenarios of usage for the stand alone desktop computer based on the amount of time that the computer is powered on. Maga concludes that there is significant environmental efficiency improvements to be made by using a thin client and server system which is corroborated by Dasilva et al. [47] and Vereecken et al. [48] who, whilst their work is not as comprehensive as Maga, arrive at the same conclusion. The main criticism of these pieces of research is that they are all at least seven years old and, as acknowledged by Maga et al., the carbon footprint of energy supply and manufacturing is predicted to decrease substantially. This research is still a very strong indicator that development of a cloud based image processing system could have positive effects in the reduction of negative environmental effects that arise from the manufacturing and usage of mobile devices.

Salmerón-Garcia et al. [49] investigate the trade-offs between using cloud based image processing and on board image processing to assist with robotic navigation. They found that remote computation of images from low powered devices was possible but that the ability of a system to process images was limited by the transmission bandwidth from the device to the cloud. They postulate that with the evolution of wireless networks over the next few years that this bottleneck could be greatly reduced. It was also discovered that the amount of computational offloading to the cloud must be carefully analysed to determine the best trade-off between communication and computation. The research outlined in this thesis also aims to determine a balance between using cloud resources and completing computations on device and in what way these may affect energy consumption.

Ablfazli et al. [50] document the challenges involved with the augmentation of cloud technologies and mobile devices. Along with other factors, heterogeneity, both in regards to the hardware and platforms used by mobile devices and also the available wireless technologies, was shown to be a significant performance limiting factor of implementing any mobile cloud infrastructure. Noor et al. [51] concur, in a more recent study, that the ability to get a wide range of hardware and software technologies, data formats and platforms to collaborate with each other provides a notable challenge. Noor also finds that energy efficiency from data transmission can offset the benefits of moving computation to a more efficient data centre. This supports previous sentiments on the importance of balancing the benefits of cloud computation and data transmission.

Matos et al. [7] carry out a sensitivity analysis on mobile cloud computing and find that the battery discharge rate when using 3G or WiFi has the highest impact on the availability of mobile cloud computing. Matos took the approach of mathematically modelling a mobile cloud computing architecture to discover the effects that different parameters have on the system. This method evaluates the potential technical problems of an established system rather than any other issues such as economic or environmental viability. This gives a deeper understanding of which technical areas should be investigated before development commences and is used in this thesis to advise on decisions made in the implementation of the mobile cloud system.

A result reported by many studies into the challenges of cloud computing is that mobile cloud computing security is a significant issue [52] [50] [53] [51]. Khan et al. [53] go so far as to say that security and privacy are one of the main reasons that companies do not want to adopt mobile cloud computing services. This report was published in 2013 and so may not accurately reflect modern viewpoints but nevertheless security still remains a major concern for developers. Taking into account this and all of the other factors mentioned above, it is important that a feasibility study should be used to decide if a cloud based system would be beneficial (Abolfazil et al. [50]). In this thesis a cloud based mobile image processing system is developed to help expand on research into the viability and performance of mobile cloud computing.

1.5 Outline of Objectives

The objectives of this thesis are:

- Conduct research into Android application design and real time video streaming systems that can facilitate the low latency data transfer needed for real time image processing. Research should also be conducted into the environmental, social and economic effects of using a cloud image processing system.

- Develop a cloud based image processing system that allows an Android device to stream a video feed from its camera to a processing server where it has some computationally expensive image processing applied to it and returned within a short time frame.
- Develop novel methods of testing a cloud image processing system including ways to test latency and how users interact with changes in latency.
- Evaluate the performance of the cloud based image processing system with a range of tests including power consumption and data transfer.

In this chapter the potential implications of using a cloud based image processing system were discussed along with a look at currently available systems that perform similar tasks. Notable challenges that need to be overcome were highlighted and reviewed. In the next chapter, these challenges are used to guide technological background research that is used as the basis for the implementation of the cloud system.

Chapter 2

Technology Background

As discussed previously in the introductory chapter, there are many challenges in creating a mobile cloud image processing system such as the one in this thesis. It is therefore important that different technologies are evaluated for their ability to meet said challenges. In this section different technologies are researched, discussed and assessed for their effectiveness in solving the problem of real time cloud based image processing.

2.1 Cloud Computing

The use of cloud computing is on the rise. In 2017 it was reported that 90% of organisations are using some form of cloud service and that the number of workloads running on a hosted cloud service would increase from 45% to 60% [54]. It has also been predicted that the global cloud computing market will grow from $272.0 billion in 2018 to $623.3 billion in 2023 [55]. Businesses have materialised to meet these markets in the form of CPaaS (Communications Platform as a Service) which provision communication services to users as and when they need them. Looking at the growth in this sector it is possible to see that the utilisation of cloud computing to stream media and process it is a financially viable strategy.

The National Institute of Standards and Technology (NIST) definition of cloud computing [56] states that four of the essential characteristics are on-demand self-service (the ability to provision computing capabilities), resource pooling (allowing multiple users to use the same cloud resource), rapid elasticity and broad network access. These are all essential properties which the system developed in this thesis needs to have if it is to be cost and environmentally effective. It is only possible to offer advanced image processing to mobile devices because cloud resources are shared and are able to expand rapidly to accommodate varying usage demands.

2.2 Internet Speeds

Internet connection speeds play a vital role in image transmission quality and overall user acceptance. When streaming video content the aim is to have the highest quality image without the user experiencing any buffering [57]. This becomes a fine balancing act as it is important to have the best quality images, not only for user satisfaction but also, in the case of the system developed here, to have high resolution images that can be inspected, processed and then sent back to the user. If the quality is not high enough then it may not be possible to discover features in an image.

The vast majority of the United Kingdom has 4G mobile internet coverage and most mobile users have access to internet speeds greater than 5Mbps which is enough to stream 720p video using H.264 encoding [57]. This image quality is high enough to enable the system developed here to identify facial keypoints and with the current internet coverage available there is little limitation to where this system can be used. Latency varies widely depending on the internet access type and what infrastructure different internet service providers have [58] meaning that user satisfaction can't be guaranteed in all situations.

2.3 Video Streaming Protocols

2.3.1 Real-Time Messaging Protocol (RTMP)

The RTMP specification, originally developed by Adobe and released in 2012, offers a reliable and low latency way of transmitting audio and video between dedicated streaming technologies such as a streaming server and the Adobe Flash Player. This method delivers streams of data smoothly, with a lot of information and is used widely to stream live video. RTMP is specifically optimised for live transmission of media with the ability to change the video quality automatically as the bandwidth changes. When using RTMP, streaming data is multiplexed meaning that it is possible to achieve higher quality images but at the cost of a latency that is too high to use in the system here.

RTMP was looked at as a potential technology to use in this project due to it being a mature technology of more than ten years and the vast amount of support available. However there are some significant drawbacks with using such a system. RTMP relies on a media server as an intermediary between the video source and the client, this adds cost, mitigates a lot of the benefits of the system described here and is a lot less scalable as well. This technology is designed for one way streaming of data where there is a source and recipient but no response from the user. This means that latency is not the biggest priority and even a highly optimised RTMP system could only hope to get just below two seconds of latency. If RTMP were to be used in the system developed here images would be recorded by the mobile device, sent to the RTMP server, processed and then sent back which would incur a four second latency penalty. In a lot of real time use cases such as mobile image processing this would be very off putting and in some instances completely unusable.

2.3.2 Apple HLS and Low-Latency HLS

Apple HLS (HTTP Live Streaming) is a protocol based on HTTP which does not technically stream media but instead small sections of video are regularly downloaded. Its main aim is to provide high quality video streaming and so has little regard for latency. HLS provides an adaptive bit rate and is widely supported which is useful when developing a system that aims to accommodate as many devices as possible. A variant of HLS exists that is optimised for the task of achieving low latency called Low-Latency HLS which can reduce the latency down to three seconds or less. This is achieved by the media server pushing data to the user instead of the usual approach of the user repeatedly requesting for the next packet which can create a lot of overhead.

Other than the fact that HLS can't get a latency much lower than three seconds, which is not ideal for the system developed here, there are some significant drawbacks to using it. HLS is not designed for real time media creation and streaming. It stores multiple media files of differing quality and bit rate on a server, these files are then also subsequently split into segments typically two to ten seconds long. Clients can request segments of different quality from the server depending on their requirements. Generating segments for even one quality level adds a lot of latency to a system that is already too far from the ideal latency to be used in the system developed here.

2.3.3 Web Real-Time Communication (WebRTC)

WebRTC is an open source project created by Google which provides applications the capability of real time communications through the use of simplified APIs. Initially designed for use in web browsers, WebRTC has since been expanded to be compatible with most mobile devices including both Android and iOS by various groups who have taken advantage of the fact that it is open sourced. It is used throughout industry for real time video communication including platforms such as Facebook Messenger, Discord, Houseparty and Google Stadia.

WebRTC functions differently to the other streaming methods described above as there are no media servers needed to host content. Instead WebRTC works by connecting participants directly to each other so that they communicate directly peer to peer. This allows the latency to reach levels as low as sub 500 ms in good network conditions. Latency is further reduced by utilising the User Datagram Protocol (UDP) transport layer to send data packets. This is used as there is no error checking or packet recovery

and re-transmission which adds large overheads to transmission. Error checking is not vitally important for streaming media data as a loss in these packets will only result in a temporary decrease in quality, which as long as it is not sustained, can be overlooked by a user. Similar to Low-Latency HLS, WebRTC implements a push strategy to reduce latency which means that data packets are constantly pushed to the user without the need for constant requests.

A server is not required to host the media that is being transmitted using a WebRTC system but several small servers are required to pair up clients so that they can establish a peer to peer network connection. Most devices with an internet connection implement a Network Address Translation (NAT) whereby they do not have a unique IP address and rely on a routing device to send them the correct data packets. These routing devices use information in the IP header of each packet to determine the correct destination. One of the servers, known as a STUN server, is responsible for finding a clients public IP address, the type of NAT that they are using and the information needed to send packets to that device. If a peer to peer connection can't be established then another type of server called a TURN server can be used to relay data. Using a TURN server introduces latency and adds large costs to the system as this server is now needed to relay data for the entire duration that communication is taking place. However it is unlikely that a TURN server will be used and so is not evaluated in this research. When a client has received the needed data from the STUN server then a signalling server is needed to connect it to another client and pass details about the type of streaming that is about to take place. Both the signalling server and the STUN server are lightweight and so have little impact on running cost unlike an RTMP or HLS media server which both need to store and stream media.

WebRTC is used in the image processing system implemented in this thesis as it is almost perfectly suited for this type of application with a very low latency, being open sourced allowing many people to build implementations of it and being supported on many mobile devices.

2.4 Codecs

After a video streaming protocol has been chosen there are important implementation factors to consider. One such factor is which codec to use. A codec is a program that is used to compress digital data so that it can be streamed more efficiently. If data is encoded by a device and then sent to another it needs to be able to be decoded to be played back. This means that both parties need to support a particular codec as well as communicate to each other which one they are using. WebRTC supports two main codecs, H.264 and VP8. WebRTC was initially designed to work in browsers and so has all the encoding and decoding performed directly in native code to increase efficiency. However in the use case outlined in this thesis, communication happens between two different non-browser devices and so choosing the correct codec is very important to enable compatibility and high performance.

2.4.1 H.264

H.264 is the most prevalent codec used today and is supported by most devices in one way or another. Historically it was the go to codec for most applications including those based on WebRTC. Apple are strong supporters of the H.264 codec with hardware acceleration for the standard available on the majority of their devices and implemented in native code in the Safari browser. All modern Android devices support H.264 but most do so only as a software implementation which results in extremely inefficient encoding and decoding and so poor power efficiency.

2.4.2 VP8

The VP8 standard has been around for a long time and so very highly optimised. Google, who own and develop VP8, have made it open and royalty free resulting in no royalty fees needing to be paid in order to use it. Unlike H.264, VP8 is not supported by hardware acceleration on Apple devices and has only recently been supported in software. This is in contrast to most Android devices where VP8 is hardware accelerated and H.264 is not.

It is difficult to compare the quality of VP8 and H.264 as it is dependant on the implementation by the device or browser. When choosing which codec to use, the target platform being developed on is the most important thing to consider. Here, Android devices were used to develop the cloud image processing system and so VP8 was chosen. Industry developments in streaming technologies are moving towards AV1, a royalty free and open codec developed by a consortium of companies including Amazon, Apple and Google along with major chip manufacturers. The fact that so many companies that are all competing in the same market have chosen to join together to create one standard strongly suggests that this will be the future for most streaming services. This standard was not incorporated here as although it offers superior performance it is still in its infancy and is not widely supported on mobile devices.

2.5 Android

The Android operating system (OS) makes up over 72% of worldwide mobile phone market share [59] and is set to continue this trend over the next few years. This makes it the ideal target platform to investigate the effects of the cloud image processing system developed here. As there is such a large range of devices running different versions of the Android OS there is potential for testing and evaluating the performance of the system with many different parameters such as camera quality, processor ability and internet connection speeds.

Android mobile phones offer the ability to develop on cheap devices, on an open source platform and have all the necessary hardware packaged into them with convenient APIs. This is much simpler compared to developing a system comprised of custom camera and processing hardware. If a codec is not supported by hardware acceleration then it defaults back to using a software encoder and greatly reduces performance. Due to the vast range of devices that use the Android OS there are many different implementations of hardware acceleration used for codecs. This means that it may be difficult to find representative results when testing performance.

In this chapter an investigation into relevant technologies was completed with justifications for their inclusion within this project. Emphasis was paid to how specific technologies could provide solutions to the challenges discussed in the introduction. In the next chapter a description is given of how these technologies were implemented to create the cloud based image processing system.

Chapter 3

Implementation

This chapter will discuss the implementation of the cloud based image processing system and the reasoning behind any technical decisions that were made. In depth details are given about the development of the system and how the supporting technologies work.

3.1 WebRTC

As discussed previously, WebRTC is a peer to peer system that enables real time communication of audio and video between clients. To initialise a connection between two devices they both must use a set of standards known as Interactive Connectivity Establishment (ICE) protocols. These standards describe how two clients can connect to each other using the shortest path. ICE protocols use third party STUN and TURN servers to help with this. A STUN (Session Traversal Utilities for NAT) server is a server that assists in the traversal of Network Address Translations (NAT) by enabling a client to discover its own public address, port and other information that will enable it to make a direct communication with other devices. If a clients IP address can't be discovered using a standard STUN server, possibly because it uses a particular NAT or firewall setup, then a TURN (Traversal Using Relays around NAT) server may be used which connects two clients by relaying data between them. The STUN server used in this implementation is a public server provided by Google. Once a client has received the information needed from a STUN server then the WebRTC connection process can begin.

There are three components needed to set up a WebRTC connection: two independent devices and a signalling server. This signalling server is used to match up devices and pass information between them in order to establish a link which can then be used to exchange media. The following explains the process of setting up a connection.

1. The caller (A) uses Session Description Protocol (SDP) which is a standard that describes the media content of the connection (such as resolution, formats, encryption, codecs, etc.) to create a message which enables both peers to understand the data that is being transmitted. As 'A' is the device making the call, this message is known as the offer. This offer is sent to the signalling server.
2. The signalling server then passes this offer on to the device 'B' that device 'A' is trying to establish a connection with.
3. Device 'B' then prepares an 'answer' using the SDP which contains all the parameters of the data that it is about to send. This answer is sent back to the signalling server.
4. The signalling server passes this answer back to the original caller 'A'.
5. Once both devices have received each others SDP messages then they can begin to stream data packets between themselves without the need for the signalling server. If however the signalling server fails to create a link between the two devices then a TURN server can be used to relay the data packets.

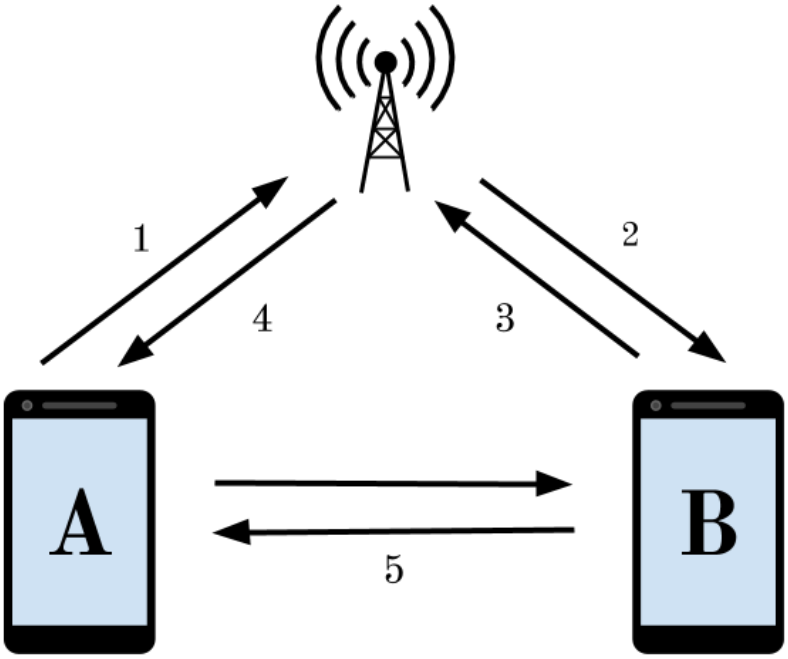


Figure 3.1: WebRTC Initial Connection Process

3.1.1 Mobile Device

A mobile application was developed based on an implementation of WebRTC by Amr Yousef [3]. This app serves the purpose of capturing a video stream from the user using the front facing camera, connecting to a STUN server and establishing a connection to another WebRTC device. The app takes on the roles of creating an offer and sending it to the signalling server whilst also listening for any response answers from the image processing server. The application also displays the video stream that has been returned after the stream has had any image processing performed on it alongside a live feed from the front facing camera so that latency effects can be directly observed.

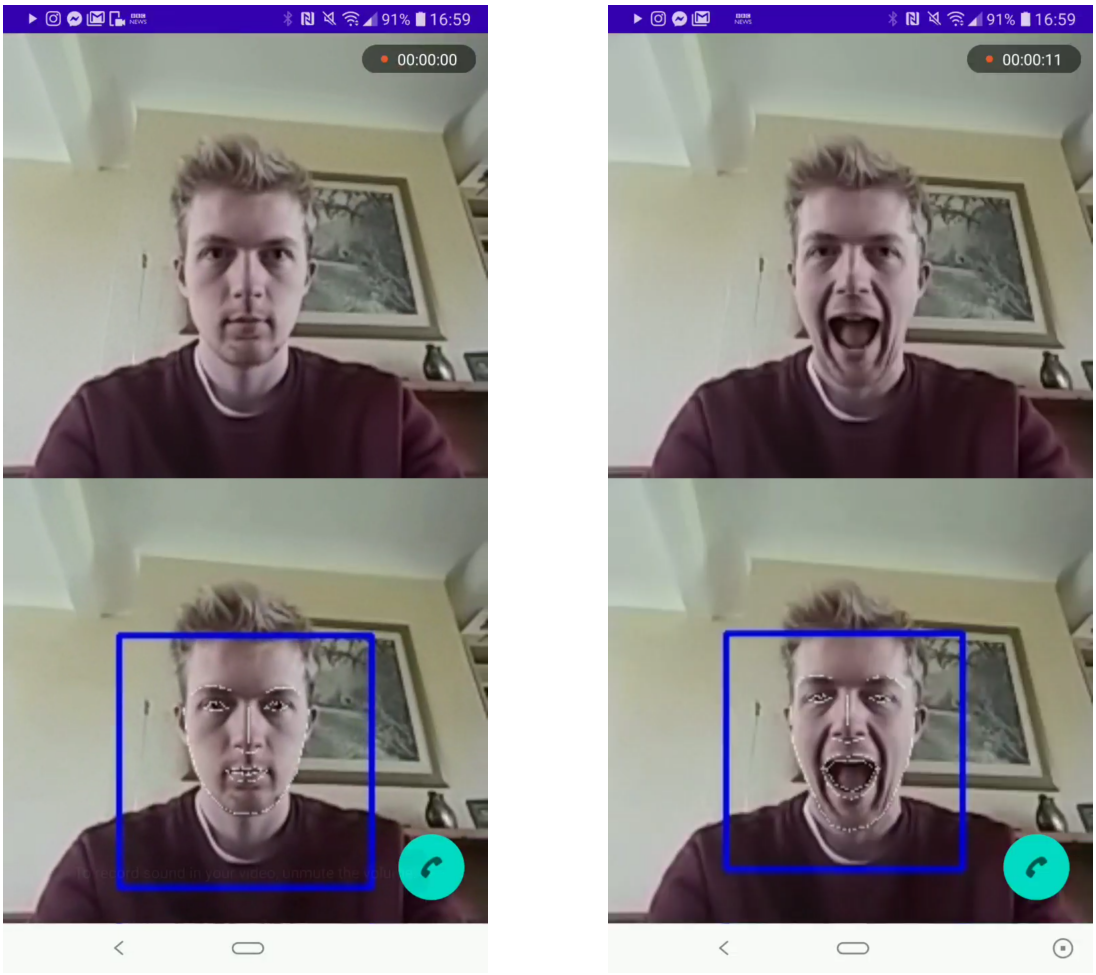
Figure 3.2 shows two examples of the application in use. White keypoints from OpenPose are plotted on the lower half of each image whilst the blue box represents the bounding box of the face.

3.1.2 Signalling Server

The signalling server used in this system is also based on work by Amr Yousef [4]. This is a very simple server that uses the WebSocket protocol to pair up any two clients that connect to it. It passes on any data that it receives on to other devices connected to the server. As the purpose of this system is to take a video feed and process it on a more powerful computer, the server will only receive an offer from a mobile device and an answer from the image processing server. This makes it very easy to pair up mobile devices with a respective instance from the image processing machine.

3.1.3 Image Processing Computer

The computer that is used as the image processing server in this implementation is a desktop computer running Ubuntu 19.04 with an Nvidia GeForce GTX 1080 graphics card, 16GB of RAM and an i7-4790K CPU which makes it very capable of running multiple instances of computationally expensive image processing tasks such as facial keypoint detection using OpenPose. A Python library called Aiortc [2] was used as a base for developing the server side WebRTC client. This library had to be modified to accommodate the differences in the WebRTC implementation between the mobile application and the original implementation by aiortc. These modifications included changing the messaging protocol between aiortc and the signalling server as well as modifying the SDP messages that were sent during the



(a) *Closed Mouth with Feature Detection*

(b) *Open Mouth with Feature Detection*

Figure 3.2: Demonstration of Application in use

initial communication set up. Further modifications had to be done to allow the incoming video stream to also become an outgoing video stream with as little delay as possible as well also allowing time for image processing to take place.

OpenPose [5], was used to demonstrate the capabilities of this system by performing identification of facial keypoints. OpenPose is a real-time multi-person system that uses deep learning techniques to detect human body, hand, facial and foot keypoints in single images. This library was compiled from source to work with the image processing server and uses CUDA (a parallel computing platform) and cuDNN (a GPU accelerated library to help run deep neural networks). The requirements for running OpenPose with the default configuration include an NVIDIA graphics card with at least 1.6GB available and a minimum of 2.5GB of RAM. These specifications are not available on any current mobile device and even the majority of modern laptops would struggle to keep up with the computational demand. OpenPose can be configured to be less computationally expensive but at the cost of reduced detection quality. Reducing the load that OpenPose has on the image processing server will improve the performance of the system but is not taken into account in this projects analysis.

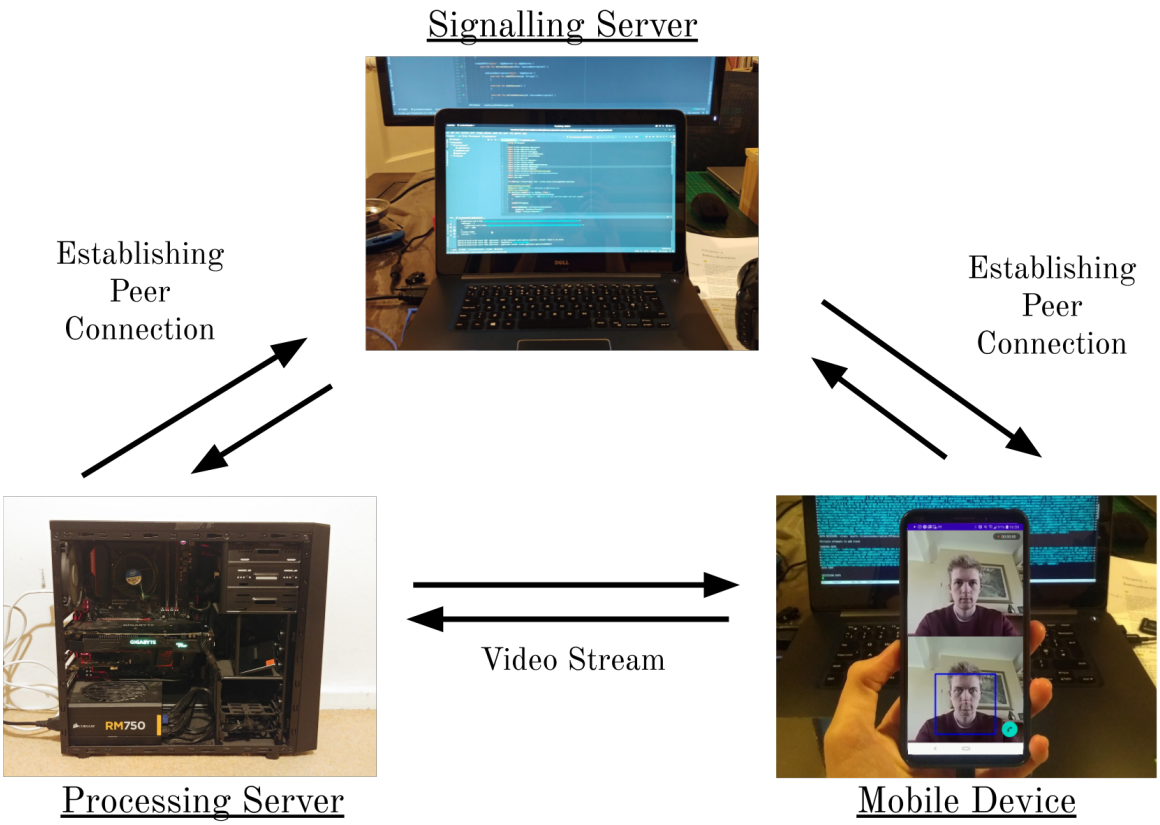


Figure 3.3: System Setup

In this chapter a prototype implementation of the cloud based image processing system described in the project was developed. Demonstrations of the system in use were also shown with an overview of the system setup. In the following chapter the system is tested and evaluated against the project goals.

Chapter 4

Evaluation

The primary objective of this thesis is to demonstrate the ability of a mobile device to complete computationally expensive image processing tasks in real time which would otherwise not be possible to do. In this chapter different performance aspects of this system are tested and evaluated. Secondary objectives such as the reduction in power consumption by devices and the goal of reducing latency are also investigated.

Due to the lack of available alternatives, multiple methods for analysis were conceived and developed. A user interaction testing method was created to explore users interaction with the system and a remote profiler was produced to accurately record device statistics.

4.1 Latency

As discussed in previously, latency is a critical factor to investigate when evaluating the effectiveness of a responsive image streaming system. This is especially true for certain applications such as Augmented Reality (AR) where high levels of latency can severely reduce the ability of participants to complete tasks. Even low levels of latency in applications such as Virtual Reality (VR) can have severe effects including feelings of nausea and the reduction in users ability to stay balanced. This shows the importance of testing any application of this technology to ensure that there are no significant negative effects to users.

In this experiment the effects of latency were measured based on users' ability to complete a task using an augmented reality application that ran using the system developed here. The results from this experiment are used to justify whether such a system is suitable for use with AR.

4.1.1 Methodology

A simple AR game was developed where users would have to move a dot on a mobile device screen along a path presented to them on a monitor. Participants were asked to use the provided mobile device to move the dot into a green starting area. This was done by pointing the device to the area of the screen. Once the dot was centred in the green square a countdown would begin. When the time read zero participants would trace the path to get to the red end square. Users were awarded a score at the end of the challenge based on their completion time and the number of incidents where the the dot left the path. This scoring system took into account both speed and accuracy. A diagram of the setup is shown in Figure 4.1.

Participants were asked to complete the task with four different levels of latency. At the lowest level (L0) the task was run on the mobile device and not using the system developed here. This gave a baseline for latency and the effects the system had in regards to any additional latency. At the next level (L1), the task was run using the cloud system at the lowest latency that it would allow, this level showed the maximum potential of the system. The next two levels (L2 and L3) used the cloud infrastructure but with artificially added latency. It is impossible to quantify the exact difference in latency between levels due to optimisations made by the system; for example the use of a buffer to compensate for different processing delays. L2 and L3 were used to demonstrate the extreme effects that latency would have on

this particular task completion.

Each user was given three practice attempts so that they could familiarise themselves with the game in an effort to reduce the effects of practice. Participants were given the same instructions of keeping behind a certain point so that each of their attempts would not be made easier by the distance between them and the monitor.

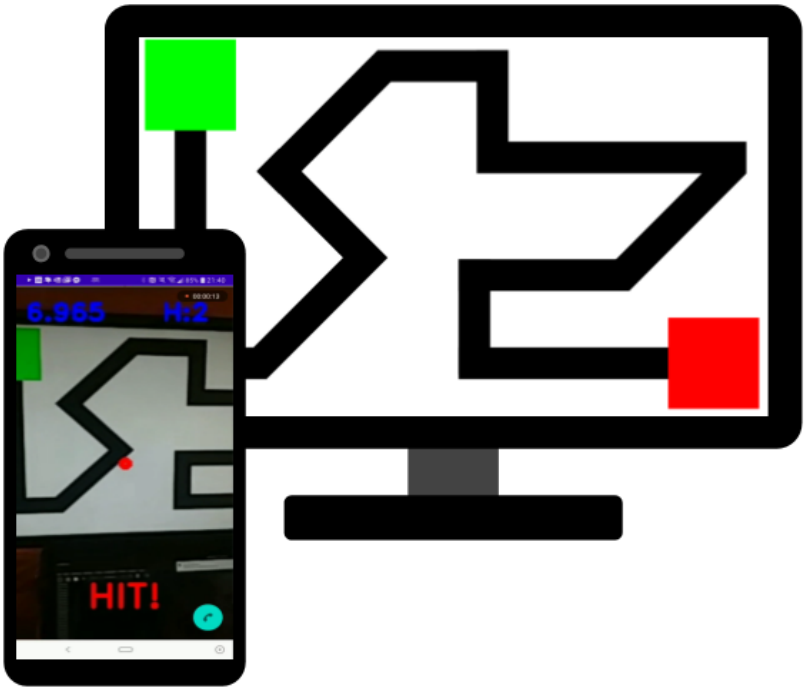


Figure 4.1: Latency Experiment Setup

4.1.2 Results

Each participant has their scores at each level of latency averaged and then for each level the percentage difference is calculated as shown in Figure 4.2. When interpreting the results from this experiment it is important to note that each participant has a different skill level and that at an individual level each person is affected by latency in different ways. This can be seen in the results of some participants where their scores at each level differ from their mean score less than other participants.

From the results seen in Figure 4.2 it can be seen that there is not a statistically significant difference between the scores obtained at L0 and L1. This would suggest that the effect of the latency penalty incurred by using the cloud system has little to no effect on users ability to perform this particular task, and if it does then it is very user dependant.

What can clearly be seen is that at higher levels of latency (L2 and L3) there is a significant decrease in performance for most users. However at L3 these same users performed better than at L2, suggesting that there may be a technique for coping with latency that exceeds a certain level.

When asked about their experience during the experiment, participants expressed how much more difficult the task had become at levels (L2 and L3). It was also noted that some participants thought that L1 was the easiest level. This could be taken as indicative that L0 and L1 were very similar or it could be due to the participants familiarity of the process.

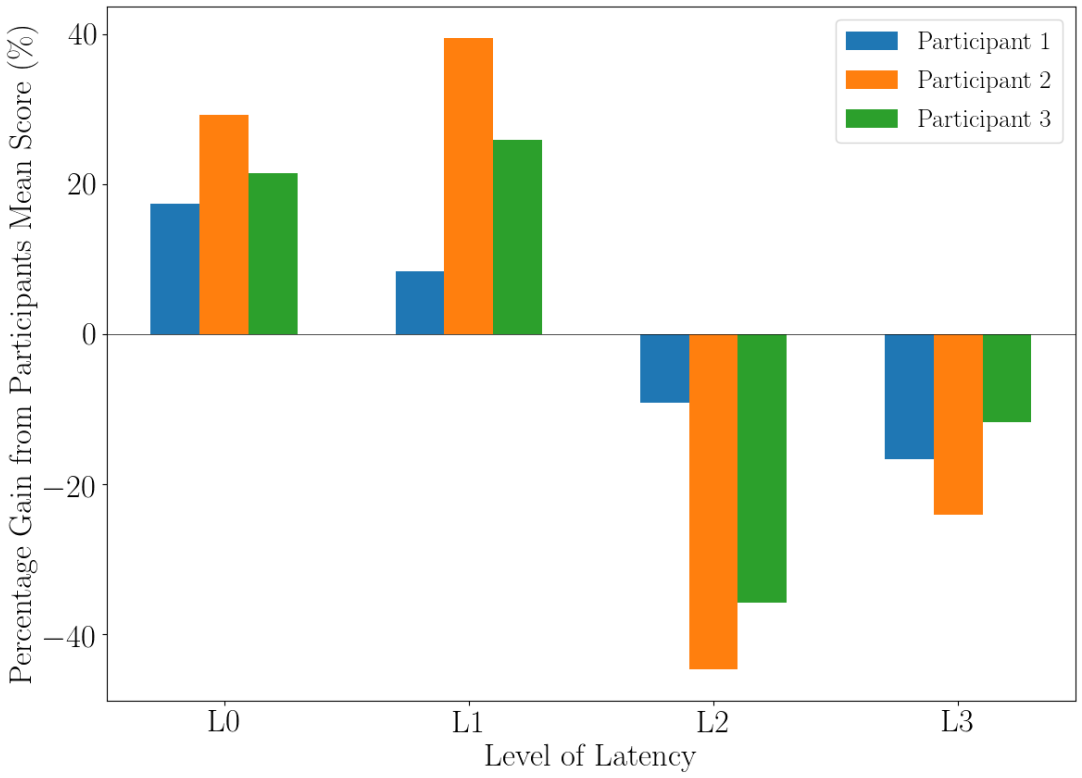


Figure 4.2: User Scores in Task

4.1.3 Conclusions

There is strong evidence that people using the cloud based system suffered no or very few negative effects when trying to complete this task. This suggests that for some augmented reality applications the system developed here could be used. Institutions such as museums and other educational bodies often look at using augmented reality to enhance learning, but with limited budgets it can be difficult for them to justify expensive mobile devices. The experiment conducted here has shown that it is possible to run AR tasks on cheaper devices with little effect on the quality of the service. However it has also shown that users response varies with different levels of latency and that for different AR tasks it is important to investigate these effects to ensure that using this system is still viable. Although these tests may show that task completing is not affected it does come at the cost of reduced picture quality which could be a limiting factor for this system's uses.

		Participant 1	**Participant 2**	**Participant 3**
L0	Time	14.93	23.65	11.08
	Number of Hits	0	2	1
	Score	**8507**	**7035**	**8592**
L1	Time	15.43	24.03	10.86
	Number of Hits	2	0	0
	Score	**7856**	**7596**	**8913**
L2	Time	25.1	45.86	27.56
	Number of Hits	3	8	9
	Score	**7856**	**3013**	**4543**
L3	Time	27.56	46.66	25.50
	Number of Hits	4	4	4
	Score	**6043**	**4133**	**6249**

Table 4.1: Raw Results from User Latency Experiment

4.2 Performance of Codecs

Different mobile devices have different implementations of processes that compress data. These processes are commonly known as codecs. Codecs are essential for media streaming as data needs to be compressed in order to conserve network bandwidth. They are also one of the main consumers of processing and power resources. This means that their power and data usage performance are critical in applications such as the one described in this thesis. If their power usage is too high then devices will not be able to complete continuous image processing tasks for extended time periods. Devices can implement codecs using specially designed parts of their processors, a practice known as hardware acceleration. Alternatively the compression software can be run on the main processor which is less efficient and so consumes more power. The aim of these experiments was to quantify the effect that using a different codec would have on various devices and demonstrate the importance of choosing the correct one when designing a video streaming system.

4.2.1 Methodology

Two tests were used to measure the effects that choosing a particular codec would have on a mobile device. One test measured the loss of charge from a device's battery and the other measured both the incoming and outgoing data. In both sets of experiments the tests were run for 20 minutes. The two codecs that were tested were VP8 and H.264 as these are both supported by the WebRTC standard and are used widely in real world data streaming applications. The two test devices chosen to run the experiments were the LG V30+, a mid ranged phone, and the Moto G6, a lower end device. Both devices have the ability to perform hardware accelerated compression with VP8 but can only complete non-accelerated H.264 compression.

Figure 4.3: Codec Experiment Setup

The same setup was used in both sets of experiments as shown in Figure 4.3. One of the ways that compression algorithms reduce the size of video data is to look for any areas of the image that have changed since the previous frame and then reduce the amount of repeated information being sent. To help ensure consistency for every test, both devices were pointed at a stationary image so that they would be compressing the same image with no movement. Similarly, light levels were kept the same as a change in brightness would change the pixel values of the image being captured and mean the reduction in efficiency of the compression algorithm. In all tests the same network connectivity method was used from the same location to ensure that the effects of different forms of connectivity did not factor into the results. Each test was carried out multiple times for twenty minutes so that the effects of each codec would be more apparent. A profiler was developed that would sample the battery's charge or the amount of data received and transmitted every twenty seconds. This profiler was designed to work wirelessly to

ensure that the devices would not be charging whilst the experiment was taking place.

The power usage test was carried out with devices starting at 90% of their full charge to reduce the possible effects of batteries exhibiting different characteristics at different levels. These could include different levels of accuracy or having different discharge rates. During these tests both phones were kept at 50% brightness to maintain consistency, as a device's screen uses a large percentage of battery charge and different screen types use differing amounts of power. To further maintain consistency, device settings were kept the the same such as Bluetooth and mobile internet being switched off. Similarly any open applications were closed.

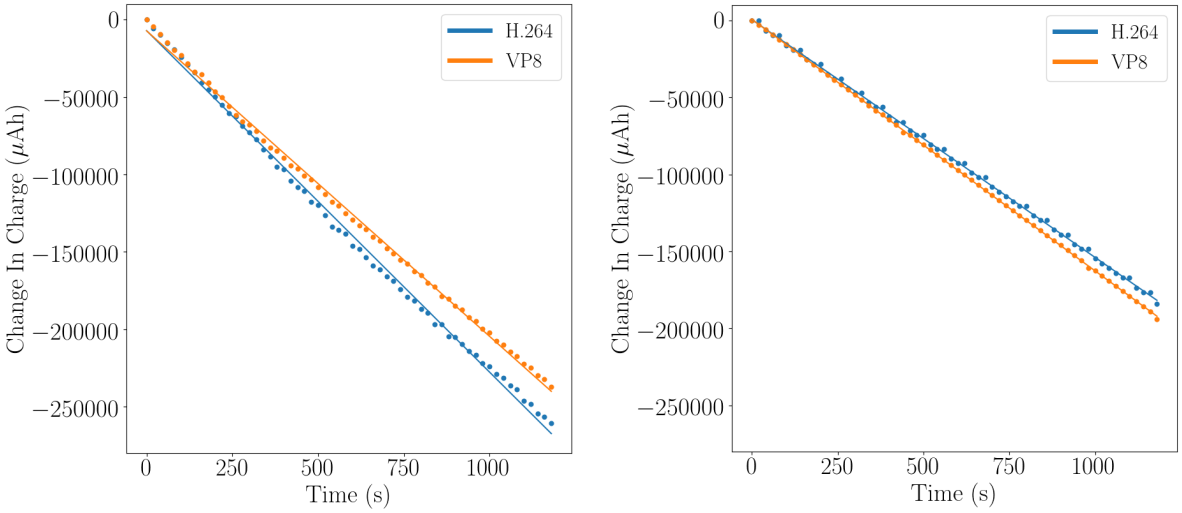
When testing the data transfer rate when using different codecs the mobile devices were kept at 100% battery charge as well as all of the other control variables discussed above. This was to help ensure that any battery conservation techniques that each phone employed did not affect performance.

4.2.2 Results

Both experiments were repeated multiple times and results were averaged to reduce the effects of outliers. These results for battery and data use with each codec can be seen in Figure 4.4 and Figure 4.5 respectively.

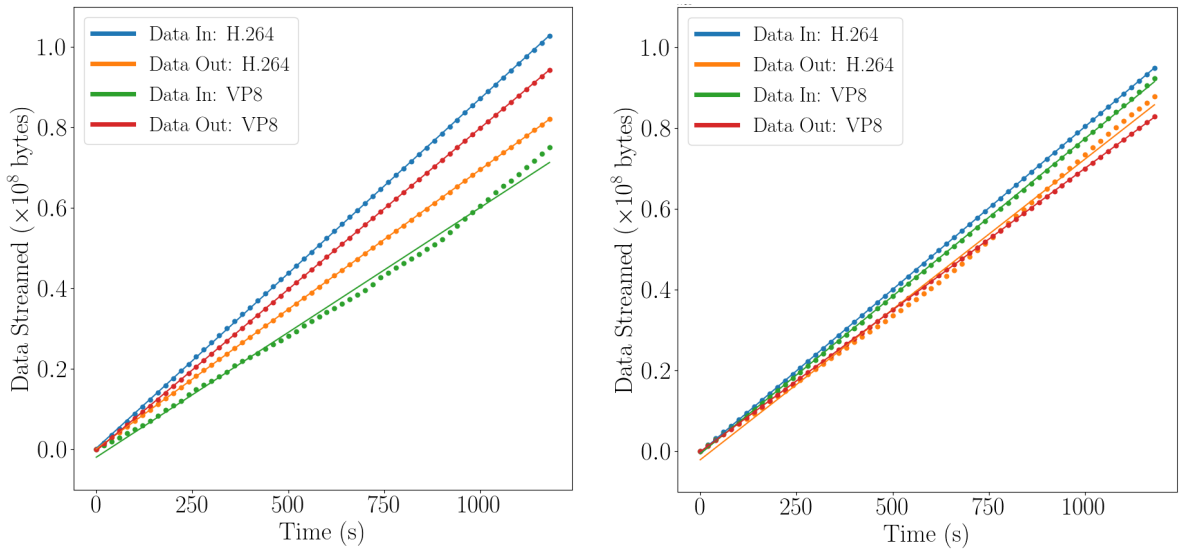
Each device presented very contrasting results when comparing power consumption by different codecs. When running the system using the LG phone there is a significant difference between using the VP8 codec and H.264. Due to VP8 being hardware accelerated this result was as expected and shows that for certain devices there are gains to be made in regards to power consumption by utilising the optimal codec. The Moto phone had a less significant difference in power usage between codecs and unlike the LG phone more power was consumed by the H.264 codec than VP8. The reasoning behind this is difficult to discern but possible explanations may include: each codec compressing data by different amounts, the phone was not making use of hardware acceleration or the data being returned by the image processing having a higher resolution and requiring decompression to be applied to more pixels. The LG device also consumed more battery charge than the Moto. This could be accounted for by the LG's higher quality and larger screen or that the battery has been used more and so holds a charge for a shorter amount of time. This all adds further evidence that codecs need to be thoroughly evaluated and tested for each use case.

Looking at both the data output and power usage by both devices it can be seen that the power consumption by each codec is inversely proportional to the data output; indicating that a codec that uses less power also produces the most data. This suggests that data compression is proportional to power use and that the more power that is used the more the data is compressed. By looking at the amount of data that is output, the maximum amount of data transferred in the twenty minutes was 100 MB which equates to an average speed of 0.6 Mb/s. This speed is far from the maximum potential of either device in these test scenarios. Reducing the amount of compression would also greatly reduce the power consumption and allow the devices to run for longer. It must be noted that the image quality was set to the lowest value. Higher quality video would increase the data being transferred and if compression is reduced, limitations of the network may come into effect. Comparing the spread of battery charge and data for both devices shows a correlation between power consumption and amount of compression. This adds further evidence to the proposition that the level of compression can be adjusted to meet power constraints.



(a) *LG V30+* (b) *Moto G6*

Figure 4.4: Device Power Consumption from the System Using Different Codecs



(a) *LG V30+* (b) *Moto G6*

Figure 4.5: Device Data Usage from the System Using Different Codecs

4.2.3 Conclusions

The twenty minute duration of the test and the varying results from different devices means that there is no definitive answer on which codec performs better in this particular application but it does demonstrate the need for extensive testing for any future uses of this cloud image processing system.

An important conclusion from this investigation is that a balance needs to be made between power usage and compression, the image quality and the amount of data transferred. This balance needs to be based on the constraints of the application being developed. For example, in a situation where internet connectivity is poor a developer may want to reduce image quality or increase power consumption. If image quality is critical to an image processing task then an increase in power consumption or data traffic may be necessary. In an ideal environment an implementation can only choose two out of three characteristics: high image quality, low power consumption and low data transfers. It is impossible to have all three.

4.3 Energy Use

One of the potential benefits of a system such as this is the energy saving capabilities that it could provide. A reduction in the use of energy opens up the possibility of many new applications of image processing technology as well as providing environmental benefits. In this experiment the amount of energy used by a device completing a simple image processing task was measured. The task was completed in two scenarios: a device using the cloud system and the device using its own resources.

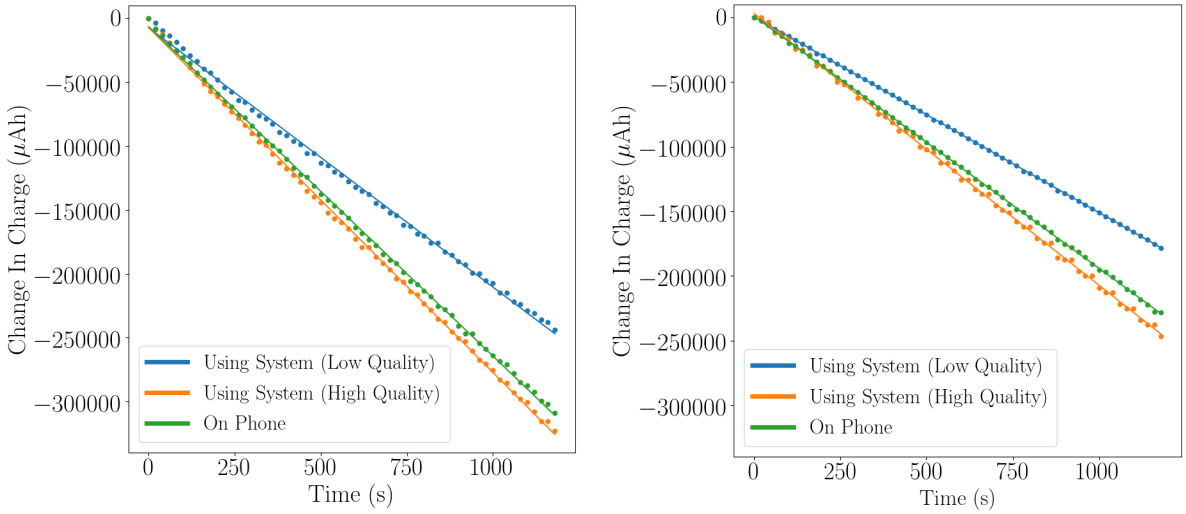
4.3.1 Methodology

The test had a similar design to the previously mentioned experiments into data and power usage with different codecs. Devices were placed facing a static image with all of the same control variables described above. Each test had a duration of 20 minutes and was completed multiple times to average out the effects of any outlier results. The experiment was completed with two different quality levels using the cloud system. This enabled observations to be made on how image quality affects power consumption. The image processing task that was completed was a Canny edge detection. This is a relatively simple image processing task that finds the edges of objects in an image that both the LG and Moto device are easily capable of performing. Examples of the results from this process can be seen in Figure 4.7.

4.3.2 Results

In both test cases where the image quality is set to low, much less power is used to complete this particular image processing task. There is a more significant difference in power consumption between using the cloud system and completing the task using the device when the quality of the image being streamed to the service is lower. When using the system with a high image quality, the power consumption is slightly more than when processing is completed on the device. As discussed in previous experiments this is likely as a result of compression needing to be applied to a larger image.

The difference in power usage between devices is most likely attributed to the LG phone having a larger and higher resolution screen as well as being an older device with a battery that has been through many more charge/discharge cycles. The total power usage by the devices when using the cloud system at low quality is very similar to the power usage in previous experiments into codec power usage where devices were also using low image quality. This shows how the device power usage is independent from the image processing task being completed when using the cloud system.

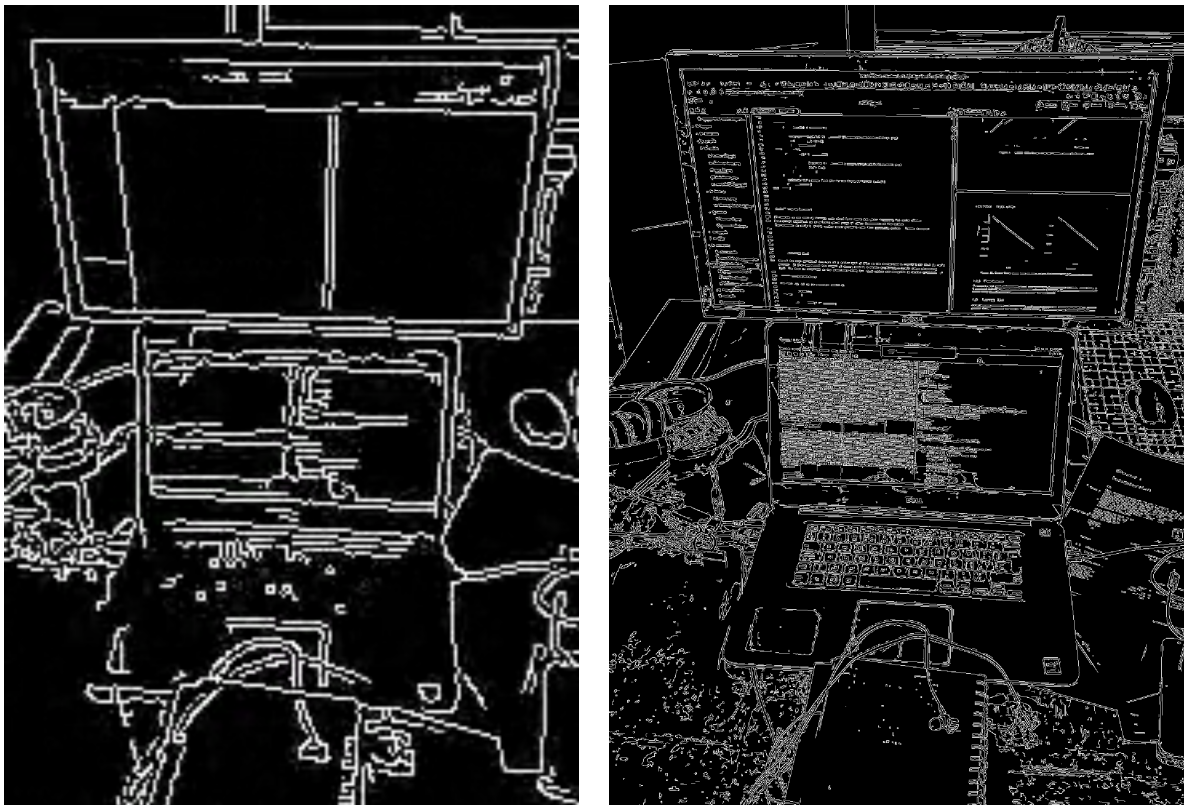


(a) *LG V30+*

(b) *Moto G6*

When the cloud systems image quality is set to low, the difference in quality of processed images is vastly different between the system and using the device as shown in Figure 4.7. This would explain why so

much more power is consumed by processing the image on the device; there are a lot more pixels that need to be processed. If the image quality is increased when using the system, the amount of power consumption is now much more similar to that of running on the device and in this scenario the image processing system performs slightly worse than the on-device processing.



(a) *On System Edge Detection (Low Quality Setting)* (b) *On Device Edge Detection*

Figure 4.7: Sample Images of the Canny Edge Detection from the LG V30+

4.3.3 Conclusions

There are opportunities for significant power savings to be made by using a system such as this. However when the power usage overheads from the signalling server, the transmission of data and the server processing the images are added together, the total energy usage of the system vastly outweighs any energy savings made on the mobile device. However if the aim is to reduce the power consumption of just the mobile device then this becomes a very viable option. Depending on the application, a trade off may have to be made between image quality and ensuring that power usage is low. For certain applications such as using OpenPose for face feature detection lower quality is not a problem but if a good user experience is required then a higher quality image will be necessary. This test only looked at one image processing technique (Canny edge detection), other processes will use different amounts of power. This means that experimentation is needed for every new application that is developed with this system to measure any benefits.

In this chapter the prototype cloud based image processing system was tested and it's performance evaluated. It was found that the system was capable of performing very processor intensive tasks and with low latency. This latency was low enough to enable augmented reality tasks to be completed. The power usage of the system was tested and it was discovered that power usage is very dependant on the use case of the system and implementation parameters.

Chapter 5

Conclusion

Mobile devices are limited in their image processing abilities by their processing power, battery capacity and thermal constraints. If a device is not capable of performing a certain task then a user would have to use a better device or not complete the task at all. The main focus of this thesis was to investigate possible ways to overcome these limitations and to determine the viability of such methods. This included the design and implementation of a cloud based image processing system along with new research methods to test latency in such a system. Research contributions were also made into the field of mobile cloud computing and the benefits of utilising such a system in every day life.

5.1 Project Achievements

This research proposed and delivered a novel approach to completing real time image processing tasks on mobile devices by streaming video frames to a cloud server to execute the image processing task and then streaming back the result. The system developed here successfully demonstrated that it is possible to complete image processing tasks that would normally be impossible by even the most high end mobile device on a low end variant. This was exemplified by utilising the OpenPose library to identify facial keypoints with trained deep neural networks which requires a desktop graphics processor to complete. This system demonstrated that a near real time image stream could be established with very low levels of latency. The ability of a system such as this allows new applications for mobile devices to be developed and can allow current applications to reach a larger market.

Investigations were conducted into the effects of latency and users ability to complete tasks. To this end a test was created that involved the development of a task that participants would complete using an augmented reality application which ran on the cloud based image processing system. This task measured the effects that latency had on participants and compared running the task using the cloud system with running the task on the mobile device. It was discovered that there was little difference in user performance between the two methods but this was at the expense of image quality. When artificial latency was added to the system, users were shown to experience adverse effects when completing a task. It was demonstrated that the system described here is suitable for certain applications but further investigation is needed for each different use case. These low levels of latency make this system suitable for augmented reality applications that can now be run on cheaper devices making them more attainable for groups such as schools with limited budgets.

Tests were conducted into the power consumption of the system with different implementation parameters and comparisons were made between the power usage of a device using the cloud system and a device completing the computations with its own resources. It was found that devices vary widely with their data compression capabilities and that each device needs to be investigated to determine the best compression method to use. It was also discovered that the power usage of a device using the system is dependent on the quality of the image that is being processed. A device that is running a very intensive task will use a lot more power than the cloud system running at a lower quality. It was discussed that implementations of the system that do not involve user acceptance of quality are better suited for the cloud based system. This could include image recognition tasks where a user does not need to see the resulting image. The OpenPose facial keypoint example demonstrated here was able to accurately dis-

cover keypoints at the lowest image quality level.

Data transfer rates were also investigated to determine the potential of the system. It was found that the system was not utilising the maximum bandwidth available and that by reducing the amount of compression applied to the video stream it may be possible to increase image quality without increasing power consumption. To investigate the power consumption and data transfer a profiler was developed to wirelessly monitor devices undergoing testing. Depending on the application and the situation, each implementation of this system needs to make a balance between network usage, image quality and power usage.

It was discussed that the overall energy usage of the system, including power consumption by the internet connection and the image processing server was much higher than any gains made by the device. It must be noted that this system is not the most efficient implementation of a cloud based image processing system as it uses a non-optimal desktop computer as the processing server and non-optimised software. This is far from the optimal situation that most cloud infrastructure is usually operated in, suggesting that with optimisation improvements there is potential for improved energy savings. As the system scales with more users, further energy savings can be made from using only one signalling server for multiple devices.

An in depth research contribution was made to the field of mobile cloud computation, evaluating the effects that such a system may have and investigating any notable challenges that this field has to overcome to be viable. An investigation into the state of cloud image processing and video streaming by reviewing primary sources was also completed and evaluated to help determine the viability of this system. This research gives insights into how cutting edge cloud streaming services are a plausible future technology.

5.2 Future Work

Future work into cloud image processing should focus on three main areas: reduction in latency, the effects of scaling the system and further analysis into power consumption across the whole system.

- The reduction in latency needs to be investigated to enable the implementation of a wider range of applications including Virtual Reality which is known to require extremely low levels of latency. This can be achieved by improving the software running on the image processing server to be optimised for its specific architecture. A field of research that should be investigated and has the potential to reduce latency times drastically is predictive image compression. For certain applications it may be possible to predict what future frames in a video feed will look like. These can then start to be processed before the actual frame arrives at the server. Even if the whole frame is not predictable, sections that have been predicted can save a lot of time processing. Game streaming platform, Google Stadia, have reportedly announced that they will be using controller input prediction to pre-compute frames of games in an attempt to reduce latency [60].
- Research needs to be conducted into the scalability of this system if it is ever to be used in real world situations. To this end an implementation of the technology developed here needs to be created to allow multiple users to access the image processing capabilities of the server at once. Systems need to be developed to efficiently scale up the number of image processing instances and also have them scale down when not in use. This needs to be carefully evaluated in terms of environmental effects and the economic viability of running on major cloud computing infrastructure as opposed to the desktop computer that was used in this research. User habits should also be researched to determine if having a device that can last for longer would reduce the frequency at which they update their devices.
- This project may be regarded as a proof of concept pilot study: demonstrating the viability of real-time image processing using cloud technologies. Future work might more rigorously evaluate sub components of the entire system with further analysis of the cloud based image processing methodology needing to be conducted to fully validate its effectiveness. This should include research into the power consumption of the whole system with various different parameters such as different

mobile operating systems and variable image qualities. A methodology for accurately calculating energy usage should be developed alongside this to allow quick and easy analysis of the effects of modifying the system.

Chapter 6

Bibliography

[1] "Webrtc." [Online]. Available: https://webrtc.org/

[2] "Python webrtc library." [Online]. Available: https://github.com/aiortc/aiortc

[3] A. Yousef, "Base app for android webrtc." [Online]. Available: https://github.com/amrfarid140/webrtc-android-codelab/tree/master/mobile

[4] ——, "Base server for android webrtc." [Online]. Available: https://github.com/amrfarid140/webrtc-android-codelab/tree/master/server

[5] T. S. S.-E. W. H. J. Gines Hidalgo, Zhe Cao and Y. Sheikh, "Openpose library." [Online]. Available: https://github.com/CMU-Perceptual-Computing-Lab/openpose

[6] "Opencv library." [Online]. Available: https://opencv.org/

[7] R. Matos, J. Araujo, D. Oliveira, P. Maciel, and K. Trivedi, "Sensitivity analysis of a hierarchical model of mobile cloud computing," *Simulation Modelling Practice and Theory*, vol. 50, pp. 151–164, 2015.

[8] Apple, "iphone xr - environmental report," Apple, Tech. Rep., 2018. [Online]. Available: https://www.apple.com/environment/pdf/products/iphone/iPhone_XR_PER_sept2018.pdf

[9] Google, "Google pixel 3a product environmental report," Google, Tech. Rep., 2018. [Online]. Available: http://services.google.com/fh/files/misc/pixel3a_productenvironmentreport.pdf

[10] Apple, "ipad air - environmental report," Apple, Tech. Rep., 2018. [Online]. Available: https://www.apple.com/environment/pdf/products/ipad/iPadAir_PER_Mar2019.pdf

[11] ——, "Environmental responsibility report - 2018 progress report, covering fiscal year 2017," Apple, Tech. Rep., 2018. [Online]. Available: https://www.apple.com/environment/pdf/Apple_Environmental_Responsibility_Report_2018.pdf

[12] M. Güvendik, "From smartphone to futurephone: assessing the environmental impacts of different circular economy scenarios of a smartphone using lca," 2014.

[13] S. Reyes, "Conflict free in the drc," *Santa Clara J. Int'l L.*, vol. 17, p. 1, 2019.

[14] Intel, "Conflict minerals report - intel," Intel, Tech. Rep., 2018. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/reports/form-sd-and-conflict-minerals-report.pdf

[15] D. Nathan and S. Sarkar, "Blood on your mobile phone? capturing the gains for artisanal miners, poor workers and women," *Capturing the Gains for Artisanal Miners, Poor Workers and Women (February 23, 2011)*, 2011.

[16] N. Tröger, H. Wieser, and R. Hübner, "Smartphones are replaced more frequently than t-shirts," *Patterns of consumer use and reasons for replacing durable goods. Vienna: Chamber of Labour*, 2017.

[17] A. Andrae, "Comparative micro life cycle assessment of physical and virtual desktops in a cloud computing network with consequential, efficiency, and rebound considerations," *Journal of Green Engineering*, vol. 3, pp. 193–218, 01 2013.

[18] D. Maga, M. Hiebel, and C. Knermann, "Comparison of two ict solutions: desktop pc versus thin client computing," *The International Journal of Life Cycle Assessment*, vol. 18, no. 4, pp. 861–871, 2013.

[19] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing." *HotCloud*, vol. 10, no. 4-4, p. 19, 2010.

[20] M. Lauridsen, G. Berardinelli, F. M. L. Tavares, F. Frederiksen, and P. Mogensen, "Sleep modes for enhanced battery life of 5g mobile terminals," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–6.

[21] V. Flexer, C. F. Baspineiro, and C. I. Galli, "Lithium recovery from brines: A vital raw material for green energies with a potential environmental impact in its mining and processing," *Science of The Total Environment*, vol. 639, pp. 1188–1204, 2018.

[22] A. E. Izquierdo, H. R. Grau, J. Carilla, and E. Casagranda, "Side effects of green technologies: the potential environmental costs of lithium mining on high elevation andean wetlands in the context of climate change," 2015.

[23] K. Takeno, M. Ichimura, K. Takano, and J. Yamaki, "Influence of cycle capacity deterioration and storage capacity deterioration on li-ion batteries used in mobile phones," *Journal of power sources*, vol. 142, no. 1-2, pp. 298–305, 2005.

[24] G. Ning, B. Haran, and B. N. Popov, "Capacity fade study of lithium-ion batteries cycled at high discharge rates," *Journal of Power Sources*, vol. 117, no. 1-2, pp. 160–169, 2003.

[25] W. Diao, S. Saxena, and M. Pecht, "Accelerated cycle life testing and capacity degradation modeling of licoo2-graphite cells," *Journal of Power Sources*, vol. 435, p. 226830, 2019.

[26] H. Naganathan, W. O. Chong, and X. Chen, "Building energy modeling (bem) using clustering algorithms and semi-supervised machine learning approaches," *Automation in Construction*, vol. 72, pp. 187–194, 2016.

[27] J. Suckling and J. Lee, "Redefining scope: the true environmental impact of smartphones?" *The International Journal of Life Cycle Assessment*, vol. 20, no. 8, pp. 1181–1196, 2015.

[28] B. Han, P. Hui, and A. Srinivasan, "Mobile data offloading in metropolitan area networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 14, no. 4, pp. 28–30, 2010.

[29] T. S. Baines, H. W. Lightfoot, S. Evans, A. Neely, R. Greenough, J. Peppard, R. Roy, E. Shehab, A. Braganza, A. Tiwari *et al.*, "State-of-the-art in product-service systems," *Proceedings of the Institution of Mechanical Engineers, Part B: journal of engineering manufacture*, vol. 221, no. 10, pp. 1543–1552, 2007.

[30] F. H. Beuren, M. G. G. Ferreira, and P. A. C. Miguel, "Product-service systems: a literature review on integrated products and services," *Journal of cleaner production*, vol. 47, pp. 222–231, 2013.

[31] M.-J. Kang and R. Wimmer, "Product service systems as systemic cures for obese consumption and production," *Journal of Cleaner Production*, vol. 16, no. 11, pp. 1146–1152, 2008.

[32] T. Joe and H. F. Montgomery, *Service innovation: Organizational responses to technological opportunities and market imperatives.* World Scientific, 2003, vol. 9.

[33] M. V. Heikkinen, J. K. Nurminen, T. Smura, and H. HäMmäInen, "Energy efficiency of mobile handsets: Measuring user attitudes and behavior," *Telematics and Informatics*, vol. 29, no. 4, pp. 387–399, 2012.

[34] O. K. Mont, "Clarifying the concept of product–service system," *Journal of cleaner production*, vol. 10, no. 3, pp. 237–245, 2002.

[35] J. O'Shaughnessy and N. J. O'Shaughnessy, *The marketing power of emotion*. Oxford University Press, 2002.

[36] "Don't forget flight mode: your mobile can rack up £1,000 of roaming charges, url=https://www.theguardian.com/technology/2020/jan/12/mobile-phone-air-sea-charges-satellite-roaming, author=A Tims, year=2020, month=jan,."

[37] B. F. Janzen and R. J. Teather, "Is 60 fps better than 30? the impact of frame rate and latency on moving target selection," in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, 2014, pp. 1477–1482.

[38] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu, "The effect of latency on user performance in warcraft iii," in *Proceedings of the 2nd workshop on Network and system support for games*, 2003, pp. 3–14.

[39] S. Kawamura and R. Kijima, "Effect of head mounted display latency on human stability during quiescent standing on one foot," in *2016 IEEE Virtual Reality (VR)*. IEEE, 2016, pp. 199–200.

[40] J.-P. Stauffert, F. Niebling, and M. E. Latoschik, "Effects of latency jitter on simulator sickness in a search task," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2018, pp. 121–127.

[41] E. Geelhoed, A. Parker, D. J. Williams, and M. Groen, "Effects of latency on telepresence," *HP Labs Technical Report: HPL-2009-120*, 2009.

[42] J. D. McCarthy, M. A. Sasse, and D. Miras, "Sharp or smooth? comparing the effects of quantization vs. frame rate for streamed video," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004, pp. 535–542.

[43] S. R. Gulliver and G. Ghinea, "Changing frame rate, changing satisfaction?[multimedia quality of perception]," in *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, vol. 1. IEEE, 2004, pp. 177–180.

[44] G. Yadavalli, M. Masry, and S. S. Hemami, "Frame rate preferences in low bit rate video," in *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, vol. 1. IEEE, 2003, pp. I–441.

[45] R. T. Apteker, J. A. Fisher, V. S. Kisimov, and H. Neishlos, "Video acceptability and frame rate," *IEEE multimedia*, vol. 2, no. 3, pp. 32–40, 1995.

[46] T. Ryan, "Variable frame rate display for cinematic presentations," *SMPTE Motion Imaging Journal*, vol. 128, no. 2, pp. 10–14, 2019.

[47] D.-A. Dasilva, L. Liu, N. Bessis, and Y. Zhan, "Enabling green it through building a virtual desktop infrastructure," in *2012 Eighth International Conference on Semantics, Knowledge and Grids*. IEEE, 2012, pp. 32–38.

[48] W. Vereecken, L. Deboosere, P. Simoens, B. Vermeulen, D. Colle, C. Develder, M. Pickavet, B. Dhoedt, and P. Demeester, "Power efficiency of thin clients," *European Transactions on Telecommunications*, vol. 21, no. 6, pp. 479–490, 2010.

[49] J. Salmerón-Garcı, P. Inigo-Blasco, F. Dı, D. Cagigas-Muniz *et al.*, "A tradeoff analysis of a cloud-based robot navigation assistant using stereo image processing," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 444–454, 2015.

[50] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 337–368, 2013.

[51] T. H. Noor, S. Zeadally, A. Alfazi, and Q. Z. Sheng, "Mobile cloud computing: Challenges and future research directions," *Journal of Network and Computer Applications*, vol. 115, pp. 70–85, 2018.

[52] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead," *Journal of Network and Computer Applications*, vol. 84, pp. 38–54, 2017.

[53] A. N. Khan, M. M. Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1278–1299, 2013.

[54] . Research, "69% of enterprises will have multi-cloud/hybrid it environments by 2019, but greater choice brings excessive complexity." [Online]. Available: https://451research.com/images/Marketing/press_releases/Pre_Re-Invent_2018_press_release_final_11_22.pdf

[55] "Cloud computing market by service, deployment model, organization size, workload, vertical and region - global forecast to 2023." [Online]. Available: https://www.reportlinker.com/p05749258/Cloud-Computing-Market-by-Service-Deployment-Model-Organization-Size-Workload-Vertical-And-Region-Global-Forecast-to.html

[56] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.

[57] J. R. C. Patterson, "Video encoding settings for h. 264 excellence," *Technical note, Lighterra, Surfers Paradise. URL http://www. lighterra. com/papers/videoencodingh264*, 2012.

[58] J. Sommers and P. Barford, "Cell vs. wifi: on the performance of metro area mobile connections," in *Proceedings of the 2012 Internet Measurement Conference*, 2012, pp. 301–314.

[59] "Mobile phone operating system market share," march 2020. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide

[60] T. McKay, "Google suggests stadia will somehow achieve 'negative latency,' perhaps via predictive button input," oct 2019. [Online]. Available: https://gizmodo.com/google-suggests-stadia-will-somehow-achieve-negative-la-1838928452