A convolutional neural network to classify the American Sign Language fingerspelling alphabet for use in an interactive learning application.

[redacted]

Supervisor: Judita Preiss

2020-2021

Final Year Project Report

BSc (Hons) Computer Science

University of Salford

Abstract

Online learners in the field of interactive topics, such as learning sign language, are hindered in the learning process by generic instructor feedback. This barrier to learning discourages practice and in turn produces and ineffective learning environment. Therefore, this work aims to improve interactivity and provide customised feedback to users in the field of learning sign language online through the use of computer vison and deep learning techniques, in an attempt to find out whether they are able to improve the overall online learning experience. An educational web application was developed which utilises convolutional neural networks to try to classify American Sign Language (ASL) letters from a live webcam. The web application told user through visual and written techniques how to correctly perform the signs, allowing the user to interactively practice and learn at the same time. Access to the application was provided to a group of testers to carry out a series of user tests. User feedback showed that 83% of users strongly preferred this method of learning over the traditional techniques, 92% of users feeling as though after 15 minutes of testing with the application they had developed a strong understanding of the subject of ASL and 83% of users feeling as though they have greatly benefitted from the interactivity of the application. Overall, from these results the conclusion can be made that improving interaction through deep learning techniques is able to better stimulate the learning process.

Contents

Chapter 1 – Introduction 5
1.1 Objectives 5
1.2 Motivation 5
1.3 Approach Adopted 6
Chapter 2 – Literature Review 7
2.1 Online Learning 7
2.2 Current State of Learning ASL Online 8
2.3 Deep Learning in teaching 8
2.4 Using Deep Learning for ASL – hand gesture recognition 9
2.4.1 LeNet-5 9
2.4.2 VGG 10
2.4.3 GoogLeNet 11
Chapter 3 – Methodology 13
3.1 Development Methodology 13
3.2 Programming Tools 13
3.3 Data set 14
3.4 Deep learning - Standards, Algorithms, Techniques 14
3.5 Learning Structure of the tool 15
3.6 Web Tool / Web Hosting 15
3.7 Evaluation 16
Chapter 4 – Requirements Specification and Design 17
4.1 Requirements 17
4.1.1 Use Case Modelling 17
4.1.2 Sign Language Classification module 18
4.1.3 Front-end Integration 19
4.1.3 Requirements list 19
4.2 Design 20
4.2.1 System Architecture 21
4.2.2 Wireframes and Designs 22
Chapter 5 – Development and Implementation 26
5.1 Sprint 1 – Classifier 26

5.2 Sprint 2 – Educational Modes 30
5.3 Sprint 3 – Web Interface 35
Chapter 6 – Testing and Analysis 41
6.1 Testing Strategy 41
6.2 Results 43
Chapter 7 – Critical Evaluation 46
7.1 Review of Project Objectives 46
7.2 Review of Project Plan 47
7.4 Lessons Learnt 48
Chapter 8 – Conclusion 49
8.1 Future Work 50
References 51
APPENDICES 54
Appendix A – Project Logbook 55
Appendix B – Project Proposal 84

Chapter 1 – Introduction

The aim of this project is to create an interactive learning application, that utilises computer vision and deep learning methods, to help people learn and practice the American Sign Language (ASL) fingerspelling alphabet. This would be done through using a webcam to detect gestures the learner is signing and feeding it through a convolutional neural network to attempt to classify the ASL fingerspelling letter, providing feedback to the learner on whether they are signing the letter correctly.

1.1 Objectives

These are the key objectives set out for the project:

- **Objective 1**: Perform a state-of-the- art review to survey the field of research in automatic detection of ASL.
- **Objective 2**: Develop a feature extraction algorithm for the detection of ASL gestures, through using existing techniques.
- **Objective 3**: Develop a CNN for the classification of finger spelling characters. Measured by ensuring the average classification accuracy is at an acceptable level.
- **Objective 4**: Create a Learning System that features multiple modes including: a learning mode, a memory mode and a spelling mode.
- **Objective 5**: Evaluate the tool with automation and human users, measuring the accuracy of the model created.

1.2 Motivation

There are two main reasons why I decided to undertake this particular project, one of which being able to develop my own skills and knowledge in areas of computing that I am captivated by and really eager to learn more about - which are data science, deep learning and computer vision. Seeing the advancements made in recent times using deep learning in areas such as facial detection and autonomous cars shows that it is an exciting field to get into and one that will continue to grow in the years to come. Therefore, it is something I look to become proficient in for the future, so I aim to use this as a learning experience as well as it potentially opening up a career in deep learning and AI.

Other than personal growth, I also see this project as something that could help contribute to an area that I would like to see grow – which is intelligent online learning. I believe that whenever you are learning something it is important that you are able to validate your own knowledge as well as being able to put into practice the things that you have learnt. This is especially relevant for when you are learning sign language, however currently when learning online there is no way that learners could get feedback to validate that what they are signing is correct, individually, learners would have to go out and practice with other people – which in a time like this during a pandemic it is not a totally viable option. So, in

creating something like this I hope that it could push learning online to be more independent and interactive to provide a better learning experience for everyone.

1.3 Approach Adopted

The approach adopted for this project was more of an experimental approach. This means that it will be more focused on the development of an application rather than on the research of technologies. This calls for an incremental workflow to be adopted, which can allow for the effective management of the project and be able to allow for the project to succeed.

Chapter 2 – Literature Review

2.1 Online Learning

Arising from the substantial developments and improved availability of technology over the last decade, the world has seen dramatic growth and demand for online learning (Bederson et al. 2015). It has provided an opportunity for many to receive a high-level education from their homes allowing the freedom to study in their own time and at their own pace – frequently acting as an alternative learning channel versus the traditional face-to-face learning (Yao and Chiang, 2017). However, despite the growing popularity of online learning there still exists challenges when it comes to the overall learning efficiency/experience for students.

Paechter and Maier's (2010) work studies and identifies the strengths and weaknesses of online learning in comparison to face-to-face learning. The study utilises the works of Ehlers (2004), on the desirable characteristic of e-learning; and of Brophy (1999), on educational teaching practices to identify four key principles of effective teaching. These key principles identified are summarised as follows:

1) Learning outcomes: The students are provided with a clear set of aims which they are expected to achieve. Brophy (1999) also emphasises the importance of a clear structure and coherence in the curriculum which allows students to have a clear path to succeed in meeting the learning outcomes.
2) Course resources: The students are provided with coherent and detailed learning materials, which could be in physical or electronic form.
3) Interaction between students and an instructor: It is the role of the instructor to support students in a variety of ways e.g., setting out the aforementioned learning outcomes, administering students with the adequate learning content, producing tailored feedback to students in ways they can improve on their accomplishments and also enabling students to be able to engage in learning activities.
4) Individual learning process: Students have the ability to put into practice what they are learning in a self-regulated manner.

Paechter and Maier (2010) evaluated these key principles with respect to online learning, through surveying students from 29 universities. The results show that the strengths of online learning include the dissemination of learning material as well as the ability to endorse self-regulated learning, mainly due to the ease of access to resource materials which can be worked through in the student's own time. Where online learning falls short is in the interaction between the student and the instructor. Students in the study felt as though the quality of feedback and interaction within activities online were limited in comparison to face-to-face learning, feeling as though feedback was more generic and less tailored to the individual.

2.2 Current State of Learning ASL Online

American Sign Language is used by over approximately 500,000 people (Mitchell et al. 2006) and has been seeing an escalation in the number of speakers and learners over the last two decades, currently having over 100,000 people learning it as a language (Nces, 2016). While primarily the main and most efficient method of learning ASL remains face-to face learning (Andriakopoulou et al. 2007), the number of online learners has risen in conjunction with the growth of online learning. Even though this has made learning ASL more accessible than ever, it is still affected by the pitfalls that come with online learning mentioned previously (Paechter and Maier, 2010) – being affected even more so than other subject areas because of the importance face-to-face communication has to learning ASL (Tigwell et al. 2020).

In recent times the Covid-19 pandemic has caused majority of face-to-face learning to move to a completely online format. This has allowed for the study of how different subject areas are affected by the change in learning style. Tigwell et al.'s (2020) recent work on documenting how ASL learners have had to adjust, verifies the findings of Paechter and Maier (2010) in terms of the limitations of online learning. Tigwell et al. (2020) declared that ASL classes "are typically held face-to-face to increase interactivity and enhance the learning experience" and that from his findings suggests the fails in these departments. It is stated in the work that although this shift to online learning was impromptu, with instructors having little time to prepare, the result concluded that the current landscape of tools and software available was considered "a major limitation in the context of using signed languages".

When surveying the main sources for learning ASL online (NAD, 2020), the main learning medium is video learning. These videos aim to show users what ASL fingerspelling signs look like and have them replicate them (Skillshare, 2020). However, there is no verification from the instructor's side or any feedback to the learning on whether they are signing correctly. This can be deemed inadequate when it comes to teaching, as it violates one of the key principles of learning (Paechter and Maier's 2010), which is interaction with the instructor/teacher – lowering the quality of learning. However, a tool can be proposed that would utilise deep learning and computer vision techniques could be proposed to increase interactivity, with the provision of personalised feedback.

2.3 Deep Learning in teaching

The rise in the use of deep learning and Artificial Intelligence (AI) of recent times has brought about significant transformations of many industries (Dam 2019). By utilising large amounts of historical data in conjunction with complex machine learning algorithms, we have been able to revolutionise technology and deliver "many fundamental breakthroughs in computer vision, speech recognition, and natural language processing" (Dam 2019. p. 34).

Little research has been done, however, when it comes to applying deep learning to the field of online educational tools. This creates a little uncertainty for whether the hypothesis of implementing deep learning techniques to the field of online education, specifically for learning ASL, to improve interaction and quality of learning is one that is able to be proved.

When comparing the trends of online education (Bederson et al. 2015) and deep learning (Dam 2019), it seems inevitable that we would start to see educational tools created which employs deep learning techniques. However, gaging why this form of educational tool hasn't been created or popularised, should be taken into account as it could present certain limitations that could hinder the effectiveness of this type of tool.

It could be implied that a tool such as this would be too computationally expensive to use on the large scale. A potential reason for this may be the computational cost of running deep learning applications on a large scale, especially when trying to classify object real time (Justus et al. 2018). It could also be implied this type of solution may not have been feasible in the past and has been overlooked with the recent advancements in deep learning.

Research has been done, however, on the effects of reciprocal interaction between a human and a computer (Giannakos et al. 2018). Giannakos et al.'s (2018) work on learner-computer interaction focuses on the make-up of creating a learning tool that would aid the human learning process, by making use of "HCI, learning technologies, learning science, data science, psychology" to capture data of the learner "allows us to better understand users' learning capacities and design meaningful experiences for them". Though, this has not been proven through research by Giannakos et al.'s (2018) the research done around the area proved that the idea of an educational tool that enhances the learning experience is one that is supported to be achievable.

2.4 Using Deep Learning for ASL – hand gesture recognition

In terms of the actual implementation of deep learning system, there are three parts that would need to be thoroughly investigated, these include: the hand gesture recognition and neural network architecture, the interface and resources for learning and the implementation, and hosting of the application.

In terms of hand gesture recognition and neural network architecture, substantial developments have been made in the fields of computer vision, image recognition/classification spurred on from the increase and ease of accessibility of processing power, making classification of objects in real time easier than ever (Mitsianis et al. 2018). This also applies to the field of hand gesture recognition which has enriched areas such as "UAV, somatosensory game, sign language recognition" (Sun et al., 2008) - sign language recognition being the area relevant to this project. In order to be able to detect and classify hand gestures, a Convolutional Neural Network (CNN) will be used. There are many ways to implement these neural networks and the network architecture used depends on multiple factors such as the dataset used, the output required and computational power available (Sun et al. 2008).

2.4.1 LeNet-5

One of these architectures that has been proven to be effective for hand gesture classification is the LeNet-5 neural network architecture (Sun et al. 2018). LaNet is a

convolutional neural network architecture originally used to classify handwritten numbers created in 1989 by Yann LeCun (LeCun et al. 1989), LaNet-5 being the latest and most effective iteration. The architecture proposed by LeCun et al. (1998) for LaNet-5 consists of a convolutional layer followed by a pooling layer, both of which being repeated two and a half times.

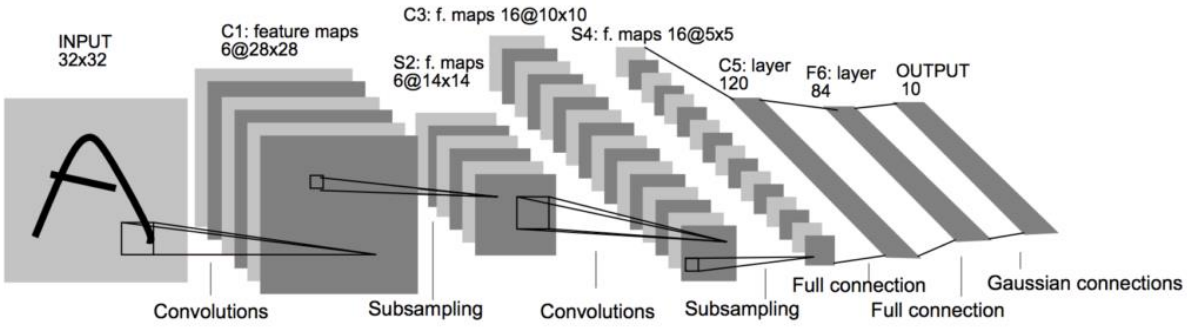


Figure 2.1: *Architecture of LaNet-5 as proposed within the work of LeCun et al. (1998)*

Sun et al. (2018) uses the the typical LaNet-5 network proposed by LeCun (1998). Sun et al. (2018) uses this architecture to be able to classify hand gestured numbers from a live video feed. The results of the research find that the average accuracy of this architecture is 98.3%. Sun et al. (2018) notes that that there was some issue when it comes to hand rotation, with the dataset unable to classify the numbers gestured properly. This is perhaps an issue on training data with the being little variation/rotations of the hand for the algorithm to learn effectively.

2.4.2 VGG

Another viable CNN architecture that can be used for the classification of ASL fingerspelling is the Visual Geometry Group (VGG) architecture. This is a model that was proposed by Karen Simonyan and Andrew Zisserman in 2014, submitted to the “Large Scale Visual Recognition Challenge 2014” (ILSVRC2014) for which is achieved a 92.7% accuracy on the ImageNet dataset (a large visual database designed for use in visual object recognition software research) and is described as a very deep convolutional neural network (Simonyan and Zisserman, 2015).

The architecture consists of passing the input “image through a stack of convolutional (conv.) layers, where we use filters with a very small receptive field: 3×3” also using filters with the size of 1x1 and a stride of 1. “Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2” (Simonyan and Zisserman, 2015). VGG with greater layers stack convolutional layers together before max pooling. Stacking is done to “approximate the effect of one convolutional layer with a larger sized filter, e.g. three stacked convolutional layers with 3×3 filters approximates one convolutional layer with a 7×7 filter” (Brownlee, 2020). This also means as the number of depths increase, the number of filters will also increase, hence it being described as very deep CNN.

The benefit of this network being very deep can mean that it would take less epochs to converge during training. This was proven by Strezoski et al. (2016) who used this architecture to classify hand gestures. The results concluded that when trained for "35 epochs with a batch size set to 128" the model they had used had converged at 17 epochs. In terms of accuracy of the test set VGG scored an accuracy of 84.33% which is high, however this architecture did have one of the slower average classification time and the training duration (2ms and 5mins respectively), out of all the CNNs tested which include GoogLeNet, AlexNet, LeNet-5 and VGG (bearing in mind that a NVidia GTX 980 Ti was used). Despite this Strezoski et al. (2016) states "VGG model had the best ratio of classification accuracy and timing making them most suitable for real time use", making it a serious contender for use in classifying the ASL fingerspelling alphabet for this project.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.2: *Architectures of VGG-11, VGG-13, VGG-16 and VGG-19 as proposed within the work of Simonyan and Zisserman. (2015)*

2.4.3 GoogLeNet

In 2015 Google had published a paper (Szegedy et al. 2015) which documents their new CNN architecture called Inception and new model called GoogLeNet. The main idea for this

architecture was to "consider how an optimal local sparse structure of a convolutional vision network can be approximated and covered by readily available dense components" and similar to VGG it is also consider a very deep CNN. This proposed model managed to achieve top results in the 2014 ILSVRC challenge (ImageNet, 2014) beating out the VGG architecture to achieve 93.33% accuracy.

The architecture of GoogLeNet revolves around the use of multiple 'Inception Modules' which consists of "1×1 convolutions are used to compute reductions before the expensive 3×3 and 5×5 convolutions" as shown in Figure 3. These modules would be stacked and connected to form the GoogLeNet model.

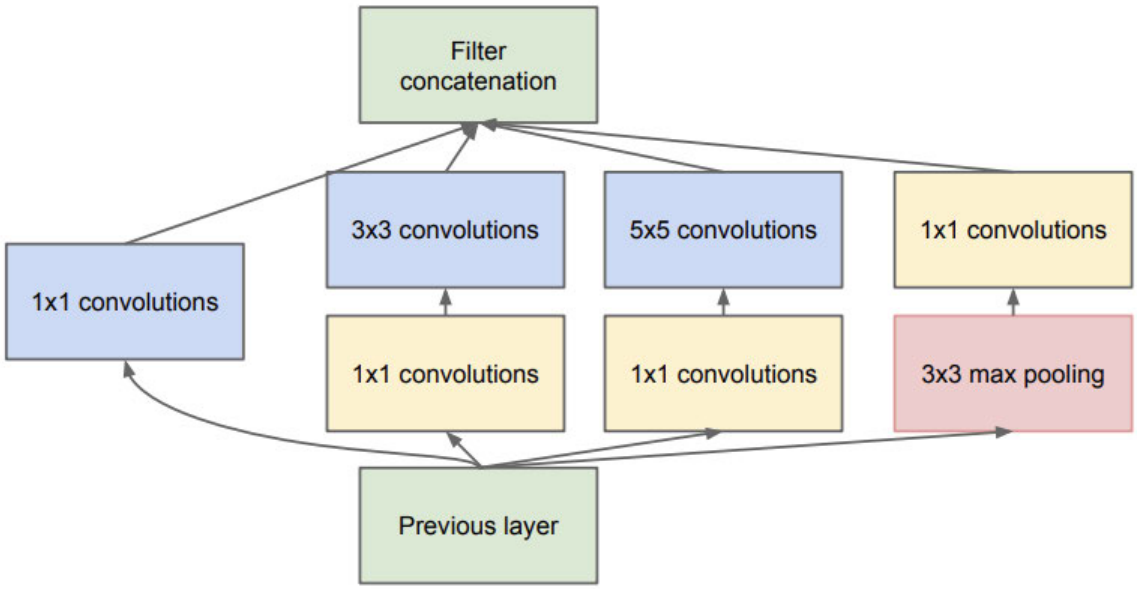


Figure 2.3: *Inception Module with Dimensionality Reduction proposed within the works of Szegedy et al. (2015)*

This model was also used in the work of Strezoski et al. (2016) for the comparison of CNNs on the classification of hand gesture recognition in real time. Strezoski et al.'s (2016) tests find that GoogLeNet had the best accuracy when compared to all the other models (at an accuracy of 90.41%), which is due to the depth and complexity of the model's architecture. However, this depth and accuracy does come at a cost which is the resulting high training time. GoogLeNet also had the highest average classification time for a test image of 2.8ms, making it less ideal for a real time classification tool than the VGG model.

CNN Summary

Overall, in terms of accuracy the VGG model seems to provide the best results based on Strezoski et al.'s (2016) findings. This, however, can be seen as subjective as it may not always be the best providing there are different project objectives and datasets. All this therefore means that experimentation is key in deep learning and utilising to a single model in every instance is not the answer.

Chapter 3 – Methodology

This section will cover the decisions made in terms of how the project will be carried out, providing justification for each of the choices made. It will discuss the development methodology, tools, standards, languages, algorithms, and techniques that will be used to achieve the objectives set out for this project.

3.1 Development Methodology

The development methodology that will be put into practice for this project is the Agile methodology. “Agile is an iterative approach to project management and software development” (Atlassian 2020), it allows for software development projects to be broken down into increments – each of which being able to be planned, implemented and evaluated individually. The benefits of this mean that projects are more flexible and are able to react to any issues or changes in requirements, that arise, more easily. This can also mean the risk involved can be minimised as adjustments to increments can be made simply (Beck and Andres, 2004).

The attributes mentioned above are vital when related to this specific project. This is because this project is heavily focused on creating multiple pieces of software that are all able to communicate with each other. Breaking the project down, simplifying the tasks allow for a more focused and productive workflow. The ability to plan more flexibly is also very important as when working in a complex field such as deep learning, technical issues can arise that could need a change in requirements and with the use of Agile, these changes can be made effectively.

3.2 Programming Tools

In developing the application, Python was chosen to be the primary programming language, specifically Python version 3.6 as it is the most stable version according to the documentations of the individual libraries that are intended to be used (TensorFlow, 2020) (OpenCV, 2020) (Django, 2020). The reason behind using Python is because of the wide array of support for machine learning libraries and frameworks that are essential for deep learning, such as TensorFlow and Keras. These two libraries were voted the most popular framework for deep learning with Python as noted on the JetBrains’ developer survey (JetBrains, 2019) making it ideal for use in this project. The OpenCV library will also be used in this project to be able to capture live stills of the hand gesture to then pass to the neural network created using TensorFlow and Keras.

The use of the Django library will also be used to create the web front for the application. OpenCV has support for the Django library allowing for ease of implantation and communication between the web front and the deep learning side.

Libraries such as Keras, TensorFlow, OpenCV and Django are all optimised and integrate with ease with the python programming language, which is one of the main reasons why Python is an extremely popular language when it comes to making machine learning applications (Beklemysheva n.d.).

3.3 Data set

An important part of this project is to choose a dataset that will allow for the creation of an efficient classification tool. There are multiple factors that need to be taken into consideration when assessing whether the dataset is appropriate for this specific project. One factor that needs to be taken into account is that test users for this project will be accessing the tool remotely via a web application, meaning that they would need to use their own webcams to provide the live video feed. The implication of this is that the video feed being passed to the neural network model will be varied in terms of resolution and in terms of quality of the image, due to differences in lighting conditions.

Another factor is the amount of data and whether or not it has variation in rotations, this is important so that the model will be able to also classify the fingerspelling letters from different angles – as all test users may not be able to move their webcam to a specific position.

Variations in Skin tones should also be taken into account as any image pre-processing done with a limited quantity of skin tones could affect the overall performance of the tool. These were all factors mentioned in the work of Szegedy et al. (2016) work on hand gesture recognition.

The data set I have chosen for this project is taken from research done by Pugeault N. and Bowden, R. (2011), in why they created their own computer vision hand gesture recognition classifier and provided great results with there being a 99% recognition rate. This dataset contains 1000 of each letter in the ASL fingerspelling alphabet in the training set, which is substantial enough for this project. The reason I have chosen this dataset is that it meets all the above requirements.

3.4 Deep learning - Standards, Algorithms, Techniques

After having chosen the dataset, the next step would be to create or use an existing CNN architecture to be able to classify the ASL fingerspelling letters. From the research done in the literature review, the most viable option for real time classification from a live video source seems to be the VGG architecture as noted by Strezoski et al. (2016). This architecture is gives depth and complexity which allows it to be useful for classifying images with a high degree of accuracy and is able to do this within a reasonable time scale. However, it is hard to tell without testing whether this architecture would be suitable for this project. Strezoski et al. (2016) in his work also created a custom CNN which had similar accuracies as VGG but has better training times and classification times, this shows that

doing some experimentation can yield better results and show which CNN model would be best suited. The VGG architecture can be used as a starting point some exploration will need to be done to find the ideal set up.

3.5 Learning Structure of the tool

After the ASL fingerspelling classification tool has been trained and tested to prove a high level of accuracy at a reasonably quick time, the implementation of the learning side of the tool can be considered. Based on the work of Paechter and Maier's (2010), in order to create an effective learning tool, it would need to provide clear learning outcomes, course resources, interaction between the learner and the instructor and the facilitation of the interactive learning process.

With the above needs in mind, I propose the creation of three different 'modes' of teaching that can be implemented in the application, these include the 'Introduction Mode', 'Memory Mode' and 'Spelling Mode'.

The 'Introduction Mode' involves sequentially display the letters of the alphabet having the user attempt to replicate the hand gesture. The users will be shown a live video feed of themselves signing and from this image data will be taken when the image has not changed for a certain number of frames. This data will then be used to classify the letter, if the user signs the letter correctly the next letter in the alphabet will be shown. If the user signs incorrectly, they will be provided feedback that they are incorrectly signing and be told to try rotating the hand slowly to make sure the camera is capturing from the correct angle.

A similar approach will be used for the 'Memory Mode', however instead of showing the learner a picture of the hand gesture, they will be shown the text for the letter and expected to replicate the sign learnt from the first mode – solidifying their memory of the letters. Word can also be provided to the users, with the expectation of them signing multiple letters consecutively. An incorrect letter in this case would result in the visual of the letter being shown after multiple incorrect attempts to the user to remind them how the letter is signed.

Finally, the 'Spelling Mode' will allow for the independent learning of the users. In this mode they will be allowed to use the letter s they have learned in previous modes to spell out words of their choice, allowing themselves to practice signing the letters.

3.6 Web Tool / Web Hosting

The actual front of the application would be coded using the Django framework which will communicate with the deep learning back end to form a complete application. In terms of hosting this application, a source that is able to give a substantial amount of graphical computing power would be necessary. For this, there are a couple of options, the first of which being hosting the site on the University of Salford's servers, though what computer

power the university is able to offer is still unknown at this point (with an enquiry being made). The alternate option is to use services such as Amazon Web Service or Google Cloud, both of which are capable of providing the computational power to host such an application (Aws, 2020) (Google Cloud, 2020).

3.7 Evaluation

When it comes to evaluation of the project a user study will be conducted, that will have users remotely accessing the web tool – having a mix of users who are familiar with ASL and users that are not. Some filtering of data may have to be done as there are issues that can arise from the application being host and users accessing it remotely, such as poor internet quality and poor webcam quality. An initial check can be made to check whether the user's setup satisfies the requirements, to which then their data will be used to evaluate the application.

The evaluation would consist of allowing users to test the application, going through each of the modes, after which they will be made to fill out a questionnaire. This questionnaire will be designed to create a score of the learning efficiency of the tool. This will be done with the inclusion of scale questions where a statement will be made, and users have a scale of one to ten/strongly disagree to strongly agree where they would answer based on their experience with the application.

Chapter 4 – Requirements Specification and Design

This section will utilise and expand on the previous chapter by taking the findings presented and transform them into a detailed requirements specification for this project, aiming to provide a justified list of requirements and an assessment how they are seen as crucial in meeting the overall aims of the project. This section will also discuss the project's design process, detailing the design decisions made and the tools intended to be employed to construct the project – providing a comprehensive justification behind these decisions at each step.

4.1 Requirements

The requirements specified in this section will be taking into account the overall aim of the project which is to develop an interactive sign language learning application which utilises deep learning techniques. In order to discover the requirements for the system and detail how they must be implemented and operate, a set of modelling techniques will be employed – these are called 'Use Case' modelling and 'User Story' modelling.

4.1.1 Use Case Modelling

When it comes to developing a list of requirements an overview of the system must take place to define what needs must be met as well as being able to identify the system's constraints on its operations and implementation. A proven method of identifying requirements in the aforementioned way, involves the creation of 'Use Case diagrams' and 'User Stories'. The benefit of this method allows for a greater understanding of how a user will interact with a system and provide clarity for the different requirements in each part of the system (Cockburn, 2000). Figure 4.1 shows the relevant use case diagram for all the interactable features for each user regarding this application. From this diagram it is clear that within the requirements specification there are two main overall requirements that are deemed crucial to the success of the project. These consist of a need to create a back-end sign language classification module and a connected front-end interactable user interface (UI) – both of which can be broken down into smaller sub requirements.

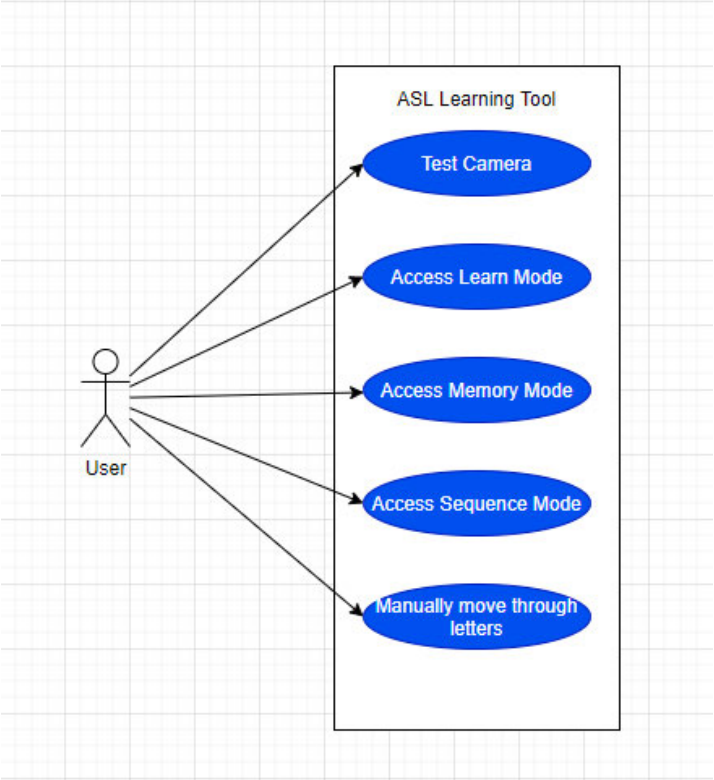


Figure 4.1: *Use case model*

4.1.2 Sign Language Classification module

Arguably the most crucial requirement, the need for a sign language classification module is the basis for the entire project – upon which every other requirement is depended on. This requirement states that the module must be able to take a still frame from a video and be able to classify what sign language letter is shown within said image. In order to do this a model must first be constructed and trained on a desired dataset containing images of the sign language alphabet letters – this can be seen as a functional requirement. A non-functional requirement related to this would be that the trained model would need to have a high degree of testing accuracy in order for the application to be successful in real life practice.

Additionally, in order for the model to keep a high level of accuracy additional data would need to be added containing different hand shapes, skin tones, lighting conditions and backgrounds which will ensure that would be able to be effective in a variety of environments. Therefore, there would be a need to have a system in place where additional images can be added to the dataset to increase the pool of data the model can be trained and tested on. The idea for this is to eventually allow users to submit their own hand images to be added to the dataset (part of the future work), but right now that would be seen out of the scope of the project, so it will currently be used to manually add images to increase the variety and size of the dataset.

4.1.3 Front-end Integration

Once a fully functional live video classification tool is made, the front-end web UI would need to be considered. The shell that will hold all the functionality of the application this would need to be in the form of a multi-page dynamic website. This website would need to able to contain multiple pages to encapsulate the different learning modes of the application whilst also allowing for navigation between these pages.

Another functional requirement is that the user must be able to access a 'Learning Mode'. This is important as it is the main way information is presented to the user other than the information on the page and it is the key-way users are going to learn the ASL alphabet. The sub requirements that are linked to this include the learning mode must being able to show the user a series of images in which the user must replicate the image, when the user has successfully replicated the image the next letter in the alphabet must be shown.

The inclusion of a 'Memory Mode' is another requirement for the system, this is important as it will test the user and simulate a greater learning experience (Paechter and Maier, 2010) – allowing the user to retain what was just learnt. This mode works similarly to the 'Learning Mode' but will show the letters instead of the sign itself asking users to replicate. The same sub requirements that come under 'Learning Mode' also apply to this mode.

A requirement related to the previous two is the mandatory inclusion of the feedback system. This feedback system is important as it tells the users the mistakes that they are making on why their had signs are not being registered, to which they are then informed on how to correct it through text pop ups.

The final mode that is deemed a requirement is the 'Sequence Mode', this mode will allow users to use what they have learnt to spell out word. This is an important part of the application as it allows for users to apply what they have learnt, greater reinforcing it in their memory and making an overall better learning experience (Paechter and Maier, 2010).

4.1.3 Requirements list

Back End

Functional requirements:

- The application must be able to train and save a model in regard to a specific dataset.
- The application must be able to capture live footage from the user's webcam and be able to capture a single frame from the live footage.
- The application must be able to classify the ASL letter using the trained model and captured image frame.
- The application must be able to pass the predicted result to the front end of the application.
- The application must be able to pass feedback to the user when an incorrect sign is predicted in reference to the current letter in either mode.

Non-Functional requirements:

- The CNN model must be trained on a large dataset with different hand sizes and lighting conditions allowing for a well-trained model.
- The classification of the image must be done relatively quickly (aim is to classify in under one second).
- The testing accuracy of the model should be to a high level.
- The feedback to the user should be instantaneous to what sign they are currently holding up (after the classification has been done).

Front End

Functional Requirements:

- The application must have a multi-page web app front that users can navigate and interact with.
- The application must have a learning mode where users are shown an ASL letter and asked to replicate.
- The application must have a memory mode where users are shown a letter in plain text and asked to remember the sign for said letter.
- The application must be able to have a sequence mode where users can use the words learned to spell out letters.
- The application must show the user feedback generated from the back end of the system.

Non-Functional Requirement:

- The navigation between pages should be instantaneous.

Usability Requirements:

- The web front must be designed in a way that maximises accessibility for users, following design techniques mentioned in the literature review.
- Instructions should be provided to the user on how to use the application efficiently.
- Information should be provided on the page to assist the learning of the user.

4.2 Design

With the requirements for the system now being explicitly defined, it is now easier to plan out the design of the application for how the different requirements need to be connected to each other within the system. This is done through the creation of user journeys, UML diagrams and wireframes, which will be implemented in a series of agile sprints. As mentioned in Chapter 3 the use of an agile development methodology is important to this project as there will be a lot of experimentation in terms of how the system is developed. This therefore means that some of the designs created in this section will be more of a guideline rather than a final detailed model.

4.2.1 System Architecture

As mentioned within the requirements specification there are two main overall systems within the desired application, which include the Web Interface and the Sign Language Classifier – a diagram of which can be seen in Figure 4.2.

Sign Language Classifier

Within the Sign Language Classifier there are 4 main elements. The first and arguably most important element is the dataset, which will contain a series of ASL alphabet images that the system will train, test and validate from. It is important that the dataset be as rich as possible in the variety of images it contains in order to produce a well-developed model. This links to the second element in this part of the system which is the AddToDataset script. This is here to add to the existing database taking from a proven and tested project to further increase the variety of images in the dataset as well as providing the basis for a system where users could potentially add their own images to the dataset.

Also within this part of the system is the Train script, with aims to utilise an existing proven CNN architecture and train it on the ASL dataset provided to create a trained model. This script will only need to be ran once and have the model be saved at the end for use in classifying test images.

The ClassifyFromWebcam script will take the saved trained model and use it to classify images that are taken as still every couple of frames from the user's webcam – the number of frames will be determined after some experimentation. After this script attempts to classify the image, the predicted result will be returned to the Web side of the application.

Web Interface

The Web Interface side's structure will follow an MVC (Model View Controller) pattern, meaning that the application would be split into a series of layers, these include the: Model, which will send request to bring back the information; Controller, which will take the result and pass it to the view; and the View, which is what the user will see. This design pattern is a proven technique in web development which allows for faster web development process and allows data to flow smoothly throughout the application in a formal manner (reference).

Expanding on what these layers will contain, firstly to describe the View layer there will be three web pages each of which will be a different learning mode of the application – this will be the entry point of the application for the user. Each of these pages should contain a webcam feed as well as the relevant supporting learning information mentioned in the Requirements section, which will be supplied by the Controller. The views should also contain navigation buttons to move between pages as well as buttons to interact with the supporting information such as skipping letters or resetting the sequence.

The Controller layer contains the code behind the button presses allowing it to switch between views and control the supporting information in terms of what is being displayed.

It will also take in information such as frame grabs of the user's webcam and pass it to the Model layer.

The Model layer takes the frame capture and initially passes to the BL where some operations are done so that it is suitable to pass of to the Sign Language Classifier (SLC), which is done after the image is then passed to the DataAccessor class – which then fires off a request to the SLC and waits for the response. This response is returned as a single character which will be the prediction for the image sent, which will then be sent all the way back down to the view for the user to see.

The entire system should work asynchronously meaning that when a frame of the user's hand is taken there should be no page refreshes/cuts/button presses needed in order for the predicted result to shown on screen again.

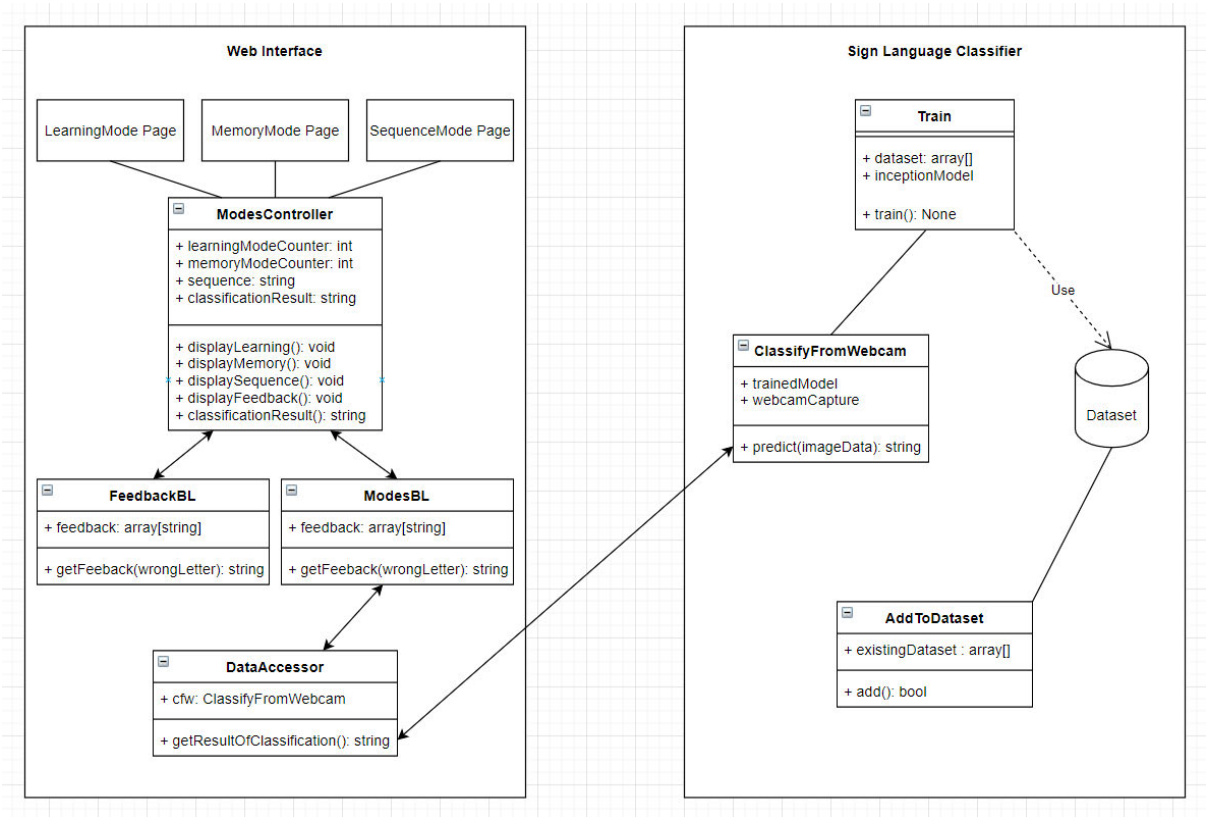


Figure 4.2: *UML diagram of system architecture*

4.2.2 Wireframes and Designs

An important part of the design process is to also be able to visualise and plan out what the users see (could put reference about system design). It allows for a better understanding of what the requirements for the system would be as well as the interactions that need to take place between those requirements, making the implementation process more focused and streamline – allowing for faster development, crucial as there is not much time to create the application. They also allow for the planning of usability and accessibility of the site, which is why a wireframe for each page of the web application has been made (Figures 4.3-4.6).

Figure 4.3 shows the home page for the application where the users can access the different parts of the system, done through the user copying the signs shown next learning mode. This screen will also act as an initial test screen where user's camera and environment setting will be tested. If the classifier is able to classify the hand correctly then the user will be allowed to move on to the next phase, otherwise they will be told that they are unable to enter.

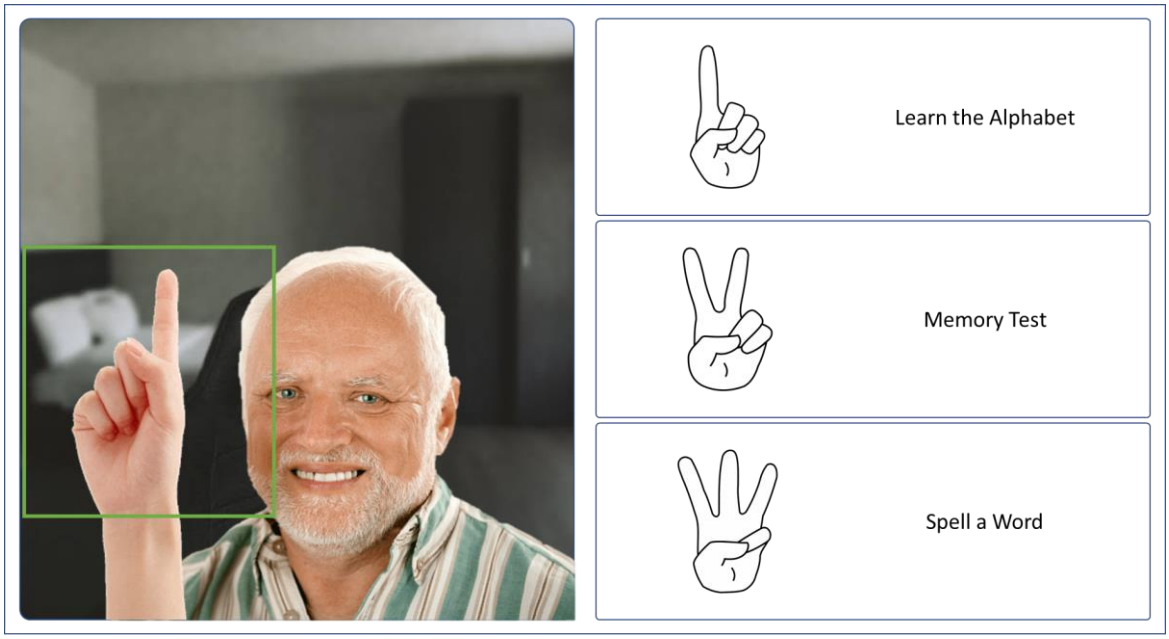


Figure 4.3: *Wireframe of home page*

Figures 4.4-4.6 show the different modes of the application, all of which follow a similar layout of showing the webcam feed on the left, mode specific area on the right along with a feedback section below it. The idea behind these wireframes were to keep the designs simple and have all the key information on the screen at one time, allowing for users to have to manually interact with the page less – allowing more focus on the learning of the ASL signs.

The pages should work independently and have the individual modes work without the user needing to interact with the page with a mouse and keyboard. Therefore, the design of the pages should communicate to the user what needs to be done and whether they are doing it right or not. One way this will be done is the green/red boxes on the webcam classifier, where green would tell the users they are signing the letter correctly and red to say they are incorrectly performing the sign. Another being the feedback section being dynamic in that it would automatically change text in regard to what sign the user has put up.

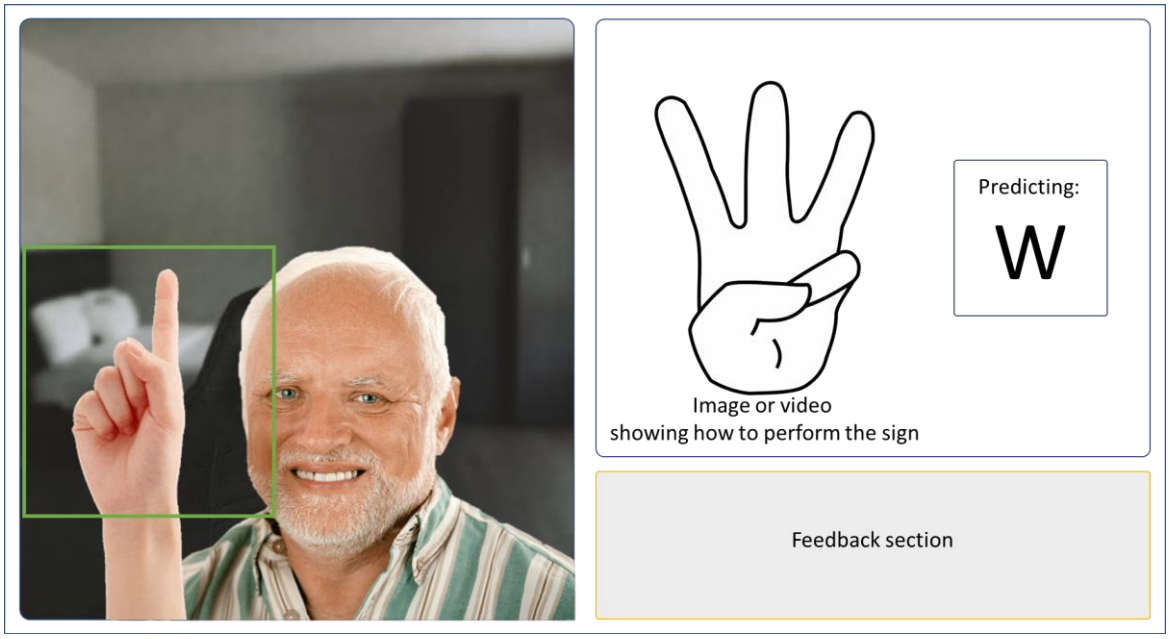


Figure 4.4: *Wireframe of home page*

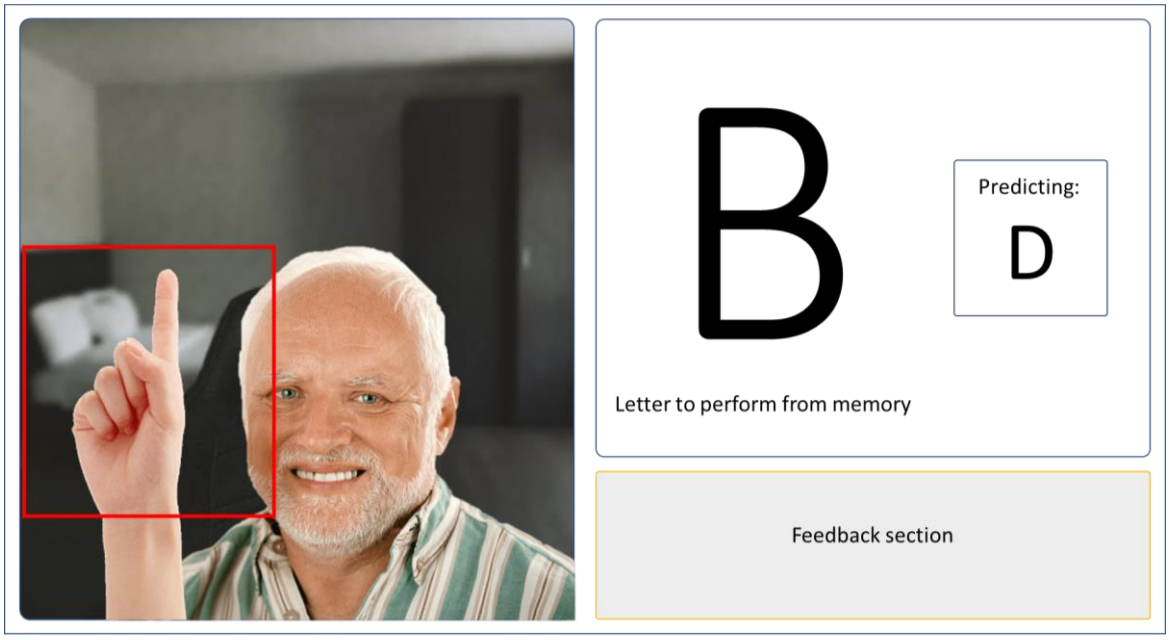


Figure 4.5: *Wireframe of home page*

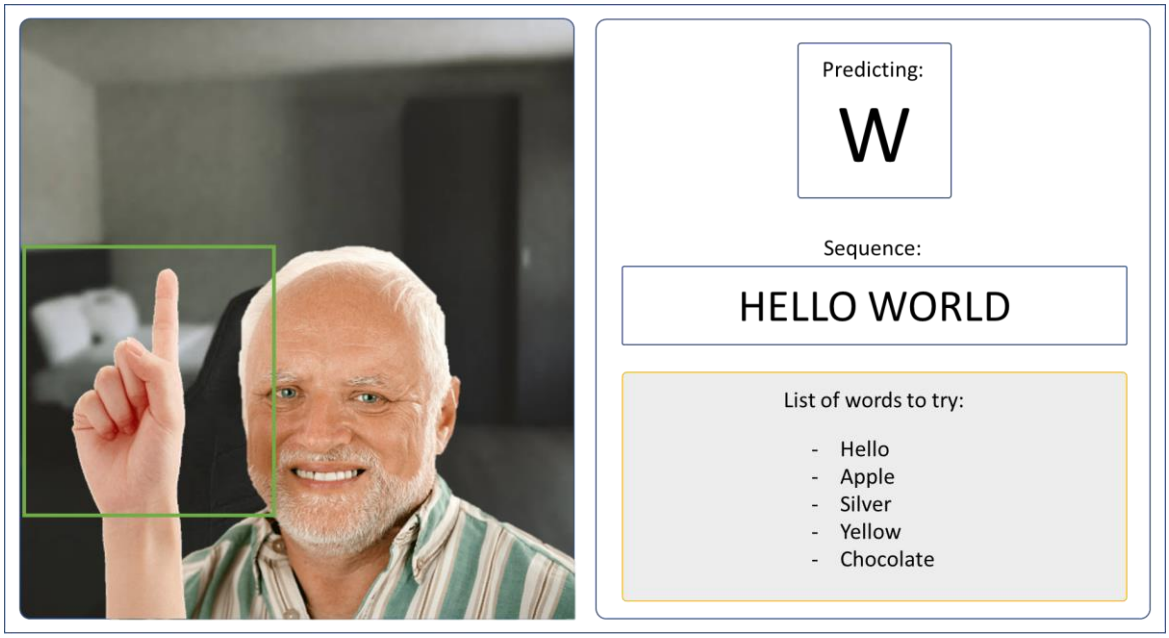


Figure 4.6: *Wireframe of home page*

Chapter 5 – Development and Implementation

This section will detail the implementation process for the creation of the proposed sign language learning tool. It will describe the development process in regard to both how it was organised/maintained, such describing the use of story boards and Kanban boards used, as well as the specifics of the implementation – such as the algorithms, data structures and design tools used at each step.

The project has employed an Agile development methodology which breaks down the overall task into more manageable iterative steps. This has led to the project being split into three different phases (also known as sprints) which need to be executed sequentially in order for the project to effectively developed within the evident time constraints. Each of these sprints have been individually planned, executed and tested and will be described below in detail in terms of the implementation and development process.

5.1 Sprint 1 – Classifier

The first sprint within the project focuses on creating the sign language classifier outlined in the previous chapter, it is the basis of the entire project and an essential component upon which every other requirement is dependent upon. The classifier consists of 3 main components which include: the dataset, the training/modelling process and the live video classification process.

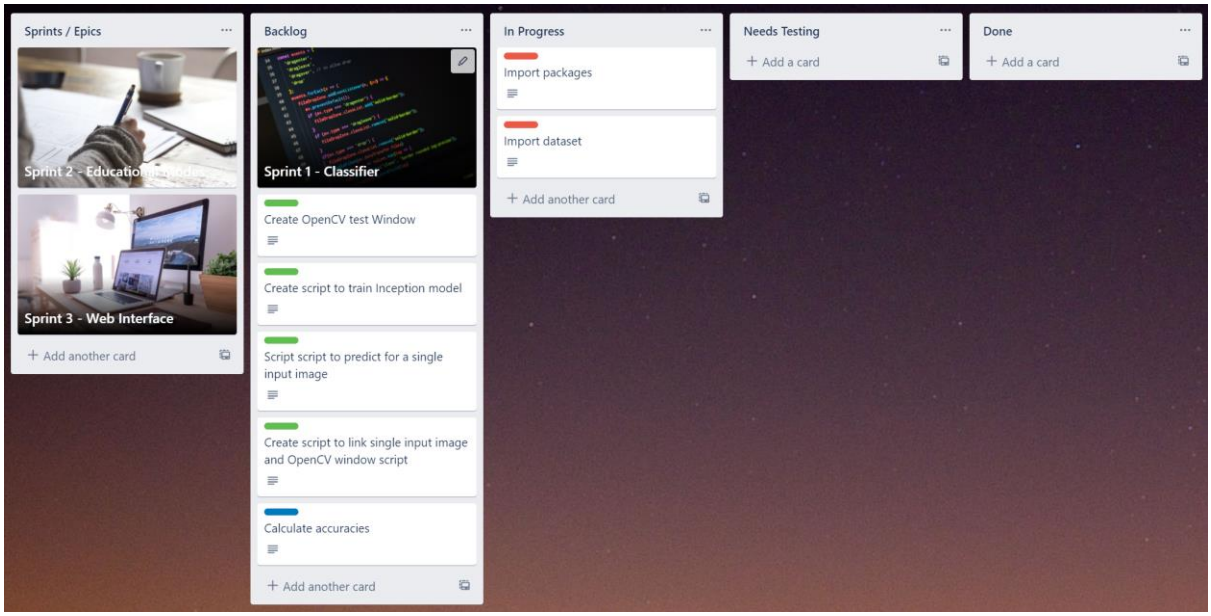


Figure 5.1: *Trello board of tasks from the second sprint.*

The Dataset

At the core of any good deep learning application there must be a diverse, extensive dataset (Szegedy et al., 2016) which allows for the training of well-defined CNN. The dataset selected for use for this project was accumulated by Pugeault N. and Bowden, R. (2011) and is a relatively decent size dataset for this project – containing on average 1000 images per ASL letter. However, browsing through the images in the dataset presents the first major issue encountered, which is that all of the images have very similar backgrounds and the lighting conditions, although they may be varied slightly, are not broad enough to create a classifier for use in a range of environments. In order to solve this issue another dataset must be used or amended to the existing dataset and after some research another two other data sets were discovered (the other datasets) that had a similar problem to the one from Pugeault N. and Bowden, R. (2011). In order to solve this issue, a python script was written to take a random selection if 500 of each ASL letter to form a new dataset.

Also, in order to further expand the dataset a second script was written that allows images from the webcam to be taken, labelled and added to the dataset – a code snippet of which can be seen in Figure 5.1. This was something that was already planned in the design phase as an additional feature but, became of greater importance in the use of improving the quality of the dataset. The implementation of this script involved the use of the OpenCV library, which is the computer vision library that will be used throughout the project to utilise the user's webcam and have code interact with the images taken, this will be described in detail in the next step of the project.

```
while True:
    ret, frame = cam.read()
    frame = cv2.flip(frame, 1)

    x1, y1, x2, y2 = 100, 100, 300, 300
    img_cropped = frame[y1:y2, x1:x2]
    cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 0), 2)

    if not ret:
        print("failed to grab frame")
        break
    cv2.imshow("Capture Images", frame)

    k = cv2.waitKey(1)

    if k != -1:
        print(k)

    if k == 48:
        img_counter = 0

    if k % 256 == 27:
        # ESC pressed
        print("Escape hit, closing...")
        break

    elif 97 <= k % 256 <= 122:
        # Any Key Pressed
        img_name = r"\dataset\{0}\{1}{2}.jpg".format(chr(k).upper(), chr(k).upper(), img_counter)
        cv2.imwrite(img_name, img_cropped)
        print("{} written!".format(img_name))
        img_counter += 1

cam.release()

cv2.destroyAllWindows()
```

Figure 5.2: *Code snippet of script that adds images to the dataset*

Training/modelling process

Within Chapter 3 it was concluded after reviewing the field of CNN architectures an experimental phase would be needed in order to find best possible implementation suited to this project. Also, from the research done it was concluded that the VGG16 model would be a good starting point based on the testing accuracy from the ImageNet challenge. The VGG16 model is an example of a Transfer Learning model, this means that the model is trained and developed for a specific task and is reused as the starting point for a second task. This is something that is popular in the computer vision field as developing models take a large amount of time and resources and using this approach improves the efficiency of feature extraction of new images. This model had been imported into the solution through the Keras model as seen in Figure 5.3 and the solution was further developed using the existing functions that come along with this the VGG16 package. The dataset images were then loaded into a variable through the VGG16 function load_img() and also reshaped each image to a 224x224 size. These images were then pre-processed to prepare the images for the VGG model. However, a roadblock was then met at the training stage. Training the model proved to be very time consuming and difficult for a development process as large chunks of time would need to be taken out to wait for the model to be retrained. This was deemed too large a disadvantage for the project as the development stage would take too long. Therefore, a decision was made to switch to using the Inception V3 model instead.

```
from keras.applications.vgg16 import VGG16
# load the model
model = VGG16()
```

Figure 5.3: *Keras imports*

The inception module was also a transfer learning model that was trained on the ImageNet dataset. Initial tests of this model proved to be very successful and quick to train and develop for.

The project then started off by utilising the OpenCV library to be able to take in inputs from the webcam and manipulate images so that they were able to be passed into the model to be trained or tested. Initially a window was used instead of a web browser to test the application (seen in Figure 5.6). An image from this webcam was captured every 4 frames using the OpenCV library and converted to a data array using NumPy and then reshaped to a 244x244 size ready to be passed into the model. Capturing every 4 frames meant that in a 60fps camera 15 classifications would be done a second (7 in a 30fps camera). This was deemed adequate as the inception model was proven to be very fast as mentioned in Chapter 2.

The Inception pre-trained model was imported using TensorFlow. The first step in order to retrain this model would be to feed images and calculate the bottleneck values for all of them. Bottlenecks is a term used for the penultimate layer before the final output layer which carries out the classification. Though initially creating these bottlenecks take a while dependent of the computer specifications, these bottlenecks will be cached so that the next time the model is retrained the training process is incredibly quick. After the bottlenecks are

created the actual training of the top layer of the model begins, the training script takes a random 50% of images in the dataset and uses them for training, the remaining 50% of images are used for validation and testing. The model was configured to be trained over 2000 steps, the results of which can be seen in Figure 5.4. The results show that the training accuracy was pretty high, with a final result of 94% and the validation accuracy stayed high also at 96% – meaning that overfitting has not taken place. The loss function (Cross entropy) also remained very low, meaning the model was trained very well. These summaries were obtained using TensorBoard.

```
Step: 0, Train accuracy: 16.0000%, Cross entropy: 3.100316, Validation accuracy: 6.0% (N=100)
Step: 100, Train accuracy: 69.0000%, Cross entropy: 2.330640, Validation accuracy: 55.0% (N=100)
Step: 200, Train accuracy: 79.0000%, Cross entropy: 1.828305, Validation accuracy: 79.0% (N=100)
Step: 300, Train accuracy: 79.0000%, Cross entropy: 1.478232, Validation accuracy: 86.0% (N=100)
Step: 400, Train accuracy: 81.0000%, Cross entropy: 1.370323, Validation accuracy: 80.0% (N=100)
Step: 500, Train accuracy: 88.0000%, Cross entropy: 1.117974, Validation accuracy: 87.0% (N=100)
Step: 600, Train accuracy: 85.0000%, Cross entropy: 1.024809, Validation accuracy: 88.0% (N=100)
Step: 700, Train accuracy: 88.0000%, Cross entropy: 1.019319, Validation accuracy: 87.0% (N=100)
Step: 800, Train accuracy: 95.0000%, Cross entropy: 0.801249, Validation accuracy: 82.0% (N=100)
Step: 900, Train accuracy: 91.0000%, Cross entropy: 0.754730, Validation accuracy: 84.0% (N=100)
Step: 1000, Train accuracy: 93.0000%, Cross entropy: 0.648651, Validation accuracy: 92.0% (N=100)
Step: 1100, Train accuracy: 92.0000%, Cross entropy: 0.699954, Validation accuracy: 87.0% (N=100)
Step: 1200, Train accuracy: 90.0000%, Cross entropy: 0.730426, Validation accuracy: 90.0% (N=100)
Step: 1300, Train accuracy: 96.0000%, Cross entropy: 0.628658, Validation accuracy: 97.0% (N=100)
Step: 1400, Train accuracy: 95.0000%, Cross entropy: 0.577774, Validation accuracy: 91.0% (N=100)
Step: 1500, Train accuracy: 96.0000%, Cross entropy: 0.463159, Validation accuracy: 95.0% (N=100)
Step: 1600, Train accuracy: 92.0000%, Cross entropy: 0.593629, Validation accuracy: 94.0% (N=100)
Step: 1700, Train accuracy: 93.0000%, Cross entropy: 0.526490, Validation accuracy: 94.0% (N=100)
Step: 1800, Train accuracy: 92.0000%, Cross entropy: 0.569649, Validation accuracy: 92.0% (N=100)
Step: 1900, Train accuracy: 96.0000%, Cross entropy: 0.382170, Validation accuracy: 93.0% (N=100)
Step: 1999, Train accuracy: 94.0000%, Cross entropy: 0.526165, Validation accuracy: 96.0% (N=100)
Final test accuracy = 93.8% (N=4768)
```

Figure 5.4: *training outputs and testing accuracy*

Live video classification process

The actual classification process was performed using the code seen in Figure 5.5. This code uses the encoded image data array and feed it into the trained model to be able to obtain a classification prediction array. The top prediction is then converted to ASCII string and the result is returned. The initial call to this method would be made from a generator script which is constantly running, using OpenCV to take in an image every couple of frames. The final result of this sprint can be seen in Figure 5.6 – showing that the requirements for the sprint have been met.

```python
def predict(image_data):
    predictions = sess.run(softmax_tensor, {'DecodeJpeg/contents:0': image_data})

    # Sort to show labels of first prediction in order of confidence
    top_k = predictions[0].argsort()[-len(predictions[0]):][::-1]

    max_score = 0.0
    result = ''
    for node_id in top_k:
        human_string = label_lines[node_id]
        score = predictions[0][node_id]
        if score > max_score:
            max_score = score
            result = human_string
    return result, max_score
```

Figure 5.5: *Code fragment used to predict the output of the image passed in*

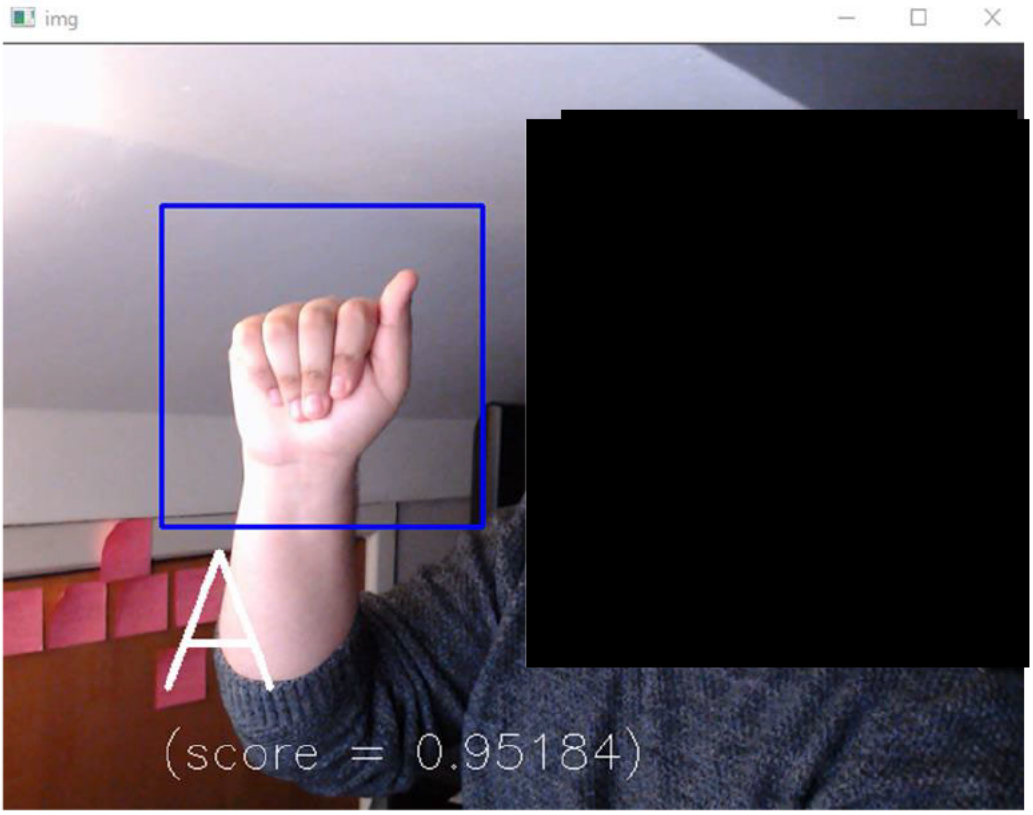


Figure 5.6: *Example of working classifier in windowed form*

5.2 Sprint 2 – Educational Modes

With the classification part of the system being complete, focus can now be shifted towards creating the different learning modes that will make use of the classifier – this being the goal of this second sprint. There are three unique modes of the system, these include the

Learning Mode, Memory Mode and Sequence Mode – selected as a result of research in the field of online learning.

The implementation of this mode, although initially planned to be part of the front-end web side of the project, is instead part of the back-end classifier side. This design change was made to reduce the number of requests being sent for the web side of the project. After some progress was made on the classifier it was clear that web side would already be making a lot of request and to improve the response times of the front end of the site, the learning mode was instead move to the classifier segment of the system.

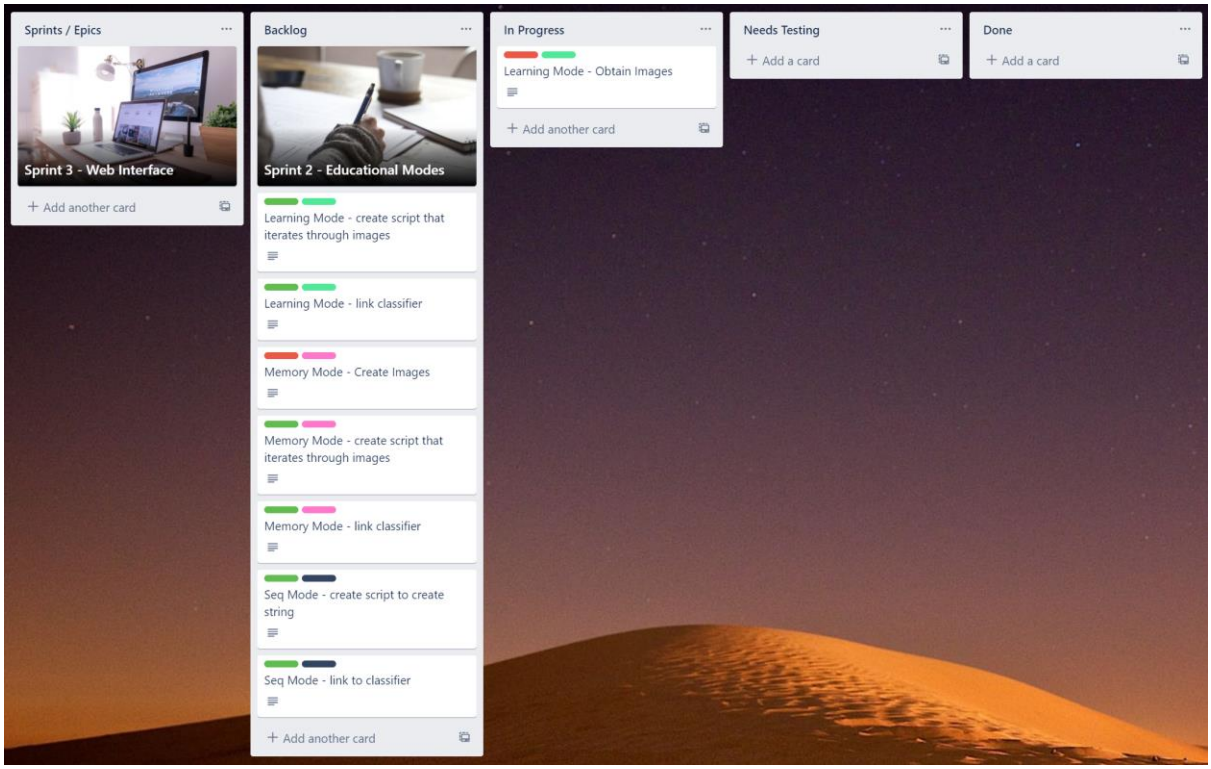


Figure 5.7: *Trello board of tasks from the second sprint.*

Learning Mode

As specified in the Requirements and Design chapter, the learning mode must be able show the user a series of ASL alphabet images for which the user must replicate in live time themselves. After the user replicates a single letter the next letter in the alphabet will be shown until the user gets to the end until which the series of letters will be replace. The learning images shown to the user have been taken from wpclipart (2021) and are high-quality images which clearly show the user how to perform the sign.

Creating the learning mode involved having an initial array of all the learning image file names in alphabetical order. This array would then be iterated through based on the current letter the user is viewing named ‘learning_counter’. The counter variable would be a representation of what letter the user is currently on. An initial implementation was made in which the result from the classification was taken and a conditional statement check was made to whether it was the letter the user was currently on – to which it the increment variable increased and moved to the next letter. However, the issue with this method was

that the letter would move to the next letter as soon as the user had signed it, meaning the next letter was shown before the user had realised it. Therefore, the solution in Figure 5.8 was used. This created a new variable known as consecutive, which was incremented each time the user was signing a single letter, this allowed for a pause to tell the user they have correctly signed the letter before moving on – allowing for a generally better user experience. An image with a temporary solution with two windows can be seen in Figure 5.9, this would then be plugged into the web front.

```
if learning_mode:
    if consecutive == 4 and res not in ['nothing']:
        if res == letters_img_arr[learning_counter]:
            learning_counter += 1
```

Figure 5.8: *learning mode counter solution*

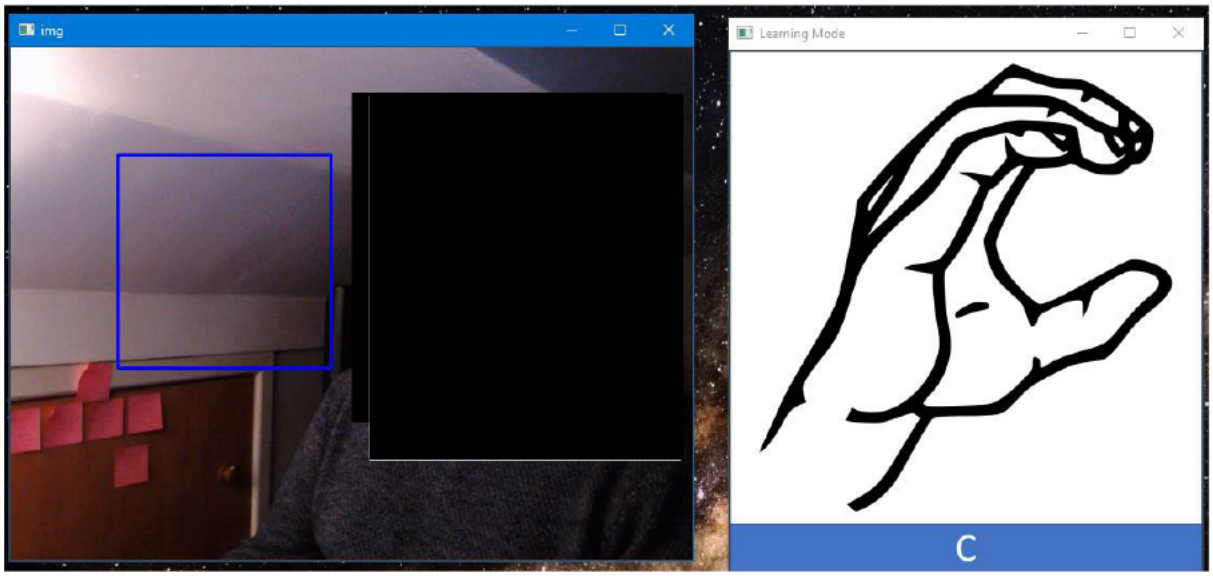


Figure 5.9: *working learning mode in temporary windowed form*

Memory Mode

The memory mode of this application follows a similar flow to that of the Learning Mode, however there are two main differences the images used and the flow of those images. Since the user's memory will need to be tested in this mode only a plain text letter will be shown instead of the sign and the user will be asked to replicate it – being even more challenging by showing the users the images in a random order.

This was done by having a similar approach to the implementation as the learning mode, as seen in Figure 5.10, but the key difference is that instead of incrementing the counter by 1 it is assigned a random value from the letter array. This is done using the randrange() function from the random library in python, selecting a random index from 0 to 25. An example of the memory method working can be seen in Figure 5.11.

```
if memory_mode:
    if consecutive == 4 and res not in ['nothing']:
        if res == letters_img_arr[memory_counter]:
            memory_counter = randrange(25)
```

Figure 5.10: *memory mode counter solution*

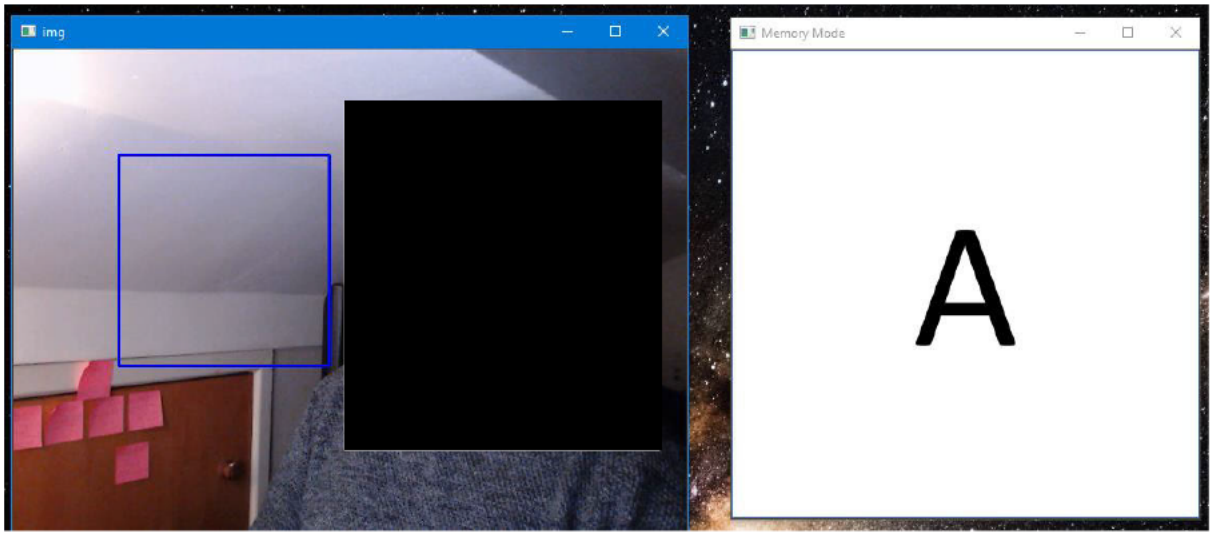


Figure 5.11: *working memory mode in temporary windowed form*

Sequence Mode

The sequence mode allows the user to use what they have learnt to further improve their understanding on the ASL alphabet as well cement the letters in memory through applying their knowledge – a proven technique of learning stated in chapter 2.

For the implementation of the sequence mode two additions needed to be made to the dataset, which include the 'delete' and 'space' signs. These two signs are not going to be taken from ASL as these specific words require movements to visually communicate, therefore 2 signs will be made up which look like the icons on a keyboard. These means that the model would need to be trained again to accommodate the two additional outputs.

In terms of implementation in the system, a string variable would hold the current sequence the user signs letter by letter taking the classification result each time, adding it to the sequence variable and then showing the newly formed string back to the user (currently in a new window, but later implemented in a web page). The code also considers the new additions the dataset and adds/removes to the string as needed – this can be seen in Figure 5.12 and the working process can be seen in Figure 5.13.

```
if sequence_mode:
    if consecutive == 5 and res not in ['nothing']:
        if res == 'space':
            sequence += ' '
        elif res == 'del':
            sequence = sequence[:-1]
        else:
            sequence += res

        consecutive = 0
```

Figure 5.12: *sequence model string builder*

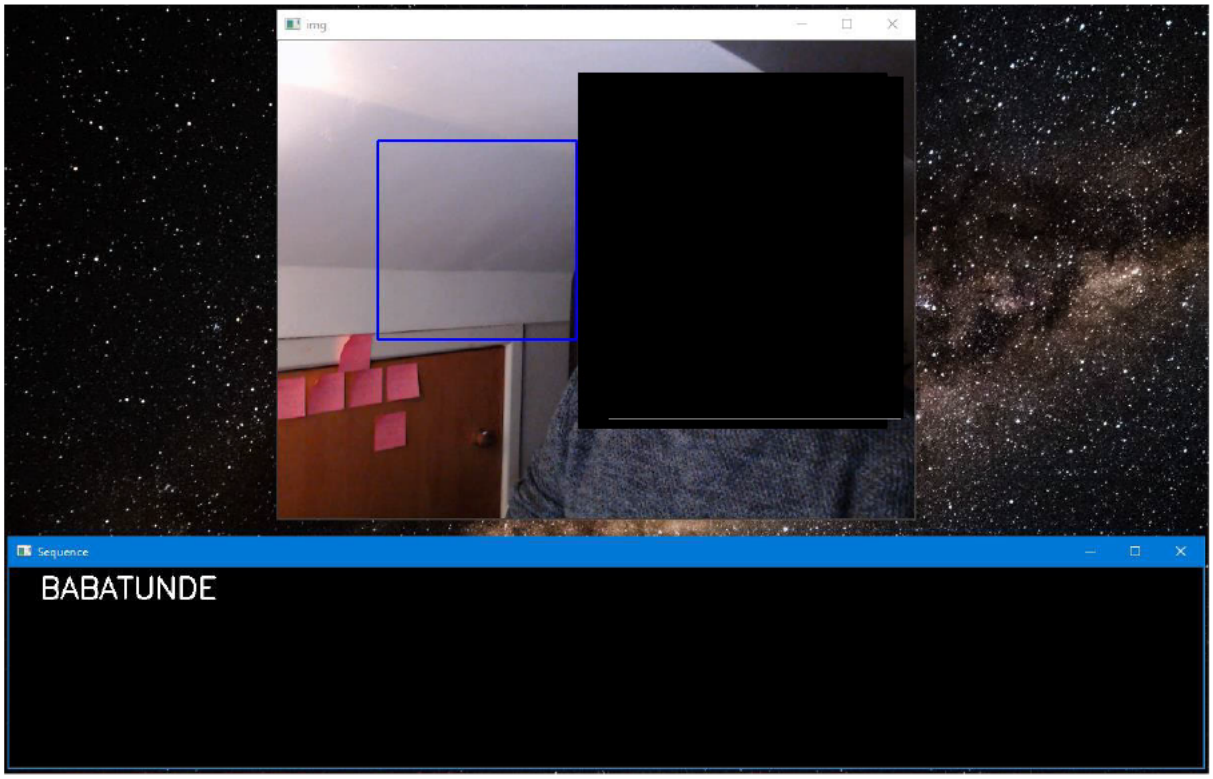


Figure 5.13: *working sequence mode in temporary windowed form*

5.3 Sprint 3 – Web Interface

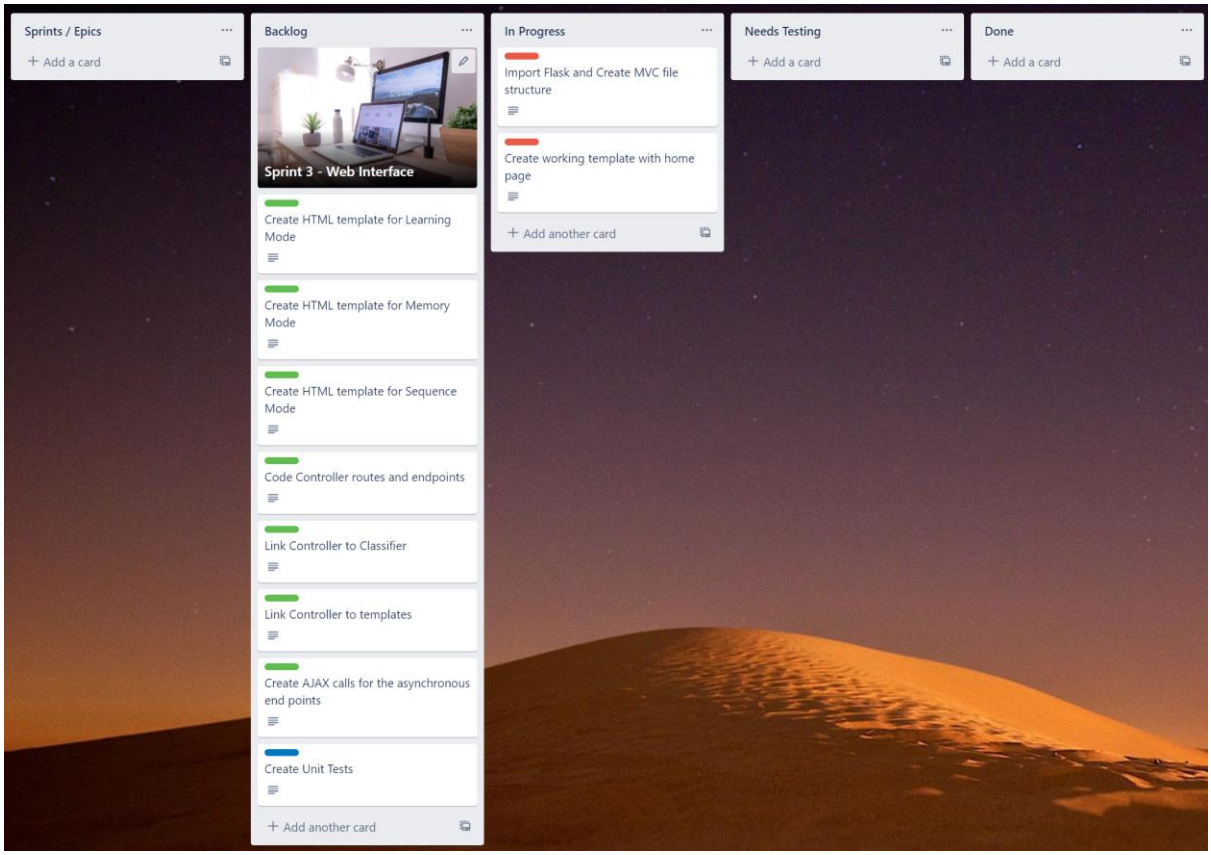


Figure 5.14: *Trello board of tasks from the final sprint.*

Although a Django MVC web interface design was planned to be constructed, a Flask MVC implementation was chosen instead. The reason being Django proved to be quite difficult to work with and had too many constraints that prevented quick development time – which was unacceptable with the development time remaining. Therefore, Flask was chosen instead as it was a more streamline lightweight framework that was easy to pick up in terms of development and allowed for the web interface to be built in a quick and efficient manner.

The architecture of the web application still followed the MVC web pattern, but now being based around the Flask framework instead. The Controller was the first part of the system to be developed as it is the connected between the html page and the information that should lie within – essentially managing the flow of data to and from the web pages. The controller accommodated a series of routes that the user could access with the default route being the home page – seen in Figure 5.15. Also, within the Controller would be the code to host the application locally on an open port as seen in Figure 5.16.

```
@app.route('/')
def index():
    return render_template('index.html')
```

Figure 5.15: route for the home page of the application

```
if __name__ == '__main__':
    app.run(host='127.0.0.1', port=5000, threaded=True, use_reloader=False)
```

Figure 5.16: code to start server

An initial implementation was made where each page load up all the elements individually (e.g., the webcam classifier, mode information and feedback), this however was incredibly slow as the start up time for the classifier was long (around 10 seconds) – this being unacceptable for a web application. Therefore, a choice was made to load the page elements in dynamically so that there would be less loading time.

As the webpage would be now developed to be asynchronous, it was decided to remove multiple pages for the different modes and eliminate page refreshes by having a single html page that would dynamically change its contents through user activated asynchronous calls. This would reduce wait times for user and make for a more fluid application. A series of html templates were made that would act as components for the different learning modes of the application (all templates seen in Figure 5.17), each of which would be displayed/removed by the controller from the single html page dependent of the route the user is currently on. This would require a series of endpoints the system would need to hit, each of which uses the setup code from the previous sprint for the different learning modes (example seen in Figure 5.19). This code would be called on button press by the user and executed through an AJAX call using jQuery – seen in Figure 5.d5.

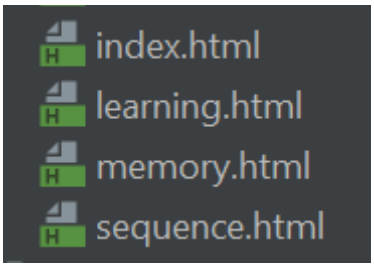


Figure 5.17: code to start server

```
<div>
    <img id="imageBlock" class="w-75 img-fluid rounded-3 border border-3 border-dark" src="http://localhost:5000/learning_mode"
        alt="Images"/>
    <div id="sequenceLetters" class="rounded-3 border border-3 border-dark p-2 fs-1"> S</div>
</div>

<div class="my-3">
    <button id="prev" type="button" class="btn btn-primary mx-3 fs-5">< Previous</button>
    <button id="next" type="button" class="btn btn-primary mx-3 fs-5">Next ></button>
</div>

<div>
    <div id="feebackres" class="alert alert-info" role="alert">
        If your hand isn't registering try moving it closer to the camera.
    </div>
</div>
```

Figure 5.18: html template

```python
@app.route('/learn', methods=['POST'])
def learn():
    setup_ler()
    return json.dumps({'status': 'OK'})
```

Figure 5.19: code to start server

```javascript
$('#ler').click(function(){
    $.ajax({
        url: '/learn',
        type: 'POST',
        success: function(response){
            console.log(response);
        },
        error: function(error){
            console.log(error);
        }
    });
});
```

Figure 5.20: code to start server

In every mode there needs to be the webcam live feed with the classifier, this therefore has been baked into the index html page from its own endpoint seen in Figure 5.21. This endpoint is connected to the classifier script which uses a generator pattern to continuously be running in the background, taking in inputs which would be the image captures from the webcam and continuously output the result asynchronously to the web application through the use of python's *yield* keyword.

Finally, the feedback module was made. The feedback comments were stored in a variable as a dictionary, with there being a sentence for each letter. This sentence was picked out depending upon the current letter the user is viewing on either the learning or memory mode (shown of Figure 5.22) and displayed to the user in an asynchronous way using AJAX calls as seen on Figure 5.23.

```python
@app.route('/video_feed')
def video_feed():
    return Response(gen(), mimetype='multipart/x-mixed-replace; boundary=frame')
```

Figure 5.21: code to start server

```python
@app.route('/feedback', methods=['POST'])
def feedback():
    feed_string = ''
    if learning_mode:
        feed_string = feedback_dic[letters_img_arr[learning_counter]]
    elif memory_mode:
        feed_string = feedback_dic[letters_img_arr[memory_counter]]
    else:
        feed_string = ''

    current = users_current_letter

    if learning_mode or memory_mode:
        if current == 'del' or current == 'space':
            current = 'nothing'

    return '<h5>Predicting: ' + current.upper() + '</h5>' \
           + '<b>' + feed_string + '</b>'
```

Figure 5.22:

```javascript
function fetchfeeback(){
    $.ajax({
        url: '/feedback',
        type: 'post',
        success: function(response){
            if (response !== undefined)
            {
                $('#feebackres').text(response)
            }
        }
    });
}
```

Figure 5.23:

In order to style to html templates, the Bootstrap framework was used, this ensured that the pages were accessible as possible and also be responsive – meaning that this application this application could be viewed on mobile devices without any design issues. Colour on the page was also taken into account choosing contrasting colours for elements that overlay making for an overall more accessible user experience.

The final outcome of the development process for each page can be seen in Figure's 5.24, 5.25 and 5.26.

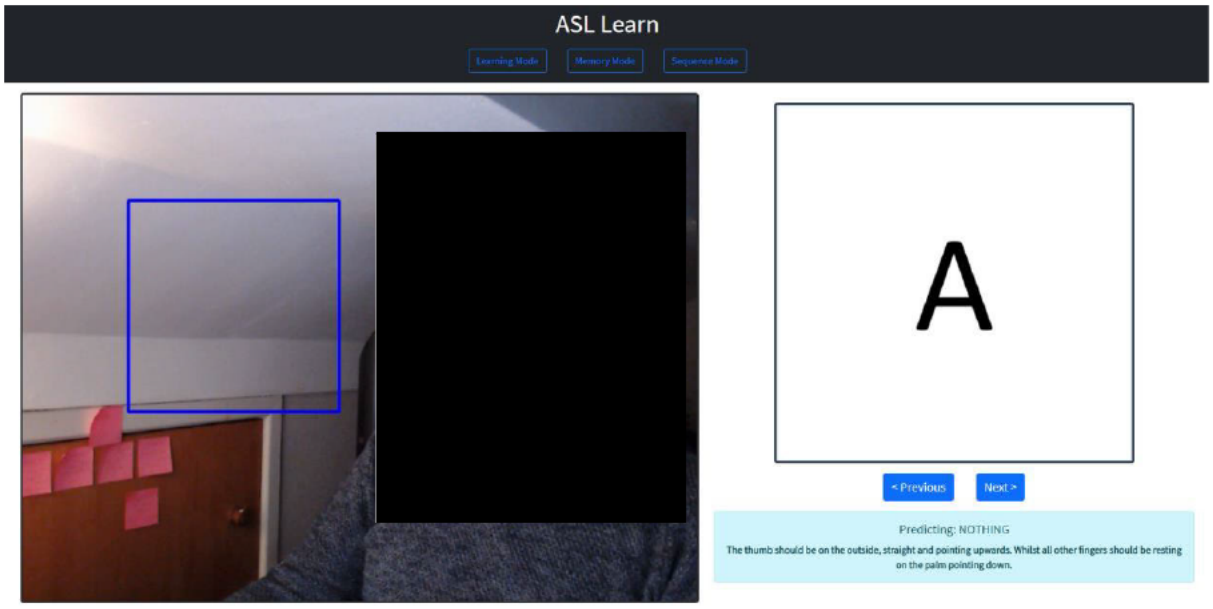


Figure 5.24:

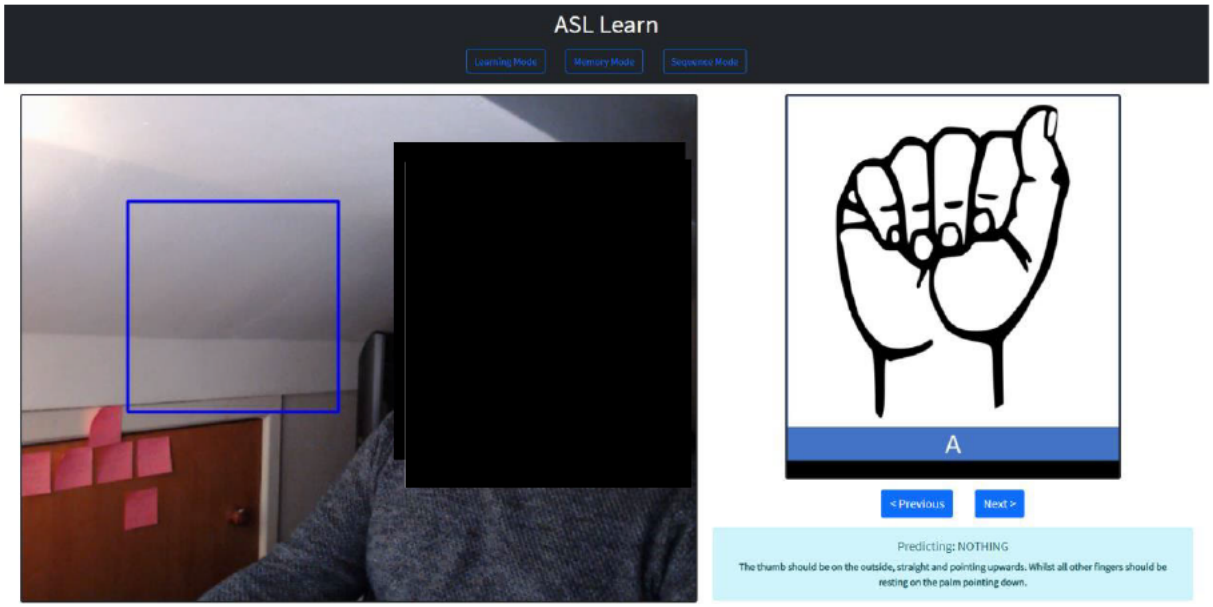


Figure 5.25:

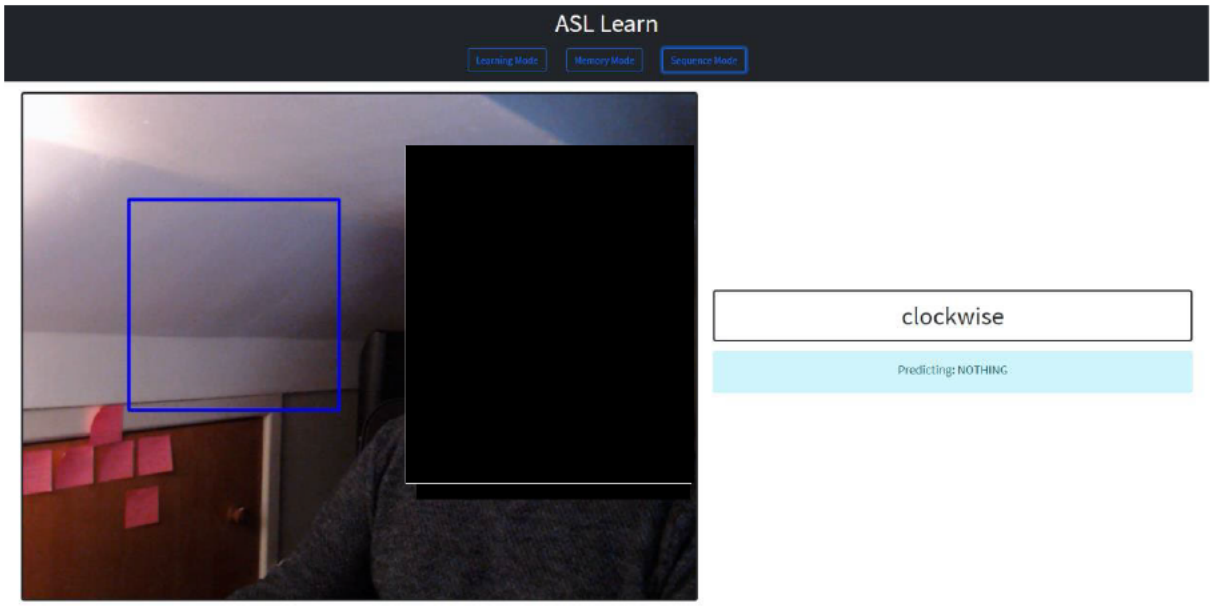

ASL Learn
Learning Mode
Memory Mode
Sequence Mode
clockwise
Predicting: NOTHING

Figure 5.26:

Chapter 6 – Testing and Analysis

This section will discuss the user testing carried in order to evaluate the application to verify whether the aims of the project were met. This will involve describing the testing strategy utilised and analysing the quantitative and qualitative data obtained to determine the success of the project.

6.1 Testing Strategy

Many tests have already been undertaken for the application, such as: performance testing, where the accuracy of the model was obtained through obtaining the average classification result of a defined test/validation dataset as well as testing the performance of the classifier in terms of speed; and unit testing, where code functions in the web application have been tested to check whether they operate as expected with different inputs, making the code reliable and future proofed.

However, as the aim of the project is to create an interactive learning application that enhances the learning experience for users, it is a usability test that would be the key factor in determining how the project has fared. This test will involve finding a group of test users who will interact with the application to provide feedback on their experiences. This overall journey can be split into three main phases – which include user screening, user testing and user feedback.

User screening

Over the duration of a week, a mix of potential testers were reached out to, some of whom had experience with the ASL alphabet and some of which did not. Since the target audience was not very specific, the process of obtaining users was relatively simple, however there were certain conditions that users must meet. It was essential that the users had access to a computer, internet connection and a webcam. It was also vital to make sure that all of these users had experience in some form or rather with online learning, whether that be an online course and some form of online educational tutorial – this would be important as it would allow users to have a basis to compare the created application with.

These users were informed about the project's testing processes through the user of the Participation Information Sheet – which outlined though a series of relevant questions and answers of what their involvement would consist of; the measure in place to keep their data protected and contact details of the researcher for any enquiries. The users were made to sign the Participation Consent Form to get confirmation that they are willing to participate in the research. The users were given a week to sign the consent form and within that week they were informed that they had the opportunity to ask any questions – which some users did.

Testing Phase

Once a pool of test users was available, the next task would be to allow for all users to gain access to the application. Since the hosting of the application was unsuccessful and there was not enough time to try alternative hosting solutions, the application was manually sent over to the users with instructions on how to set it up. At this stage a there was a small drop off in terms of test users who were unable to set up the application and get it working either because of hardware or software issues – but in the current climate with the restrictions put in place, there was little that could be done, meaning they were excluded from the tests. These excluded users were then used for an alternative test where they only try to learn the ASL letters by following a reputable online learning course such as SignLanguage101 (2021). These alternative users would then be given a separate questionnaire to gauge how well they feel they have learnt the ASL letters and the opinions on the learning techniques used. This would be used as a basis to compare the created product against.

In an ideal scenario, without the restrictions of the current climate, only one computer would host the application and the users would show up in person to test the application on that computer. This scenario would also have allowed for some observations to be made by the researcher in the habits of users in terms of them interacting with the application. However, carrying out the test this way does have some degree of bias as the user may think/act differently with the researcher in the room – letting the user test the system at home provides more of a realistic situation when it comes to online courses.

Nonetheless, the testing still took place with the number of users who were able to set up the application. In order for the user to have an experience of all parts of the system, a list of objectives/task were made. These include:

- Get through the entire alphabet in the Learning Mode – this will test the classifier on a unique environment with a new set of hands shapes and skin tones. If the classifier works successfully, it will test whether the feedback of knowing you are signing the letters correctly provides for a greater learning experience.
- Get through at least 10 letters in the memory mode – this is again a test on the classifier as well as the test whether it learning mode has functioned in helping the user learn the content.
- Complete the list of recommended word on the Sequence Mode – this will allow for the user to apply what they have learnt to a real-world scenario of spelling words.
- The user was then instructed to use the application for a further 10-15 minutes to go over anything they liked.

Feedback Phase

The feedback phase of the project would entail the users filling in a questionnaire on their experiences. This questionnaire would gauge how the user felt about the different parts of the system, through a series of questions that would provide quantitative results and qualitative results – discussed in the following section.

6.2 Results

As the questionnaire contained a series of statements that the user must specify which predefined answer best suited their experience (these answers included Strongly Disagree, Disagree, Neither Agree nor disagree, Agree and Strongly Agree), it was possibly to calculate the averages for each question which can be analysed.

The questionnaire initially asked the users if they were able to get past the camera test. Out of the 15 users, 3 were unable to get past the camera test and therefore had to be excluded from the test. Some users were in contact with the researcher over getting to application ready to be able to classify and it was noted that many users had to toggle the lighting conditions in the room to get classifier to work.

The questionnaire then began by asking the users how they felt about the web interface's UI and general user experience, through the statement-answer format. 75% of the users strongly agree' that the application was user friendly and accessibly (the other 25% selected that they 'Agree'). Users also mainly agreed (83% strongly agreed) that the user interface was fast and responsive. These results may be skewed in that since the websites are being hosted locally everything would be fast and responsive compared to it being hosted on a web server.

The users were then asked about the classifier, independent of the modes. From the feedback it was possible to analyse that the classifier was a little inconsistent in producing the correct output to what the users were signing. 50% of test users disagreed with the statement of the classifier being able to identify the user hands correctly a majority of the time. However, there were a couple users that strongly agreed with the statement, therefore showing that the classifier does work but needs to be trained on a more comprehensive diverse dataset in order to be more of a professional product.

This can be further supported by the reviewing the average classification result from the user tests. Before the user testing an additional code, fragment was added to the back end of the code that would calculate the average test score of the classifier based on the images that the classifier was successfully able to predict. This code was then linked to an endpoint in the Flask controller and was called by a piece of jQuery AJAX call, which would then print the accuracy to the browser's console. During the instruction process users were also told to access this console menu and note down the final average classification result for their session. The results noted by each user was noted and a test accuracy based on user input was calculated to be 56% - much lower than the test result calculated in the implementation stages. This shows that there are still some room for development in the trained model to generalise better with a unique set of images. This can be done by improving the diversity of images within the dataset further by adding a greater number and wider range of images. Improve the accuracy of the model, will lead to an increase in the user experience as the issue many users had was that they classifier was just not reliable enough.

Users were also asked about the ASL feedback section that would provide users feedback on what the users were signing. From the qualitative feedback from users, the general perception was that, although this section was helping in describing how to carry out the

letter, the was a lot of missed opportunities in how detailed this system of feedback could be. This was supported by 75% of the test users, who disagreed that the feedback system was able to correct letters that they struggled with. One user state that they felt as though a more visual approach should have been taken on the feedback system, they described a system where a translucent overlay of the ASL letter could be applied on to the webcam allowing users to more accurately correct their hand position rather than having to interpret from text. This is a type of system that would have taken a lot of work to implement but would be plausible and beneficial, something that could be added as a feature later on.

The next part of the questionnaire then asked the users how they felt about the different modes of the application. When asked about the ease of the getting through the learning mode, 58% of users strongly agreed or agreed that they found getting through the entire alphabet relatively easily. The feeling towards the memory mode and the learning mode was mostly similar. This could be attributed to the fact that around half of the test users had difficulty getting the classifier to return the correct result most of the time.

However, when asked about the overall learning experience, the user feedback was mainly positive. User's feedback shows that 83% agreed or strongly agreed that this method of learning the ASL letters would have been better than just reading of watching videos of the same topic, finding it more enjoyable overall and all of the users that agreed also agreed that they would like to see this type of learning with instant feedback be used in other areas of online learning and though it would greatly benefit user interaction. This shows that although users had difficulties with the classifier, they mostly agreed that there was potential around this area of online learning that could benefit people, learning in a more hands on way that can provide verification the learner without manual intervention.

When comparing the responses from the user testing of the application made against the responses of the testers from the second part of the test (where they were made to use an existing online ASL learning course), there was a pattern that showed that users much preferred learning via the created interactive application. This is shown from the responses the second set of testers gave, which states that 92% of users did not feel as though they have been able to correctly understand and replicate the ASL signs. In contrast to this, 83% of users from the first test felt as though the system allowed for them to feel validated in the sense that they are performing the ASL signs correctly and were confident in what they had learnt. This shows that having a deep learning/computer vision application in place to validate user inputs allows for an improved learning experience – therefore meeting the aims of the project.

Summary

Overall, by using the testing results to measure whether the project has accomplished it goals it can be said that the project was mostly a success. The projects aim was to create an interactive application that utilises deep learning and computer vision technique to enhance learning and give a give an overall better user experience. This was proved by from the user feedback from the questionnaire, where a majority of users (83%) preferred this method of learning to more traditional ways. Normally, other studies would be used to support the

results found here, but as mentioned previously in the review of the field there seems to be a lack of research in this area. Something that does support this research would be from Paechter and Maier (2010)'s work of defining the key principles of online learning, where they mention that interaction with a system is essential to a user's learning process – something that was proved by the results of this project.

Chapter 7 – Critical Evaluation

This section will contain a personal evaluation for the entire project and will discuss the following in detail: a review of the project objectives, a review of the project plan, an evaluation of the product developed and finally the lessons.

7.1 Review of Project Objectives

Firstly, I will be going through each objective that was set out at the start of the project and discussing whether what was accomplished meets what was planned.

Objective 1 - Perform a state-of-the- art review to survey the field of research in automatic detection of ASL

This objective was completed in Chapter 2, where research was carried out to all of aspects of the project including the: online learning, HCI and CNN aspects. The research was carried out in a professional manner and positive feedback was received from the project supervisor. This research allowed for an overview of the field was able to be obtained, which allowed for a clear requirement draw up and a cleaner development process.

Objective 2 - Develop a feature extraction algorithm for the detection of ASL gestures, through using existing techniques.

A feature extraction system was not exactly developed over the course of the project and instead an existing feature extraction tool was used. A choice was made to use a transfer learning convolutional neural network as they are incredibly fast, reliable and quicker to develop for compared to having created your own CNN or feature extraction techniques. In this case the Inception V3 neural network was used. This was already trained on the large ImageNet dataset (with over 10 million images) meaning that the feature extraction for this model would be generic and can be reused for a multitude of other projects. Therefore, technically the objective was met, however a change for the project's development was required in order to create a professional application. The feature extraction from the model used was to a high quality and was able to pick out help classify the ASL letters easily.

Objective 3 - Develop a CNN for the classification of finger spelling characters. Measured by ensuring the average classification accuracy is at an acceptable level.

Similar to the previous objective, a CNN wasn't developed from scratch as I learnt that it would take a substantial amount of time and resources for the model to be trained to a high quality. Instead, the Inception V3 neural network pre-trained on the ImageNet dataset was used, training a new top layer to the model with the images from the ASL alphabet dataset. The implementation and training of this model were very successful as the test accuracy of the model was approximated 94% and in personal test the classifier worked perfectly. However, this was proved a little deceptive as the results from the user test prove that the model may not be trained on a diverse enough set of images, with a number of users having difficulty getting the system recognising their hands.

Objective 4 - Create a Learning System that features multiple modes including: a learning mode, a memory mode and a spelling mode.

This objective was something was also very successful in development, all of the modes were created to their full extent and allowed for the user to be able to take in, memorise and practice what they have learnt. In the testing phase the majority of users agreed that the interface was well developed and the different learning modes, in conjunction with the classifier, provided enough content to be able to learn the ASL alphabet effectively. Overall, this objective was successfully met.

Objective 5 - Evaluate the tool with automation and human users, measuring the accuracy of the model created.

This objective was something that was also achieved successfully, the automated evaluation of the tool was done through a series of: performance tests, which tested the accuracy of the model created as well as the speeds the classification process; and through unit tests which test the websites endpoints and functions to make sure all edge cases were delt with – making sure there were no faults with the system. The project also performed human user tests where users were given a chance to interact with the system created and were able to give their feedback through a questionnaire, the results of which were analysed and documented in Chapter 6.

7.2 Review of Project Plan

Looking back at the plan proposed at the start of the project a lot of modifications had to be made throughout the project. The progress made up until December was on in line with the initial schedule created, however due to unforeseen personal circumstances the next few months little to no project work was being done. I was able to pick the final year project back up at the end of February where all of the remaining tasks had to be rescheduled and a new plan had to be created which can be seen in Figure 7.1.

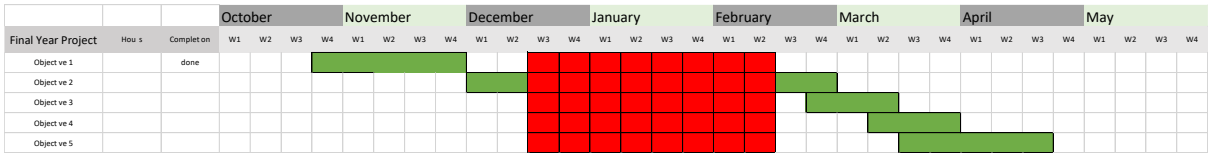


Figure 7.1: *new Gantt chart visualising the new schedule.*

This new schedule was more focused and had little flexibility which meant that in order to complete the project this exact schedule would need to be followed explicitly. Because of my circumstance I only had two months to deliver the final product and finish writing a dissertation. Intelligent planning and organisation helped me in completing the project in time. I was able to keep myself organised through a series of Kanban board using Trello and this allowed me to keep focus on the task at hand and finish on task at a time (Trello boards can be seen in Figures 5.1, 5.7 and 5.14).

By the end of the project, I was able to develop the application to the specification I had outlined in the original project proposal. Albeit the website was not able to be hosted in time, the product itself turn out extremely well, surpassing even my own expectations based on the time I had remaining. The classifier was a huge achievement and a personal leap for me in terms of improving my own coding ability and the entire project around it is something that I am proud to have done for my final year project.

7.4 Lessons Learnt

Undertaking this project has allowed me to learn a number of things relevant to project management and software development. It also was able to help me develop my own personal skills throughout the project whether it be: technical skills, such as creating and training neural networks; or soft skills, such as time management and task organisation.

One key lesson I was able to take away from this project, for future work, was the importance of time management. As time was already limited and then shortened by my personal situation, I was able to observe how planning time out efficiently allows for projects to be streamlined, focused and finish on time – utilising Agile methods. If I had more time, more time would probably be spent of experimentation where none was needed, and tasks would have taken longer to complete. Therefore, observing the agile development methodology at work has given me more of a desire to use it in future projects.

Another lesson I have learnt is how important user testing is in understanding and figuring out how a project has fared. In my performance test, the classification result was pretty high and in my own tests with my own webcam and set up I was getting good results. However, my perspective changed when the application got put into real life practice and users were experiencing difficulties that I thought would be nullified. This showed me that the user testing process is an essential part of any project development as it allows you to identify any mistakes/issue – letting you to rectify these issues before launching the product to a production environment.

Although I had learnt so much, If I had to choose one final lesson to describe it would be the importance of surveying the field of your chosen project. In order to properly know the bounds of you project and to construct something of meaning it is important to consider what is already out there. Carrying out research will also allow for you to improve your own knowledge in the area which can help in the development and problem-solving aspects of the project. This can save time and resources and allow for the creation better plans and objectives.

Chapter 8 – Conclusion

Overall, throughout the course of this project there have been many notable achievements. The project began by surveying the chosen field to find a gap in knowledge, which was to apply the fields of computer vision and deep learning/image recognition to online learning via the method of the ASL sign language alphabet. The next phase of the project was to create a literature review of the fields of online learning, deep learning, computer vision and human computer interaction - which became the basis for drawing up and creating a requirements specification for the project to be developed.

The project developed utilised all of the above techniques and was able to take user hand images as inputs and output the classification result after being passed into the trained model. The output was then coloured up and displayed to the user via an online web application that focuses on user interaction which aims to help users learn the ASL alphabet in a more interactive way. Overall, according to the performance tests and user testing done the project was successful in meeting its goals – however, further work was required for this product to become professionally viable.

Reflecting further on the product created, there are certain ethical issues that come into play. One being the security of the image data sent to and from the web client to the classifier. In real world practice images of the user would be considered sensitive data, thus there should be not way that the user's data should be stored without the user's consent. It should also be the responsibility of the developer/researcher to make sure that application is secure and should have features in place to prevent the user's information being hijacked. If a feature was to be developed that stored user information there would be certain legal issue that may arise, one of which of being the need to be compliant with GDPR regulations.

Socially, I believe this application can be very beneficial. This is because it allows for sign language to be learnt and practiced in small, short sessions and the user testing has proved that it also enhances the learning experience making it a fun way to learn. All this combined can allow for a greater population to learn sign language, making it more commonplace in everyday society. This would immensely help deaf people in their daily lives as they are able to communicate with people more effectively.

In terms of comparing this application to the state of the art there are two aspects that can be discussed. Since there is not really an application like this out there, comparing the application would need to be done by breaking down its parts and then comparing that to the state of the art. So, for example firstly comparing the classifier made to the state of the art discussed in Chapter 2 from Strezoski et al. (2016) who created a hand gesture classifier, had a test accuracy of 84.33% - using a dataset that was also used as a part of the final dataset in this project. In comparison the model created in this project had a test accuracy of 93.8%, beating the state of the art. Secondly, the product made can be compared to other learning applications out there in terms of design and interactivity. The comparison in this case can be subjective in that it is dependent on the user and how they prefer to learn.

But, in the tests carried out in this project – users much preferred the interactive approach used by this project over existing courses available online.

8.1 Future Work

There are many additions that can be made to this project that could allow it to become more professional and practical. Taking from the need for additional images needed to better train the model, a feature on the site could be put in pace that allows users to submit their own images. This would work similarly to the image capture script created during the implementation process, where a box would be shown as an overlay for the webcam and the users would perform the sign in that box and the application could take captures of whatever is inside the box and save it to a database (something like an Amazon S2 Bucket). These images could then be used to train the model on a regular schedule to improve the classification result in a variety of situations. However, there would be certain issue related to this feature, one of which being that the images submitted would need to be filtered so that anything that is not an ASL hand gesture would need to be removed. There would also be the security and ethical issues with this feature, for example users may send images they did not mean to send and would not be able to undo without contacting the creator. These edge cases would need to be further considered and developed before something like this gets put into practice.

Another addition that could be made is to create a more user centric experience, where a possible login session can be made for the user and they would have tailored courses/exercises in a chosen sign language. For example, instead of just having the users carry out the signs shown on screen, more intricate exercises would be used such as having a conversion using sign language letters or have users read a sentence and fill in the blanks. These features would help the users practice the sign language further and have examples of real-life examples they can learn from.

Final Word Count: 14996

- **Dissertation Part 1: 4999**
- **Dissertation Part 2: 9997**

(Excluding References and Appendices)

References

Bederson, B. B., Russell, D. M. and Klemmer, S. (2015). Introduction to Online Learning at Scale. ACM Transactions on Computer-Human Interaction, 22(2), doi:

Yao, J. F. and Chiang, T. (2017). Teaching Online Courses 101. Journal of Computing Sciences in Colleges, 32(5), doi:

Paechter, M. and Maier, B. (2010). Online or face-to-face? Students' experiences and preferences in e-learning. The Internet and Higher Education, 12(4), pp. 292-297, doi: 10.1016/j.iheduc.2010.09.004

Brophy, J. (1999). Teaching, Educational practices series, Vol.1

Ehlers, U. (2004). Quality in e-learning. The learner as a key quality assurance category. European Journal of Vocational Training, 29, pp. 3-15

Mitchell, R.E., Young, T.A., Bachleda, B. and Karchmer, M.A. (2006). How Many People Use ASL in the United States? Why Estimates Need Updating. Sign Language Studies, 6(3), pp. 306-335. doi: 10.1353/sls.2006.0019.

Nces. 2016. Number and percentage distribution of course enrollments in languages other than English at degree-granting postsecondary institutions, by language and enrollment level: Selected years, 2002 through 2016. [Online]. [8 December 2020]. Available from: https://nces.ed.gov/programs/digest/d18/tables/dt18_311.80.asp

Andriakopoulou, E., Bouras, C. and Giannaka, Eri. (2007). Sign Language Interpreters' Training, in IICL2007, Austria

Tigwell, G. W., Peiris, R. L., Watson, S., Garavuso, G. M. and Miller, H. (2020). Student and Teacher Perspectives of Learning ASL in an Online Setting. ASSETS '20: The 22nd International ACM SIGACCESS Conference on Computers and Accessibility, Virtual Event, Greece

NAD 2020. Learning American Sign Language, viewed 5th December 2020, <https://www.nad.org/resources/american-sign-language/learning-american-sign-language/>

Skillshare 2020, "ASL | American Sign Language | The Alphabet", viewed 5th December 2020, <https://www.skillshare.com/classes/ASL-American-Sign-Language-The-Alphabet/1933820887?via=search-layout-grid>

Quinto-Pozos, D. (2011) "Teaching American Sign Language to Hearing Adult Learners," Annual Review of Applied Linguistics. Cambridge University Press, 31, pp. 137–158. doi: 10.1017/S0267190511000195.

Dam, H. K. (2019), 'Artificial Intelligence for Software Engineering', XRDS, vol. 25, no. 3, pp. 34-37

Justus, D., Brennan, J., Bonner, S. and Mcgough, A. (2018). Predicting the Computational Cost of Deep Learning Models. 3873-3882. Doi: 10.1109/BigData.2018.8622396.

Giannakos, M., Sharma, K., Martinez-Maldonado, R., Dillenbourg, P., and Rogers, Y. (2018). Learner-computer interaction, NordiCHI '18, Norway

Mitsianis, E., Spyrou, E., Giannakopoulos, T., Niafas, S. and Perantonis, S. (2018). Deep learned features for image retrieval, SETN'18, Greece

Sun, L., Zhang, L. and Guo, C. (2008). "Technologies of Hand Gesture Recognition Based on Vision [J]", Computer Technology and Development, vol. 18, no. 10, pp. 214-216

Sun, J., Ji, T., Zhang, S., Yang, J. and Ji, G. (2018). "Research on the Hand Gesture Recognition Based on Deep Learning," 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China, pp. 1-4, doi: 10.1109/ISAPE.2018.8634348.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". Neural Computation. 1 (4), pp. 541–551. doi: 10.1162/neco.1989.1.4.541

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, doi: 10.1109/5.726791.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv, 1409.1556.

Strezoski, G., Stojanovski, D., Dimitrovski, I. and Madjarov, G. (2016). Hand Gesture Recognition using Deep Convolutional Neural Networks. International Conf. on ICT Innovations, pp. 49-58, doi: 10.1007/978-3-319-68855-8_5.

Brownlee, J. (2019). Convolutional Neural Network Model Innovations for Image Classification, viewed 10 December 2020, <https://machinelearningmastery.com/review-of-architectural-innovations-for-convolutional-neural-networks-for-image-classification/>

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9

ImageNet (2014). Large Scale Visual Recognition Challenge 2014 (ILSVRC2014), viewed 12 December 2020, <http://image-net.org/challenges/LSVRC/2014/index>

Atlassian 2020, *What is Agile?*, viewed 13 December 2020, <https://www.atlassian.com/agile>

Beck, K., Andres, C. (2004). Extreme Programming Explained: Embrace Change (2nd Edition), Addison-Wesley Professional

OpenCV 2020. Open Source Computer Vision version 4.x, viewed 13 December 2020, <https://docs.opencv.org/master/d5/de5/tutorial_py_setup_in_windows.html>

TensorFlow 2020, TensorFlow Core v2.4.0, viewed 13 December 2020, <https://www.tensorflow.org/api_docs/python/tf>

Django 2020, Django documentation, viewed 13 December 2020, <https://docs.djangoproject.com/en/3.1/>

JetBrains 2019, Python Developers Survey 2019 Results, viewed 13 December 2020, <https://www.jetbrains.com/lp/python-developers-survey-2019/>

Beklemysheva, A. n.d., Why Use Python for AI and Machine Learning?, viewed 13 December 2020, <https://steelkiwi.com/blog/python-for-ai-and-machine-learning/>

Pugeault N. and Bowden, R. (2011). "Spelling it out: Real-time ASL fingerspelling recognition," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, pp. 1114-1119, doi: 10.1109/ICCVW.2011.6130290.

Amazon 2020, Machine Learning on AWS, viewed 14 December 2020, <https://aws.amazon.com/machine-learning/>

Google 2020, Google Cloud AI platform, viewed 14 December 2020, <https://cloud.google.com/ai-platform>

Cockburn, A. (2000). Writing Effective Use Cases, The Crystal Collection for Software Professionals. Addison-Wesley Professional Reading

Wpclipart 2021, American Sign Language alphabet, viewed 10th April 2021, <https://wpclipart.com/sign_language/American_ABCs/index.html>

SignLanguage101 2021, Learn ASL for free, viewed 10th April 2021, <https://www.signlanguage101.com/free-lessons/asl-level-1>

APPENDICES

Appendix A – Project Logbook

Week 1

Hours spent of project this week: 10

28/09/2020

Progress Made / Ideas I had

As the Final Year Project module kicked off this week, I began to go about researching, constructing and evaluating a couple project ideas I would be interested in undertaking. I did not have any concrete idea coming into this week but, I had established a list of areas in computing that I was curious in and would like to explore further and wanted to create a project around.

One of which, was the field of web development. Coming off a placement where I had gained a lot of knowledge of web development and practiced coding a lot for the web, I initially began to think of projects revolving around this area. One thing that was new and appealing to me during my placement was the concept of NoSQL. I had worked with a couple NoSQL databases including RavenDB and MongoDB and thought up an idea where I could explore the capabilities of SQL vs NoSQL – such as comparing speeds, storage efficiency and scalability. Though, upon doing some research it seems like this area was well documented and wasn't too confident that this would make an attractive novel final year project. Papers found:

- https://www.researchgate.net/publication/261079289_A_performance_comparison_of_SQL_and_NoSQL_databases
- https://www.cs.rochester.edu/courses/261/fall2017/termpaper/submissions/06/Paper.pdf

On the other hand, I also want to try to challenge myself for this module and create a project in a new area that I find intriguing and one in which I would be more motivated to delve deep into and learn more about. So, I also have started doing some research into areas of computing I am interested in such as AI, Machine Learning, Computer Vision and Neural Networks. I don't have too many ideas as of yet, but I will try to form some more during the course of this week.

To Do List

- Research areas of deep learning in which I could create a project around.

30/09/2020

Lecture

This week's lecture was essentially just an introduction into the module, laying out what we will be doing throughout the year as well has guidance on how to make a good start. There were a couple good points made that I have listed below which I though could help me choosing a final year project. These include:

- Difficult/challenging thing to do

- Must be realistic in terms of size, scope, time/resources available
- Produce software that has quality, reliability, timeliness and maintainability
- Identify the client for the project

We were told that this week there will be some seminars taking place that we would need to attend and decide on which project supervisory group we want to be placed in.

Progress Made / Ideas I had

After the lecture this morning I was convinced that I should choose a project idea that would be more on the challenging side that I would be motivated about, one which would also help me expand my knowledge as well as being able to meet the criteria from the lecture and project guide. So, as I mentioned on Monday (28/09/2020) I was interested in the field of AI and Computer Vision and had been doing some research into potential areas that I could make a project around and develop on. Here are three ideas that I thought would make good final year projects.

- Computer Vision RC Car – This Idea would revolve around being able to strap a camera on a RC that would in essence act as a self-driving car. It would use Convolutional Neural Networks (CNN) to detect paths to follow, obstructions to avoid whilst trying to reach an objective end point. Could possibly turn this into some sort of experiment where other cars would be doing the same but using different object detection algorithm, testing which ones worked the best.
- Automated Greenhouse – This idea maybe a bit far fetched but I think could work on a small scale. So, the idea is to use CNNs to monitor the growth of plants inside a small self-made greenhouse environment in which it would be possible to alter factors such as the temperature, the humidity and the light all through an application on something like a raspberry pi. An example of it working would be the program would detect a low level of moisture in the soil based of it learning from some training data and have it trigger the watering mechanism. Though, there are a lot of problems that I feel I would encounter for this idea, one being setting this entire environment up – which I guess would require some experience with getting a raspberry pi hooked up to these electronics.
- Sign Language Tool – My final Idea that I had been looking into was being able to use CNNs to develop an application that could detect and understand certain sign language signs. This would basically be a classification problem and from my research I had found that there have been problems when trying to classifying movements so I may have to stick to classifying stills such as letters in sign languages. From my research I had also found that hand gesture recognition is a field where quite a bit of work has been carried out, so for this idea I would have to some ways that I could make it different and original.

Supervisor Seminar

This week there were multiple seminars taking place, so I had to decide which one most suited me and attend that. I decided to attend the AI & Data Mining, Web Development and Data and Visualisation seminars. Today was the AI and Machine Learning Seminar, it was a

good chance for me to get some feedback in the 3 ideas I had above as well as be able to listen in on other people's ideas and possibly take some inspiration. Though I we couldn't get to my ideas within the seminars one-hour period I had email some supervisors to get feedback that way. After having pitched the 3 previously mentioned ideas I had to some of the supervisors, I was able to get a better idea of what of the 3 proposed would be the most realistic and achievable. I was recommended by some supervisors to talk to Dr Hughes and attend his Data and Visualisation group as they believed it was the most suited for me – which is what I will be doing on the Friday (02/10/2020).

To Do List

- Use the points from this morning's lecture to scope out these ideas, assessing the feasibility of the project idea as well as being able to identify and evaluate objectives that could be set out for the project – narrowing the list of ideas down to one.

02/10/2020

What was done

Having thought about these projects in more detail I had concluded that the sign language tool would be the most suitable project for me to undertake. I can to this conclusion as I genuinely have an interest in making something this and it would be something I would be motivated to plan and execute through the course of this academic year. I also felt as though in terms of the other project ideas I would need to buy resources and manually need to combine them together which could take up a reasonable chunk of time.

Therefore, I began to think of the achievability of this project. I knew that in order for any deep learning application to be successful I needed to find a reasonably sized dataset that I could use to train the model. Having looked around I had found this https://www.kaggle.com/grassknoted/asl-alphabet dataset which includes around 3000 images for each ASL letter. I have also seen solutions where people create their own dataset using OpenCV by capturing images of the hand gesture every couple of frames from different test subjects. I will continue to look for more datasets as I know the more labelled data that is available the more accurate the model will be.

Supervisor Seminar

During the Data and Visualisation meeting today, I had spoken to Dr Hughes and Dr Preiss about my sign language project idea and they both seemed very positive about it but were aware that it was a field in which quite a bit of work had been done, so they had encouraged me to find a new approach to it – finding a way to use it that has not been done before.

To Do List

- Do some research and thinking by next week into how I could add some novelty to this project idea.
- Create a list of at least 3 ways that I could use the tool in a creative and unique way.

Week 2

Hours spent of project this week: 12

07/10/2020

Lecture

This week's lecture focused on the project proposal. We were told the project proposal is an important part of any project as it sets out the background for the project as well as the objectives for the project. We were also told that creating a project proposal is important as it:

- Reduces the risk of there being radically different expectations for the project between the student and supervisor
- Helps to keep you focused on the important goals
- Timebound objectives help you keep track of progress

The structure of the project proposal was also discussed, for me I found this quite important as it will allow me to develop my project idea further and it gives me questions that I can look into and think about that will flesh out my project as well as my own understanding of what would need to be done.

These questions include:

- What is the aim of the project?
- Who is the project for: the client and their customers?
- Why is the project being undertaken?
- How will the client and customers benefit from the project?
- How am I going to benefit from undertaking the project?

What was done

Following up from last week, where I set myself an objective to think of a novel approach to this sign language project, I had created a list of ideas that I though would add some uniqueness to this idea. So, the base of the idea would be to use deep learning and computer vision in order to be able to classify the ASL/BSL fingerspelling alphabet.

One way I thought I could change this idea to be more unique would be to use it in a VoIP application. This could be in the form of a mobile/web/desktop application that would allow for face-to-face communication where when someone signs some letter it would appear on the other person's screen for them to read. However, in thinking of a practical use for this the other person would also need to communicate using text on the screen for the sign language user to read. In this case I would also need to explore text to speech as well as natural language processing (NLP) which in my opinion would bloat the overall scope of this project and make it much too complex.

I also had though of shifting the focus of the project to one that is more focused on the technologies used, such as carrying out a comparison of frameworks. So, for example, I could try and compare existing feature extraction systems for hand gestures to see which

are the most effective in different situations and which provide the best accuracies. Another thought would be finding an alternative to using convolutional neural networks to see whether that has a better ability to classify objects. Though I am not very keen on this idea, it is something I will keep in mind as a backup if I can't get another idea approved.

Another idea I had and one which I was more confident in achieving was to be able to use the classification of the ASL/BSL alphabet in order to create a learning application that would help people get started in learning ASL/BSL. How I imagine it acting would be that a user would turn their webcam or camera on to face themselves, they would be shown a letter from the ASL/BSL alphabet and they would need to try replicate it. Measuring whether they were successfully able to replicate the letter would be done through the CNN in saying whether it was able to classify the letter correctly. I had done some research on the official ASL site as well as some popularly listed sign language learning sites and had not found any feature like this – these sites were all using images and videos to teach.

I will be emailing Dr Hughes about my idea to get some feedback on which one of the above seem to most achievable and try to get some validation if these types of ideas are what he meant by making the project novel.

To Do List

- Make a start on the project proposal, starting to think about the project idea further by trying to answer the questions from this morning's lecture.
- Email supervisors to get feedback on project ideas I had from this week.

09/10/2020

Supervisor Seminar

In this week's supervisor seminar, there wasn't enough time to get through everyone, so I booked a meeting with Dr Hughes later in the day. During the meeting I went through my project idea and wanted to get some feedback on what he though, he mentioned that the learning application would be quite a good approach and that I be trying to scope out this project further to see what is out there already that could help me with the classification as well as in terms of hand gesture recognition.

To Do List

- Do some more research into the field of hand gesture recognition to see how feasible this project idea is as a final year project.

Week 3

Hours spent of project this week: 16

14/10/2020

Lecture

This week's lecture was about helping us think about the work methodology we should be using to manage the final year project. It went into detail about Agile and how an incremental approach to carrying out the project by splitting it up into multiple parts having a goal of each increment and focusing on accomplishing that. Agile is something I have already experienced from the second year as well as practicing it during my placement and I find it a good way to work for these large projects that need to be broken down so that each part can be planned, designed, built and tested individually allowing for a more focused approach to carrying out the project. It is definitely something I will be using to organise and execute this project efficiently.

What was done

This week I had begun writing a project proposal based off the questions that were present in the mark scheme and from last week's lecture.

- What is the aim of the project? – the aim of the project is to create a learning application using deep learning techniques that will help people learn in a more efficient and responsive manner that current online learning techniques.
- Who is the project for: the client and their customers? – the project is aimed at anyone who is looking to learn ASL/BSL online as well as people who want to practice what they have learnt in a way that provides you feedback on whether you are signing correctly.
- Why is the project being undertaken? – Because it would be a huge improvement to the current way of learning sign language online as there is no feedback or validation on whether you are signing properly without having to practice it with someone who already know sign language.
- How will the client and customers benefit from the project? – They will have an overall better learning experience that will allow them to learn in a more effective manner.
- How am I going to benefit from undertaking the project? – By learning a completely new stack that I am eager to learn and benefiting in an area which I would like to see grow which is online learning.

Supervisor Seminar

Last week I set a goal to for myself to do some research on how feasible the project idea was, though from what I found reading papers on hand gesture recognition it seemed doable, but I thought I should clarify the project in its entirety with Dr Preiss who is experience in doing projects such as this to get a better understanding the steps I will need to take. I talked to Dr Preiss in the meeting as well as well as through an email and gave me information about how I could check whether this project was feasible. The first being to find and appropriate dataset that I could use for training and the other being an existing feature extraction system that I could use to detect the hand gestures. I had already found the first through my initial research (https://www.kaggle.com/grassknoted/asl-alphabet) so it would be the case of just finding a feature extraction system that I could use.

To Do List

- Research feature extraction systems that would be appropriate for hand gesture recognition for next week.

Week 4

Hours spent of project this week: 16

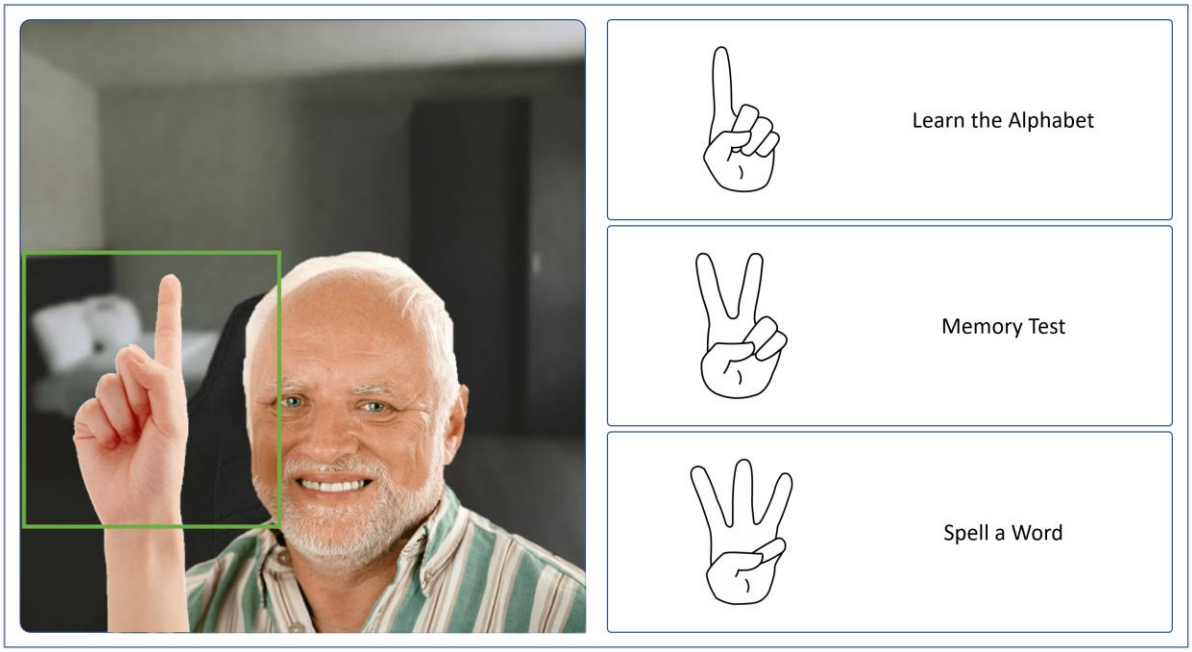
19/10/2020

What was done

This week I began thinking about the objectives that I should set for my project, so it really made me think how the final project would look and the different features the application would have. So, for example I had thought that after I have created the classification model, I could use it in different ways such as having different 'modes' for the application. These could be for example: a learning mode where it goes through all the letters in the alphabet asking you to replace them; a memory test mode where it will show you a letter in English and you would have to sign the ASL/BSL equivalent, helping test what users have learnt; and finally a spelling mode where users can practice and spell works with the letters they have learnt, further helping them memorise the letters.

I had also been looking into possible extensions to the project I could have as optional objectives that I can try to achieve if there is time remaining at the end of the project. One of these optional objectives I had thought of would be to provide varied feedback to users based on the accuracy of the classification. This would help users know how far off they are in terms of matching the letter helping them adjust and improve.

Another optional objective I had thought of was to create a web app front for this project that could be totally controlled through the use of hand gestures. I have created a wireframe below of how I imagine the menu for this web front to look. On one side would be the video feed from the webcam an on the other would be instructions to navigate through the different menus. I had talked this through with Dr Preiss and it was raised that a possible concern would be a conflict between the gestures to navigate the menu the gestures from ASL/BSL – this would be something that would need to be tested and played with to see whether it can be realisable.



To Do List

- Continue researching feature extraction methods as specified last week.

21/10/2020

Lecture – Research methods

This week's lectures involved the research methods for the project. Research process:

- Review the field – what is already out there and what is the most state-of-the-art technique/solution in this field.
- Build a theory
- Test the theory/hypothesis
- Reflection and integration

What was done

Last week I set out an objective for myself to have a look into the different types of researching different types of feature extraction systems. Some feature detection technique I had found include:

- Histogram of Gradients (HOG)
- Principal Component Analysis (PCA)
- Local Binary Pattern (LBP)

Through I have read through what each one of these are and how they differ, I currently am not sure how these techniques would be implemented in an application. This is something that I plan to research more about understanding and comparing the advantages and disadvantages of each in more detail over the next week. Some of the links used:

https://www.sciencedirect.com/science/article/pii/S187705092031526X

https://pdfs.semanticscholar.org/3089/e741052155313c99a72e6ecdfe12c14d81e3.pdf

https://towardsdatascience.com/feature-extraction-techniques-d619b56e31be

The rest of this week was spent focusing on finishing the project proposal in time for the deadline on the 23rd.

Week 5

Hours spent of project this week: 27

26/10/2020

What was done

With the project proposal completed I was now focused on carrying out all the research required for the remainder of the project. The plan I have in order to get all aspects of the project planned out in the next few weeks was to start off by going through all the things I would need in my minimum viable product. This would research would also be what I will be discussing in the literature review which is the next deliverable due on in Week 12. The research topics that need to be explore involve:

- existing tools already created for online learning and online learning in general
- what is the state of the art when it comes to learning ASL?
- has deep learning been implemented to improve online learning?
- how convolutional neural networks are made
- what convolutional neural network architectures are best suited for this project
- human computer interactions (HCI) involved with web design
- the web tools needed to create the web application
- the algorithms and techniques needed in order to develop the application

This week I began by researching tools that have been used to develop using OpenCV as it piqued my curiosity as well as creating my test applications to test a range of function available with OpenCV. From my tests I have learnt that there is not too much functionality with OpenCV itself to be able to integrate a convolutional neural network. This means that I will need to create the convolutional neural network myself. Some research will need to be done in order to final out what CNNs are and how they should be created, I will be adventuring these grounds in the coming weeks.

I encounter a large number of errors this week, however, when trying to set up the OpenCV package on my python configuration. This took a lot of hours to fix and was a very frustrating issue. Luckily I was able to face this issue in the beginning, where I also time to find alternatives otherwise, I would have panicked and had to have looked at alternative solutions.

To do list

For the next week I aiming to

- research exiting online learning tools and find

Week 6

Hours spent of project this week: 22

02/11/2020

What was done

Last week I defined a list of things that need to be research, the plan for this week will be to research existing online learning tools and what the key features involved are in an online learning experience.

The library I used to search all the papers was the ACM library, one that has millions of research paper and is a professional organisation that specialises in managing and storing papers. The first thing I wanted to find was just about online learning in general. I found a couple papers talking about the growth of online learning and from these the most cited was the paper by Bederson et al. called Introduction to learning at a scale. There was a lot of historic information in this paper that can be used as an introduction to my own dissertation. One thing in that interests was the table of statistics showing the growth in the online course market from 2005 to 2010, showing the rate of growth multiplying almost tenfold every year. The reason for this being, according to the paper, is because of the growth in the amount of people that have access to the internet – of which many being from poorer countries which have poor educational structures in place. This information can be useful in setting up the literature review and give some background context of the scale of online learning.

To do list

- later this week go over what learning applications need to have in order for learning to take place.

05/11/2020

What was done

Today I went over the principles of online learning. I believe this is an important talking point and factor in the development of the application and would be the justification for my many features later developed are essential to the created product. The first paper I found was by Stevens, E (2000) who talks about learning in a classroom setting. He mentions that the optimal learning pattern would be students start off by taking in information (which can be done through multiple channels), they then need to apply what they have learn to reinforce the knowledge. This lines up with what I already have planned for the application which is to develop a learning mode to show the users letter for them to replicate. After they have learnt the letter a memory mode would be used to help reinforce the knowledge

and make them think more specifically about the letters. Then the spelling mode would let the users apply their knowledge further reinforcing their knowledge.

Another noteworthy paper I found was from Paechter and Maier, who do more of a comparison between online learning and face to face learning. This is very useful in the sense of talking about ASL as it is mainly and most effectively taught in person, so comparing how it generically differs can be talking point in the literature review. Also, within this paper it talks about the what the key principles of online learning are and how they promote effective teaching. These principles include: well-defined learning outcomes, detailed course resources, interaction between student and instructor and finally the individual learning process. All of these can be used to relate to the to the needs of the ASL learning tool and what features need to be added as a result.

To do list

- work on finding what the state of the art for ASL tools involves

Week 7

Hours spent of project this week: 19

09/11/2020

What was done

This week I did a deep dive into all the ASL tools out there already. Unfortunately, there were very little work in in the subject areas in terms of learning. There was work done however in the classification the ASL sign language as a whole on the classification of the ASL letters. From the research done I was able to deduce that it was not effectively possible to classify the entire ASL alphabet with a generic webcam – something I had already figured out. It was possible however to use more advanced sensors such as a Kinect to classify entire sentences, but work like this was still experimental and had a way to go before it can be deemed effective and useful in the real world.

Since I was not able to find anything surrounding ASL learning tools, I moved on to do some research into what online courses where offering and what users felt about them through gauging online reviews. Most sites had mixed reviews and the majority of positive reviews were from live in person classes people were holding.

Surveying a popular ASL learning website “NAD” showed that the current main way ASL is taught online is through videos and text. There is not emphasis on helping people practice these letters. This can be seen as a barrier to entry for sign language is it is not too easy to pick up online because not many people are able to validate what they are learning.

Overall, this week a lot of valuable research has been done for the literature review, that will be very beneficial.

To do list

- start researching what CNNs are and how they are made

Week 8

Hours spent of project this week: 22

16/11/2020

What was done

This week consisted of both a theoretical research aspect as well as an experimental development aspect.

As there was still a lot of research that needed to be done on convolutional neural networks, I decided that I should use what I already have researched and start writing the literature review as I go along, saving me time and helping me finish on time.

I started off by researching what convolutional neural networks are and how they are developed. As I was doing the Deep Learning module, I already had some experience in creating models but not really any on CNNs yet. The place where I was able to obtain the most basic analysis was from the TensorFlow website. I was able to follow a couple tutorials from there and was able to develop a CNN of my own. However, when it came to training, I realised that training your own CNN from scratch would be immensely costly in terms of time and would depend on my pc's resources available.

This was a problem that I needed to overcome as without the CNN I would not be able to carry out this task. I started doing some research into popular CNNs used by professionals and came across and article that explained why creating a CNN from scratch was a bad idea. This article then mentioned transfer learning approaches to CNNs where a model is trained on a large number of images which allows them to have great general feature extraction. Then a new top layer is developed for the model which is trained on your own dataset and set of categories.

To do list

- Research a list of the most popular transfer learning models.

Week 9

Hours spent of project this week: 19

23/11/2020

What was done

This week a list of transfer learning methods was researched and wrote about in the dissertation. The models discussed included:

- VGG (e.g. VGG16 or VGG19).
- GoogLeNet (e.g. InceptionV3).
- Residual Network (e.g. ResNet50)
- LeNet-5

Starting with the VGG model, I was able to find out that this model was model that was proposed by Karen Simonyan and Andrew Zisserman in 2014, submitted to the "Large Scale Visual Recognition Challenge 2014" (ILSVRC2014) for which is achieved a 92.7% accuracy on the ImageNet dataset. The ImageNet dataset is large dataset with over a million images of different things. The architecture consisted of a series of configurations the one we will be discussing in the literature review will be the VGG-16 variant. The architecture looks as follows:

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Further research was done about this model, regarding its testing performance when it has been retrained. A paper was found from Strezoski et al. who used this model and retrained it for hand gesture recognition. This model was able to provide an average of 84% test accuracy, with an approximate training duration 2 minutes for a dataset of over 10,000 images – which is relatively impressive.

The second model researched this week was the LeNet-5 model, though this is not a transfer learning model, I saw that many reported it was effective when it came to image recognition. The model was developed by Yann LeCun, a notable figure in the deep learning

space. A paper was found where this model was used to classify images of the hand holding up numbers. This model bode well in the test having a test accuracy of 98.3%. However, it is to be said that the training times for the model were incredibly slow and did would not be suitable for a project like this were there is not much time for development.

To do list

- Finish researching the individual models

25/11/2020

Meeting

A meeting was held with Dr Preiss this week, where we discussed details of how the dataset needs to be used in the chosen CNN, mentioning that images may need to be converted to an array and reshaped to a specific size.

The topic of the web side of the application was also discussed. Dr Preiss mentioned that I should be aiming to add a section in the literature review about what would be the best possible option for the web side. Also, in relation to this, it was also mentioned what options were available in terms of hosting, whether it be through the university server or externally through AWS. Some research into the desired specs would need to be done in the coming weeks.

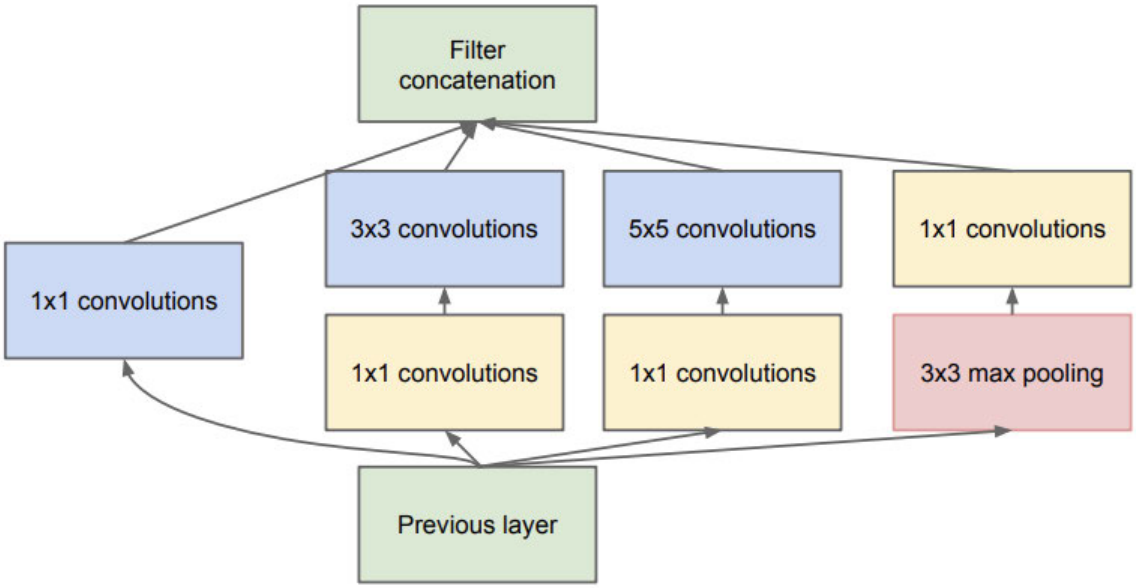
Week 10

Hours spent of project this week: 18

30/11/2020

What was done

This week I went over and looked again at another transfer learning model called the GoogLeNet or alternatively named Inception V3. This model has a vast model that is composed of a series of inception modules, seen below.



This module features a bunch of convolutions and max pooling at each layer in the model which all then concatenate and repeat. This model was used by Strezoski et al. in a previously mentioned paper and the model's accuracy when retrained on his custom dataset of hand gestures was 90% more than that of VGG and had an incredibly fast classification time of 2.8ms. This would be ideal for use in the ASL tool as we will be using a live webcam many images would be passed into the classifier and at this speed the response time between the front end of the code and backend would be miniscule.

Meeting

Another meeting was held this week, with the focus this week more on the modes that will be developed for the project. We discussed what they would do and specific features that would be possible to include. In addition to this we also discussed how the user studies would take place and whether or not a test should take place on whether a subset of users should be made to learn via traditional techniques and the other others made to test to product created with the experience being compared. Dr Preiss said it would be difficult to carry out this sort of test and would require more than just a questionnaire to come to a solid conclusion. Therefore, some though needs to be put into how the tests will be carried out in the second half of the module in an attempt to evaluate the product created in a fair and reasonable way.

To do list

- Finish writing up literature review and fill in the CNN section with the findings from the last two weeks.
- Start talking about the methodology of the product in each area of the project.

Week 11

Hours spent of project this week: 17

07/12/2020

What was done

This week was spent on researching sections of the methodology. The intended sections that will be discussed in the methodology would be:

- The development methodology
- The programming tools to be used
- The data set to be used
- The deep learn CNN
- The different modes of the application
- The web application
- The evaluation of the product created

Some research was done initially for the development methodology where notes were made on the agile process such as the techniques involved like sprints and Kanban planning. Papers were then used to support the use of agile stating it benefit and when it works best.

Research was also done regarding the web application in what frameworks were available with python that I could develop with. I was able to find two frameworks that could be implemented quite easily, the first being Django and the other being Flask. The research proved that Django is the more complete framework, whereas Flask was more lightweight meaning that I will be using the Django framework to code the web side of the application.

Meeting

During the meeting with the project supervisor, I was able to clear up what needs to be added to the literature review in terms of HCI. The rest of the meeting I just went over the things I have included in my literature review with Dr Preiss giving me small bits of feedback for each section.

To do list

- Finish writing the first part of the dissertation

Week 12

Hours spent of project this week: 12

14/12/2020

What was done

This week was focused on getting the first part of the dissertation out of the way, most of the research had already been done from the previous weeks, all that was required now was the write up.

Literature Review Deadline on the 18th.

To do list

- Start working on a minimum viable product

Week 13-21

As planned on the initial Gantt chart 5 weeks will be used to focus on the first Semester assignments for other modules.

In addition to that, unfortunately, due to personal circumstances an additional 3 weeks were taken off.

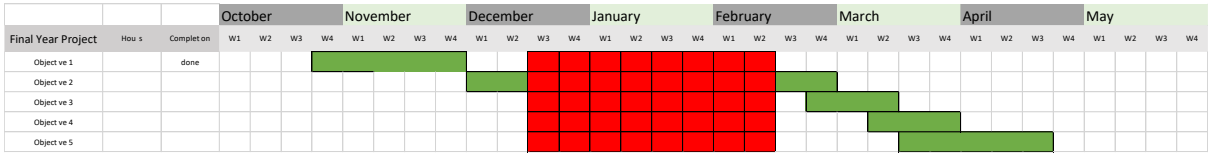
Week 22

Hours spent of project this week: 5

26/02/2021

What was done

Due to illness, I was had missed out a large chunk of time, which I had originally planned for the development stage. However, with the time remaining there was still enough time to get all parts of the project completed. A new plan was drawn up that if followed would allow me to meet all the objective of the project. The new schedule can be seen below.

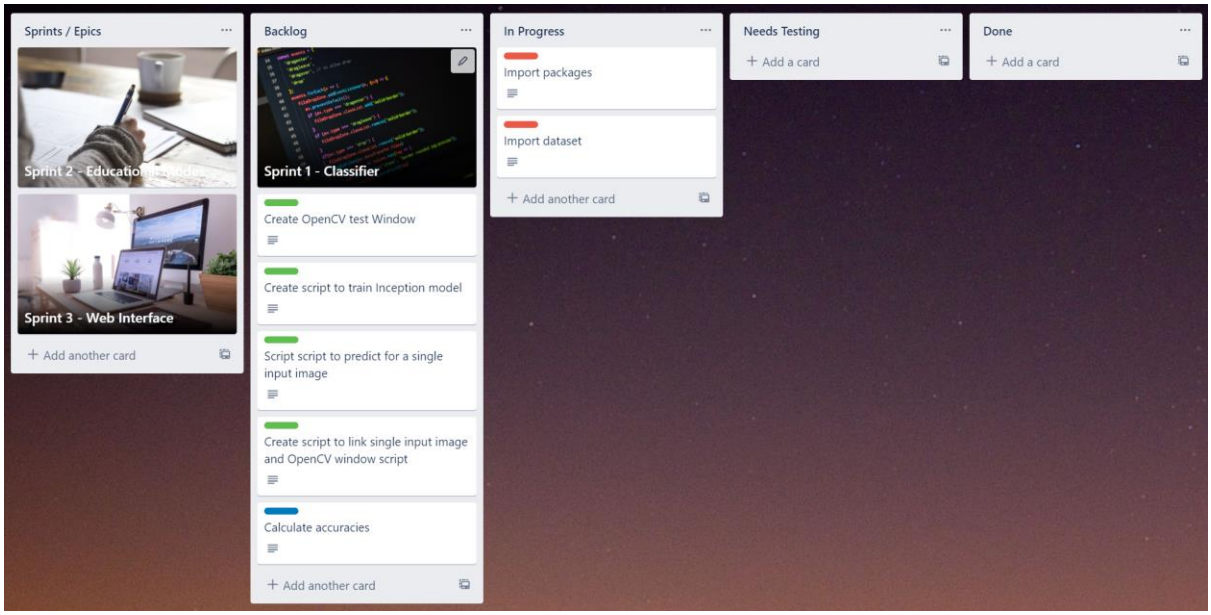


I had already started on an implementation of the deep learning neural network in the beginning of December, which gave me a something that I could follow up on and continue to develop.

I planned out my time into a series of three sprints, which is as follows:

- Sprint 1 – Classifier
- Sprint 2 – Learning Modes
- Sprint 3 – Web App

The task in the first sprint are outlines in this Trello board:



To do list

- Pick up from where I left off

Week 23

Hours spent of project this week: 28

01/03/2021

What was done

Picking up from December where I had started making a minimum viable product, I had started using the VGG-16 model in order to get a working classifier. I continued on from this work importing numpy reshaping the images from the dataset and starting to train the model. However, this is where the first major issue was encounter. The training process for this model was taking too long, at the rate it was going it would've taken hours to train once. At this stage of development, I could not afford the waiting times as I needed to test the system multiple times. I searched for a answer to solve where the model was going wrong and taking too long but after a two days with no luck I resorted to switching models.

I then decided that I would try using the Inception V3 model. This model utilises bottlenecks as a form of cache that allows for a slow training process the first time around but relatively quick for the next time the model is trained. As before I imported the Inception model using TensorFlow and began following the guide of the TensorFlow website to implement the code. This week I was able to user the guide to create a training algorithm which creates bottlenecks for every single image and caches them. TensorBoard was imported in order to keep track of the accuracies of the models. When training the model created a test accuracy of 93.8% was achieved.

```
Step: 0, Train accuracy: 16.0000%, Cross entropy: 3.100316, Validation accuracy: 6.0% (N=100)
Step: 100, Train accuracy: 69.0000%, Cross entropy: 2.330640, Validation accuracy: 55.0% (N=100)
Step: 200, Train accuracy: 79.0000%, Cross entropy: 1.828305, Validation accuracy: 79.0% (N=100)
Step: 300, Train accuracy: 79.0000%, Cross entropy: 1.478232, Validation accuracy: 86.0% (N=100)
Step: 400, Train accuracy: 81.0000%, Cross entropy: 1.370323, Validation accuracy: 80.0% (N=100)
Step: 500, Train accuracy: 88.0000%, Cross entropy: 1.117974, Validation accuracy: 87.0% (N=100)
Step: 600, Train accuracy: 85.0000%, Cross entropy: 1.024809, Validation accuracy: 88.0% (N=100)
Step: 700, Train accuracy: 88.0000%, Cross entropy: 1.019319, Validation accuracy: 87.0% (N=100)
Step: 800, Train accuracy: 95.0000%, Cross entropy: 0.801249, Validation accuracy: 82.0% (N=100)
Step: 900, Train accuracy: 91.0000%, Cross entropy: 0.754730, Validation accuracy: 84.0% (N=100)
Step: 1000, Train accuracy: 93.0000%, Cross entropy: 0.648651, Validation accuracy: 92.0% (N=100)
Step: 1100, Train accuracy: 92.0000%, Cross entropy: 0.699954, Validation accuracy: 87.0% (N=100)
Step: 1200, Train accuracy: 90.0000%, Cross entropy: 0.730426, Validation accuracy: 90.0% (N=100)
Step: 1300, Train accuracy: 96.0000%, Cross entropy: 0.628658, Validation accuracy: 97.0% (N=100)
Step: 1400, Train accuracy: 95.0000%, Cross entropy: 0.577774, Validation accuracy: 91.0% (N=100)
Step: 1500, Train accuracy: 96.0000%, Cross entropy: 0.463159, Validation accuracy: 95.0% (N=100)
Step: 1600, Train accuracy: 92.0000%, Cross entropy: 0.593629, Validation accuracy: 94.0% (N=100)
Step: 1700, Train accuracy: 93.0000%, Cross entropy: 0.526490, Validation accuracy: 94.0% (N=100)
Step: 1800, Train accuracy: 92.0000%, Cross entropy: 0.569649, Validation accuracy: 92.0% (N=100)
Step: 1900, Train accuracy: 96.0000%, Cross entropy: 0.382170, Validation accuracy: 93.0% (N=100)
Step: 1999, Train accuracy: 94.0000%, Cross entropy: 0.526165, Validation accuracy: 96.0% (N=100)
Final test accuracy = 93.8% (N=4768)
```

However, this result was only achieved after images from another dataset and images saved by me though a custom script were made. The original image dataset was too blurry and feature extraction was difficult on those images meaning, the classification accuracies were low. This was overcome and the mode is working perfectly now.

To do list

- Create a script that uses the webcam and is able to classify what the user is signing.

Week 24

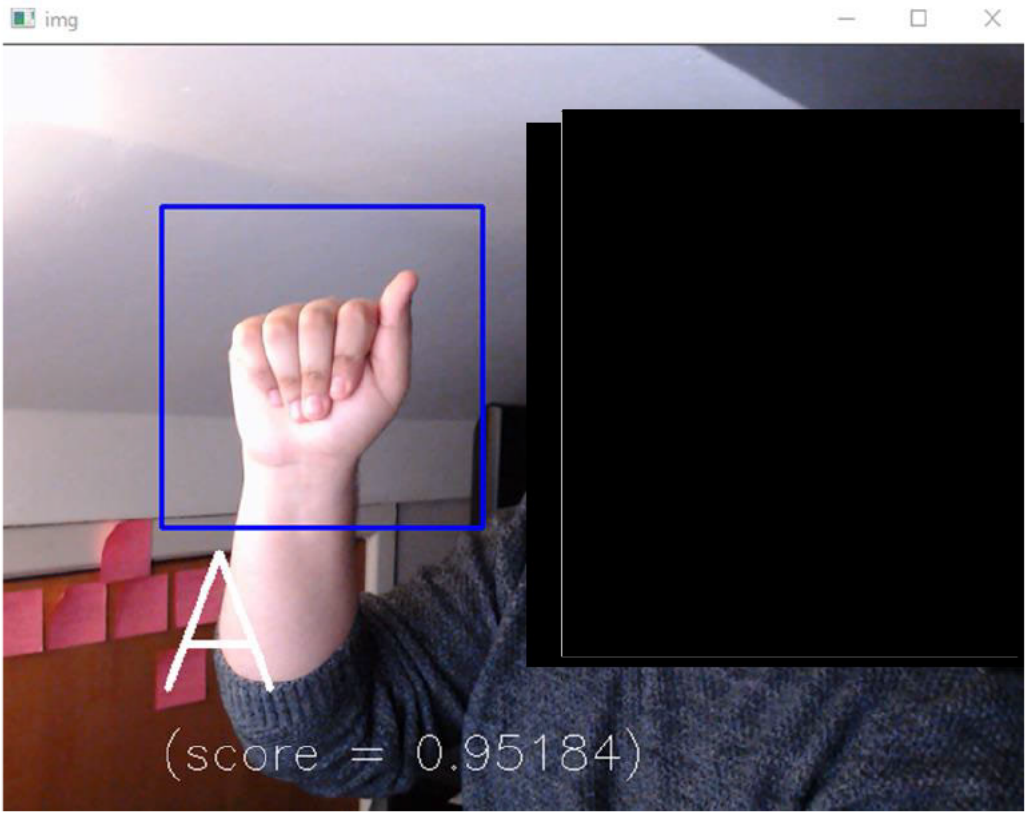
Hours spent of project this week: 31

08/03/2021

What was done

This week I managed to utilise the OpenCV application created in the first semester and add a box to the live webcam feed. This box is what will be fed into the trained model to get the classification result. The way this was done was by initially using the generator pattern to have a continuous piece of code running that is able to take inputs and output the result. A function was created that took a capture of the blue box every couple of frames (4 frames worked best after some testing) and converted the captured image to a numpy data array. This data array was then reshaped t a 224x224 sized image that would be ready to be passed into the retrained model to get the result. As seen in the image below I displayed the classification result as well as the accuracy of the classification as an overlay of the live

webcam feed user OpenCV. This would only be here for testing purposes as the final product this information would be displayed on a webpage.



To do list

- The next step was to develop the learning modes of the application and then integrate this whole back-end side into a front facing web application.

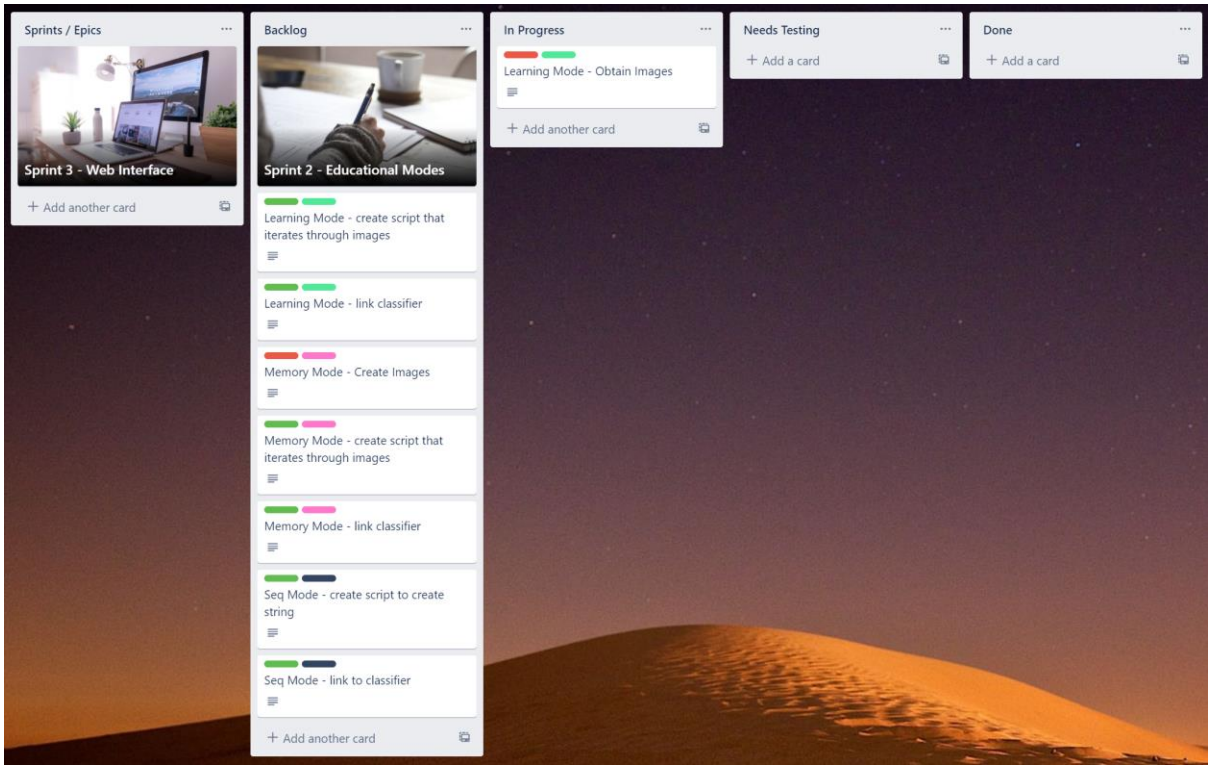
Week 25

Hours spent of project this week: 29

15/03/2021

What was done

Below are the tasks that were set out for the learning modes.

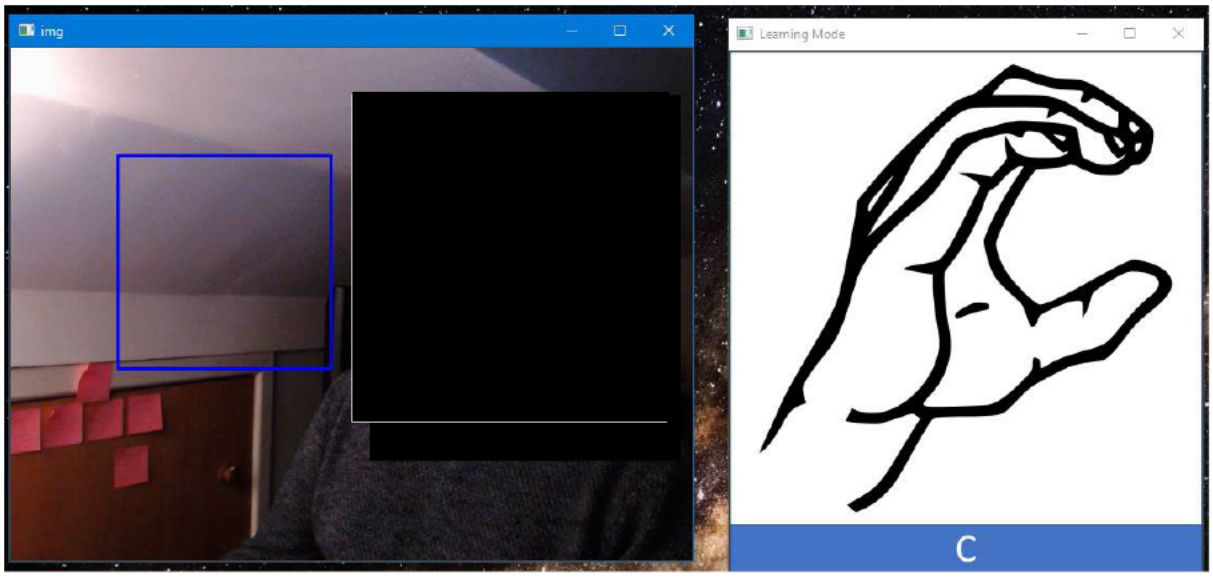


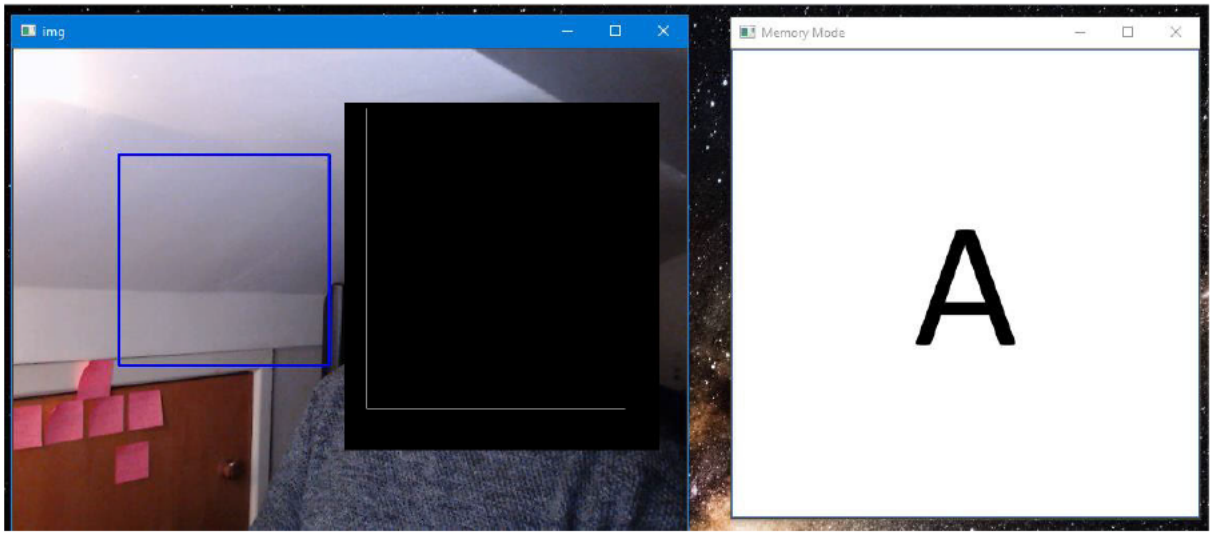
This week I began by going over the requirements for all of the different modes. As seen in the Trello sprint 2 image, there were a couple precursors that needed to be knocked out first. These included collecting the learning material such as the images to be used on the site which after some deliberation I decided upon using the ASL images from https://www.signlanguage101.com/free-lessons/asl-level-1 who had clear and accurate images. Also, images for the memory mode were created by having a letter on a white background.

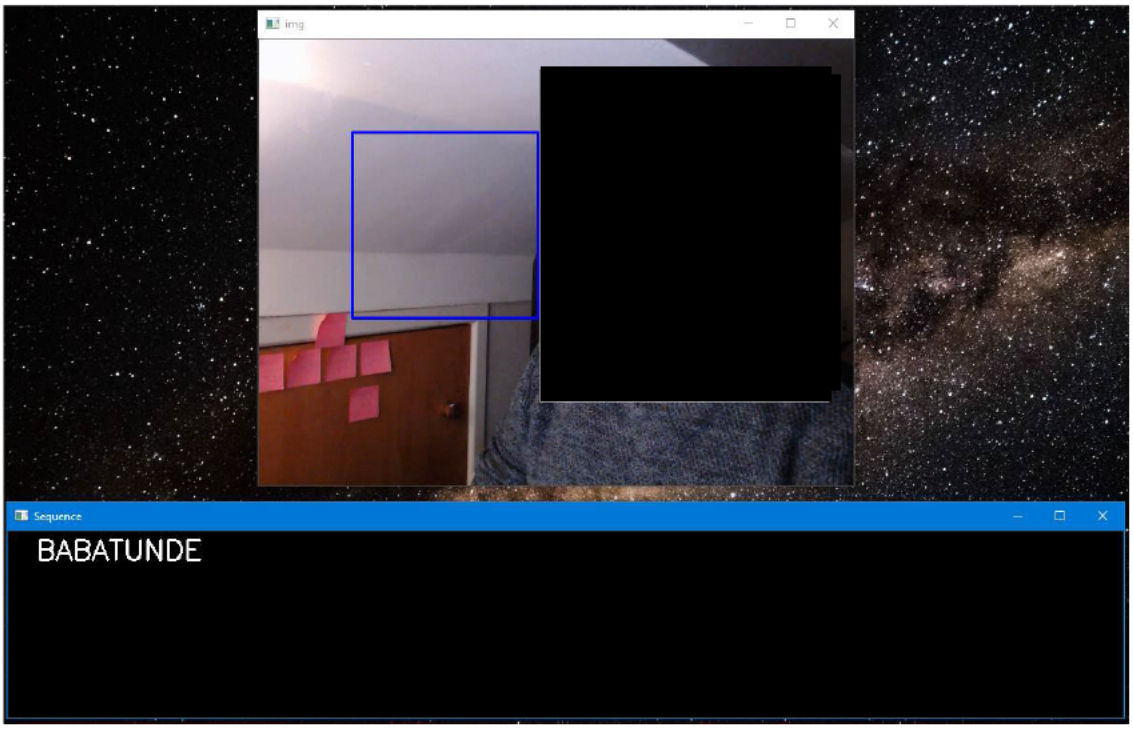
After this was done, I began working on the learning mode. These modes were relatively simple in that they only needed the output from the classification result and needed to be matched up with the array list of letters created. This then allowed for whenever, the classifier and the learning array were on the same letter the letter in the learning array would change and so would the image shown to the user. This method was a little too quick and there was not enough time for the user to realise the image has changed. Therefore, a condition was made that whenever the sign was being held up for more than 4 consecutive classification then the image would change. The blue box would also change to green whenever they are correctly performing the sign, giving some visual feedback.

The memory mode was developed very similarly but just had a different image array and the images were randomised instead of being in an alphabetical order like the learning mode. The sequence mode was quite similar too however whatever the user was signing was converted to plain text and appended to a string variable which was then shown to the user.

The results of this sprint can be seen in the images below.


img
Learning Mode
C


img
Memory Mode
A


img
Sequence
BABATUNDE

An initial attempt was made to create the web application using Django, but soon after importing it. It was clear that the there would be a learning curve and would take some time to get used to. Therefore, the this was then abandoned, and the Flask framework was used instead. Flask was really simple to set up and creating an MVC structure was straight forward.

After the structure was created the HTML templates were created for each section of the page including the classifier and each of the learning modes. The classifier routes were then created which led the application to the different modes. The controller was then linked to the classifier code and wired up so that the input of the classification can be shown on screen.

The feedback section was also created this week by having a separate class that would take the input from the classifier and output the necessary feedback as requested.

To do list

The modes that would be switched in and out using AJAX calls to the controller which switches out the content instantly will be done next week.

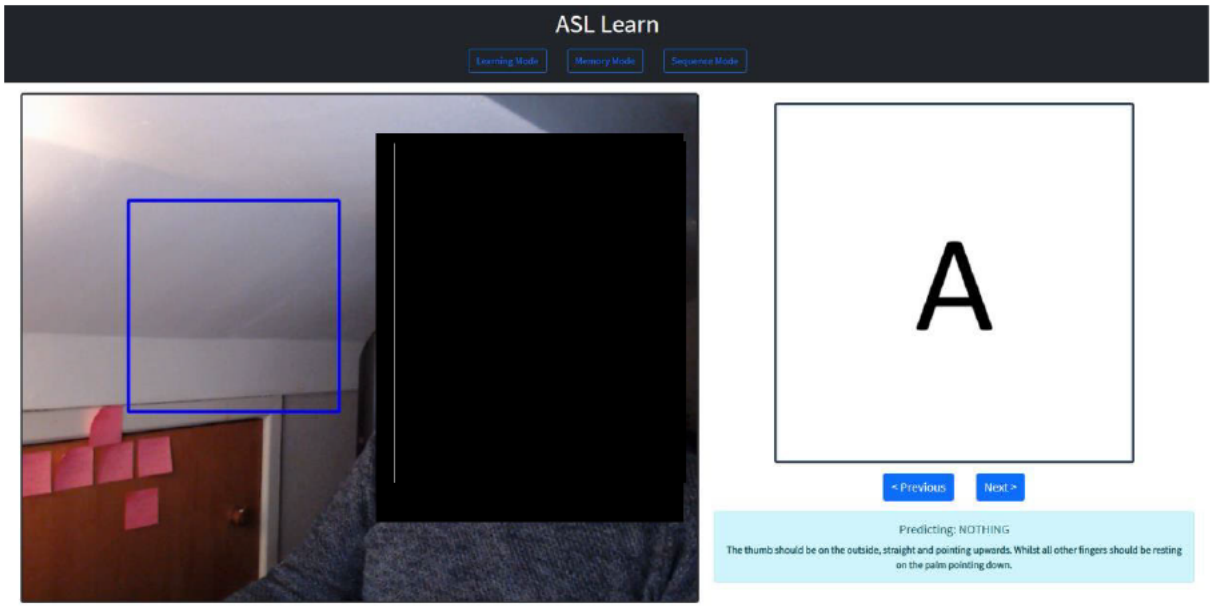
Week 26

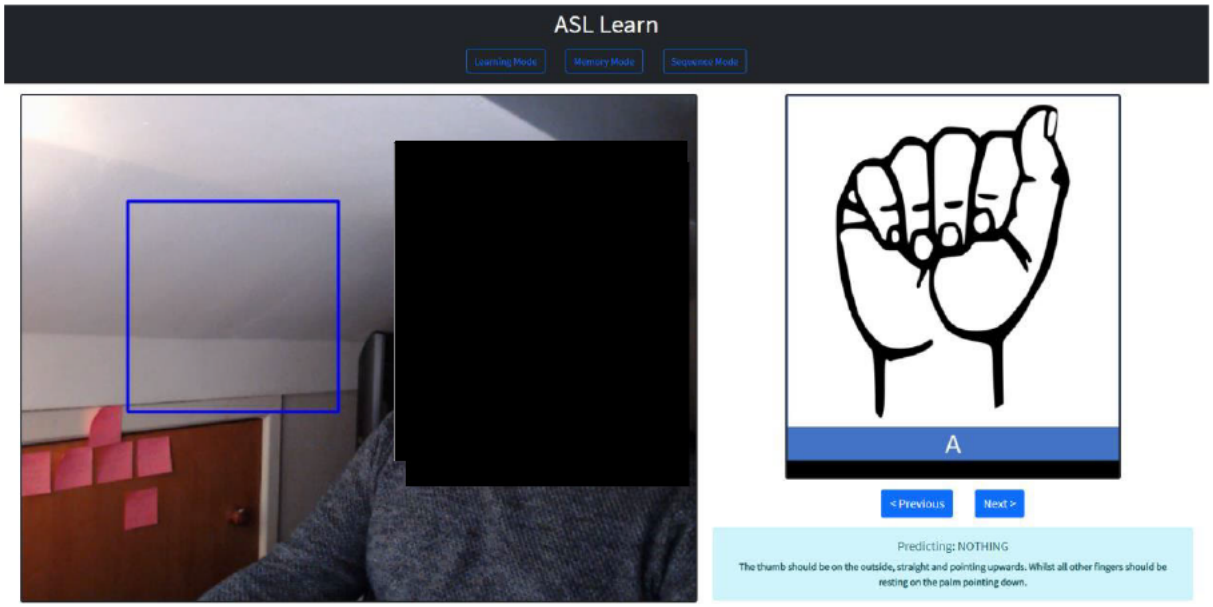
Hours spent of project this week: 27

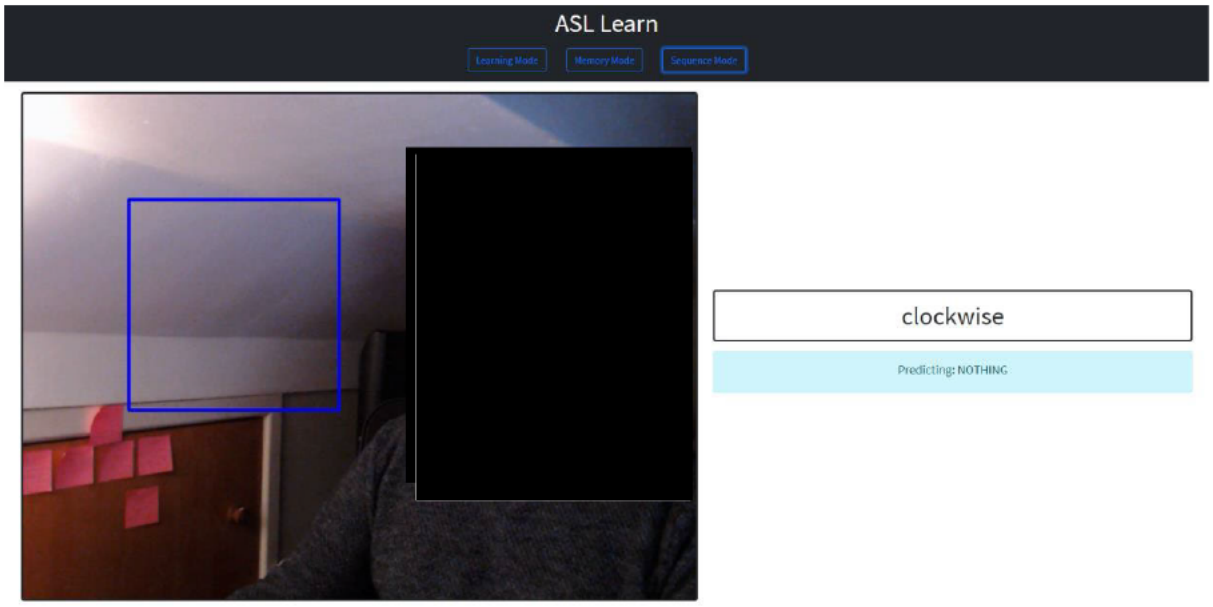
22/03/2021

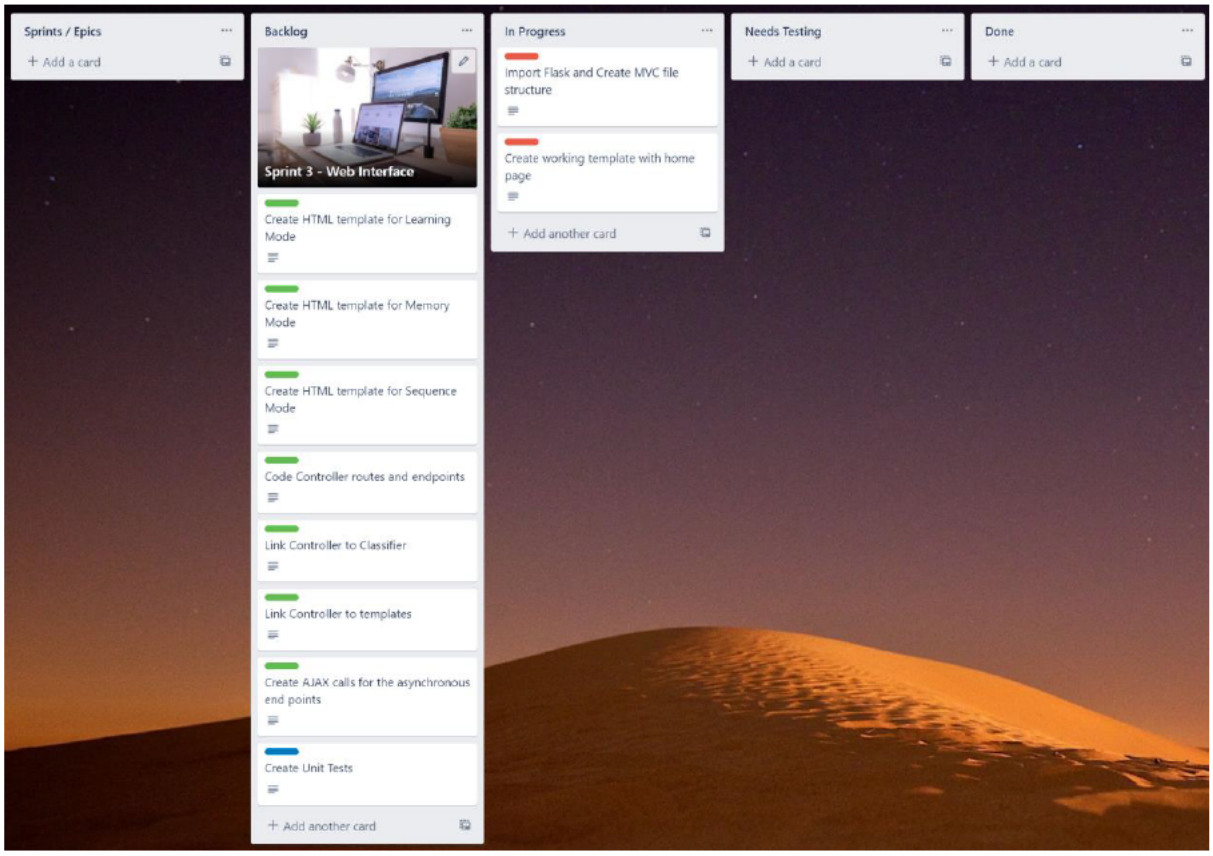
What was done

This week I began by picking up from the last week and continued wiring up the asynchronous part of the website. This was done by using jQuery and AJAX call to the controller endpoints to get the educational mode, the feedback and the classification result. The final images of the website can be found below.


ASL Learn
Learning Mode
Memory Mode
Sequence Mode
A
< Previous
Next >
Predicting: NOTHING
The thumb should be on the outside, straight and pointing upwards. Whilst all other fingers should be resting on the palm pointing down.


ASL Learn
Learning Mode
Memory Mode
Sequence Mode
A
< Previous
Next >
Predicting: NOTHING
The thumb should be on the outside, straight and pointing upwards. Whilst all other fingers should be resting on the palm pointing down.





After the product was now developed it was time to get the user study underway. A questionnaire was created in that featured a series of questions that would provide quantitative and qualitative feedback that can be used to evaluate whether the aims of the objective were met. Users were reached out to in a group of programming discord channels and friends of friends. I was trying to avoid getting close acquaintances as it would probably lead to bias results. Around 25 users had signed up for the project and they were given copies of the participation information sheet as well as a copy of the consent form. They were then given a week to reply on whether they would like to take part.

To do list

- Start writing the dissertation part 2

Week 27

Hours spent of project this week: 22

29/03/2021

What was done

Now that the product was finished in terms of development, I now focused on writing up the second part of the dissertation. I began by writing the requirements specification of the project and the process that led to the requirements for the project. I also mentioned the wireframes that were used which can be seen in from the early weeks of the logbook. This section also went over the structure of the code in the project where a series of UML diagrams were created and detailed.

During this period a number of users had contacted me with inquiries about the project and their involvement. Some of the users at this stage had dropped out as they had other commitments.

To do list

- Next week, develop the user study

Week 28

Hours spent of project this week: 18

05/04/2021

What was done

The final number of users that had signed up to participate was 19. The only issue at this point was that any attempt to host the site failed. I had tried AWS and the university server, and I could not get the classification code to run. Therefore, I had to create an executable of the project, which would run the classifier in the background and host a Flask server locally, the only caveat being that the users would need to install python for the project to work. Since most of the users were programmers, they had no issue, however a few more users dropped out at this stage.

The users that had dropped out were given an alternative test where they carried out just learning from an online ASL course and the feedback to the two groups was going to be compared.

After the users had set everything up, they were given a list of tasks that would make them go through all the sections of the product, after they had completed these tasks, they were then given a questionnaire that they had to fill out with a week to complete it.

Week 29

Hours spent of project this week: 21

12/04/2021

What was done

The development and implementation section of the dissertation was the focus of this week. The processes and method used and described in this logbook was further detailed in the dissertation. This section also discusses the algorithms, data structures and design tools used throughout the project.

During this week the user feedback was also return and are ready to analyse.

Week 30

Hours spent of project this week: 19

19/04/2021

What was done

This week revolved around creating the 'Testing and analysis' and 'Critical evaluation' sections. The user feedback was used to analyse whether the objectives of the project have been met. This week I will also be discussing the review of the project plan and the lessons learnt.

Week 31

Hours spent of project this week: 18

26/04/2021

What was done

With the product finished, the only part left was the dissertation. This week I focused on the conclusion and writing the abstract of the project. The conclusion discussed what was done throughout the project was reflected upon using the user results. This section also discussed future work for the project as well as the legal/social/ethical/professional issues related to the product.

Week 32

Hours spent of project this week: 6

03/05/2021

What was done

Now that the dissertation was finished, final checks were made to all parts of the project before handing it in.

Appendix B – Project Proposal

Using a convolutional neural network to classify the American Sign Language fingerspelling alphabet for use in an interactive learning application.

Student: [redacted]

Supervisors: Judita Preiss and Chris Hughes

Word Count: 1365

Introduction

The aim of this project is to create an interactive learning application, that utilises computer vision and deep learning methods, to help people learn and practice the American Sign Language (ASL) fingerspelling alphabet. This would be done through using a webcam to detect gestures the learner is signing and feeding it through a convolutional neural network to attempt to classify the ASL fingerspelling letter, providing feedback to the learner on whether they are signing the letter correctly.

Who the project is for and how they will benefit?

The project is aimed at anyone looking to start learning the basics of ASL, specifically the fingerspelling alphabet, in a more interactive and individual way - compared to currently existing methods available online. Currently, the majority of learning resources available online for ASL fingerspelling use videos to teach how to correctly sign the fingerspelling alphabet, however they do not test whether the learner has correctly taken understood this information nor do they test whether the learner is able to sign the letters correctly. Through carrying out this project I hope to fill in the gap in this area of online learning so that learners are able to get individual feedback when they are learning ASL to better their signing and to strengthen their knowledge as well as make them more practiced and adept.

The project will benefit these users by providing an overall online better learning experience which would be done through multiple ways. The main one involves the ability for the application to communicate and feedback to the user when they are incorrectly signing a letter, showing them the correct way and allowing them to practice and alter how they are signing to be more accurate and understandable. Another way is to allow for the ability to practice what they have learnt through different areas in the project such as a spelling area where they could spell out words with the letters they have learnt. This will help them retain information better as they could test what they have learnt immediately and also receive feedback immediately rather than waiting to go out and test it with someone else who knows sign language – which can be difficult during a time like this, where we are in the middle of a pandemic.

Motivation

One of the main reasons why I decided to undertake this project was to be able to develop my own skills and knowledge in areas of computing that I am captivated by and really eager

to learn more about - which are data science, deep learning and computer vision. Seeing the advancements made in recent times using deep learning, such as facial detection and autonomous cars, show that deep learning is an exciting field to get into and one that will grow rapidly in the years to come. Therefore, It is something I look to become proficient in for the future which is why I believe choosing it as a final year project would a good challenge for me to get to grips with the practical and technical aspects of deep learning, as well as it potentially opening up a career into deep learning and AI.

Other than personal growth, I also see this project as something that could help contribute to an area that I would like to see grow – which is intelligent online learning. I believe that whenever you are learning something it is important that you are able to validate your own knowledge as well as being able to put into practice the things that you have learnt. This is especially relevant for when you are learning sign language, but currently when learning online there is no way that learners could get feedback to validate what they are signing is correct, individually, learners would have to go out and practice with other people – which in a time like this during a pandemic it is not a totally viable option. So, in creating something like this I hope that it could push learning online to be more independent and interactive to provide a better learning experience for everyone.

Objectives

These are the key objective that I have set out for the project:

- **Objective 1**: Perform a state-of-the-art review to survey the field of research in automatic detection of ASL. *Estimated time: 5 weeks.*
- **Objective 2**: Develop a feature extraction algorithm for the detection of ASL gestures, through using existing techniques. *Estimated time: 5 weeks.*
- **Objective 3**: Develop a NN for the classification of finger spelling characters. Measured by ensuring the average classification accuracy is at an acceptable level. *Estimated to take about 6 weeks.*
- **Objective 4**: Create a Learning System that features multiple modes including: a learning mode, a memory mode and a spelling mode. *Estimated time: 5 weeks.*
- **Objective 5**: Evaluate the tool with automation and human users, measuring the accuracy of the model created. *Estimated time: 3 weeks.*

These are some optional objectives that I hope to achieve if there is time to do so before the end of this project:

- **Optional Objective 1**: Develop a hand-gesture-controlled web front that allows navigation between the application's different modes. *Estimated time: 4 weeks.*
- **Optional Objective 2**: Develop a more advanced feedback system that utilises the classification accuracy value to present different levels of feedback. *Estimated time: 4 weeks.*

Development Requirements

The programming language of choice for this project will be Python, this is because of the extensive selection of libraries and frameworks that are available to implement deep

learning applications. Some of the libraries I plan to use include Keras, TensorFlow, Jupyter, NumPy and OpenCV – which I plan to manage with the use of Anaconda in order to ensure all package versions installed are compatible with each other. The IDE I will be using is PyCharm as it is a fast IDE to develop with and has support for all the packages listed above.

In terms of hardware, I will be developing the project on my own PC which specs include an I7-7000k 4.2Ghz Intel CPU and a Nvidea GTX 1050-TI GPU. I believe that these will be suitable in developing the project as well as provide relatively quick time to train the algorithm. I will also be using a Logitech 1080p 60fps web cam which will help capture clear images of hand gestures.

Managing the reporting side of the project will be done through the use of Microsoft 365 tool such as MS Word, which will be used to keep a logbook and write the final report the project and Excel to maintain a Gantt chart which will be used to schedule objective

Methodology

I believe that in order to achieve the objectives I have set out for this project; an Agile approach would be the most adequate working methodology. One reason why I believe this is because of the nature of this project. The project I will be creating consists of developing multiple parts that would merge and form a complete application. So, using Agile in this case would be beneficial as I would be able to split each part of the project into increments in which I could plan, analyse, design, build, test and demo to supervisors individually. This would allow for the project to be built in a more focused and productive manner, delivering smaller pieces of work that I could get feedback on and be able plan accordingly for the future increments. This would also help reduce risk in the project, as I could ensure whether or not I am on the right path and make adjustments if needed, which will be important when working in a complex field like deep learning.

Time Plan

Below I have a Gantt visualising how I will be taking on the project. As in the event that I am not able to meet the deadlines I have set out for myself, I have left a period of time at the end of march which I could feed into for my main objectives in order to make sure those are met.