Winter 11-8-2020

Deep Learning Application On American Sign Language Database For Video-Based Gesture Recognition

Muhammad Murtaza Saleem
Sheridan College, salemuha@sheridancollege.ca

Deep Learning Application On American Sign Language Database For Video-Based Gesture Recognition

A Thesis

presented to

School of Applied Computing, Faculty of Applied Science and Technology

of

Sheridan College, Institute of Technology and Advanced Learning

by

Muhammad Saleem

in partial fulfilment of the requirements

for the degree of

Honours Bachelor of Computer Science (Mobile Computing)

June 2020

Deep Learning Application On American Sign Language Database For Video-Based Gesture Recognition

by

Muhammad Saleem

Submitted to the School of Applied Computing, Faculty of Applied Science and Technology
on December 13, 2020, in partial fulfillment of the
requirements for the degree of
Honours Bachelor of Computer Science (Mobile Computing)

Abstract

ASL speaking individuals always bring a companion as a translator [1]. This creates barriers for those who wish to take part in activities alone. Online translators exist, however they are limited to the individual characters instead of the gestures which group characters in a meaningful way, and connectivity is not always accessible. Thus, this research tackles the limitations of existing technologies and presents a model, implemented in MATLAB 2020b, to be used for predicting and classifying American sign language gestures/characters. The proposed method looks further into current neural networks and how they can be utilized against our transformed World Largest – American Sign Language data set. Resourcing state of the art detection and segmentation algorithms, this paper analyzes the efficiency of pre-trained networks against these various algorithms. Testing current machine learning strategies like Transfer Learning and their impact on training a model for recognition. Our research goals are: 1. Manufacturing and augmenting our data set. 2. Apply transfer learning on our data sets to create various models. 3. Compare the various accuracy's of each model. And finally present a novel pattern for gesture recognition.

Keywords: American Sign Language, Deep Learning, Convolutional Neural Network, Models, ASL Data set, Computer Vision, Transfer Learning, WL-ASL. ...

Thesis Supervisor: Dr. Abdul Mustafa
Title: Professor, School of Applied Computing

Acknowledgments

This is dedicated to Richard Comeau. Many thanks to my Advisor: Dr. Abdul Mustafa for assisting me in this project especially during such a difficult time. Including Dr Rachel Jiang who assisted me in the first half

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Sign language is a form of visual communication which involves a complex combination of hand movements. Certain placements of fingers can represent individual characters, while a complete motion of characters and phrases translate to a full sentence or gesture. According to World Health Organization, the number of hearing-impaired individuals has reached over 400 million [5]. While it continues to rise the fact is that signers, impaired individuals, will always bring a companion to translate for them. This is due to the lack of accessibility in closed and social areas. In the modern world, 2 to 3 out of every 1,000 children in the United States are born with a detectable level of hearing loss in one or both ears [6]. With many innovations in technologies, there are plenty of mitigation to hearing loss including cochlear implants. However, these are meant for the modern world where healthcare is essential. Countries like Pakistan unfortunately do not provide the same luxury to children affected by hearing loss, whether it be through pollution or trauma [7]. Thus, many children are taught at a young age to sign in native 'Urdu' and 'English' [7]. Retail-environments and walk-in businesses struggle with the barriers associated to speech too, thus losing valuable connections with hearing-impaired individuals. Alfonso Castillo reported that signed language interpreters are lacking [8]. Not only retail, but debates, conferences along with legal proceedings provide limited accessibility for the hearing-impaired. Many

current solutions have been built on glove-based techniques. These computers read the embedded sensor signals in the glove[9], but it is highly dependent on a product and cost which limits impaired and non-impaired individuals. Cameras are what everyone has, luckily most devices have camera's and are also programmed for accessibility [10]. With that in mind, there should be a proposed solution that can be efficient in an environment of point-point communication. This would provide a much higher quality of living for folks that regularly communicate using ASL. With advancements in technologies, human computer interaction, HCI, has greatly improved. Automated systems and computer generated task now handle most analysis and predictions, thus giving the rise to the era of Machine Learning [11]. With an endless level of research in the field of machine learning. The motivation behind this study was to create our own machine designed to learn human features gestures for signed interpretations. Establishing a novel model that can be used for predicting and classifying ASL without the need of a physical translator.

1.2 Approach

Computers and machines have the ability of learning based of images, signals, tabular data, etc. This method of learning is also known as machine learning[9]. We can train our machine to learn/re-learn the nuances of signers. Existing applications that utilize computer vision include but are not limited to:

1. License Plate Detection,
2. Face and Emotion Detection,
3. Stop Sign Detection,
4. Plant Detection

Most of these applications are also created in a similar manner, such that the machine is fed an array of labeled images or raw data. These images are then trained by our machine using a high level abstraction algorithm that seeks to find relationships

amongst features [11]. Once completed our machine will be able to apply an algorithm best suited to pattern recognition. These patterns and algorithms are crucial for our machine to make informed decisions about unsupervised data. Where unsupervised data would be unlabeled data not fed to our machine for training. Application that are built for ASL characters have proved fruitful in recognizing static images. Alphabets, and Numbers can be predicted with 80% accuracy [12]. This paper will investigate the applications of machine learning on a much larger data set. In order to develop a unique pattern for gesture recognition. Our workflow can be described using the following use-case diagram:

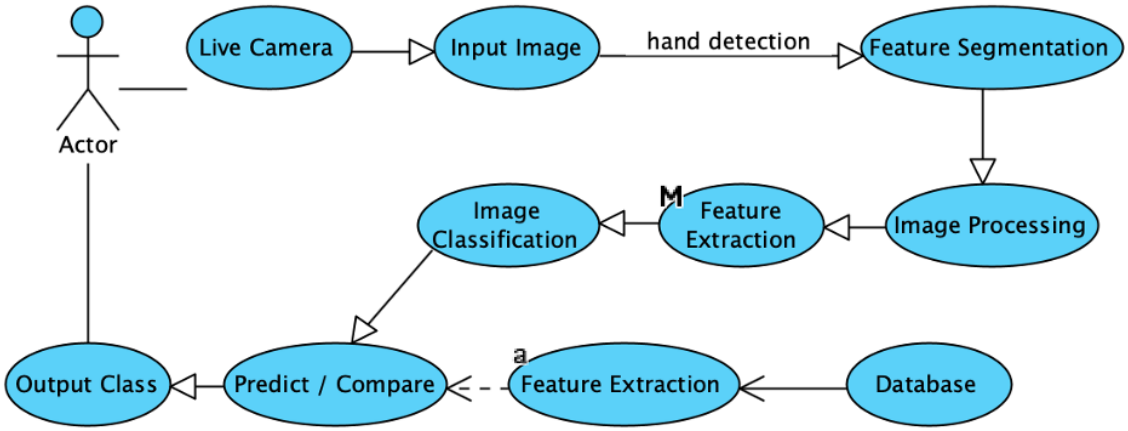

Figure 1-1: Provides an overview of the program workflow. Where each bubble is it's in own class, and the arrows represent the flow of data between classes until it comes back to the Actor, which represents either the end-user or the mobile device.

a large and publicly available dataset is needed. WL-ASL also known as World's Largest – American Sign Language data set released Jun 2020 [4], is studied in this research. Using MATLAB tools for Deep Learning and Image Processing, [10], MATLAB tools provide a solution to retraining available networks, and this subset of deep learning is known as Transfer Learning [13]. Deep Learning, as a branch of machine learning, uses a series of high-level abstraction against a network of filters. The network can thus translate processed data to output an accuracy to a number of classes. The network not being limited to vision/audio or un-ordered list. Developments in deep-learning and computer vision have already provided valuable products in ve-

hicle detection, face recognition, and audio manipulation. The goal is to provide a novel solution to pattern recognition in videos or images. The best way to describe a pattern is to develop an arithmetic algorithm. These arithmetic models get

1.2.1 Transfer Learning and Neural Networks

Figure 1-2: Visual to transfer learning, Building on top of pre-built models [1]

Transfer learning can be described as starting on a mountain, similar to Fig 1.2 where most of the work that will be done relies on the back of giants in the data science field. Transfer Learning is a subset of Deep Learning. It provides a new method of enhancing neural networks. The process of transfer learning involves an existing network and fine-tuning that network by making adjustments to training parameters [1]. This adjustment can improve feature representation so that new task can be supported by existing neural networks. In this research, the transfer Learning method is applied to enhance our network by retraining it to a new augmented data set. Our network can have a wide array of layers, however, the precision of our data set and the variability of our data heavily influence our machines.

Although being cautious of not over fitting the network with too many features and too little variation [1]. Our goal is to work on top of current neural networks that have been trained on a million images. Creating a process similar to

Current applications apply transfer learning on networks to detect Regions of Interest or further enhance Object Detectors like YOLO for real-time detections [14], this can have a significant effect in our segmentation phase. Improving real-time detection and classification is the goal of this research thus our Neural Network will be fine-tuned by using our training enhancers and custom training data. By adjusting the network, we find approaches to further mitigate any issues that can occur during neural network training. One of the concerns that this research will look out for is under fitting and over fitting during our training. With such an extensive list of features there is a possibility of under fitting our neural networks. The best way to spot these errors to understand how the data plots. Fig 1.3 provides a visual to what over fitting and under fitting looks like versus a fit/Robust model.

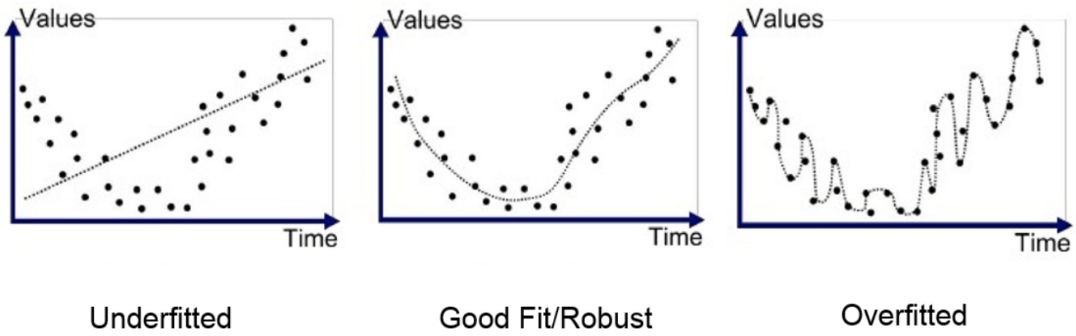

Figure 1-3: "Under fitting, Over fitting in machine learning and how to deal with it" [11]

Neural Networks have a huge role to play in our classification algorithm. Many neural networks currently proposed in Sign Language recognition have been trained on static alphabets, but not gestures. One of the most novel sign language recognition algorithms was introduced by SignFi, [15], a Convolutional Neural Network, that could establish 276 sign gestures. Although novel for its work, their model tested with 60% accuracy [15]. This research will find a way to improve on the SignFi system with

the use of Transfer Learning. To conclude, this thesis presents a model tested against the processes of Deep Learning to prove the reliability of transfer learning on a set of large features. The research that lead to this process is discussed in our Literature Review in Section II. Our process is illustrated in Section III, Methodology. Findings are described in Section VI, followed by the conclusion of our work.

Chapter 2

Literature Review

Machine learning, computer vision and deep learning are all closely related principles, heavily influenced by statistics and expanding the field of data science which all conveniently can be computed using MATLAB 2020b. MATLAB can allow users to create/edit networks in order to build new models [16]. Provided by their extensive toolbox for computer vision and deep learning. Current MATLAB techniques allow explicit and implicit monitoring, where we can isolate moving and non-moving parts. HCI technologies present Vision Based Recognition as a 'challenging interdisciplinary research area'[17]. A successful working system is dependent on the robustness of the available data, efficiency of the algorithms, tolerant against mistakes and scalable to all machines. Current machine learning solutions for sign language interpretation address the issues of gesture recognition such as color, texture and position inconsistency images [10]. Many have focused on individual characters in sign language interpretation using video or images as inputs. An approach using machine learning can be split into computer vision and hardware-based solutions. Hardware-based sensors relied on accelerometers and flex sensors are capable of detecting precise movements of fingers and hands [18]. Thus, hardware technology provides higher accuracy by capturing data directly [10]. The limitations to this form of detection is the semantic evaluation of the data [18]. In addition, these sensors are limited by the abilities of combining the facial expressions with gestures to accurately perform the translation in real-time [19]. Glove-based techniques have a proven 100% accuracy due to accelerometer, the

research conducted was focused around vision-based detection algorithms [18]. This section will provide a historical analysis based off of our current system flow diagram.

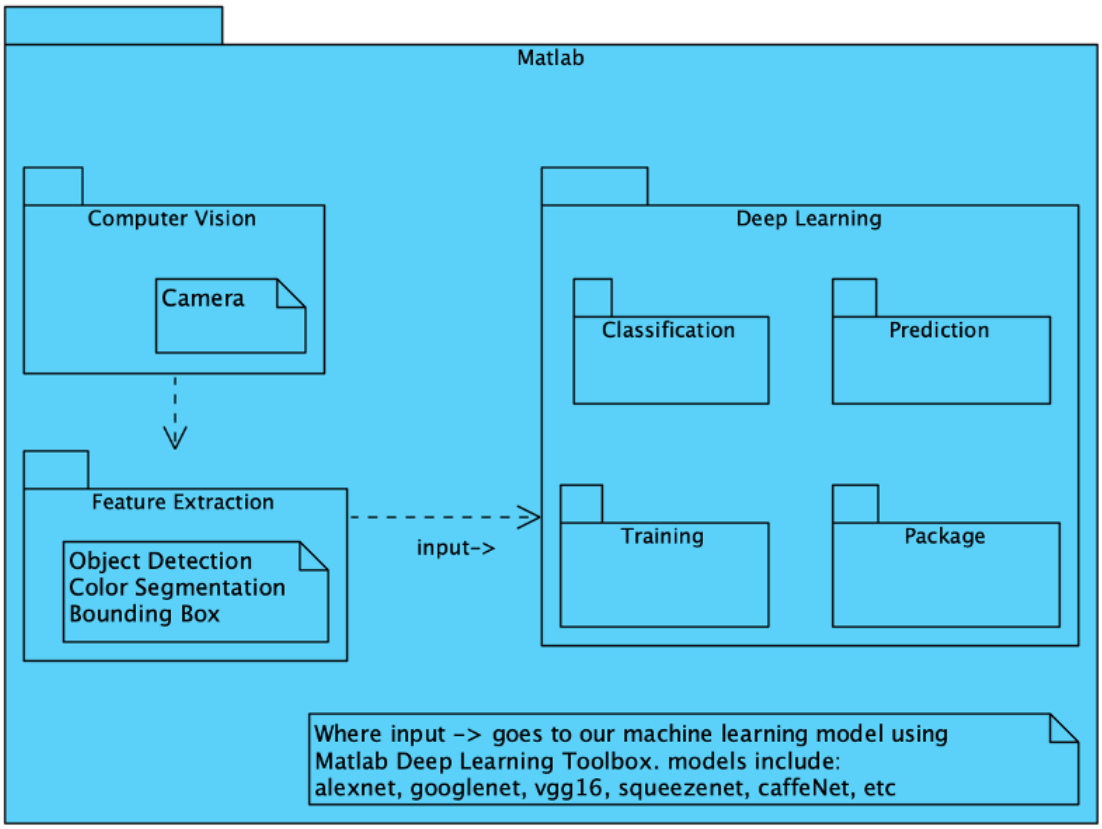

Figure 2-1: Where each package represents the individual files and functions associated with that class.

2.1 ASL Detection Systems

Hasanizzaman. [20] was one of the early presenters to vision based ASL system. Using image-property analysis and skin color segmentation as two separate features to recognizes 14 gestures [20]. HMM, also known as Hidden Markhov Model, is one of the most popular in vision-based gesture recognition [9]. Prof Chen was able to present a hand gesture recognition that provides real time hand tracking, HMM training and gesture recognition. Although the paper utilizes the spatio-temporal characteristics provided by Hidden Markhov Model, it maintained an 80% accuracy

on just 15 gestures [21]. Another one of the earliest papers on vision sensors, tackled by Prof. Starner T.E, applied the Hidden Markov Model [9]. This model became the steppingstone to realizing non-linear filtering problems, improving reinforcement learning. The work established the use of the model to translate American Sign Language Vocabulary using a single camera module [9]. Drawbacks of this approach included only recognizing a subset of characters and required users to wear dark-colored gloves [22]. Similar work was performed by Prof. Pratibha Pandey, who was able to establish an efficient algorithm for sign language recognition by applying 'glove-based' techniques for real-time video capture [2]. Pratibha applied several techniques including color-markers against gloves, the accuracy of her algorithm proved 90% accurate on her test-set of characters from 0-10 [2]. Figure 2 below provides a visual to the color marking detection algorithm.

Figure 2-2: Color Marker based hand Gesture Recognition [2]

2.2 Segmentation & Extraction Algorithms

Skin segmentation research involved finger/hand detection to spell ASL through character translation [12, 22]. There are many datasets available in these types of research in terms of characters and different skin color. The American Sign Language dataset is comprised of many different characters under different lighting and was used in

many peer-reviewed papers most notably done through Dr. A. Jalal using Deep Learning, a subset of machine learning [13]. Published 2018, the research is up to date to latest technologies in machine learning, deep learning being one of them which will be explored later in this review. Jalal's approach was effective and employed a built-in non-linear algorithm achieving 70% overall accuracy [13]. Further research can improve the accuracy by incorporating skin segmentation methods and a more appropriate dataset for gestures. Many skin segmentation and detection methods will always use an algorithmic approach. In machine learning these algorithms can vary in computation time, and the more complex the algorithm, slower computation times [12]. To counter the limitations multiple algorithms will be explored. This research conducted with ASL algorithms will be applied to lower-latency devices for deployment in work-force settings such as mobile devices. Research shows that efficient models, which involves less computations and processing time, are necessary for lower-latency devices, such as mobile devices [23]. Mobile applications later incorporated these lightweight models [14], providing users with the capability of machine learning through their devices. Google was able to create a very useful model for low-powered devices, MobileNet which could be easily integrated with IOS/Android devices [23].

2.3 Object Detection & Recognition

Recognition in machine learning can be tackled in various forms, most notable forms are through Skin Segmentation and Feature Detection. Finger tracking and color analysis yield 85% accuracy [24], many other detection systems exist in using built-in programs like MATLAB, such as:

1. Object-detection using ROI [25]

2. Pixel segmentation [2]

3. Point-tracking segmentation [25]

State-of-the-art features like Histogram of Oriented features (HOG)[**?**], Scale-invariant feature transform (SIFT) and speed up robust features (SURF) [26], that provided an accuracy of 70% which is acceptable in semantic classification. The work studied elaborated on the purpose of detection and how it can provide an edge in classification, especially if the study is to be performed in various external conditions. Further evaluation shows the importance of classification, and how detection is crucial for classification purposes. Prof. Parama notes: 'In order to improve classification our model for detection must fit extracted data for classification'[12].

2.4 Classification Models

Existing classification methods included CNN, Adaboost, Deep Forest, Eigen Distance Classifiers, K-Nearest Neighbors, Principle Component Analysis, etc. [26]. All of which were developed under NLP, natural language processors. Among these classifiers, most popular of them being SVM, support vector machines [27], which works well against multiple features to classify multiple objects. SVM and deep forest are chosen as typical classifiers to make an in-depth analysis [18]. SVM was invented in 1963 it tackled linear problems, come 1992 developers applied a kernel trick making it kernel-linear [30], suitable for multiple classes it would pave the way for new technologies in machine learning. The classifiers studied are also the neural networks that our processor will be training against. Kenshaw University studied these classifiers against the American sign language data set [25], noted that in order to create an accurate model for classification the efficiency of the neural network is dependent on the layers the neural network presents. Layers can be extensive depending on the filters associated with the network. CNN, Convolutional Neural Networks work best with problems that are associated with multiple filters [28]. Many of the classifiers previously stated fall under the same category of CNN, like SVM. Shahriar's paper in Convolutional Neural Networks study's this area and notices that CNN would be more efficient if the network itself was recursive in information gathering, such that training, and testing would be intertwined [29]. The answer to the solution be-

ing Deep Learning, which was briefly mentioned before using Dr. Jalal's paper for segmentation and deep learning [30].

2.5 Deep Neural Networks

Deep Learning rose during the 21st Century and presented a recursive neural network language, superior to natural language processing, NLP, as it could obtain semantic and syntactic information. Unlike NLP models it can vectorize multiple features against a sequential dataset, significantly improving Unsupervised Learning [13]. This branch of machine learning uses a series of high-level abstraction models and in this research will be our success factor. Developments in deep learning and computer vision have already provided valuable products in vehicle detection, face recognition, and audio manipulation. Deep Learning Framework act as steppingstones for designing, training and validating neural networks. Using this technology real-time recognition can perform faster and provide an accurate translation, and since signs are static, we can focus on hand posture and placements. Deep learning can provide these manipulations by extracting spatial and temporal features, and to further this research multiple forms of image classification will be applied [22].Krizhevsky developed the first successful image classification model in 2012 (ILSVRC-2012), using a deep learning algorithm. Surprisingly superior to other classifiers, the process itself was smart enough to extract multiple features in a matrix, augment them while preventing overfitting [11]. Applying these concepts to this research will challenge the efficiency of Deep Learning using CNN [13]. Deep Neural Networks can improve the accuracy of existing networks by using another method known as transfer learning, by building on top of existing models we can add new datasets or faster networks to increase our accuracy. Tayyip Ozcan's paper in transfer learning uses CNNs for heuristic optimization [31]. Utilizing alexnet, further enhancing accuracy to 80% accuracy over ASL alphabets. Cote-Allard proposed transfer learning during his research into hand gestures, as it further improved feature extraction techniques [32]. Increasing accuracy to above 80% while factoring in areas of Overfitting. It was found

that MATLAB's vision and deep learning toolbox provided extensive evaluation of current pre-trained networks. Many of those included:

1. AlexNet: 8 layered architecture, trained on over 1000 images from ImageNet database [32].

2. GoogleNet: 22 layered architecture, trained on over 100000 images from ImageNet [32].

3. ConvNet: 3 layered archietecure, trained on over 1000s of images for object detection, most commonly used to predict digits and stop signs [3].

4. CaffeNet: 7 layered architectured DAG netowrk. Trained on over 10000 images of hand written digits for pattern recogntion.

Due to the extensive work done on them. Many networks already trained against thousands of images. However, there are still limitations to deploying these networks for current vision-based devices. In order to optimize these networks, following the research similar to Tayyip to provide multiple ways to retrain networks [31]. With research conducted on optimal gesture recognition algorithm, our methodology can build off of current trained networks and improve the detection of our ASL classes. ConvNet proved to give better results to recognizing hand gesture, face and any object. An example of CaffeNet was applied by multiple researchers to recognize handwritten numbers. With the advantage of CNN not requiring any feature extraction to train the model, it provides our research a greater opportunity to train networks against augmented images. Take Fig 4. Which looks at the network architecture of ConvNet. For this research, our fully connected layer would need to incorporate signed characters and then finally output the correct sentence. The image also provides a good description of Convolutional and Pooling layers and how matrix computation is applied which will be further analysed in our methodology section.

To solve the problem of classification our research will apply transfer learning. This method will allow our machine to learn new tasks based on the previous task.

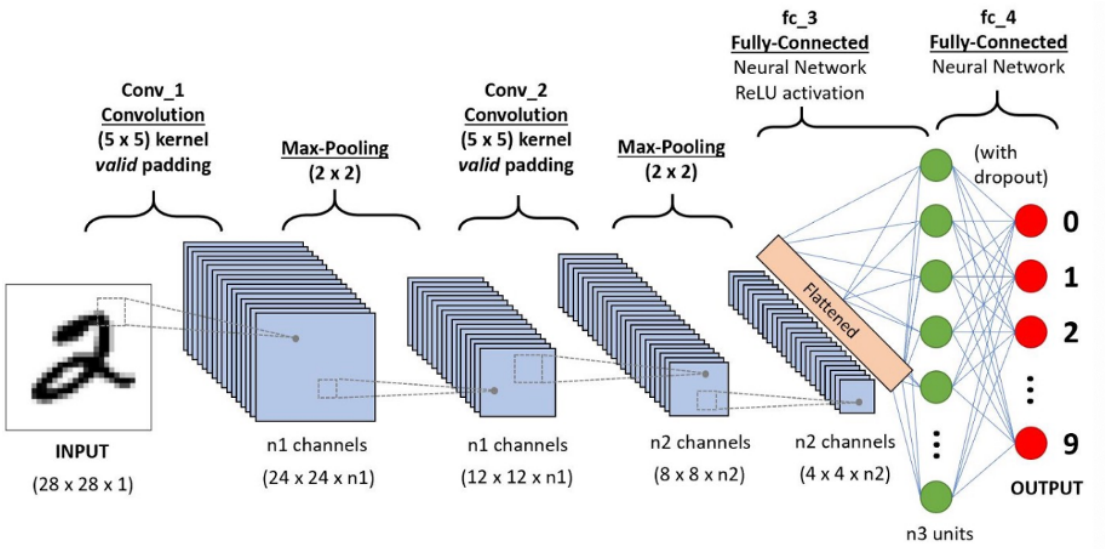

Figure 2-3: ConvNet Sequence to classify handwritten digits [3]

MATLAB available neural networks will be used along with predefined networks from existing projects [17].

Chapter 3

Methodology

3.1 Initial Phase

Originally this work was first tested out using a learning tool called DataCamp [17]. DataCamp is a resource made for students to learn and understand core programming concepts. The ASL project by DataCamp helped in understanding complex neural network architectures using Keras model, and specifically for understanding subtle patterns in streaming video [17]. The project introduces One-Hot Encoding, a coding scheme meant to compare each level of categorical variable [33] One-Hot encoding is an interesting technique used in machine learning, and most common due to the binary observation made by the schema. Figure 5 illustrates a visual example one-hot encoding categorical variables.

Considering that our sign language data set is a categorical data set where each folder consists of an array of static images that is named at the folder level: One-Hot encoding is a suitable fit [17].

The work itself presents an example of having an augmented data set that will be converted to an array for our Keras model, as Keras does not accept just one element [**?**]. This same concept can be applied to our current problem however this solution that was generated relied solely on 3 distinct characters: A, B, C [17]. The work completed gave an insight into what kind of networks to expect that would best suit the dataset.

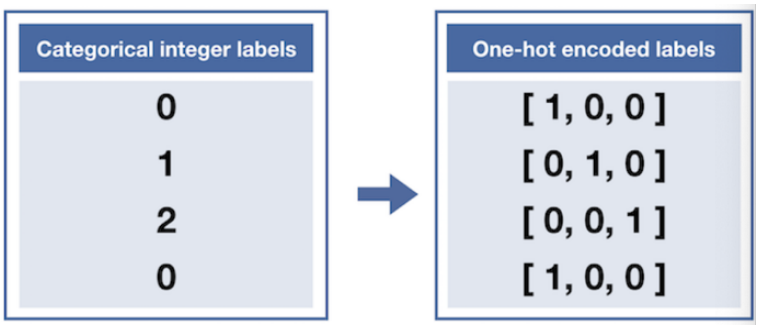

Figure 3-1: Conversion of a one-dimensional array of labels into a two-dimensional array.

Since signs are sequenced, we can focus on hand posture and placements. To understand the basics of our process it can be better broken down into phases:

1. Feature Extraction & Data Processing Phase
2. Training & Testing Phase
3. Classification Phase

3.2 Process

ASL characters can be detected in multiple forms, this research benchmark's three significant detection and extraction algorithms. All three which were implemented in MATLAB.

1. RGB to Gray-scale conversion,
2. Gray-scale to Binary image data,
3. Optical Flow Data from RGB Video Data[34],
4. I3d Network to extract spatio-temporal features from Video Data [35],

The reason behind these extraction algorithms is to improve the accuracy of our models. Many current detection systems rely solely on static images [9]. Whereas our area of research will focus on a sequence of images, similar to many current activity recognition algorithms [35].

The Feature Extraction and Data Processing phase will augment our video data sets to images. The Training & Testing phase of our program will involve training our pre-trained 'network' against a split data set for validation. A network in MATLAB is referred to as a Convolution Neural Network [36]. Deep Learning tools in MATLAB allow users to customize existing or new networks. A detailed description of the networks and their functions are described in Section III. The most popular networks included but were not limited to:

- MobileNet for low-powered devices by Google Researchers [3].
- Sign-Fi CNN, used with WiFi Connectivity [15].
- Pre-trained Matlab networks: including AlexNet, googleNet, etc [36].

After the completion of the first two phase, our program can be tested in real time using computer vision. Converting the video to images and finally re-scales the images. The purpose is to fit the data taken in real time to our network. Once the input size matches our network input, the next phase can take place. The Classification Phase predicts the class of our validation and live data [22]. Segmentation and Feature Extraction are similar due to the process involved with Data Manipulation. However, segmentation applies techniques like region of interest, color/pixel segmentation, and re scaling properties [22]. The Feature Extraction Phase focuses on an array of images and produces multiple data sets against the original. Augmenting the brightness and noise and changing reflection of images finally to create an array of original images and augmented images [22, 28]. The purpose of this wide array of images is to give our network a greater training set on various conditions, this way our classification accuracy is higher on distorted data. In a regular setting the AI should be able to recognize hand placements under different reflections and contrast. Current applications only provide monolithic features that translate alphabets [10], this study will go beyond that.

3.3 Data Set

The data set first analyzed was from Microsoft's labelled MS-ASL data set [37]. This data set covers over 200 signers with a large labelled sign count of 1000 words [37]. Microsoft has also published a pre-trained model for us to experiment with in our evaluation. Provided by The British Machine Vision Conference in 2019 [37]. However, the data set contained YouTube video links for each sign instead of raw multimedia. In order to fully process the data set: we have to look into YouTube downloaders, specifically Pytube: a python script that downloads public links [37]. Although, MS-ASL data set has been integrated into Kinect camera's using Pose Detection there are still limitations towards its full-scale use [37]. The data set and machine learning involved limits to characters alone and although the data set was vast but come a year later and come a larger data set was found. It is good to note that the deaf community actively uses public video sharing platforms for communication and study of ASL. An Australian Centre for Robotic Vision was able to propose a data set comprising of about 2000 signed words called WLASL [4].

3.3.1 WLASL

Also known as World-Level American Sign Language video data set. The largest publicly known data set available early 2020. Using a similar Pytube approach, we were able to download the data set and were given much assistance from the public GitHub of this project [4]. The signed characters were translated to proper mp4 format. Using Pycharm, an IDE for python, we were able to process the data set further. With the diversity of the data we found they were organized in a JSON format as seen in Fig 7.

Utilizing PyCharm the data was extracted in order for us to fully use in our processing stage in MATLAB. Each video in our dataset ranged between 60-100 frames. We were successfully able to extract 1957 videos through the online link available in the dataset json file. each video pointing to one word or character. The videos comprised of over 119 individuals, and continues to grow to this day [4].

```
"gloss": "book",
"instances": [
    {
        "bbox": [
            385,
            37,
            885,
            720
        ],
        "fps": 25,
        "frame_end": -1,
        "frame_start": 1,
        "instance_id": 0,
        "signer_id": 118,
        "source": "aslbrick",
        "split": "train",
        "url": "http://aslbricks.org/New/ASL-Videos/book.mp4",
        "variation_id": 0,
        "video_id": "69241"
    },
```


Figure 3-2: Screenshot of the WLASL v03 json which holds information and url to the original data set

[?]

3.4 Data Processing

MATLAB offers many tools for processing video data, image data, binary data, and many more [38]. For our procedure we needed to organize the WLASL data set in order to utilize them for training our neural networks. So the first step required the data set to be organized and renamed based off the 'WLASL_v0.3.json' file. Using MATLAB vector manipulation on 'class names' and 'bbox', they were renamed and finally processed to images [38].

Along with the bounding box data, we were further able to process and augment our data set. MATLAB tools for augmentation of Image Data-stores was suitable for our project, allowing us to utilize these feature extractors:

1. Skin Segmentation
2. Color Augmentation

Fig 3-3 and Fig 3-4 provide those visual aids from our program, all which are also available under the student GitHub. The data set was more focused around the hands and face of our user, noting that the position of the body would not matter. The reason being is due to the location of hands and head, for now we focus on a

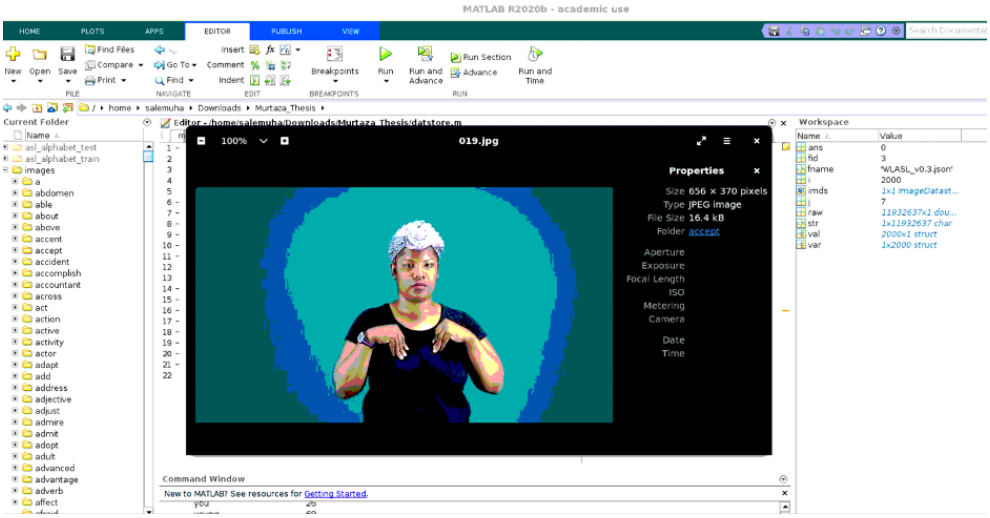

Figure 3-3: MATLAB work space showing the augmented images in their respective folders with an example of a static image from the WLASL data set.

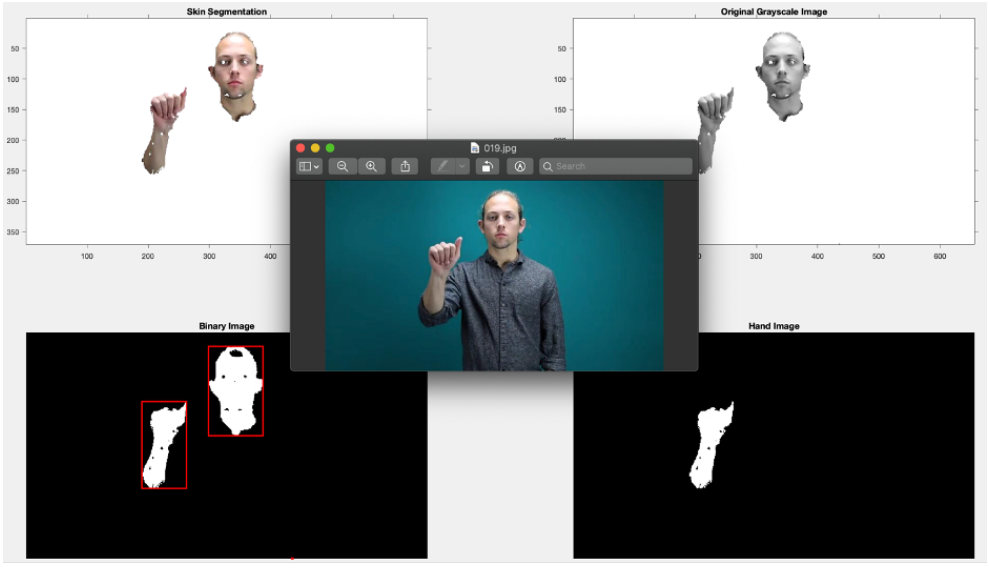

Figure 3-4: MATLAB Processing on static image of sign 'a'. Includes skin detection phase, using color tracking and finally hand detection.

straight forward view of our user. Thus our dataset and test input must be similar in terms of background conditions. With our test on pattern recognition, we will find out more if the hands and face alone have an impact in our recognition.

With our data ready and labeled for our neural networks. We would need to choose the most appropriate network. MATLAB offers many pre-trained networks. Many of which would assist in our research for fine-tuned hand detection.

3.5 Convolutional Neural Networks

Using neural networks, we can isolate regions of interest to better distinguish between characters and gestures. Most pre-trained networks by MATLAB perform vision tasks which require a convolutional layer, which are highly effective in image classification [3]. Using a convolutional layer, we can extract features frame by frame of our image/video data set. Our feature selection will require a temporal convolutional network to encompass spatial and local features. Our CNN network will also comprise of a max pooling layer, the reason is to down-sample our inputs, this will reduce the dimensionality of our input features allowing assumptions within sub-regions [37, 4]. In mathematics the idea is to take the filtered input from the Convolutional layer, which normally decodes everything in matrices. These matrices need to be reduced in scale for our network and thus the pooling layer operates on these matrices, the operations performed are normally Max or Average, thus dubbed max-pooling or just pooling. Fortunately, as a student many of these pre-trained networks are readily available on MATLAB's student license. Such networks include:

1. AlexNet
2. Google-Net
3. Caffe-Net

Many of these networks are set to accept images of certain size and variance, and thus our image augmentation in our data processing stage is crucial in order to kick

start our training. To note however, the current solution only utilizes color augmentation. Rotation and obfuscation was not included in our dataset purely because our research wasn't ready to prove the effectiveness of augmented video dataset.

MATLAB's pre-trained networks are trained on over 1000s of images [41]. Many current networks work well with object detection, and in our case, we want to utilize a coherent hand detection module. Using other models like of ResNet15, ResNet50, Vgg16, etc, all which are available on MATLAB, have very few features for classification which would seem problematic for our problem of detecting about 2000 different features....That would fit well with our input. AlexNet and GoogleNet being one of the most popular neural networks, pre-trained on more than a 1000 features from the ImageNet database. Comprised of 8 layers with the input layer accepting an image of size 227-by-227 [38]. Googlenet can classify over 1000 object categories, and proved useful for our research, as we were able to train these networks by applying a unique aspect of deep learning.

3.6 Deep Learning

Loading these networks on to our MATLAB workspace was feasible. MATLAB comes with built-in applications in order to test and edit neural networks, specifically the classification learner app and the deep network designer [38].

Using these tools and the research provided it was crucial that this network be retrained on to our augmented dataset. Thus, comes the transfer learning phase of our research. In order to prove the effectiveness of these pretrained CNNs, the layers associated to alexnet were modified to fit the new classes. For both DAG and CNN network programs, the layers replaced were the last three layers represented in Fig 3-5. Where we will have a fullyConnectedLayer that provides our network the number of classes and the learning factor, a new softmaxLayer and finally a new classificationLayer [41]. Fugure 3-6 displays the MATLAB code that replaces these layers.

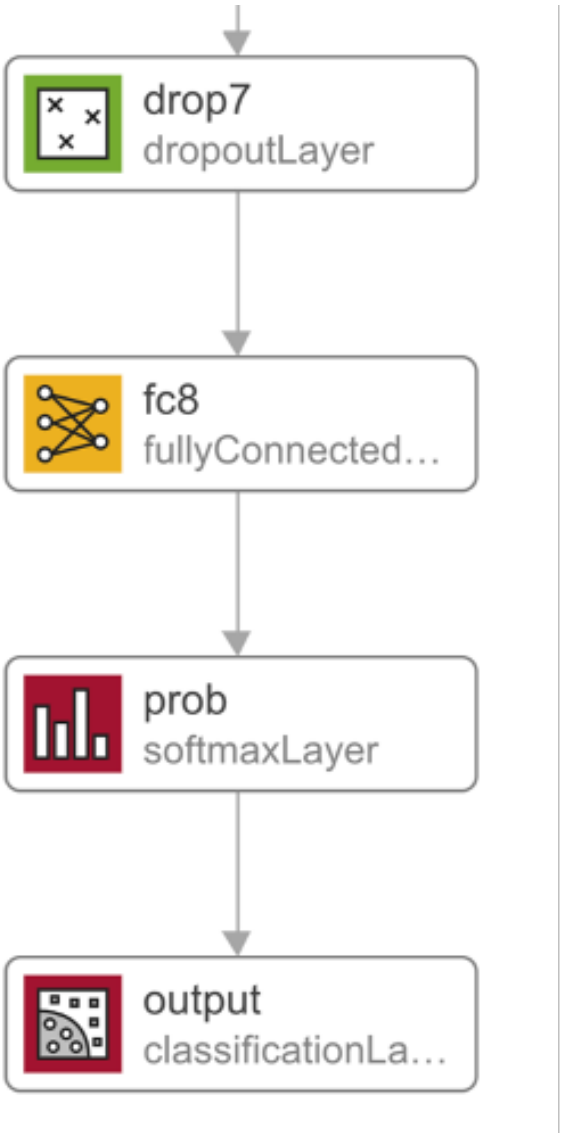

Figure 3-5: GoogleNet Architecture: last 4 layers using "Deep Network Designer" by MATLAB. Back box are the layers altered for transfer learning.

With our network ready and our dataset ready and split 80/20. The training process was able to begin. Training began on our nvidia DGX stations. Utilizing the 4 gpu's, our training environment was set to 'multi-gpu' and 'auto' for CPU testing purposes. Although the process was long due to the vast dataset. With the Finally, we were able to validate our network by testing the prediction on our validation dataset which would be 20% of our original dataset. The next section provides insights to our findings.

Chapter 4

Findings

With an extensive dataset, comprised of over 1700 words/gestures, a wide array of features was to be detected by our neural networks. Our program retrained three networks used on this new preprocessed dataset. The dataset itself was feasible to extract using Pytube. With the original dataset comprised of links and labels. Using *VideoReader* & *imwrite* an augmented dataset was formed for our transfer learning process. Our research provided two different *working* models, one trained for live video and the other trained for static images. Each image was analyzed by frame-sequencing and changes within our frame were noted in our sequence length, described in Fig 4-1.

Creating a range for our sequence between 30-130. These sequences were then extracted to individual frames and were augmented using 'augmentData.m', a method utlitizing skin segmentaion and color tracking algorithms. Fig 4-2 provides a visual to the framing of our video files. The extraction and detection process of hands and skin is described in . For most of the models we were training with we had to play around with the input of the datastore. CaffeNet being the one network that required grayscale images, such that they were two dimensional vectors.

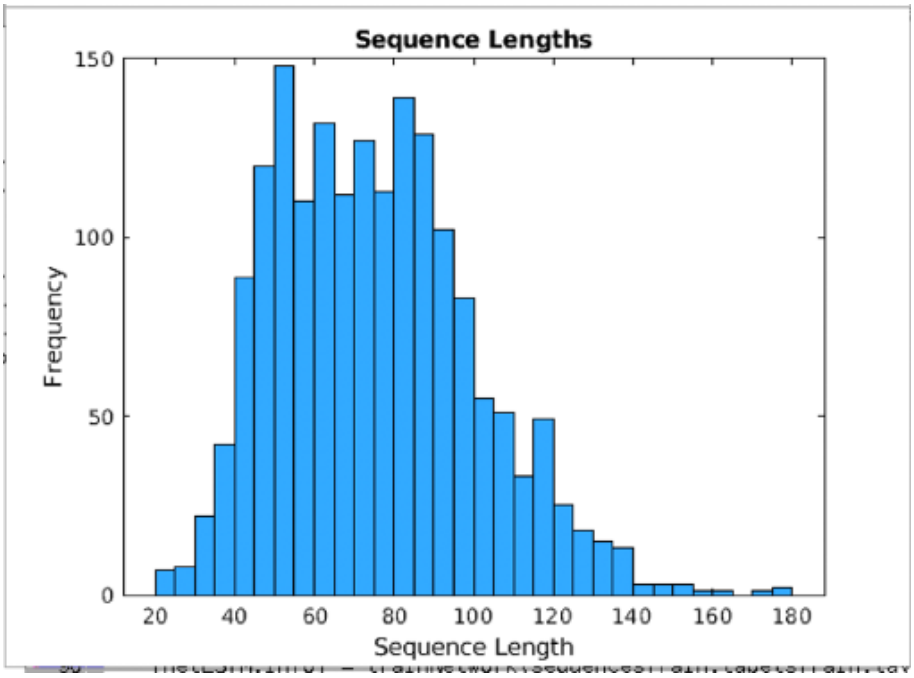

Figure 4-1: Sequence length of all the videos with our data set.

Figure 4-2: ASL signs "read" top row and "dance" bottom row. notice how they differ in orientation of the hands [4]

4.0.1 Alexnet

AlexNet being network with the least amount of layers, it was however the network that took the most amount of time. This came into question on how transferable are the features from our ASL dataset to Alexnet. Figure 12 provides the screenshot to the lagged attempt at training by alexnet on our Nvidia GPU. Our terminal at that point remained idle for over 3 days. Noting heavy over fitting as was expected from intial research in our literature review.

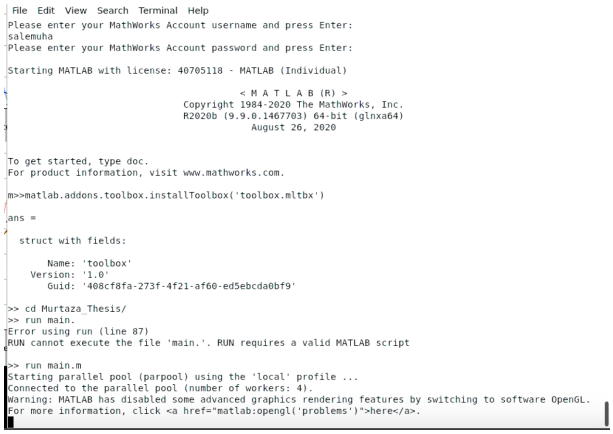

Figure 4-3: MATLAB Cloud Computing with Nvidia GPU using MATLAB Command Terminal

In order to confirm that the training was off we were able to utlize MATLAB's visualization system to pinpoint what was happening during training. Fig 13 proves the failed attempt at training alexnet to our ASL dataset.

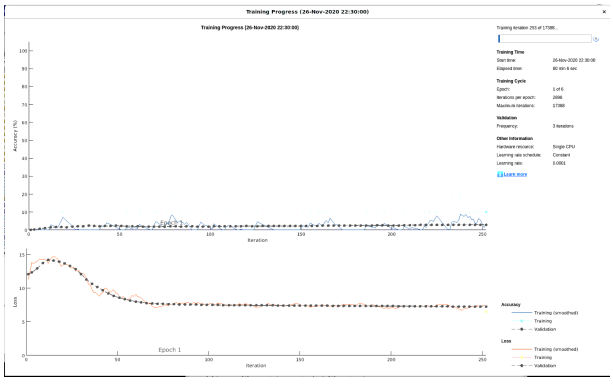

Figure 4-4: Training on single CPU using MATLAB software for visualization

In figure

With initial test it was noted that AlexNet would not perform well with pattern recongition. Thus, the two pretrained networks loaded on to our MATLAB system were:

1. Google-Net
2. Caffe-Net

4.0.2 Google-Net

The best of the three architectures. Performed with similar training parameters as alexnet. Proved to provide results greater than ¿80% and was much faster at outputting a model, a total of 25 minutes was taken for training, in comparison to the eternity from alexnet. While running the program on our DGX Nvidia Container. The processing behind googlenet's architecture was more flexible in terms of manipulating the input of our data set. Even testing one-hot encoding proved fruitful for our research as the categorical labeling was acceptable by its training parameters. While Alexnet would return an exception. Once training was complete our model tested on a test data set that held 20% of the original WL-ASL. Our results when applying tranfer learning to googlenet's architecture proved useful indeed as the model resulted an accuracy of 0.8061 on our test set. Fig 4.5 displays the prediction of our googlenet mode. Fig 4.6 provides the confusion matrix on our test data set vs the true labels.

Figure 4-5: Matlab Terminal computation for accuracy = mean(PredY == ValidationY). accuracy = 0.8061

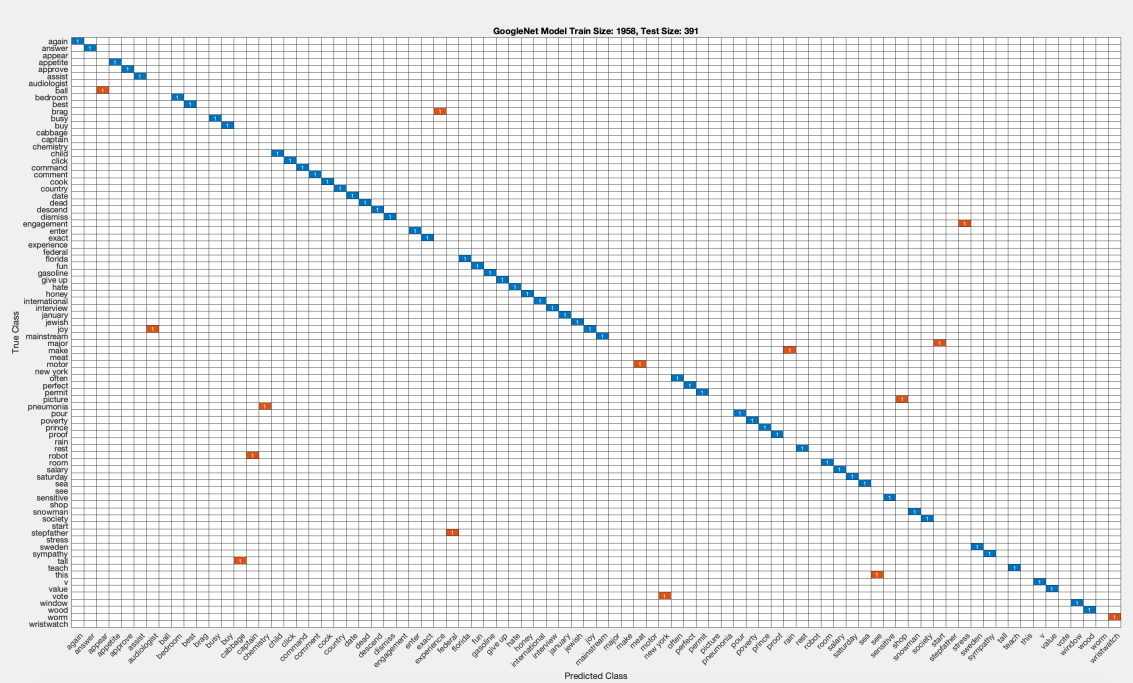

Figure 4-6: Confusion matrix of our test set predicted by our googlenet model. **Note: Confusion Matrix is reduced to fit page, size: 200**

Similarities between googlenet's and caffenet's architecture are due to them being DAG Networks. As using directed graphs isolate patterns for recognition in image sequences, and that is why our training with Caffe Network and GoogleNet was much faster with the amount of features we had in comparison to the CNN alexnet.

4.0.3 Caffe-Net

This particular network, although a DAG network, required our images to be gray scale [nxn] cell array. Caffe-Net was optimal at evaluating deep architectures. Although in this method the input of our data was preprocessed using skin segmentation and color augmentation, Fig 4.5 & Fig 4.6 provides a preview of the binary datastore.

The new datastore of images were then sequenced to indivudal frames resulting in a total 40,000 augmented hand detected images. The network trained on this large data was shuffled every epoch and was deployed on our GPU Nvidia tesla cores. However, the lack of clear validated dataset caused under fitting for our model. Thus resulting in the lowest accuracy found at 40%, which fails the model.

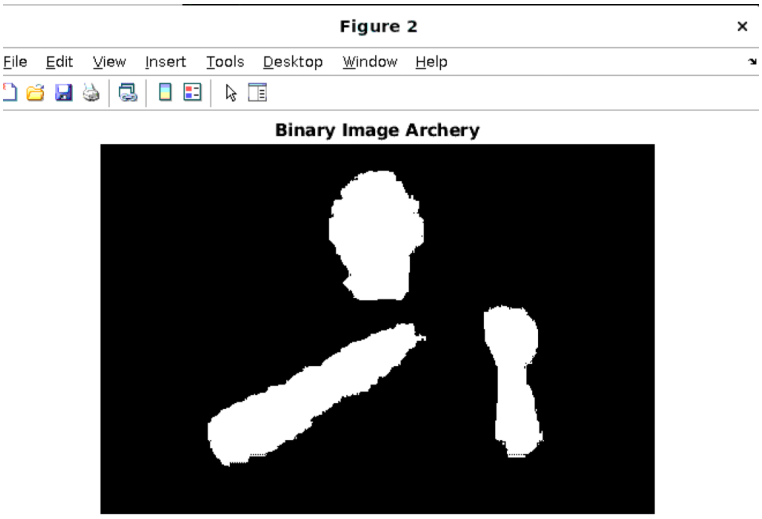

Figure 4-7: preview of binary image 'Archery' Result of 'augmentData.m'.

Figure 4-8: preview of binary image 'Aware'. Result of 'augmentData.m'.

```
Warning: Directory already exists.
> In main_3 (line 37)
Training data size: 3904
Validation data size: 586

accuracyEval =

    0.4022

>> main_3

tempFile =

    'sequencedVideo.mat'
```

Figure 4-9: Testing Caffe-Net Model on full binary image data store

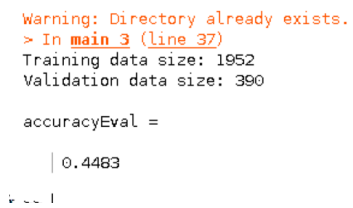

Figure 4-10: Testing our caffe net model on a reduced binary image datastore

Thus with our findings we can safely conclude that the fastest and ideal neural network is GoogleNet. Not only was it able to train with precision but it was also more rigorous at pattern recognition than Alexnet and Caffe-Net. Googlenet's architecture works well with RGB frames and with Gray-scale. Making it much more convenient and flexible for testing a larger data set. With all the research conducted, a fair generalization can be made on our GoogleNet model: That it is the best standard for pattern recognition in a sequence of frames. Below represents the results from all three of our models. Unfortunately alexnet had failed throughout every test Given, most of the issue surrounding alexnet's training was the GPU. As using MATLAB terminal for GPU cloud deployment limits the toolbox associated with MATLAB add-ons.

Table 1: Classification Table

Network	Skin Dataset	Binday Dataset	Hand Dataset	RdFcn(video)
GoogleNet	0.806	0.4	0.4	0.8061
CaffeNet	0.3	0.4	0.45	Invalid
AlexNet	TimedOut	TimedOut	TimedOut	TimedOut

Chapter 5

Conclusion

5.1 Summary

Understanding image processing and video processing is crucial to working with computer vision. The researched studied a novel data set released June 2020 of this year. With this data set we were able to train, test and prove the rational behind transfer learning. Although our results for each network varied greatly, the research behind these networks gives a clear understanding in how to assess machine learning applications. Each network is unique in its design and thus can apply varying filters that best accommodate the dataset. Of the three networks, googlenet being a directed graph network had the most to gain from this research. Googlenets architecture is very useful in pattern recognition due to the spatio-temporal features that googlenet uniquely extracts. Having many layers of abstraction also provides a greater level of variance amongst the dataset [32]. With our results we can conclude the following from our research:

1. Static Images and Sequence Images must be treated as separate data stores. Although a video can be an image and vice-versa, training to find patterns amongst these images require proper 2d or 3d convolutions.

2. Data Augmentation plays a huge role in Under-Fitting or Over-Fitting, The best type of data is short concise but consistent data,

3. Having too many features can slow down the performance of this classifiers,

4. Our model can be implemented in real-time and deployed using MATLAB cloud,

5. The classifier implemented is a novel contribution. No other model exist with an 80% accuracy on a dataset of 2000 ASL gestures and characters.

Thus going back to our original goal of our thesis: which was to analyze gesture recognition and present an algorithmic model that would output an accuracy of 80% and greater. It can be said that our research hit the passing line in terms of generating a well trained model. Although there are definitely more areas to improve.

5.1.1 Advantages and disadvantages of our pre-trained networks.

Advantages:

- GoogleNet: Faster response and can handle much larger classes. Googlenet also has the capability of analyzing video frames as an input through sequencing.

- CaffeNet: Although more faster than googlenet.

- AlexNet: Ideal for few features but many files.

Disadvantages:

- GoogleNet: Varying input size and network architecture fits for video based analysis.

- CaffeNet: This network experienced the most notable loss. Training on a 2 dimnesional image array seems to provide no clear indication of pattern recogntion

- AlexNet: Requires high computational power, generally runs on a powerful computer.

5.2 Future Work

In future implementations of the gesture recognition system, many improvements could be made on the systems presented in this paper to aid in increasing the accuracy of ASL classification. Balancing our dataset would prove useful for future training on neural networks, Originally this paper tried to segment frames of interest using our detection algorithm, isolating key hand points, but one can argue that more data of greater variability would prove beneficial for boosting accuracy. Our model, through googlenet, can be implemented in a straight camera view setting. We believe that with a greater dataset, consisting of rotated videos, will provide greater accuracy for detection in various environment. Looking at Fig 5.1 our result was more accurate in test that matches the same variability our dataset had. Thus proving that with an augmented video data set we can further improve the training and detection process of this pattern recognition solution.

[width=12cm]taz/Taz/Chapters/Figures/test.png

Figure 5-1: Fig 5.1 Testing on live recorded 2 sec video. Signing 'skate' on right with classified label left

It would also be more advantageous if there were more datasets per feature. Using just one video per word can limit our network from learning the so called 'nuances' of sign. So far our training dataset is just comprised of a video per word. With a greater number of videos or files per class, we would definitely improve our training process. In fact, if we have fewer features and more data files, alexnet would've outperformed both networks [32]. Unfortunately one of the goals that this research was not able to complete was to further analzye and improve the accuracy of the googlenet models. Unfortunately due to constraints and issues surrounding the pandemic, many of the next steps were put on hold.

In the spirit of furthering science and this work, the source code for the classifier, models, and data sets will be openly available on the author's and/or journal's website. We hope this will encourage other researchers to extend and explore our work and to test and compare our classifier with other ASL gesture algorithms.

Bibliography

[1] Ulysse Cote-Allard, Cheikh Latyr Fall, Alexandre Campeau-Lecours, Clément Gosselin, François Laviolette, and Benoit Gosselin. Transfer learning for semg hand gestures recognition using convolutional neural networks. pages 1663–1668, 2017.

[2] Erizka Banuwati Candrasari, Ledya Novamizanti, and Suci Aulia. An efficient algorithm for sign language recognition. *Telkomnika*, 18(5), 2020.

[3] Khalil Bousbai and Mostefa Merah. A comparative study of hand gestures recognition based on mobilenetv2 and convnet models. In *2019 6th International Conference on Image and Signal Processing and their Applications (ISPA)*, pages 1–6. IEEE, 2019.

[4] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. pages 1459–1469, 2020.

[5] Suzanne Nichol. Exploring factors influencing canadian families' decisions of communication mode for children who are deaf/hard of hearing. 2020.

[6] Quick statistics about hearing.

[7] Amirita Dewani, Sania Bhatti, Mohsin Ali Memon, Wajiha Arain Arif, Quratulain Arain, and Sayyid Batool Zehra. Sign language e-learning system for hearing-impaired community of pakistan. *International Journal of Information Technology*, 10(2):225–232, 2018.

[8] A. Castillo. Deaf frustrated by scarcity of sign language interpreters at news briefings. pages 114–117, 2020.

[9] C. Vogler and D. Metaxas. Adapting hidden markov models for asl recognition by using three-dimensional computer vision methods. 1:156–161 vol.1, 1997.

[10] P. Sridevi, T. Islam, U. Debnath, N. A. Nazia, R. Chakraborty, and C. Shahnaz. Sign language recognition for speech and hearing impaired by image processing in matlab. pages 1–4, 2018.

[11] Tom Mitchell. Introduction to machine learning. *Machine Learning*, 7:2–5, 1997.

[12] Kohsheen Tiku, Jayshree Maloo, Aishwarya Ramesh, and R Indra. Real-time conversion of sign language to text and speech. pages 346–351, 2020.

[13] Md Asif Jalal, Ruilong Chen, Roger K Moore, and Lyudmila Mihaylova. American sign language posture understanding with deep neural networks. pages 573–579, 2018.

[14] S. Kim, Y. Ji, and K. Lee. An effective sign language learning with object detection based roi segmentation. pages 330–333, 2018.

[15] Hasmath Farhana Thariq Ahmed, Hafisoh Ahmad, Swee King Phang, Chockalingam Aravind Vaithilingam, Houda Harkat, and Kulasekharan Narasingamurthi. Sign language gesture recognition with bispectrum features using svm. 2233(1):030001, 2020.

[16] Matlab optimization toolbox. ¡The year of your version, you can find it out using ver¿.

[17] American sign language recognition of characters using datacamp. *DataCamp Projects*, 2020.

[18] S. S Kumar, T. Wangyal, V. Saboo, and R. Srinath. Time series neural networks for real time sign language translation. pages 243–248, 2018.

[19] Hamzah Luqman, El-Sayed M El-Alfy, and Galal M BinMakhashen. Joint space representation and recognition of sign language fingerspelling using gabor filter and convolutional neural network. *Multimedia Tools and Applications*, pages 1–22, 2020.

[20] Celso M de Melo, Brandon Rothrock, Prudhvi Gurram, Oytun Ulutan, and BS Manjunath. Vision-based gesture recognition in human-robot teams using synthetic data.

[21] Vinay Jain Pratibha Pandey. Hand gesture recognition using discrete wavelet transform and hidden markov models. 18(5), 2019.

[22] S. Shahriar, A. Siddiquee, T. Islam, A. Ghosh, R. Chakraborty, A. I. Khan, C. Shahnaz, and S. A. Fattah. Real-time american sign language recognition using skin segmentation and image category classification with convolutional neural network and deep learning. pages 1168–1171, 2018.

[23] Setiawardhana, R. Y. Hakkun, and A. Baharuddin. Sign language learning based on android for deaf and speech impaired people. pages 114–117, 2015.

[24] Rasha Amer Kadhim and Muntadher Khamees. A real-time american sign language recognition system using convolutional neural network for real datasets. *TEM Journal*, 9(3):937, 2020.

[25] RS Sabeenian, S Sai Bharathwaj, Tamil Salem, and M Mohamed Aadhil. Sign language recognition using deep learning and computer vision.

[26] Shaon Bandyopadhyay. A study on indian sign language recognition using deep learning approach.

[27] Rasha Amer Kadhim and Muntadher Khamees. A real-time american sign language recognition system using convolutional neural network for real datasets. *TEM Journal*, 9(3):937, 2020.

[28] Harsha Vardhan Guda, Srivenkat Guntur, Kunal Gupta, Priyanka Volam, Sudeep PV, et al. Hardware implementation of sign language to text converter using deep neural networks. 2020.

[29] Dec 2020.

[30] Md Asif Jalal, Ruilong Chen, Roger K Moore, and Lyudmila Mihaylova. American sign language posture understanding with deep neural networks. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 573–579. IEEE, 2018.

[31] Tayyip Ozcan and Alper Basturk. Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition. *Neural Computing and Applications*, 31(12):8955–8970, 2019.

[32] Yang Yang, Lin-Feng Yan, Xin Zhang, Yu Han, Hai-Yan Nan, Yu-Chuan Hu, Bo Hu, Song-Lin Yan, Jin Zhang, Dong-Liang Cheng, et al. Glioma grading on conventional mr images: a deep learning study with transfer learning. *Frontiers in neuroscience*, 12:804, 2018.

[33] Kedar Potdar, Taher S Pardawala, and Chinmay D Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International journal of computer applications*, 175(4):7–9, 2017.

[34] Jordan J Bird, Anikó Ekárt, and Diego R Faria. British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language. *Sensors*, 20(18):5151, 2020.

[35] Amin Ullah, Khan Muhammad, Javier Del Ser, Sung Wook Baik, and Victor Hugo C de Albuquerque. Activity recognition using temporal optical flow convolutional features and multilayer lstm. *IEEE Transactions on Industrial Electronics*, 66(12):9692–9702, 2018.

[36] Fu-Lian Yin, Xing-Yi Pan, Xiao-Wei Liu, and Hui-Xin Liu. Deep neural network language model research and application overview. pages 55–60, 2015.

[37] Hamid Reza Vaezi Joze and Oscar Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv preprint arXiv:1812.01053*, 2018.

[38] MATLAB team. Mathworks documentation for deep learning and image processing.