National College of Ireland

# Hand Gesture Recognition and Classification using Computer Vision and Deep Learning Techniques

MSc Research Project
Data Analytics


Bharti Mehatari
Student ID: X20175825

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Bharti Mehatari |
| **Student ID:** | X20175825 |
| **Programme:** | MSc in Data Analytics |
| **Year:** | 2021-22 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Dr. Catherine Mulwa |
| **Submission Due Date:** | 31/01/2022 |
| **Project Title:** | Hand Gesture Recognition and Classification using Computer Vision and Deep Learning Techniques |
| **Word Count:** | 8167 |
| **Page Count:** | 24 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | Bharti Mehatari |
| **Date:** | 30th January 2022 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | ☐ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | ☐ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | ☐ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Hand Gesture Recognition and Classification using Computer Vision and Deep Learning Techniques

Bharti Mehatari
X20175825

**Abstract**

Hand gesture recognition systems have gained a huge momentum in the field of computer vision over the last few decades having its application in various fields. Considering the recent pandemic situation, the need for touch-less devices have become key concern for people's hygiene and safety. Also, the idea of devices that function via hand gestures have also attracted people to adopt such technology thus creating a demand of gesture recognition systems in the market. Recent research has shown the dominance of deep learning algorithms on feature analysis and image classification. Therefore, these techniques have been developed in this research project, to create models that can detect and identify gestures used on a daily basis. The dataset consists of ten different individuals performing various gestures. The implementation also demonstrated several image processing operations such as smoothing of images, greyscale image conversion, image thresholding as well as data augmentation. The hyperparameter optimised 2D convolution model outperformed other implemented models that provided highest accuracy of 99.88% and only 7 cases misclassified out of 20000 images. The confusion matrix and learning curve were also considered for the performance evaluation.

## 1 Introduction

Due to the ongoing technological advancement, human-computer interaction has gained quite the attention with systems like face detection and recognition, gesture recognition, posture recognition, etc. Deep learning algorithms have paved the way for various applications for gaming, virtual reality environments, driver-less cars, etc. which are using gesture recognition technology. Taking into account of the life post pandemic, it seems necessary to replace the usage of touchscreen devices from public places with gesture recognition systems to curb the problem concerning people's safety and hygiene. The concept of gesture detection and classification aims to interpret human gestures through various deep learning and machine learning algorithms. It is a pivotal technology in human computer interaction (HCI) and helps improving systems for sign language recognition and other commercial areas. Two of the most powerful architectures i.e. the convolutional neural network and VGG16 which have shown outstanding performance in Computer Vision for detection, recognition and classification systems (Redmon et al. (2016)) are applied to this research. There are quite a few challenges faced in the recent related works with regards to the lighting variations, the difficulty in detecting movements or combination of various movements, position of the hand and other environmental factors. This report addresses some of these challenges and develops several

deep learning techniques i.e. convolutional neural networks, transfer learning VGG16 network and other models with hidden layers that processes the raw images using data processing algorithms, detects the hand and interprets the gesture from a set of several gesture classes using classification techniques. This project provides an effective solution to enhance the recognition that can be applied to systems in multiple domains.

## 1.1 Project Requirement Specifications

The research question aims at applying deep leaning techniques with the objective of detecting and accurately identifying the hand gestures as per the defined class labels.

**Research Question:** To what extent can detection and classification of hand images using deep learning techniques (convolutional neural networks, VGG16, data augmentation) enhance recognition of hand gestures to support stakeholders in gaming and virtual reality applications and improve sign language recognition?

The research question was solved by implementing project objectives (refer Table 1).

Table 1: Overview of research objectives

| Obj. | Name | Description | Methods |
|---|---|---|---|
| 1 | Literature review | Critically review the literature on image processing and gesture recognition | |
| 2 | Generate dataset | Download hand images data from Kaggle and prepare a balanced dataset | |
| 3 | Data pre-processing | Normalise the input images and perform necessary image processing techniques. Prepare dataset for training neural networks. | Image thresholding, gaussian blur |
| 4 | Implementation of CNN architecture (feature extraction) | Implementation of deep CNN model for extracting features from the images | Evaluation: confusion matrix, accuracy and validation loss |
| 4.1 | Implementation of classifier (classification) | Implementation of classifier for accurately identifying the gesture as per class label | |
| 4.2 | Hyperparameter optimisation | Hyper-tuning the parameters such as no. of epochs, learning rate, batch size, etc | |
| 4.3 | Implementation of VGG16 network | Implementation of pretrained model for comparison with base model. | |
| 5 | Data visualisation | Design and process flow diagrams and evaluation plots in applicable sections. | |

**Contributions:** The contribution of this project involves the implementation of deep learning models for the identification and classification of hand gestures. These models will allow the systems to recognise and classify various gestures performed on a daily basis for the ease of operations. It will also support stakeholders of gaming industry and virtual reality environments as well as enhance sign language gesture recognition.

The rest of the technical report is organised as follows: Chapter 2 presents the thorough review of literature in hand gesture recognition systems involving several criteria. Chapter 3 involves the research methodology and the design specifications consisting of distinct diagrams depicting the entire process. In chapter 4, the implementation and usage of techniques is explained and chapter 5 concludes the evaluation and results.

# 2 Related Work

This section represents a detailed study of research in gesture recognition technology and puts forth a critical literature review of the project topic and its related key areas. The review is divided into sub-sections which cover the following important areas: image processing, various methodologies used in detecting hand gestures and gesture recognition models. (2.1) sub-section involves detailed discussion on image processing. (2.2) sub-section involves literature review of both deep learning and machine learning algorithms used for detecting hand gestures. (2.3) subsection mentions pretrained gesture recognition models and different architectures.

## 2.1 Review of Literature on Image Processing

Image processing is the first and foremost step in any project as transforming the input images so as to be understandable by the machine learning and deep learning algorithms and is a critical task. This method involves various techniques that help to perform operations on the raw images in order to get an enhanced image and extract useful information from them. One of the major challenges is to eliminate noise from the image due to varied background and lighting conditions. Extreme noise cancellation can lead to loss of information which may cause issue in detecting the object as well as identifying the underlying gesture. Therefore, careful image processing techniques must be applied for data pre-processing.

A novel approach has been adopted by Sharma et al. (2020) of identifying hand gestures based on several techniques involved in its implementation. As part of image pre-processing, the RGB image was initially converted to a single-channel greyscale image. Canny edge detection was applied on the transformed grayscale image to generate only the strongest edges in the image. Canny edge detection is a commonly used and effective approach for detecting edges in images which also helps to reduce background noise, allowing for more effective use of other approaches. For extracting features, techniques like Oriented Fast and Rotated Brief and bag of words were deployed. Finally, in order to generate effective results, the processed data was sent through various classifiers like Logistic Regression, Random Forest, Support Vector Machine, k-Nearest Neighbours and Multilayer Perceptron.

Using a boundary histogram, Wysoski et al. (2002) proposed rotation invariant postures. The input images were captured with a camera, a filter for skin colour detection was applied, and then a clustering method was used to locate the boundary for each group in the clustered image using a standard contour-tracking algorithm. Grids were created from the image, and the boundaries were standardised. By splitting the image into number of regions N in a radial form, according to a certain angle, the boundary was represented as chord's size chain later used as histograms. MLP neural networks and Dynamic Programming (DP) Matching were employed in the classification process.

Another experiment of using scaled normalisation was offered by Hasan and Misra (2011) to recognise gestures based on brightness factor matching. The images are segmented using the thresholding approach with a black background. Any segmented image is trimmed and the image's centre mass is identified, so the coordinates are moved to match the centroid of the hand at the X and Y axis origin. Because this method is based on the object's centre mass, the generated images are of varying sizes. To overcome this problem, a scaled normalisation operation is used to maintain image dimensions and time, with each block of the four blocks scaling with a factor that is different from the other blocks' factors. To extract the features, two methods are used: first, edge mages, and second, normalised features, in which only the brightness values of pixels are calculated and black pixels are ignored to shorten the feature vector. The database comprises of six gestures, with ten samples per gesture, five for training and five for testing. The recognition rate for the normalised feature problem outperformed the base approach generating a 95% success rate.

The main focus of Panwar and Mehra (2011) in their experiment was on primary operations viz. picture enhancement and segmentation and orientation detection. In order to accurately determine the hand patterns occurring in the provided image, a lot of work was put into the features analysis by executing several approaches such as centroid, peak or finger region detection, Euclidean distance and presence of thumb. The model gave an accuracy of 92%, with 360 photographs properly identified from 390 images, misclassifying 60 cases. It uses a shape-based approach rather than the typically used skin-color identification approaches since unpredictable light and backdrop conditions cause problems when detecting the hand or any other item. As a result, the shape-based technique benefitted the research in its early pre-processing stages thus making it easier for the model to draw out features and classify.

A lot of efforts were put in image processing for the detection of hand gestures by Chen et al. (2014). Firstly, they have adopted the background subtraction method for detecting the hand and then transformed it into a binary image. Secondly, using the finger segmentation method, the palm and fingers were segmented in order to facilitate the identification process. The palm and fingers were distinguished by mathematically computing the central distance of the detected hand image with the help of inner circle of maximal radius. Finally, the gestures are recognised using a simple rule classifier. This method was helpful in accurately classifying the gestures as per the labels thus achieving and accuracy of 96.6% which also outperformed the state-of-the art approaches. In another approach, the researchers Kumar et al. (2018) have provided a robust model for the invariant Sign Language Recognition framework that detects occluded hand gestures. The project involved the extraction of skeletal data from the Kinect sensor and was tested on a large dataset consisting of about 2700 motions. These motions were recognised and classified with the help of Hidden Markov model and the results were accomplished with 83.77% of accuracy on occluded gestures.

## 2.2 Critical Analysis of the various Deep Learning and Machine Learning Models

Numerous models have been created by researchers for the identification of hand gesture recognition using both machine learning as well as deep learning models. There have also been some hybrid approaches combining both these methods which have shown promising results in the analysis of hand gestures. With conv1D, the conv2D pyramid, and the LSTM block, Do et al. (2020)presented a multi-level feature LSTM. From skeletal data, they extracted skeletal point-cloud features and from the hand component segmentation model, they used depth shape features to come up with a solution. On the Dynamic Hand Gesture Recognition (DHG) dataset consisting of 14 and 28 classes, the approach attained accuracy of 96.07 and 94.40 percent, respectively. The LSTM model with two-pyramid convolutional blocks was used to extract diversity of dynamic hand motions from 14 depth and 28 skeletal data. 18 classes had a 94.40% accuracy rate. Similarly, Abhishek et al. (2020) developed an HCI-based recognition system that employs a variety of detection approaches, including skin-color detection, skeletal structure detection and camera and light effects for picture processing. The computer's ability to recognise behaviours such as scrolling up/down pages and switching pages was improved with the help of 3D CNN model.

Another project developed by Islam et al. (2019) opted for minimal pre-processing by eliminating the background using background subtraction technique in order to minimise the computational complications. This technique is based on the k-gaussian distribution that selects appropriate gaussian distribution for each pixel and provides a better adaptability on varying scenes due to illumination changes. This subtraction technique was first introduced by Zivkovic (2004). After eliminating the background, the images were reduced to only one colour channel by converting it to greyscale images. This transformation helps the network to better learn and understand the input data. Finally, they were resized to a specific dimension before passing it to the convolutional network layer. A comparative analysis of the data augmented model and a base model revealed that the augmented model achieved higher precision and recall measures. This augmentation brought variations in the data thus improving the learning scope of the CNN model.

Apart from the typical neural network techniques, there has been a lot of work done using machine learning algorithms. With the help of RapidMiner tool, Trigueiros et al. (2012) have extracted the hand features into two separate datasets. These excel sheets contained several values for the angle, mean and variance of the greyscale hand image, bin values obtained from histogram orientation and area of the binary hand. A comparison of 4 machine learning algorithms viz. k-NN, naïve bayes, ANN and SVM were employed for classification. The ANN method yielded the best result of 96.99% accuracy in dataset1 and 85.18% in dataset2 respectively. Rahim et al. (2019) investigated the translation of a sign word's gesture into text. The authors used a fusion segmentation method of YCbCr and SkinMask. Support vector machine (SVM) was used as classifier for the recognition of the signs acquiring an accuracy of 97.28% wherein the dataset consisted of 9 sign gestures from double hands and 11 gestures from a single hand. John et al. (2016) offered a vision-based system for intelligent vehicles as they are used in car user interfaces to improve driver's comfort. They addressed the problem of multiple frames causing low accuracy and rising computing complexity by representative frames. Using innovative tiled picture and binary patterns within a semantic segmentation-based framework, the deconvolutional network, the frames were obtained.

## 2.3 Critical Review of Gesture Recognition Models and its Architecture

The research conducted in the very early stages of image analysis and recognition experiments used the Haar wavelets for feature extraction instead of the pixels. Studies proposed by Chen et al. (2007) and Barczak and Dadgostar (2005), uses the AdaBoost algorithm that is apt for adaptively selecting the best features in each step and integrate them into a strong classifier. This helped in increasing the overall accuracy of classification as well as the model's performance. The AdaBoost learning algorithm is used to train a collection of "positive" and "negative" samples differentiating amongst the object of interest. The haar wavelets were used in early face recognition algorithms to create the haar-like characteristic. The researchers performed this experiment by taking a two-level approach. The first level classifies the postures with AdaBoost learning and haar-like features. For the next level of gesture recognition, they implemented context-free grammar to examine the syntactic structure based on the detected postures.

In the past few years, the gesture identification technology has attracted a large audience and shown great development in the AI business with several real-world applications. For the efficient working of such tools and systems, these models need a large amount of learning based on the training data. As a result, different models have been pre-trained using a collection of different class labels and categories, such as VGG19, AlexNet, LeNet, Inception network, Residual network or ResNet50, Xception, and so on. An application demonstrated by Elboushaki et al. (2020) relied on pretrained convolutional residual networks (ResNets) for training extraordinarily deep models and ConvLSTM for dealing with time-series connections to learn high-level representations of gestures. Firstly, the architecture learns the spatial and temporal information from the images and depth sequences simultaneously using 3D ResNets, followed by ConvLSTM layer that captures the temporal correlations between them. Secondly, to reduce noise from backdrop and other variations, these temporal data are encoded into a motion representation, which is then used to extract features using a two-stream architecture based on 2D-ResNets. Finally, various fusion strategies were applied at different levels for the purpose of classification which led to stable and robust recognition system effectively functioned on four extensive datasets.

A comparative experiment conducted by Zabir et al. (2018) for gesture classification makes use of two pretrained models- GoogLeNet and AlexNet along with a few additional layers on top of these models. The effectiveness of all the models in achieving the goal of gesture recognition is critically analysed and evaluated in this research. Because the pretrained models were trained on the benchmark dataset ImageNet Large Scale Visual Recognition Challenge (ILSVRC) containing millions of images with various class labels, they are expected to produce higher results. The Caltech101 dataset was utilised in this experiment, which consisted of around 100+ class labels. The research focussed on measuring the loss, accuracy as well as time consumed by each model throughout the training and classification tasks. The pretrained model was found to have a 99.65% accuracy rate, whereas the customised CNN model had a 91.05% accuracy thus revealing that, GoogLeNet was the best of the bunch because to its constant early output and low loss rate.

Using an extensive dataset of sign language depicting English alphabets, Agrawal et al. (2020) have proposed a hand gesture recognition system. The input data is a combination of images captured from a live-stream video of a webcam and some images

from Kaggle dataset. Their objective was accomplished by applying custom Convolution Neural Network layers on top of a well-known pretrained model called Inception V3. These pretrained models have already gained information on large datasets thus allowing users to retain existing layers as well as construct more custom layers for deep learning of the model. They introduced some kind of variations in the background as well as data augmentation method to eliminate ambiguity and improve the overall accuracy of the model. However, no major change was observed in the results. The comparison of related work done using different kinds of approaches to solve the problem at hand is provided in Table 2.

Table 2: Comparison of Literature in Gesture Recognition.

| **Authors** | **Methods used** | **Dataset** | **Accuracy** |
|---|---|---|---|
| Sharma et al. (2020) | Histogram of Gradients, RF, SVM, MLP, Logistic KNN, Naïve Bayes | American Sign Languange | 96.96% |
| Hasan and Misra (2011) | Peaks and bit sequence | 39 hand gestures | 92% |
| Chen et al. (2014) | FEMD, Rule classifier | 0-9 number gestures and 13 sign gestures | 96.60% |
| Trigueiros et al. (2012) | ANN, k-NN, SVM, Naïve Bayes | Dataset1 and Dataset2 | 96.99% |
| Rahim et al. (2019) | SVM | 9 single hand and 11 double hand sign gestures | 97.28% |
| Zabir et al. (2018) | CNN, GoogLeNet | Caltech101 | 99.65% |
| Agrawal et al. (2020) | CNN, InceptionV3 | Sign Language MNIST | 69.01% |

## 2.4 Identified Gaps and Conclusion

The above literature review gives a brief knowledge about the research performed for image processing as well as various methods involved in the hand gesture recognition domain. From this review, it is learnt that although numerous methods and techniques have been applied for the initial critical phase of data pre-processing, there is insufficient research accomplished in the gesture recognition systems with images in varied lighting and background conditions. Many papers have mentioned having inculcated the variations in background. However, they have not clearly explained such scenarios. Most of the experiments have followed the skin-colour detection and edge detection techniques. There are also many pretrained networks (e.g Inception network, YOLOv3, etc.) available for image analysis that show remarkable performances in gesture recognition but do not account for noisy images. Some of the models tend to increase the computational cost as the number of cameras increases, thus rendering the system inoperable in real-time situations. It is also observed that in some of the experiments, the models only work on a smaller dataset and tend to overfit as the size of the data increases thus deteriorating the accuracy rate.

# 3 Research Methodology and Design Specification

## 3.1 Introduction

The methodology section is the most critical part of the project as it represents an overview as well as detailed approach of the entire project model along with the techniques used. It covers all major aspects such as image analysis, implementation of the deep learning algorithms, summary of the methodology that comprises of feature selection and classification methods; with the goal of evaluating the performance of such models while grouping the hand gesture images.

## 3.2 Hand Gesture Recognition Methodology Approach

The main goal of this research is to perform the task of gesture recognition from images with the help of deep learning techniques applied on a set of ten labelled images dataset. The methodology uses Python language and Google Colaboratory to cater the computational needs of the experiment. The models are constructed using the Keras package. For image pre-processing, the PIL and OpenCV libraries are utilised as they are best suited for unstructured input data of images and videos. Figure 1 depicts the entire workflow of the proposed approach for the detection, identification and classification tasks. The sub-sections that follow provide step-by-step information regarding the workflow strategy.

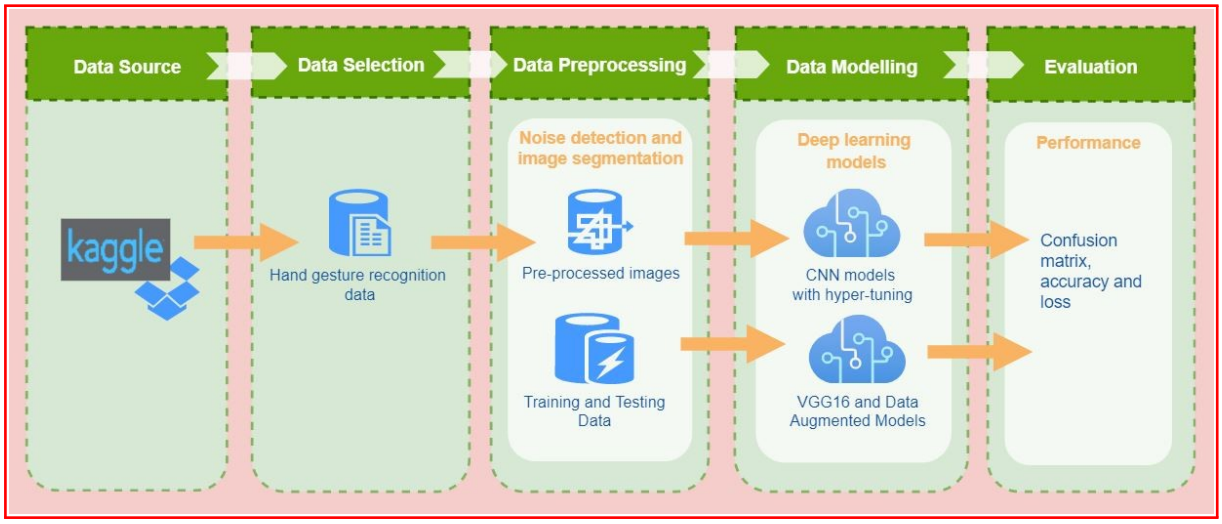


Figure 1: Brief Overview of the Research Methodology

## 3.3 Data Collection of Images

A standard dataset named Hand Gesture Recognition Database[1] (public in nature) is obtained from Kaggle.com and employed to carry out this research. The dataset is originally of about 2 gigabytes and is compressed to 1.62 GB after downloading. There are ten different types of gestures in the dataset viz. fist, palm, thumb, okay, letter L, letter C, fist moved, palm moved, down and index wherein each gesture is performed by

[1] https://www.kaggle.com/gti-upm/leapgestrecog

multiple subjects to maintain variations. The input dimensions of all these images are of 640 × 240 and each class has 2000 samples. Hence the total is 20000 images and no data imbalance within the data files. These hand gestures are depicted in Figure 2.

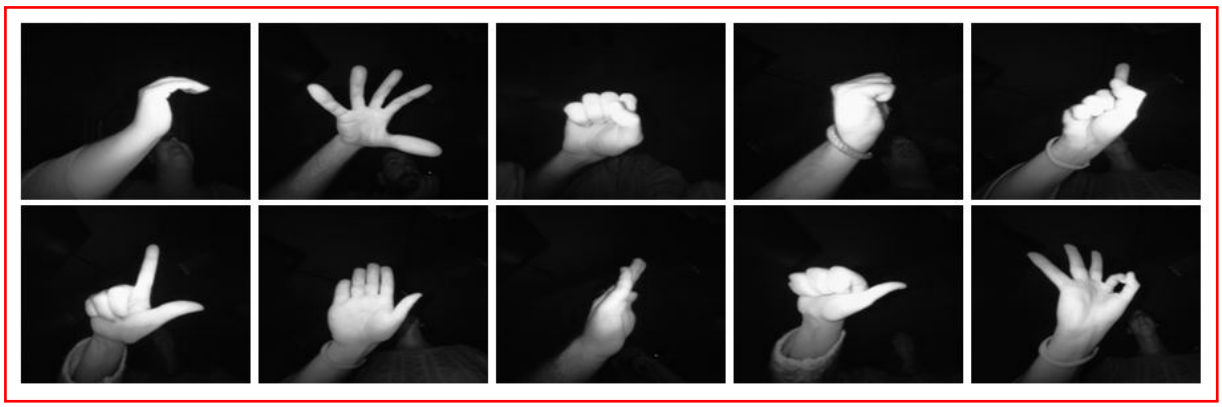

Figure 2: Gesture samples from source data labelled as classes- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

### 3.4 Image Processing

Upon loading the dataset into Google Colab Pro, it was accessed and resized to a particular shape and dimension as accepted by the input layer of the models. All the resized images are then stacked after converting them into arrays. The gesture labels are stored separately that access the image based on the first two characters of the gesture name. The performance of the introduced models improve depending upon the applicable preprocessing techniques as well as the amount of data used for training. Hence, the following steps are carried put as part of the image processing using the OpenCV library which are as follows:

- Conversion of RGB images to single channel grey scale images
- Checking and elimination of noise in the input images
- Smoothing images using the GaussianBlur method that reduces high-frequency noises
- Normalisation of data with the help of image thresholding technique which helps in removing lighter or darker regions and contours of image based on pixel value
- Resizing the images to fit according to the input dimensions of the models

As seen in Figure 3, the image represents the 'okay' gesture and has been applied several image processing techniques as part of image analysis necessary before feeding in to the deep learning model. First, the image is converted into greyscale and flipped using the OpenCV library. Then, unnecessary noise and/or high frequency edges are detected and eliminated using the GaussianBlur technique. This method uses the Gaussian kernel and convolves the image by blurring it thus acting as filter to remove all the impurities. After this, the process of image thresholding takes place which transforms the pixels as black or white by specifiying the range of values as 150 to 255, thus retrieving only the hand object. Finally, the image is resized to a particular dimension of 224 × 224 as accepted by the relative network model.

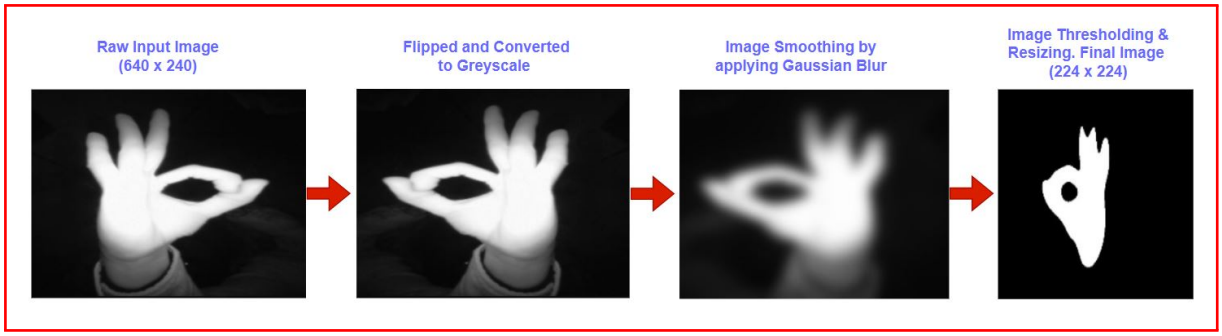


Figure 3: Gesture samples from source data labelled as classes- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

## 3.5 Modelling of the Deep Neural Networks

The traditional machine learning techniques require feature importance and dimensionality reduction techniques to be conducted externally. However, this research project was solved with the help of deep learning techniques such as the convolutional neural network. These networks are capable of gathering information from images while passing through various hidden layers until the classification process in the final layer. 2D CNNs and pretrained model VGG16 are utilised for the purpose of feature mapping and dense layers along with dropout layer are implemented for classifying the multi-class labels.

## 3.6 Evaluation of the Deep Learning Models

The evaluation phase of a data analytical project is highly critical and interprets the entire functioning and results of the developed model. This helps in knowing the performance of these models with the help of evaluation metrics such as rate of accuracy, confusion matrix, several components of the confusion matrix like precision, recall, sensitivity, etc. as well as loss measures. The outcomes of the implemented models are evaluated and reported in chapter 5. For testing the performance of all the models, the confusion matrix as well as plot of validation loss and accuracy are presented. The most widely used evaluation metric in classification type of problems which is the rate of accuracy is also calculated.

## 3.7 Design Specification System Flow

The design flow of the detection and classification of hand gestures using deep learning algorithms is visualised in the form of a three-tier architecture consisting of the Presentation, Business Logic and Data Layer as represented in below Figure 4. The first layer describes the visualisations of input data, pre-processed data as well as the results which are obtained with the help of seaborn and matplotlib libraries. Different libraries such as keras and tensorflow are used in the Business Logic layer for coding. This tier explains the technical aspect of the system implementation and involves the various deep learning models that are utilised. It consists of the CNN model, transfer learning VGG model and data augmentation model. The third layer defines the various processes involved in data processing; right from acquiring the dataset from Kaggle, performing exploratory analysis to understand the nature of data, followed by preliminary data cleaning steps and finally, image transformation so that useful information can be extracted from these images and sent over to the Business Logic Tier.

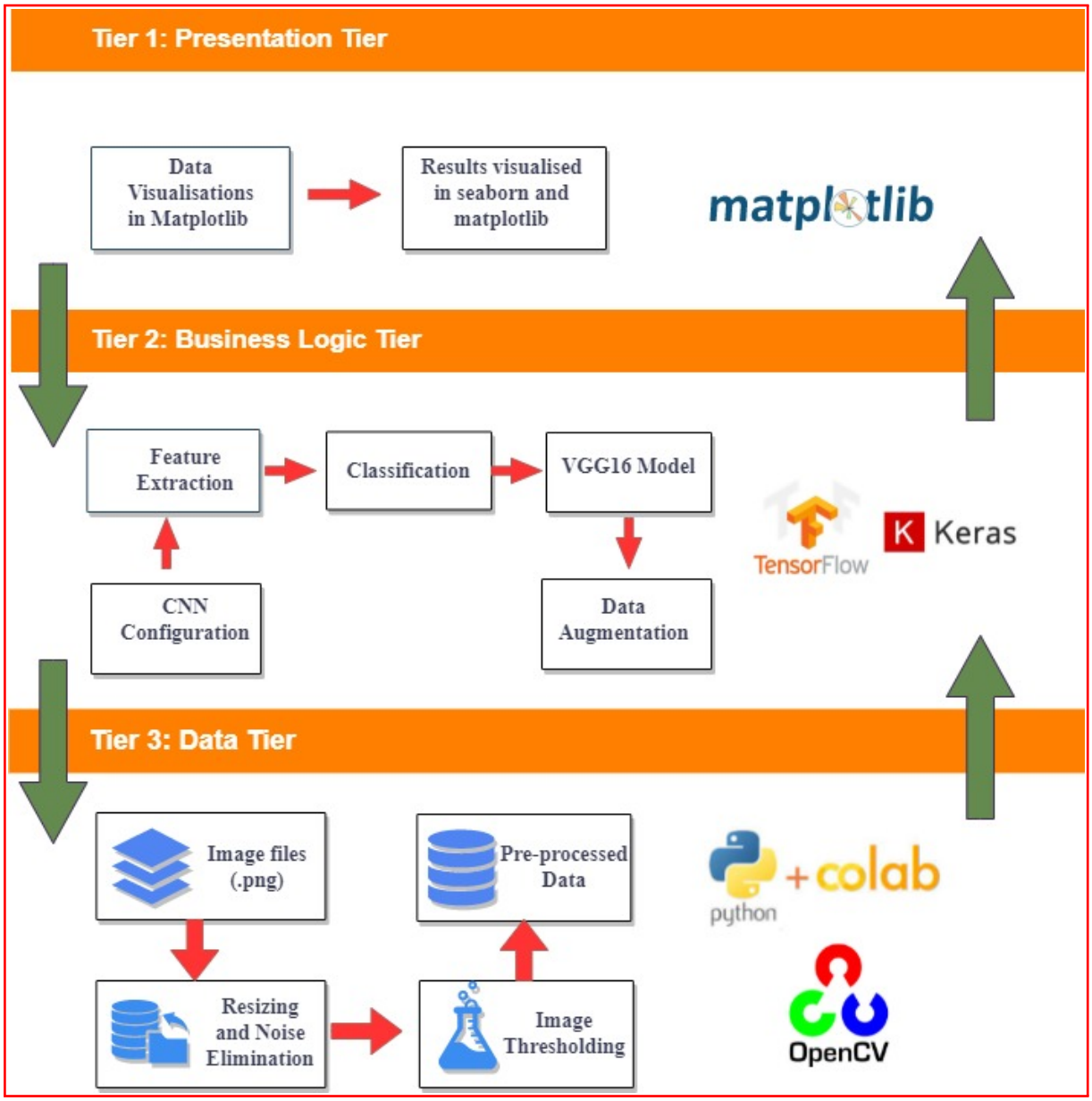


Figure 4: System Design for Recognition of Gestures

## 3.8 Conclusion

The hand gesture recognition research methodology is thoroughly explained in this chapter which meets the project's requirements. The data is collected from Kaggle and all the preliminary steps involved in the research project are briefly performed and showcased in this section, thus achieving the objectives 2 and 3 stated in Table 1.

The core implementation of the various deep learning models with respect to the project topic is illustrated in the next section i.e. chapter 4.

# 4 Implementation of Hand Gesture Recognition and Classification Models

This section describes how the ICT solution for gesture recognition that can assist the dumb-deaf community for devices with sign language identification as well as stakeholders in various businesses with computer vision technology. In the experiment, the dataset of ten hand gestures is used to evaluate the performance of the introduced methodology. These gestures are assigned numeric values as 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 in the ICT solution in order to prepare the data for classification.

## 4.1 Training the Model

The convolutional neural networks (CNNs) are a subset of deep learning algorithms that have presented huge success and innovation in the field of image identification. They are commonly used for analysing visual imagery and image classification as these networks are patterned to resemble the functioning of a human brain. The three major components of a CNN architecture are stated below:

- The convolutional layers are implemented with a specific number of filters (also called kernels) to the raw image. These layers perform a set of mathematical operations such as the matrix multiplication and sum of pixel values that forms into a feature map. The ReLU activation function is also applied to bring up some non-linearity into the model.
- The pooling layers are introduced to subsample or downsample the data extracted by the convolutional layers so as to reduce the dimensionality of feature map. Max pooling is the most widely used algorithm that helps by keeping only the maximum value from the feature thus decreasing the processing time.
- The fully connected layers (dense layers in this project) are employed for the sole purpose of classification. It takes the output generated from the previous layers and uses the softmax activation function for multi-label classification.

Various models have been implemented in this project on the basis of this architecture for the learning of hand gestures and its multi-category classification. The code artefacts for training and testing is prepared by performing the following tasks:

1. After performing all the image processing steps, the pre-processed data is converted into numpy arrays and stored as X data. The shape of the array is (20000, 224, 224)
2. The gesture labels are obtained on the basis of the initial two characters of the file using regular expressions and stored as Y data. This numpy array contains the number of images and the corresponding gesture label. The shape of the array is (20000, 10)
3. Both X and Y data are then split into training and testing sets with a ratio of 70: 30 using the scikit learn library from Python.

**Model Construction:** First is the base model which is built on the traditional 2D CNN architecture having input layer of dimension 224 × 224. A 5 × 5 filter (kernel) is used in the first convolutional layer. This filter convolves i.e. scans the pixels over the input image and extracts the features to produce a feature map. The next layer is the max pooling layer that uses a 2 × 2 filter as it reduces the number of pixels by downsampling the output from feature maps. Similarly, two more conv and max pooling layers are added that keep convolving and extracting features. A visual representation of this model architecture is shown in Figure 5. Then, the Flatten layer flattens the output from these previous convolutional layers into a single vector as the dense layer starts at this point and expects a 1D vectors of numbers. This dense layer takes the input from feature analysis and appropriate weights are applied which helps in predicting the correct label. A dropout layer as well as batch normalisation are added alternatively. This is done to regularise the inputs and take control of the model from overfitting. Finally, the stochastic gradient descent is used for optimisation during the model compilation process and tries different learning rates for the models.

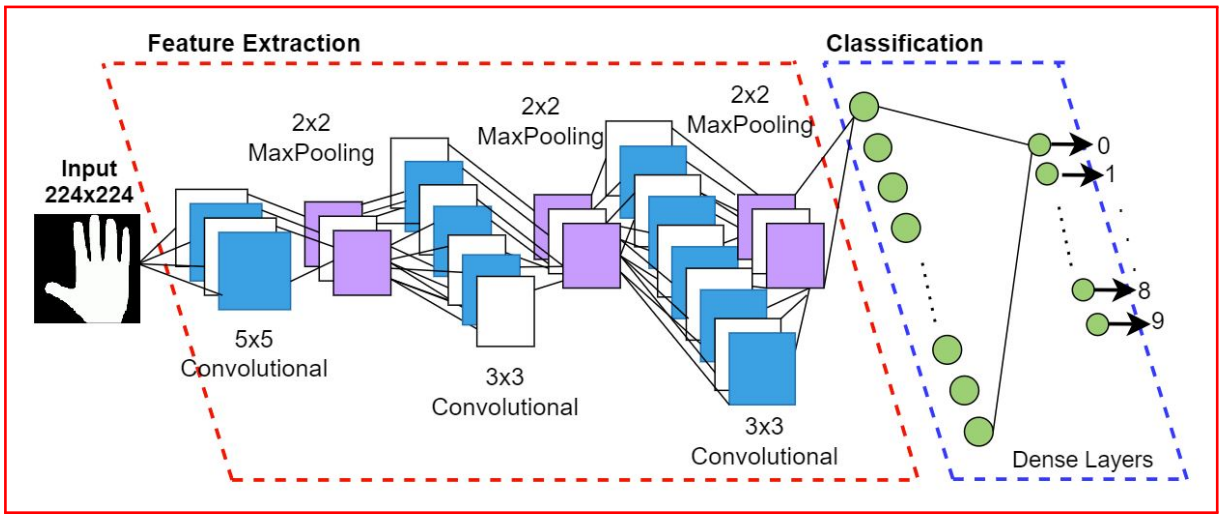


Figure 5: Convolutional Neural Network Architecture

## 4.2 Feature Extraction of Hand Images

The core block of a CNN model is its convolutional layers. It is not easy to discover important features automatically from an image and its label. The first layer identifies edges, the second layer integrate them to recognise forms, followed by other layers that use this information to figure out the underlying object. It learns to recognise them as a feature after viewing a lot of them in images. In this research project, few varieties of CNN models are used for feature extraction. These models are base model, pre-trained, data transformed or fine-tuned with parameters. To produce feature vectors for each of the hand images in the dataset, the data is pre-processed and given to these models. The convolution and pooling layers perform feature extraction. For example, given an image, the convolution layer detects features such as the fingers, edges, shape changes based on the pixel values, etc. and creates the feature mapping which is then fed to the dense layer.

## 4.3 Classification of Gestures

The dense layers learn how to use the features produced by convolutions in order to correctly classify the labels. These layers act as a classifier on top of the features, and assign a probability (weight) for the input image being a 'palm' gesture in this case (refer Figure 5). Dropout is a regularisation technique for preventing overfitting in a model. As the name suggests, this method works by temporarily "dropping" a neuron or disabling it with probability p, at each iteration during the training phase. In this model, the dense layer is assigned with 128 neurons and the activation function as 'relu', the final layer is another dense layer with ten neurons, because the goal is to classify images into ten gesture categories. As it is a multi-class classification problem, the 'softmax' function is utilised.

## 4.4 Transfer Learning

The VGG is a pretrained convolutional neural network from researchers at Oxford's Visual Geometry Group, hence the name VGG. It is a 16-layer deep neural net, containing stacked convolutions and pooling layers throughout the network therefore it is also referred to as VGG16. This network has been trained on the enormous Imagenet dataset and the first convolutional layer accepts a fixed size input of dimension $224 \times 224 \times 3$. Therefore, the data is accordingly processed by stacking the numpy arrays to form this shape and the network is loaded with the help of Keras library. Due to the stacking, the dataset expanded to a huge size. In order to deal with this issue as well as the limitations of GPU in Google Colab, the data was reduced from 20000 images to 10500 images that could be accommodated by available resources.

The major objectives 4 and 4.1 are successfully achieved with the brief explanation as well as diagrammatic representation of the end-to-end 2D CNN model implementation along with additional pretrained network implementation. The results of each of these models are plotted and interpreted in chapter 5.

# 5 Evaluation, Results and Discussion

Various intermediate models are developed using the 2D CNN as well as pretrained network architecture. The different kinds of models based on the modifications of its parameters, number of layers as well as data augmentation are performed, evaluated and presented in the following subsections. Each subsection consists of the appropriated evaluation measures with suitable plots that depict the performance of every model.

## 5.1 Experiment 1- Base Model 2D CNN

### 5.1.1 Model Configuration

The first CNN model was built using three 2D convolutional layers and three max pooling layers for feature mapping. Three layers of batch normalisation are also added for regularising the training data. For the classification process, fully connected layer consisting of two dense layers and a dropout layer to reduce overfitting of the model are included. Dropout helps in getting rid of some of the learned neurons, thus decreasing the overall number of trained neurons.

### 5.1.2 Evaluation and Results

Since this was the initial model, it was fitted with a batch size of 64 and 5 number of epochs to visit the training data. After evaluating the model on unseen testing data, the accuracy was received as 99.53%. Although the accuracy rate is quite high, the confusion matrix suggests that the model was able to learn from the training set and classify most of the gesture samples however, it misclassified appproximately 60 samples of gesture as seen in Figure 6.

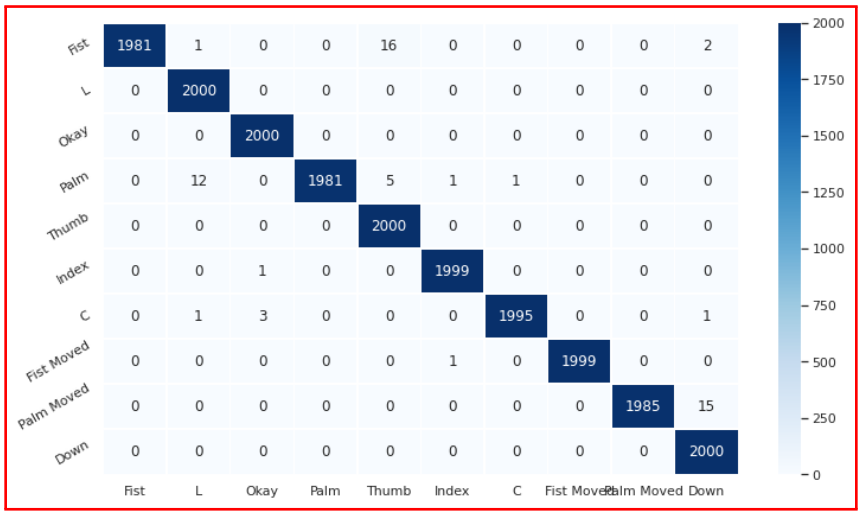


Figure 6: Confusion Matrix for Base 2D CNN Model

The plot in Figure 7 indicates that the model is tending to overfit after the third epoch meaning that the model is unable to learn parts of the unseen data. This is known by the generalisation gap between both the curves. The difference between the initial value of training and validation loss is also quite high. Therefore, it suggests that the model requires some tuning of the parameters in order to learn better.

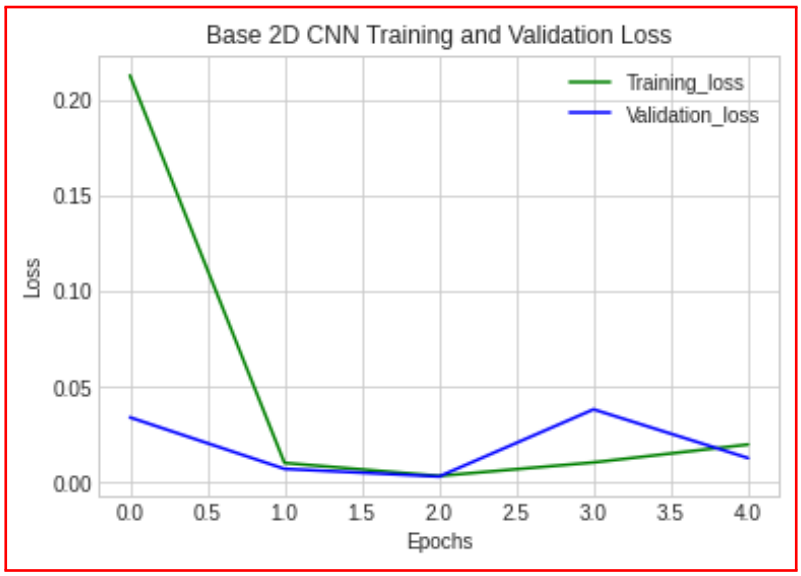


Figure 7: Loss Validation Curve for Base 2D CNN Model

## 5.2 Experiment 2- 2D CNN with hyper-tuning parameters

### 5.2.1 Model Configuration

For the purposes of improving the model performance and expanding the scope of the model, a few modifications are done to the model's hyper-parameters such as the learning rate, optimiser, number of neurons, batch size or number of iterations in order to fine-tune the model. This model has three conv + three max pooling layers with two dense layers and a dropout to avoid overfitting. The neurons in third convolutional layer are increased from 64 to 128 for effective learning. This optimised model is compiled with the stochastic gradient descent function (learning rate of 0.001) and loss function was set to categorical crossentropy. The purpose of using this optimser is that, instead of running through all of the data, stochastic gradient descent calculates the gradient using a random small subset of the observations thus reducing the computing time.

### 5.2.2 Evaluation and Results

After increasing the batch size to 256 and number of epochs to 15 while fitting the model, it achieved an accuracy of about 99.88%. Although there is only a slight change in the rate of accuracy by decimal points, but as for the confusion matrix depicted in Figure 8, the classification has drastically improved. The value of dropout is also increased to 0.5 and there are hardly 6-7 misclassified samples while most of the images are correctly predicted as per their gesture labels. The model has been built up and learnt very well on the training data hence proved from the matrix.

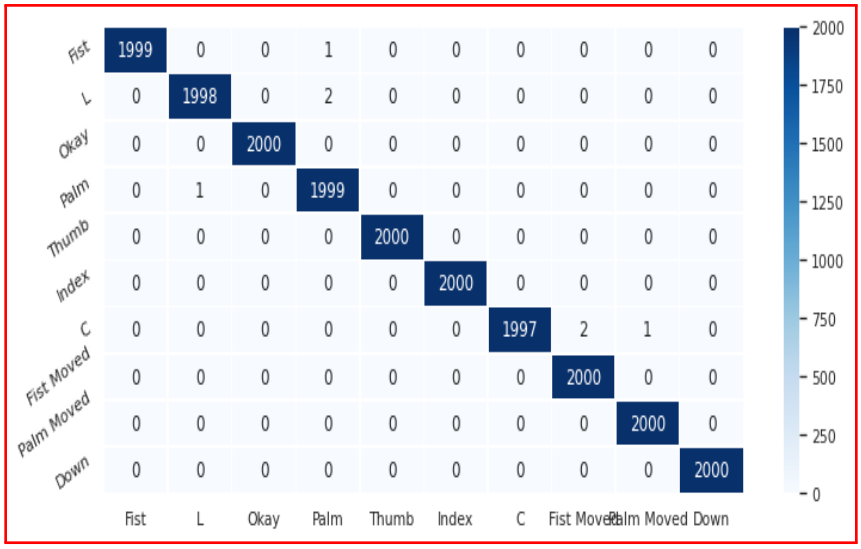


Figure 8: Confusion Matrix of Optimised 2D CNN Model

The plot in Figure 9 represents the training verses validation loss and the curve suggests that the model is an optimal fit. This is proved by the absence of generalisation gap between both the curves meaning that the model does not tend overfit or underfit after a certain run of epochs. The curves are consistent and do not divulge even after the 10th epoch which means that it the model is fit enough to be used on more data.

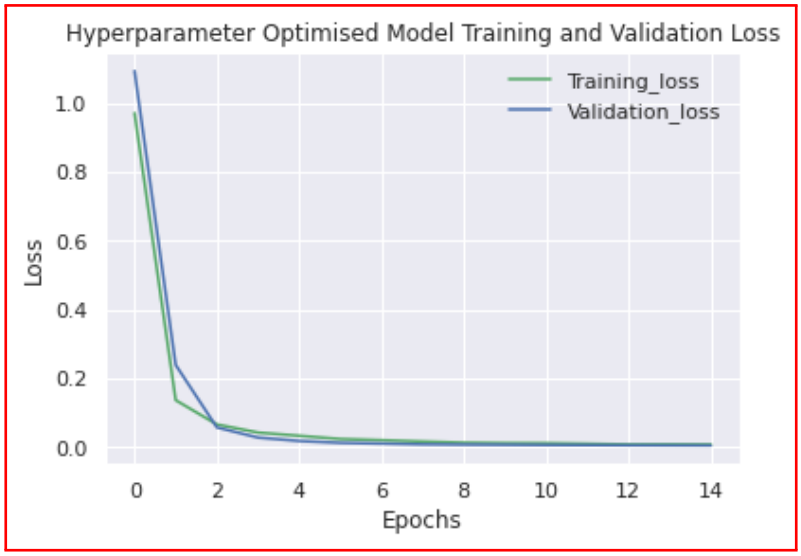


Figure 9: Loss Validation Plot of Optimised 2D CNN Model

## 5.3 Experiment 3- Pretrained VGG16 Model

### 5.3.1 Model Configuration

The main purpose of using pretrained models that are part of transfer learning is that they have already been trained on millions of image data and varieties of class labels and are capable of identifying the problem on other kinds of new data. Therefore, this model is built on the pretrained VGG16 network obtained from Keras that contains 16 deep layers for convolution of images. The images are reshaped to $224 \times 224 \times 3$ as it is the required dimension of the network and additional layers are added on top of it for gesture recognition. The output layer consists of three dense layers and a dropout layer of 0.5 value. As stated in chapter 4.4, the computation requirements were limited for this project, hence the gestures were reduced from ten to seven classes (10500 sample images). Stacking the X data arrays to transform into the necessary input shape also consumed a lot of GPU memory. Hence, Python's garbage collector method helped in releasing some of the unused space. Additionally, other hyperparameters like early stopping and model checkpoint are also set wherein the best weights can be saved.

### 5.3.2 Evaluation and Results

The pretrained VGG16 model is fitted on a batch size of 128, 20 number of epochs and Stochastic gradient descent optimiser with a value of 0.001 which achieved an accuracy of 99.68%. This high accuracy rate is justified as the pretrained networks have already gained knowledge on massive image information thus they are capable of learning and predicting easily on such data. Early stopping technique is also added to the model as it stops the network stops running when it does not show any improvement in the consequent iterations. The confusion matrix in Figure 10 obtained from the seaborn and matplotlib libraries represents the most accurate classification of all the gestures of the seven classes with very few samples of fist, letter L, palm, thumb, palm and index (approx. 20 cases only from a total of 10500) incorrectly identified.

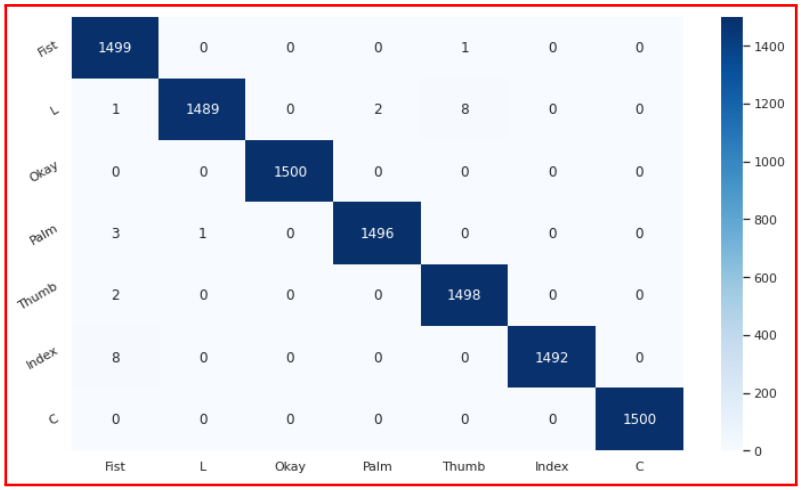


Figure 10: Confusion Matrix of Pretrained VGG16 Model

The plot in Figure 11 indicates that in the initial epochs, the model faced some issue in learning the data as there is a visible generalisation gap amongst the lines throughout the epoch runs. However, it gained momentum after consistently training on the data thus the gap shrinks in the very later epochs. The graph is still better as the performance as well as accuracy kept increasing after each epoch run, without harming the loss metrics.

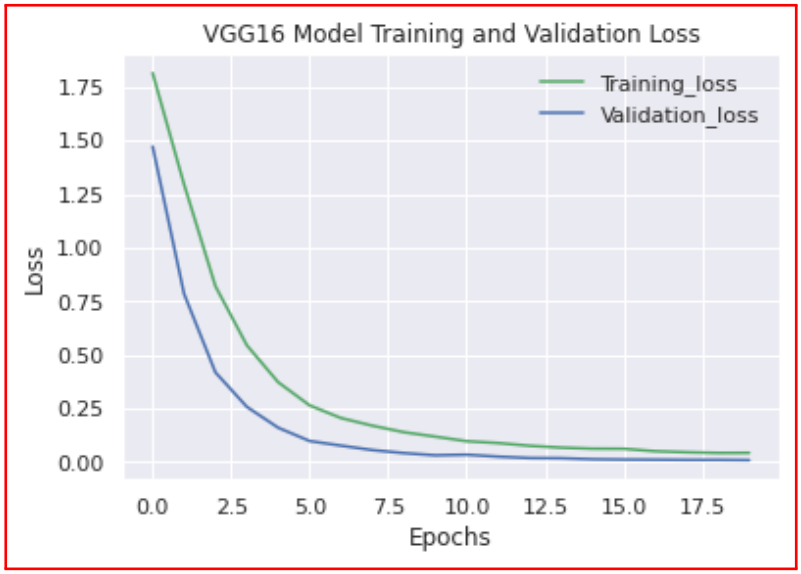


Figure 11: Loss Validation Curve of Pretrained VGG16 Model

## 5.4 Experiment 4- Model with Data Augmentation

### 5.4.1 Model Configuration

Data augmentation technique is applied to this study, to check the robustness of the model where real-time data transformations happen. The input images are flipped, rotated or shifted to certain angles, zoomed and many other variations are added. These augmentations are randomly transformed and directly applied on the training data while fitting the model thus providing an easy way of implementation. The model is fitted with

a batch size of 128 and 15 epochs keeping the model configuration the same as that of pretrained model.

### 5.4.2 Evaluation and Results

The data augmented model shows a slight decline in the accuracy rate and was achieved to be 97.23%. The reason could be the introduction of new data variations, however, the classification as per the confusion matrix (Figure 12) shows a considerable result with only some number of gestures misclassified. Taking into account the real-time data augmentations and correct predictions, the model still performs fine.

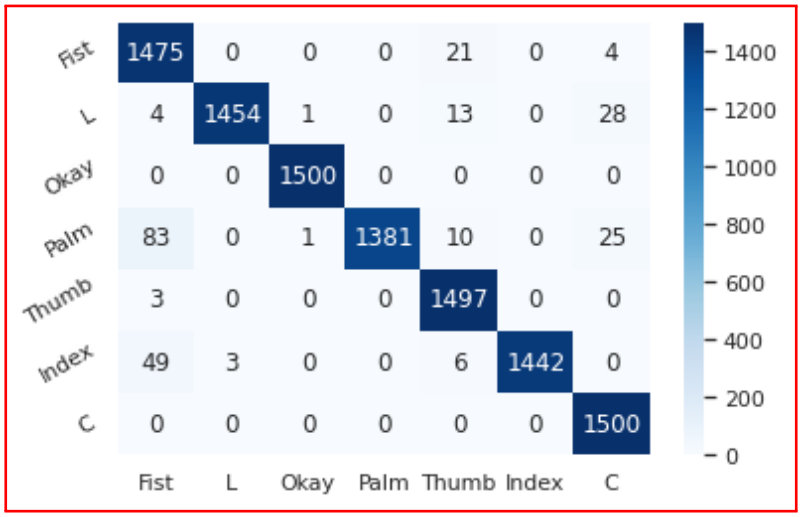


Figure 12: Confusion Matrix for Data Augmented Model

The model tends to have certain amount of generalisation gap and shows some case of underfitting (Figure 13). The model is able to predict the validation data well but is having difficulties learning the training data due to its new data transformations.

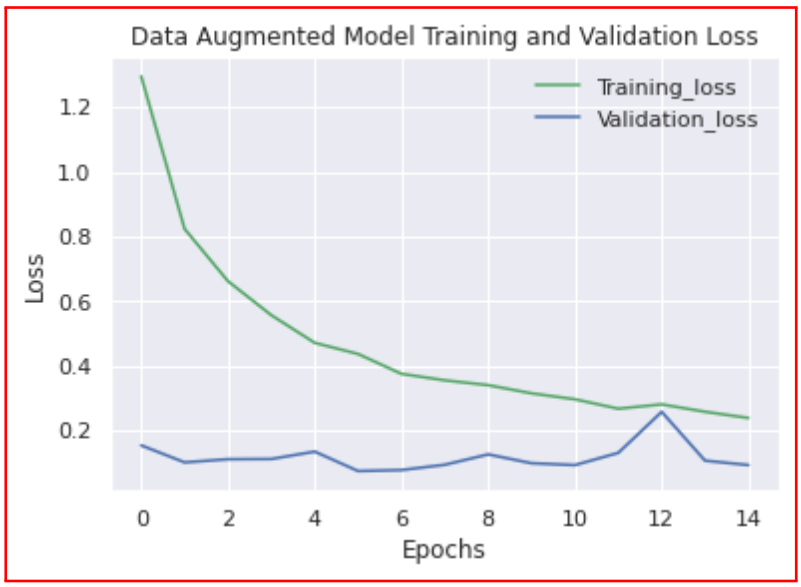


Figure 13: Loss Validation Plot for Data Augmented Model

## 5.5 Comparison of all the Implemented Models

A comparison of the models implemented for this project, namely, 2D CNN model, model with hyper-parameter tuning, pretrained VGG16 model and data augmented model is shown in Table 3.

Table 3: Comparison of the different models implemented

| Sr. No | Model Name | Model Parameters | Accuracy |
|---|---|---|---|
| 1 | Base 2D CNN model | Batch size: 64, Epochs: 5, Adam optimiser | 99.53% |
| 2 | 2D CNN model with hyper-tuning | Batch size: 256, Epochs: 15, SGD optimiser | 99.88% |
| 3 | Pretrained VGG16 model | Batch size: 128, Epochs: 20, SGD optimiser | 99.68% |
| 4 | Data Augmented model | Batch size: 128, Epochs: 15, SGD optimiser | 97.23% |

## 5.6 Discussion

The main purpose of this research project was to solve the problem of hand gesture recognition using state-of-the-art approaches within deep learning techniques and support the applications used in gaming and virtual environments as well as help sign language identification. It makes use of the Hand Gesture Recognition dataset that consists of gestures performed by 10 different persons. In this project, the 2D convolutional neural network is implemented that automatically extracts the features from the raw images and passes it to the next layer for classification. The classification task is performed by setting up different numbers of dense layers and regularisation methods that obtained best results. The base 2D CNN model itself gave a decent accuracy of 99.53% which is very ideal due to the usage of deep learning neural networks. The classifications were also fairly performed with minimal misclassification of labels. Therefore, the next model is optimised with fine-tuning and some changes in the hyperparameters and the accuracy improved to 99.88% showing absolute performance quality in terms of the training and validation loss learning curves. These conclusions were drawn with the help of learning curve plots and the confusion matrix. Apart from the customised CNN models, networks based on transfer learning are also implemented and the famous pretrained VGG16 model with additional top layers for classifier achieved an accuracy of 99.68%. However, the validation curve plot did not suffice with the performance due to the presence of some overfitting across the epochs. The next model is developed using the data augmentation technique which is most commonly used to add new transformations in the existing data and validate the performance in such scenarios. It provided an accuracy of 97.23% with a considerate number of misclassifications in the confusion matrix. Finally, a concise tabular format of the implemented models is also depicted for comparison purposes. The scope of this project was to perform a multi-category classification to recognise and classify ten hand gestures performed by different individuals and it provided the results with innovative and customised models.

All the implemented models performed reasonably well in performing the multi-class classification, although it fell short in terms of change in rate of accuracy across all the

models. Meaning that, all models' accuracy rates do not have a major change as the model performed well in the initial phase itself. Comparing the works of Trigueiros et al. (2012) and Sharma et al. (2020), the model could be implemented using the machine learning classifiers to see the change in performance. One more shortcoming is that these types of problems require much larger datasets which were also used in some of the literature reviews. This study compares multiple techniques for the job of classification, demonstrating the disparities in performance among all of the models used. It also helps in knowing how the use of transfer learning techniques can improve classification results although it required more amount of memory resources, thus resulting in increase of the computational requirements.

The hyperparameter optimised 2D CNN model outperformed the existing developed models on the basis of evaluation and performance metrics i.e. highest accuracy, classification matrix as well as loss validation measures. This model obtaining 99.88& classification accuracy may be applied to classify any new hand gestures performed by other subjects, adding to the plausibility of the model. The performance of the model can also be tried and tested on different hand gestures performed in other background settings to see how it performs with different data from multiple data sources. Finally, this study addressed both deep learning and tranfer learning methods through models and provided a comparative analysis of the actual outcomes of each model.

The major objectives 4.2, 4.3 and 5 are successfully achieved with the brief evaluation and interpretations of the results of all the implemented models along with suitable figures and plots to indicate the performance visually. All the model parameters and results are thoroughly discussed in 5.6 to conclude this important chapter. The final conclusion and future works are stated next in chapter 6.

# 6 Conclusion and Future Work

The aim of the project put forward and solved the research problem of identifying and classifying the hand gestures of people used in a daily basis through deep learning algorithms. The research question introduced in chapter 1.1 has been entirely answered and all the research objectives stated in chapter 1 Table 1 with regards to the key tasks performed in this project have been accomplished throughout the technical report. The literature review in chapters 2.1, 2.2 and 2.3 highlighted a number of potential feature characteristics that could influence the choice of dataset selection. It also discovered a wide range of data-processing algorithms that were helpful in gaining knowledge as to how raw image data is cleaned and processed for machine learning. With respect to the goals of data preparation and modelling, a dataset of hand gestures appropriate to address and solve the problem of recognition and classification models utilising a variety of day-to-day gestures contributed to the body of knowledge. These gesture recognition systems have a huge scope in the market and various application areas such as automating home devices, gaming systems, robotic control as well as sign language recognition.

In this project, the process flow of the introduced research methodology (chapter 3), the implementation of all the the models (chapter 4) as well as critical evaluation of these developed models (chapter 5) are precisely elucidated with understandable diagrams and

plausible reasons for its usage. All the objectives and sub-objectives are successfully accomplished as the models are able to categorise and predict hand gesture recognition using detection and classification models by study the underlying relationships in the dataset. The utilisation of multiple models yielded a variety of outcomes thus each model interpreted the underlying assumptions of the dataset in several ways. These models that are capable of identifying gestures can help the people with speaking-hearing disabilities to operate such systems as well this technology will prove as a boon to the existing gaming and virtual reality systems. The key findings of this data analytical project can be put forward as the producing a robust and over-achieving 2D CNN model for the usage of raw images taken in different settings that can be utilised to build non-touchable gesture recognition systems. This is made possible by the high-end deep learning architectures that have made significant progress in these types of problems and still evolving. Through this project, many lessons and techniques around key areas like image analysis, image processing, unconventional forms of feature extraction through the convolutional neural networks, implementation of classifiers and other branch of transfer learning were learnt. Apart from all these skills, there were also some challenges faced while solving this research problem. As the nature of the problem involves unstructured data and functions on deep learning architectures, there is the need for higher level computational resources. Therefore, one of the major challenges included the limited amount of GPU and RAM memory. Even after using the Google Colab Pro which offers higher RAM and GPU compared to the free version, the models' processing time reached the limits of RAM memory and also took a lot of time for the model to complete the number of iterations.

For future work, the work can be extended to include dynamic hand gestures as well as real-time gestures from video sequences directly. The dataset can be expanded with involvement of some evnironmental disruptions in the background to see how the implemented model performs in such a setting. Massive datasets could be used since the deep neural models work best when trained on large amount of information. Along with this, the usage of other data processing techniques such as background subtraction, skin colour detection, fingers and thumb detection, etc. as well as feature extraction methods like history of gradients and region of interest can be explored. Another approach would be to create hybrid models, inspired from the literature work of Trigueiros et al. (2012) and Sharma et al. (2020) that involves machine learning algorithms like SVM, k-NN, Random Forest, etc. as classifiers. This project could also be expanded by using other transfer learning models used by Zabir et al. (2018) and Agrawal et al. (2020) and see the change in classification in terms of recognition and accuracy measures. Another approach would also be embedding this gesture recognition methodology into automating home devices.

# Acknowledgement

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Catherine Mulwa for the unwavering support, insightful comments, and challenging questions throughout the research project. During the course of 13 weeks, ma'am has provided invaluable assistance for the accomplishment of this work. Lastly, special thanks to my parents and friends for their constant love and support as well as inspiring me with positive encouragements.

# References


Abhishek, B., Krishi, K., Meghana, M., Daaniyaal, M. and Anupama, H. (2020). Hand gesture recognition using machine learning algorithms, *Computer Science and Information Technologies* **1**(3): 116–120.

Agrawal, M., Ainapure, R., Agrawal, S., Bhosale, S. and Desai, S. (2020). Models for hand gesture recognition using deep learning, *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, IEEE, pp. 589–594.

Barczak, A. L. and Dadgostar, F. (2005). Real-time hand tracking using a set of cooperative classifiers based on haar-like features.

Chen, Q., Georganas, N. D. and Petriu, E. M. (2007). Real-time vision-based hand gesture recognition using haar-like features, *2007 IEEE instrumentation & measurement technology conference IMTC 2007*, IEEE, pp. 1–6.

Chen, Z.-h., Kim, J.-T., Liang, J., Zhang, J. and Yuan, Y.-B. (2014). Real-time hand gesture recognition using finger segmentation, *The Scientific World Journal* **2014**.

Do, N.-T., Kim, S.-H., Yang, H.-J. and Lee, G.-S. (2020). Robust hand shape features for dynamic hand gesture recognition using multi-level feature lstm, *Applied Sciences* **10**(18): 6293.

Elboushaki, A., Hannane, R., Afdel, K. and Koutti, L. (2020). Multid-cnn: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in rgb-d image sequences, *Expert Systems with Applications* **139**: 112829.

Hasan, M. M. and Misra, P. K. (2011). Brightness factor matching for gesture recognition system using scaled normalization, *AIRCC's International Journal of Computer Science and Information Technology* **3**(2): 35–46.

Islam, M. Z., Hossain, M. S., ul Islam, R. and Andersson, K. (2019). Static hand gesture recognition using convolutional neural network with data augmentation, *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, IEEE, pp. 324–329.

John, V., Boyali, A., Mita, S., Imanishi, M. and Sanma, N. (2016). Deep learning-based fast hand gesture recognition using representative frames, *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, pp. 1–8.

Kumar, P., Saini, R., Roy, P. P. and Dogra, D. P. (2018). A position and rotation invariant framework for sign language recognition (slr) using kinect, *Multimedia Tools and Applications* **77**(7): 8823–8846.

Panwar, M. and Mehra, P. S. (2011). Hand gesture recognition for human computer interaction, *2011 International Conference on Image Information Processing*, IEEE, pp. 1–7.

Rahim, M. A., Islam, M. R. and Shin, J. (2019). Non-touch sign word recognition based on dynamic hand gesture using hybrid segmentation and cnn feature fusion, *Applied Sciences* **9**(18): 3790.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Sharma, A., Mittal, A., Singh, S. and Awatramani, V. (2020). Hand gesture recognition using image processing and feature extraction techniques, *Procedia Computer Science* **173**: 181–190.

Trigueiros, P., Ribeiro, F. and Reis, L. P. (2012). A comparison of machine learning algorithms applied to hand gesture recognition, *7th Iberian conference on information systems and technologies (CISTI 2012)*, IEEE, pp. 1–6.

Wysoski, S. G., Lamar, M. V., Kuroyanagi, S. and Iwata, A. (2002). A rotation invariant approach on static-gesture recognition using boundary histograms and neural networks, *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02.*, Vol. 4, IEEE, pp. 2137–2141.

Zabir, M., Fazira, N., Ibrahim, Z. and Sabri, N. (2018). Evaluation of pre-trained convolutional neural network models for object recognition, *International Journal of Engineering and Technology* **7**(3.15): 95–98.

Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction, *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 2, IEEE, pp. 28–31.