SIGN LANGUAGE PREDICTOR

A Project Report

Submitted to the Faculty of Engineering of
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

By

R. BHAGYA SRI (19481A05K6)
R. PRASANTH SAI (19481A05K7)
Sk. DAVOOD ALI (19481A05L7)
P. PRAVEEN KUMAR REDDY (19481A05K1)

Under the guidance of
Dr. G. SRIDEVI, M.Tech, Ph.D
Professor of CSE Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SESHADRIRAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRIRAO KNOWLEDGE VILLAGE
GUDLAVALLERU – 521356
ANDHRA PRADESH
2022-2023

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)

SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "SIGN LANGUAGE PREDICTOR" is a bonafide record of work carried out by R. Bhagya Sri (19481A05K6), R. Prasanth Sai (19481A05K7), Sk. Davood Ali (19481A05L7), P. Praveen Kumar Reddy (19481A05K1) under the guidance and super vision in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University Kakinada, Kakinada during the academic year 2022-23.

Project Guide	**Head of the Department**
(Dr. G. SRIDEVI)	**(Dr. M. BABU RAO)**

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragements crown all the efforts with success.

We would like to express our deep sense of gratitude and sincere thanks to **Dr. G. Sridevi,** Professor, Department of Computer Science and Engineering for her constant guidance, supervision and motivation in completing the project work.

We feel elated to express our floral gratitude and sincere thanks to **Dr. M. Babu Rao**, Head of the Department, Computer Science and Engineering for his encouragements all the way during analysis of the project. His annotations, insinuations and criticisms are the key behind the successful completion of the project work.

We would like to take this opportunity to thank our beloved principal **Dr. G.V.S.N.R.V Prasad** for providing a great support for us in completing our project and giving us the opportunity for doing project.

Our Special thanks to the faculty of our department and programmers of our computer lab. Finally, we thank our family members, non-teaching staff and our friends, who had directly or indirectly helped and supported us in completing our project in time.

TEAM MEMBERS

R. Bhagya Sri	**(19481A05K6)**
R. Prasanth Sai	**(19481A05K7)**
Sk. Davood Ali	**(19481A05L7)**
P. Praveen Kumar Reddy	**(19481A05K1)**

INDEX

LIST OF FIGURES

LIST OF ABBREVATIONS

Abbreviation	Explanation
CNN	Convolutional Neural Networks
ASL	American Sign Language
ReLU	Rectified Linear Unit

ABSTRACT

Since that sign language is not a common language, quite few people can comprehend it. The majority of hearing societies find it challenging to communicate with the deaf group because of this. One can anticipate that within a few decades, digital technology would have a significant impact on how people go about their daily lives and that everyone will communicate with machines either by gestures or speech recognition. If we are able to foresee such a future, we should consider the physically disabled and take action to help them. Hence, computerized recognition systems provide a novel technique to interpret deaf signals without the aid of a professional. The 26 hand gestures with in dataset correspond to the letters of the English alphabet, from A to Z. This paper took into account the Hand Gesture Recognition standard dataset from the Kaggle website. Convolutional Neural Network (CNN), the type of neural network and its pretrained MobileNet model used in this study, improve the predictability of the alphabet in American Sign Language (ASL).

CHAPTER - 1
INTRODUCTION

1.1 INTRODUCTION

Unlike spoken languages, that utilize auditory-verbal capabilities for communication, sign languages are complete, natural languages that rely on visual-manual modalities. In contrast to spoken languages, sign languages possess distinct linguistic features and vocabulary items. Despite the similarities between sign languages, communication between signature language speakers might not always be possible. Interpreters believe sign languages to just be natural languages because they developed through time in an unconstrained and unstructured approach. Neither of these is sign language based on spoken languages, nor are they only a condensed version of them. Instead, they have evolved independently and have complex grammatical standards. Body language, a form of nonverbal communication that employs facial expressions, gestures, and postures to convey meaning, is distinct from sign language. Sign language is an entire language that has unique grammar, syntax, and vocabulary. The classification of sign languages may be done using their origins. Some of the most commonly used sign gestures, American sign gestures and French sign gestures, both emerged alongside the spoken languages of their respective nations. French sign gestures had an impact on the development of American sign gestures in the United States in the early 19th century, but French sign gestures originated in France in the late 18th century. There are other different sign languages that are utilized across the globe in addition to these, each having its own vocabulary and syntax. Nicaraguan Sign Gestures, which were formed in the 1980s by a group of deaf adolescents who gathered in a school for the deaf in Nicaragua, are one example of a sign language that has grown among tiny, isolated communities of the deaf. They created their own system of motions and gestures to interact, as they had no prior experience with sign language.

Sign languages have developed independently and have complex grammatical standards. They have evolved through time in an unconstrained and unstructured approach, and are not based on spoken languages. Therefore, sign language cannot be considered as a mere form of body language. Sign languages can be classified based on their origins. For instance, American sign language and French sign language emerged alongside the spoken languages of their respective nations. While French sign language originated in

France in the late 18th century, it had an impact on the development of American sign language in the United States in the early 19th century. Other sign languages also exist worldwide, each having its own unique vocabulary and syntax.

Sign languages have become effective means of communication for people who can't speak or hear. Deaf people use these languages to communicate with one another and with hearing people who have acquired the language. Local Deaf cultures have adopted sign languages as an essential component, and they differ depending on the area and the deaf group that utilises them. Along with those who are deaf or hard of hearing, those who cannot physically communicate, have challenges using spoken language due to their condition or have deaf family members in their family interact with each other with the help of gestures. Those who might not have access to other modes of communication can express themselves, engage with others, and fully participate in society by learning sign languages.

The variety of gesture languages used nationally and globally varies by nation, and most nations have their own native sign language, and some may have multiple languages. For instance, whereas British sign gestures are used in the United Kingdom, American sign gestures are used in the United States and some regions of Canada. Even among the close relatives of the deaf, these languages are frequently unable to be understood by the non-disabled despite their widespread usage. This may result in communication hurdles inside the deaf community as well as between the deaf and hearing populations. There is no universal sign language, unlike spoken languages. This implies that communication might be difficult and inefficient amongst people who use various sign languages. Although textual communication is an option, it can be time-consuming and not always useful, especially when travelling.

The use of gesture languages varies from nation to nation, and most countries have their own unique sign language, with some having more than one. For instance, British sign language is used in the United Kingdom, while American sign language is used in the United States and some parts of Canada. Despite their widespread use, these languages may not be comprehensible to non-disabled individuals, even among close relatives of the deaf. This can lead to communication barriers within the deaf community and between deaf and hearing populations. Unlike spoken languages, there is no universal sign language, making communication challenging and less efficient among individuals who use different sign languages.

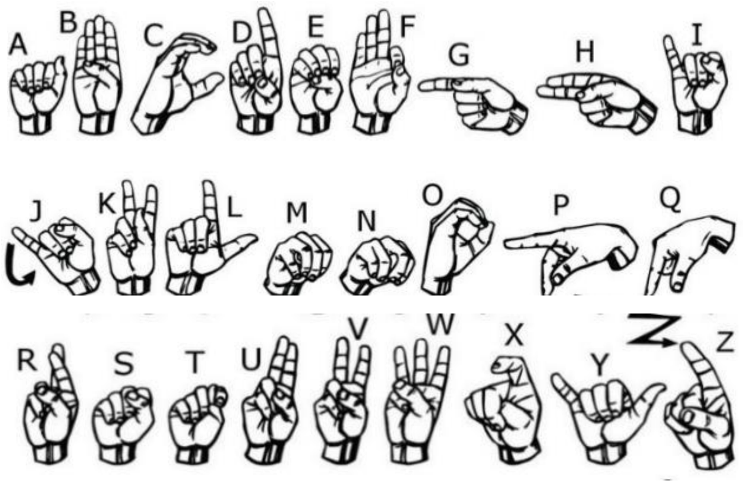


Fig: 1.1: American Sign Alphabets

In this digital technology, system also have to understand what deaf people are saying in front of camera. Hence, a recognition system is needed. The two different sorts of approaches for recognising signs in sign language are sensor-based and image-based. The first technique depends on wearing certain gloves or localised sensors. This method's key advantage is its capacity to accurately convey information regarding signals or gestures, such as the movement, rotation, orientation, and positioning of the hands. Second method is built on image processing, which does not require sensors or other hardware. This method solely uses pattern recognition and a variety of image processing techniques.

1.2 OBJECTIVES OF THE PROJECT

- To recognize and interpret the gestures of American Sign Language users and accurately translate them into text.
- Improving real-time recognition: The model should be able to recognize signs in real-time, allowing for more natural and efficient communication.
- Reducing the need for manual transcription: Sign language interpreters may not always be available, so the ability to automatically transcribe signs into text or speech can be highly beneficial in some situations.
- Optimizing the accuracy and speed of the model to reduce errors and ensure that sign language users can communicate effectively and efficiently.
- Making the system accessible and user-friendly for sign language users.

1.3 PROBLEM STATEMENT

Sign language is a visual language that is used by deaf and hard-of-hearing individuals to communicate with each other and with hearing people. Sign Language Recognition is the process of interpreting and translating sign language gestures into spoken or written language. Sign Language Recognition has numerous applications, including communication between deaf and hearing individuals, improving accessibility for deaf and hard-of-hearing individuals, and enabling sign language users to interact with computers and other technologies.Less people can understand sign language because it is not often spoken. The deaf community and the hearing society are now separated in terms of communication. With the speed at which digital technology is developing, it is likely that soon, humans will use gestures or speech recognition to interact with robots.

The problem of gestures recognition involves developing algorithms and systems that can accurately recognize sign language gestures and translate them into spoken or written language. This requires overcoming several challenges. The goal of Sign Language Recognition is to create robust and accurate systems that can recognize a wide range of sign language gestures, with high accuracy and in real-time. Achieving this goal can improve the quality of life for deaf and hard-of-hearing individuals, and enable more inclusive and accessible communication for all.

CHAPTER - 2
LITERATURE REVIEW

[1] The glove-based sign-to-text/voice converting system by for deaf and dumb is described in this research. The glove presents Arabic sign language characters as text on LCD and produces sounds through the speaker, reducing the communication gap between the deaf and the hearing. [2] The project transforms user-performed American Sign Gestures signs into sentences of English language using a sensor glove. [3] The performance evaluation of the features and classifier is described in the work that Siddharth Kaslay, Tejal Kesarkar, and Kanchan Shinde have proposed. The features utilised in the proposed system are the Gray Level Cooccurrence Matrix (GLCM), Hu moment, and Color moment. To classify data, Support Vector Machine (SVM) is used. The suggested solution makes use of a collection of images depicting American alphabet gestures. Utilizing predictive measures of precision, recall, and f1-score, the system's performance is assessed. Their suggested system had an accuracy rate of 87%. [4] A paper by Apoorva, Harshitha, Chaitra, Akshitha and Rajath discusses about a system that enables regular people and the deaf to communicate with one another. The study's objective is to provide a seamless discourse for those who do not know sign language. In this study, the skin's RGB value and the YCbCr colour space are used as a segmentation method. [5] The Support Vector Machine (SVM) Classifier, which is frequently used for classification and regression analysis, as well as Canny's edge detection and gradient histograms for feature extraction, are employed in the system's architecture. The main feature in this project is local contour sequence (LCS). In order to improve LCS and increase recognition accuracy, it can be utilised with other characteristics. [6] Three frames for every second of the video feed are captured. The frame with the static pose denoted by the hand was then identified by looking at three consecutive frames. A sign motion is recognised in this still position. Its meaning is subsequently determined by comparison to the database of gestures that have been previously stored. It uses Principal Component Analysis. [7] Using skin colour segmentation, it is able to extract indications from video sequences with dynamic and slightly crowded backgrounds. The suitable feature vector is extracted after making the distinction between static and dynamic gestures. Support vector machines are used to categorise these.

CHAPTER - 3
PROPOSED METHOD

3.1 METHODOLOGY

We have used two models. One is CNN(Convolutional Neural Network) and the other is its pre-trained model MobileNet.

3.1.1 CNN (Convolutional Neural Network)

A CNN (Convolutional Neural Network) is a Deep Learning training algorithm which can take an image as input, given various elements and objects like biases and weights in the image importance and distinguish between them. The layers used in our model are shown in figure 3.1.1. A convolutional neural network (CNN) is a type of artificial neural network that is commonly used for image and video analysis. It is designed to automatically learn and extract features from the input data in a hierarchical manner.

CNNs are made specifically to handle input data that has a grid-like layout, like pictures. They are made up of several layers that separate the features from the incoming data before categorising it into one or more groups. Backpropagation is a technique used to train CNNs, where the algorithm modifies the elements of the network like weights to reduce the discrepancy between the expected and actual output. To make sure the network learns accurate representations of the input data, the training procedure can take a while and necessitates a lot of labelled data. Convolutional, pooling and fully linked layers make up a CNN's fundamental building elements.

The key component of a CNN is the convolutional layer, which applies a set of learnable filters (also called kernels or weights) to the input image. The filters slide across the input image, producing a feature map that highlights the areas where the filter pattern matches the input. The resulting feature maps can be fed into additional layers for further processing.

The pooling layers are used to downsample the feature maps and reduce the number of parameters in the network. The most common type of pooling layer is the max pooling layer, which selects the maximum value from each pool of values in the feature map. Finally, CNNs use fully connected layers at the end of the network to classify the input. These layers take the output from the preceding layers and transform it into a set of probabilities for each possible output class.

In comparison to other classification methods, this method requires significantly less pre-processing. CNNs have been very successful in image recognition tasks, such as object detection, face recognition, and image segmentation. They have also been used in natural language processing and speech recognition tasks.

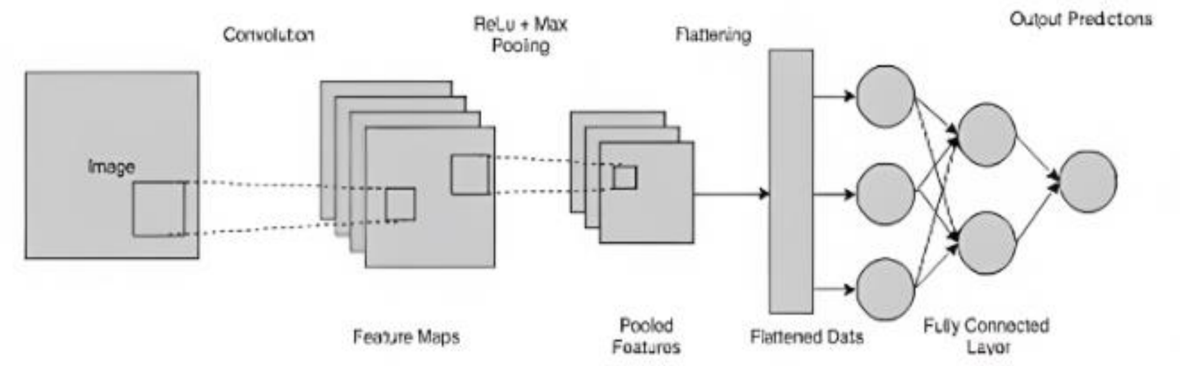


Fig: 3.1.1.1: CNN Design

CONVOLUTION : Convolution is used to extract features from the input with the help of a convolutional layer. Filters are employed in feature extraction. Filters can alternatively be referred to as "masks," "feature detectors," and "kernels." For feature extraction in this convolution procedure, it passes a similar mask through the image. The kernel values in the picture that the mask covers that correspond to the values in the mask are multiplied. The total of the multiplied values, which is referred to as the response, yields the output pixel value that corresponds to the solution. To extract features in this convolution process, a mask is applied to the image. The mask has similar values to a kernel that passes through the image. The kernel values in the image that correspond to the mask values are multiplied. The resulting product of these multiplied values, known as the "response," provides the output pixel value that corresponds to the solution.

The example demo is given below in figure 3.1.2.

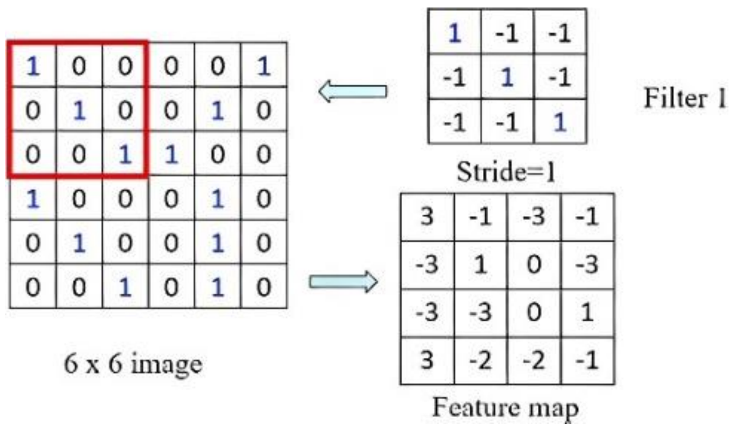


Fig: 3.1.1.2: Convolution

(1x1)+(0x(-1))+ (0x(-1))+ (0x(-1)) + (1x1) + (0x(-1)) + (0x(-1))+(0x(-1))+(1x1) = 1+0+0+0+1+0+0+0+1 which gives 3 as the value of a first pixel in the output(Feature map) when the 3x3 filter (Filter 1) is centered on the highlighted part of the input image (6x6 image) in figure 3.1.2. The stride in the figure represents the number of pixels the filter has to move over the input image during the operation. In our instance, 3x3 filters were employed. Convolution operations have been performed using a 2D convolutional layer, also known as conv2D.

POOLING : Similar to convolution, the pooling process also utilizes a filter/kernel, albeit one without any elements (sort of an empty array). It essentially involves sliding this filter over sequential patches of the image and processing pixels caught in the kernel in some kind of way; basically the same as a convolution operation.There are mainly two types of pooling operations used in CNNs, they are, Max Pooling and Average Pooling.Max pooling entails scanning over an image using a filter and at each instance returning the maximum pixel value caught within the filter as a pixel of its own in a new image.

In the figure 3.1.3., an empty (2, 2) filter is slid over a (4, 4) image with a stride of 2 as discussed in the section above. The maximum pixel value at every instance is returned as a distinct pixel of its own to form a new image. The resulting image is said to be a max pooled representation of the original image. Just like Max Pooling, an empty filter is also slid over the image but in this case the average/mean value of all the pixels caught in the filter is returned to form an average pooled representation of the original image as illustrated in figure 3.1.3.

From the illustrations in the figure 3.1.3, one can clearly see that pixel values are much larger in the max pooled representation compared to the average pooled representation. In more simple terms, this simply means that representations resulting from max pooling are often sharper than those derived from average pooling.

Convolutional Neural Networks (CNNs) are a type of artificial neural network that are well-suited for image processing tasks. The process of convolution is a fundamental operation in CNNs, which involves sliding a small filter (also called a kernel or a weight matrix) over an input image, computing a dot product between the filter and a small patch of the image at each position, and producing a new output feature map. The filters are typically learned during training, and are designed to extract different features from the input image, such as edges, corners, and textures. By using multiple filters of different sizes and orientations, the CNN can extract a rich set of features that capture the underlying structure of the input image.

After the convolutional layers have extracted these features, the next step in the CNN is usually pooling. Pooling is a way to reduce the spatial dimensionality of the feature maps while retaining the most important information. The most common type of pooling operation is max pooling, which partitions the feature map into non-overlapping rectangular regions and selects the maximum value in each region. The result of pooling is a smaller, downsampled feature map that captures the most salient features of the original image. In this way, pooling can be seen as a form of dimensionality reduction, similar to principal component analysis (PCA).

Overall, CNNs are able to learn hierarchical representations of images, where low-level features (such as edges and corners) are extracted in the early layers, and high-level features (such as object parts and textures) are extracted in the later layers. By using convolution and pooling, CNNs are able to capture the spatial structure of images and learn representations that are robust to variations in lighting, scale, and orientation.

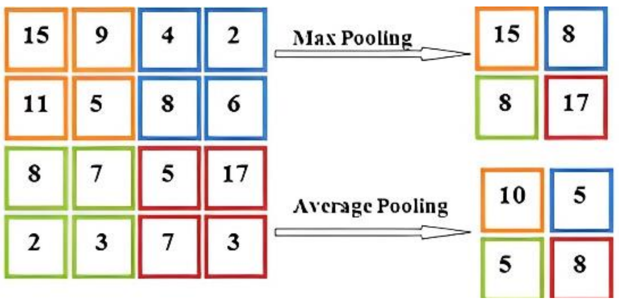


Fig: 3.1.1.3: Pooling

FLATTENING : After performing a polling operation, the resulting two dimensional arrays are flattened to create a single, extended, continuous linear vector. The fully linked layer receives this flattened value obtained by flattening as input to identify the picture (the dense layer).

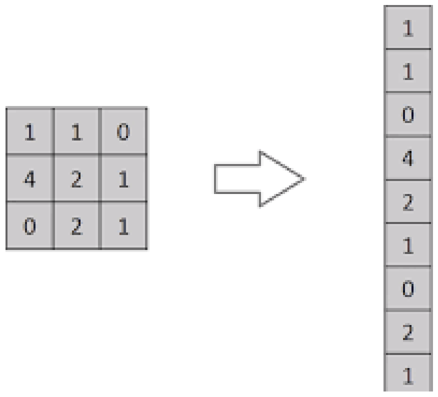


Fig: 3.1.1.4: Feature map is converted into linear vector

DENSE : In neural networks, fully-connected layers, also referred to as linear layers, are frequently employed. They link every input neuron to every output neuron. The input image is classified into a label using the fully connected layer. This layer links the data that was extracted in the earlier phases to the output layer, ultimately labelling the input according to the classification.

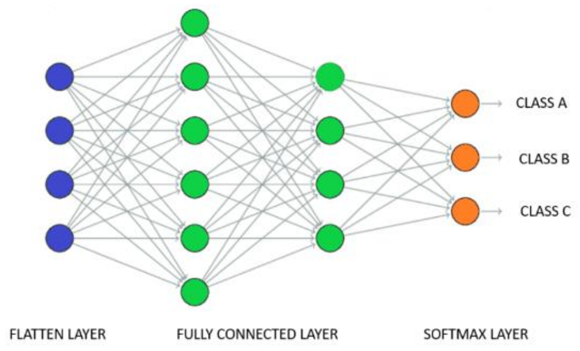


Fig: 3.1.1.5: Fully Linked Layer

ACTIVATION FUNCTIONS : One of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network. It adds non-linearity to the network. It’s just a thing function that you use to get the output of node. These are employed to understand and estimate any form of continuous and intricate correlation between the variables within the network. To put it simply, they determine which information in the model should activate during forward propagation and which information should not be activated at the end of the network. This introduces non-linear characteristics to the network. It is a mathematical function used to produce the output of a node. It is also known as Transfer Function. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred for a multi-class classification, generally softmax is used.

There are two types of activation functions. They are:

1. Linear Activation Function
2. Non-linear Activation Functions

Linear Activation Function: The function is a line or linear. Therefore, the output of the functions will not be confined between any range.

Equation: f(x) = x Range: (-infinity to infinity) It doesn’t help with the complexity or various parameters of usual data that is fed to the neural networks.

Non-Linear Activation Function: The Nonlinear Activation Functions are the most used activation functions. It makes it easy for the model to generalize or adapt with variety of data and to differentiate between the output.

The main terminologies needed to understand for nonlinear functions are:

Derivative or Differential: Change in y-axis w.r.t. change in x-axis.It is also known as slope.

Monotonic function: A function which is either entirely non-increasing or nondecreasing. The Nonlinear Activation Functions are mainly divided on the basis of their range or curves.

1. Sigmoid or Logistic Activation Function
2. Tanh or hyperbolic tangent Activation Function
3. ReLU (Rectified Linear Unit) Activation Function

The Sigmoid Function curve looks like a S-shape. The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output. Since, probability of anything exists only between the range of 0 and 1, sigmoid is the right choice. The function is differentiable. That means, we can find the slope of the sigmoid curve at any two points. It has some disadvantages like slow convergence, vanishing gradient problem or it kill gradient, etc. Output of Sigmoid is not zero centered that makes its gradient to go in different directions. The function is differentiable. The function is monotonic but function’s derivative is not. The logistic sigmoid function can cause a neural network to get stuck at the training time. The softmax function is a more generalized logistic activation function which is used for multiclass classification. It has some disadvantages like slow convergence, vanishing gradient problem or it kill gradient, etc. Output of Sigmoid is not zero centered that makes its gradient to go in different directions. The function is monotonic but function’s derivative is not. The logistic sigmoid function can cause a neural network to get stuck at the training time. The softmax function is a more generalized logistic activation function which is used for multiclass classification.

Sigmoid Activation function can be represented as:

$f(x) = 1 / 1 + \exp(-x)$

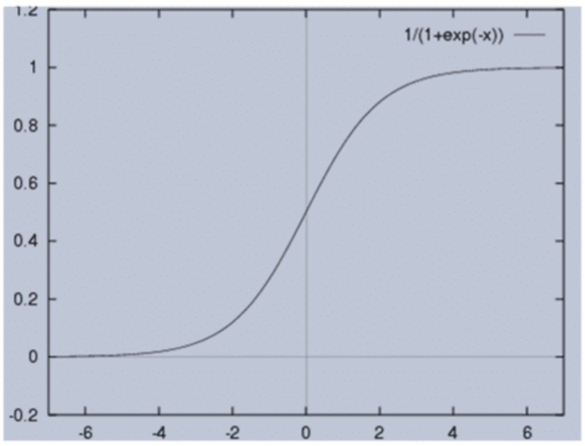


Fig: 3.1.1.6: Sigmoid Activation Function

Tanh or hyperbolic tangent Activation Function is like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped). The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph. The function is differentiable. The function is monotonic while its derivative is not monotonic. The tanh function is mainly used classification between two classes. Both tanh and logistic sigmoid activation functions are used in feed-forward nets.

Tanh activation function can be represented as:

f(x) = 1—exp(-2x) / 1 + exp(-2x)

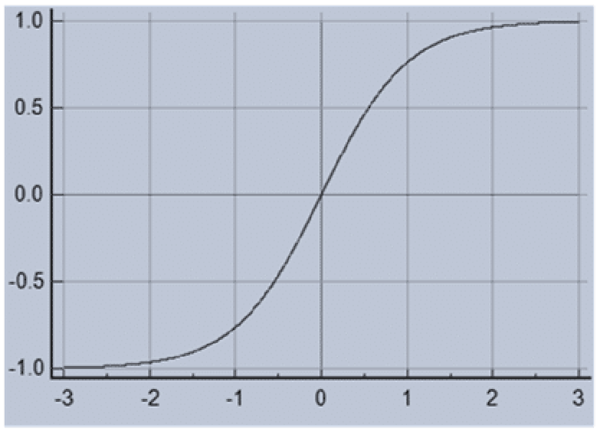


Fig: 3.1.1.7: Tanh Activation Function

Rectified Linear Unit, or ReLU, function may be used the most frequently for buried layers. We employed ReLU in hidden layers to solve the vanishing gradient issue and speed up computation performance. If the input is negative, the function gives zero,otherwise it returns x. The function and its derivative both are monotonic.

ReLU activation function can be represented as:

R(x) = max(0,x) if x < 0 , R(x) = 0 and if x >= 0 , R(x) = x

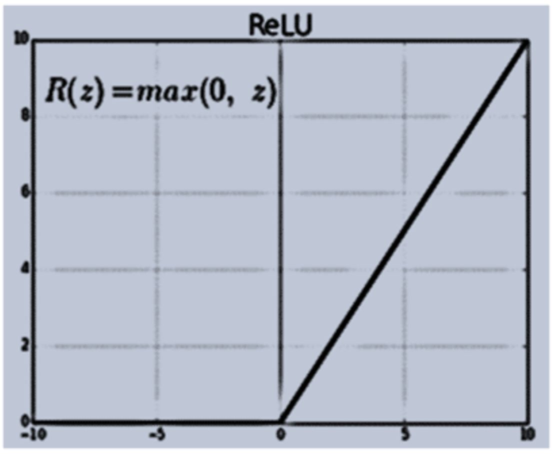


Fig: 3.1.1.8: ReLU activation function

OPTIMIZERS : Optimizers are algorithms or methods used to change the attributes of neural network such as weights and learning rate in order to reduce the losses.

LOSS FUNCTION : The loss function is the function that computes the distance between the current output of the algorithm and the expected output. It's a method to evaluate how your algorithm models the data. It can be categorized into two groups. One for classification (discrete values, 0,1,2…) and the other for regression (continuous values). *CROSS-ENTROPY :* This function comes from information theory where the goal is to measure the difference between two averages of the number of bits of distribution of information. The cross-entropy as the Log Loss function (not the same but they measure the same thing) computes the difference between two probability distribution functions. Entropy is the number of bits required to transmit a randomly selected event from a probability distribution. A skewed distribution has low entropy, whereas a distribution where events have equal probability has a larger entropy.

In information theory, we like to describe the "surprise" of an event. Low probability events are more surprising therefore have a larger amount of information. Whereas probability distributions where the events are equally likely are more surprising and have larger entropy.

- Skewed Probability Distribution (unsurprising): Low entropy.
- Balanced Probability Distribution (surprising): High entropy.

The cross-entropy is a class of Loss function most used in machine learning because that leads to better generalization models and faster training. Cross-entropy can be used with binary and multiclass classification problems (many classes with one label, different from many classes with multilabel called multilabel classification).

Types of cross-entropy:

- Binary cross-entropy: for binary classification problem
- Categorical cross-entropy: binary and multiclass problem, the label needs to be encoded as categorical, one-hot encoding representation (for 3 classes: [0, 1, 0], [1,0,0], …..)
- Sparse cross-entropy: binary and multiclass problem (the label is an integer — 0 or 1 or … n, depends on the number of labels)

Range of values for this class of Loss function:

- 0.00: Perfect probabilities
- < 0.02: Great probabilities
- < 0.05: In a good way
- < 0.20: Great • > 0.30: Not great
- 1.00: Hell
- > 2.00 Something is not working

MOMENTUM : Momentum was invented for reducing high variance in SGD and softens the convergence. It accelerates the convergence towards the relevant direction and reduces the fluctuation to the irrelevant direction. One more hyperparameter is used in this method known as momentum symbolized by ‘γ’.

$V(t) = \gamma V(t-1) + \alpha.\nabla J(\theta)$

Now, the weights are updated by $\theta=\theta-V(t)$. The momentum term γ is usually set to 0.9 or a similar value.

Advantages:

1. Reduces the oscillations and high variance of the parameters.
2. Converges faster than gradient descent.

METRICS : Evaluating a machine learning algorithm is an essential part of any project. An algorithm can be evaluated with the help of metrics. There are many metrics to evaluate an algorithm. We use classification accuracy as the metric in our project.

ACCURACY : The proportion of samples that are properly classified is the most basic evaluation criterion for classification issues. The likelihood that a sample will really be positive among all those that are predicted to be positive is known as precision. Recall rate, which pertains to the original sample, is the probability of being accurately predicted

as a positive sample in the actual positive sample. The kappa coefficient is a statistical consistency indicator that may be used to assess categorization accuracy.
The F score is determined by averaging recall and accuracy harmonically. When the f1 score is near to 1, which is the average f1 value, the score indicates great performance.

3.1.2 MobileNet (A Pre-trained CNN Model)

MobileNet is a CNN architecture that is efficient and adaptable for real-world uses. It is a neural network architecture designed for efficient mobile devices such as smartphones, tablets, and embedded devices. It was developed by Google researchers to address the challenge of running deep learning models on devices with limited computational resources and memory.There are many MobileNet variants that range in accuracy and computational complexity. The depth multiplier parameter of the original MobileNet design, which controls the number of channels in each layer, allows for trade-offs between model size and accuracy. The 27 convolution layers of the MobileNet model are composed of 13 depthwise convolutional layers, one average pooling layer, one fully connected layer, and one softmax layer.

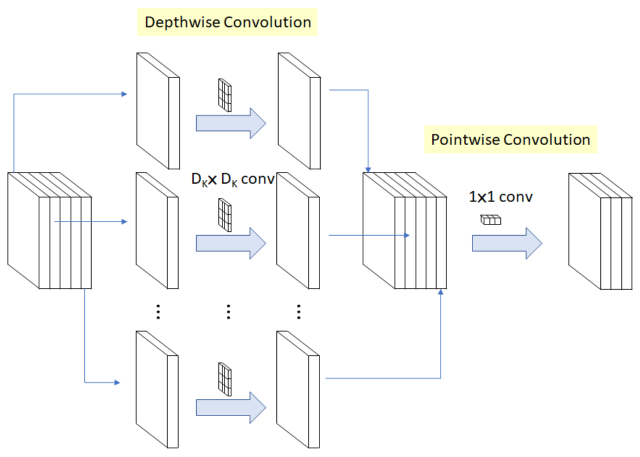


Fig: 3.1.2.1: Depth-wise and Pointwise convolution in MobileNet

It uses depth-wise separable convolutions to reduce the number of computations required for a convolutional layer while maintaining high accuracy in tasks such as image classification and object detection. Depth-wise separable convolutions consist of a depth-wise convolution that applies a single filter to each input channel followed by a point-wise convolution that combines the outputs of the depth-wise convolution using a 1x1 kernel. The depthwise separable convolutions that form the foundation of MobileNet's topology are divided into two primary operations: depthwise convolutions and pointwise convolutions. A single filter is applied to each channel of the input tensor in a depthwise convolution, whereas a set of 1x1 filters are applied to the output of a depthwise convolution in a pointwise convolution. This approach drastically lowers the amount of parameters and computational cost compared to traditional convolutions, while still preserving accuracy. A single filter is applied to each channel of the input tensor in a depthwise convolution, whereas a set of 1x1 filters are applied to the output of a depthwise convolution in a pointwise convolution. This approach drastically lowers the amount of parameters and computational cost compared to traditional convolutions, while still preserving accuracy. The term depth-wise convolution refers to the channel-wise DK x DK spatial convolution. If there are five channels as in the figure 3.1.7, we will have five DK x DK spatial convolutions. The 1 x 1 convolution in figure 3.1.7 used to adjust the dimension is called pointwise convolution.

The MobileNet architecture also includes a technique called "bottlenecking," which reduces the number of parameters and computations required by using 1x1 convolutions to reduce the number of channels before applying a larger convolutional layer. MobileNet has several variants, including MobileNetV1, MobileNetV2, and MobileNetV3, with each iteration improving on the previous one in terms of accuracy, efficiency, and complexity.

3.2 IMPLEMENTATION

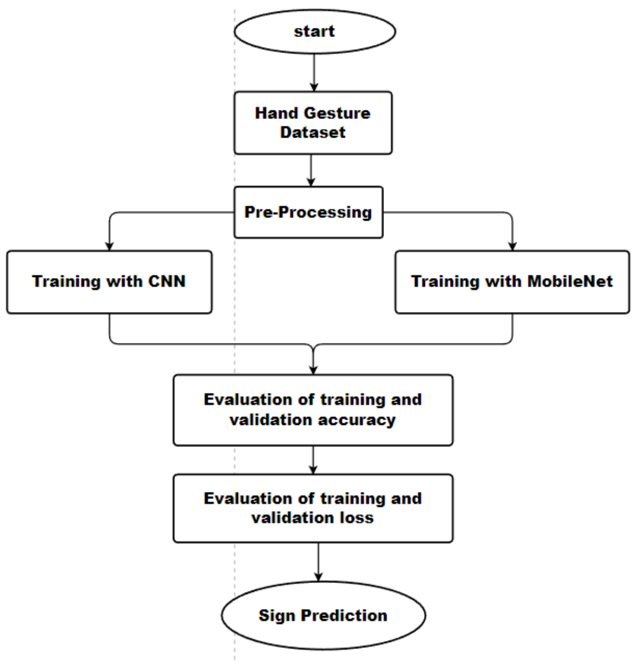
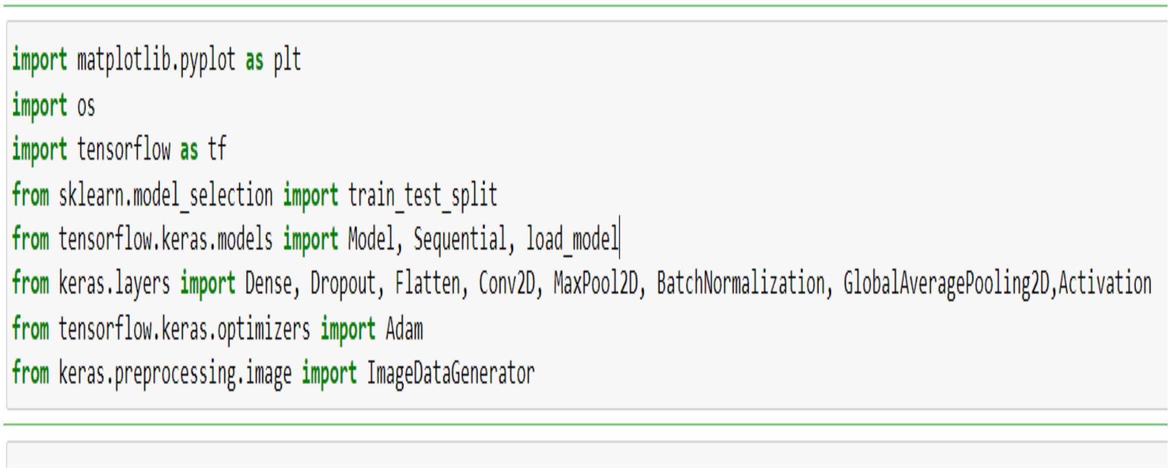


Fig: 3.2.1: Implementation

Hand Gestures Dataset: We have downloaded dataset from Kaggle website https://www.kaggle.com/datasets/ayuraj/american-sign-language-dataset

We have taken folders named with alphabets from a-z. Each folder contains 700 files.

The following libraries were imported for model building



```
import matplotlib.pyplot as plt
import os
import tensorflow as tf
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Model, Sequential, load_model
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization, GlobalAveragePooling2D,Activation
from tensorflow.keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
```

Fig: 3.2.2: Imported libraries

Matplotlib is a popular data visualization library for Python. It is used to create high-quality graphs, charts, and other visual representations of data. It was originally created by John D. Hunter in 2003 and is now maintained by a large team of developers. It provides a wide range of functions and tools for creating different types of visualizations, such as line plots, scatter plots, bar charts, histograms, and more. These visualizations can be customized extensively using a variety of formatting options, such as colors, line styles, markers, fonts, and annotations. It is designed to be highly compatible with NumPy, another popular Python library for numerical computing. This makes it easy to create visualizations from data stored in NumPy arrays or other data structures. It also provides support for creating animations and interactive visualizations, making it a powerful tool for data exploration and analysis. It can be used in a variety of contexts, from scientific research and engineering to data science and machine learning. Its ease of use, flexibility, and powerful features have made it one of the most widely used data visualization libraries in the Python ecosystem. This makes it easy to create visualizations from data stored in NumPy arrays or other data structures. It also provides support for creating animations and interactive visualizations, making it a powerful tool for data exploration and analysis. It can be used in a variety of contexts, from scientific research and engineering to data science and machine learning. Its ease of use, flexibility, and powerful features have made it one of the most widely used data visualization libraries in the Python ecosystem.

OS module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system.The OS comes under Python's standard utility modules. This module offers a portable way of using operating system dependent functionality. The Python OS module lets us work with the files and directories. It allows Python programs to perform various operating system related tasks, such as accessing files and directories, managing processes, and working with environment variables. The module also provides functions for working with file permissions, accessing environment variables, and managing processes. Additionally, the os.path submodule provides functions for working with file paths in a platform-independent way. We can use the OS module to navigate through the file system, create directories, list directories and files, rename files and directories, delete files and directories, etc. OS module can be used to run system commands and shell scripts. We can use the OS module to get and set environment variables. Is can be used to start and stop processes, get information about running processes, and communicate with them. OS module gives information about the current user, the current working directory, the system platform, and much more. Like this it peforms many activities.

TensorFlow is an open-source machine learning framework developed by Google that is widely used in research and industry. The module is the Python API for TensorFlow, which allows you to build and train machine learning models using TensorFlow in a Python environment. TensorFlow uses a computation graph to represent a machine learning model as a set of mathematical operations that are executed in sequence. This allows TensorFlow to efficiently compute the gradients needed for optimization and to perform distributed training on multiple devices. TensorFlow includes a built-in automatic differentiation engine, which allows you to easily compute the gradients of complex functions with respect to their inputs. It provides several high-level APIs, such as Keras and Estimator, which make it easy to build and train machine learning models without needing to write low-level TensorFlow code. TensorFlow can take advantage of GPUs to accelerate computation, making it possible to train large-scale machine learning models much faster than on a CPU.

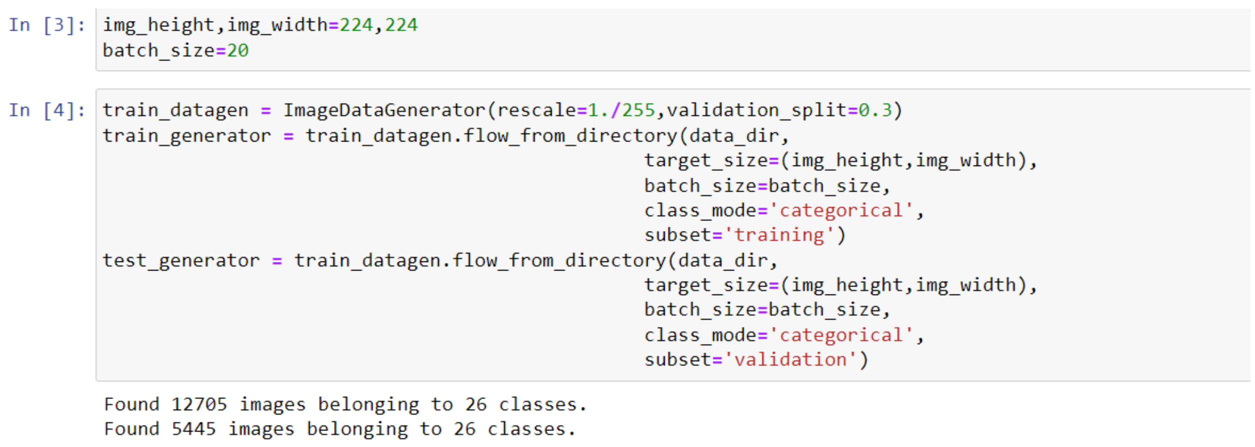
Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It was developed with a focus on enabling fast experimentation, and its ease-of-use and modularity make it particularly suitable for building and testing deep learning models. It also supports multiple backend neural network computation. Keras also provides a variety of utilities and pre-processing functions for working with data, such as data loading, image augmentation, and text preprocessing. Overall, the Keras module in Python makes it easy to build and train deep learning models, even if you are new to deep learning. It also supports multiple backend neural network computation. Keras also provides a variety of utilities and pre-processing functions for working with data, such as data loading, image augmentation, and text preprocessing. Overall, the Keras module in Python makes it easy to build and train deep learning models, even if you are new to deep learning. Keras is relatively easy to learn and work with because it provides a python frontend with a high level of abstraction while having the option of multiple back-ends for computation purposes. This makes Keras slower than other deep learning frameworks, but extremely beginner-friendly.

Scikit-learn (Sklearn) is a popular open-source machine learning library for Python. It provides tools for data preprocessing, feature extraction, supervised and unsupervised learning, model evaluation, and data visualization. It is built on top of NumPy, SciPy, and matplotlib. It offers a variety of machine learning algorithms for classification, regression, clustering, dimensionality reduction, and model selection. It also includes tools for model selection and evaluation, such as cross-validation, grid search, and metrics for classification and regression problems. In addition, Scikit-learn provides utilities for working with datasets, such as loading and splitting data, and for preprocessing data, such as scaling and normalization. Scikit-learn is a powerful and easy-to-use machine learning library that is widely used in both academia and industry. It offers a large number of tools and algorithms that make it possible to quickly and easily build and evaluate machine learning models in Python. Scikit-learn provides utilities for working with datasets, such as loading and splitting data, and for preprocessing data, such as scaling and normalization. Scikit-learn is a powerful and easy-to-use machine learning library that is widely used in both academia and industry. It offers a large number of tools and algorithms that make it possible to quickly and easily build and evaluate machine learning models in Python.

Image height and width are set to (224, 224): This means that all the images in the dataset are resized to a height of 224 pixels and a width of 224 pixels. Resizing the images to a fixed size is necessary to ensure that they all have the same dimensions, which is required for feeding them into a neural network.

Batch size is set to 20: This means that during the training process, the neural network will process 20 images at a time before updating its weights. The batch size is a hyperparameter that can affect the performance of the model, and it is typically chosen through experimentation.

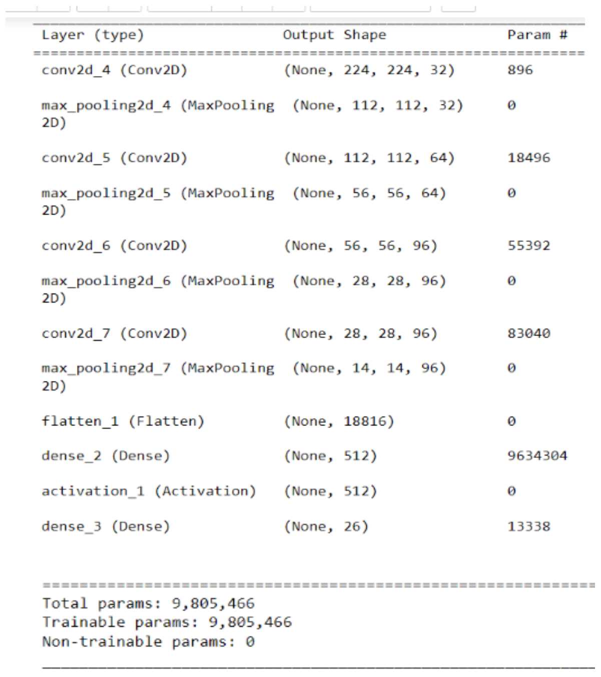
18,150 images were split into 30% and 70%: This means that the dataset was split into two parts, one for training the model and the other for testing its performance. The split was done randomly, with 30% of the images being used for testing and 70% for training. 70% (12,705) of the data is for training, and 30% (5,445) of the data is for testing: This means that 12,705 images were randomly selected to be used for training the model, and 5,445 images were selected to test the model's performance. The testing set is used to evaluate how well the model generalizes to new, unseen data that it has not been trained on.



```
In [3]: img_height,img_width=224,224
        batch_size=20

In [4]: train_datagen = ImageDataGenerator(rescale=1./255,validation_split=0.3)
        train_generator = train_datagen.flow_from_directory(data_dir,
                                                            target_size=(img_height,img_width),
                                                            batch_size=batch_size,
                                                            class_mode='categorical',
                                                            subset='training')
        test_generator = train_datagen.flow_from_directory(data_dir,
                                                            target_size=(img_height,img_width),
                                                            batch_size=batch_size,
                                                            class_mode='categorical',
                                                            subset='validation')

        Found 12705 images belonging to 26 classes.
        Found 5445 images belonging to 26 classes.
```

Fig: 3.2.3: Data splitting

After splitting the model, we have trained the model using CNN(Convolutional Neural Network) and the other is its pre-trained model MobileNet.

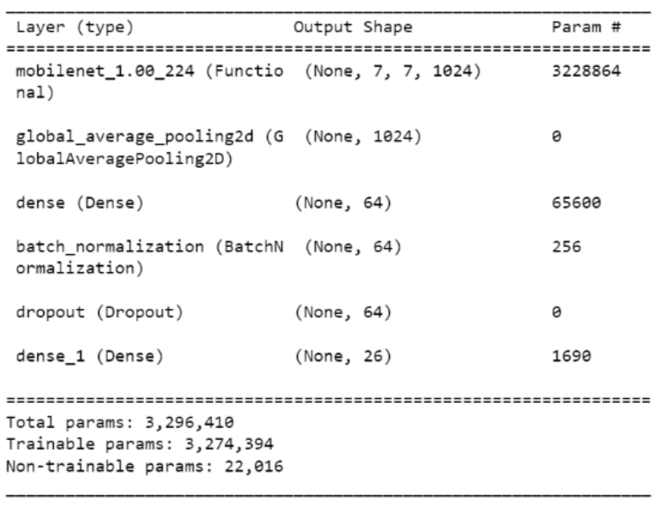


```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 224, 224, 32)      896

max_pooling2d_4 (MaxPooling  (None, 112, 112, 32)      0
2D)

conv2d_5 (Conv2D)            (None, 112, 112, 64)      18496

max_pooling2d_5 (MaxPooling  (None, 56, 56, 64)        0
2D)

conv2d_6 (Conv2D)            (None, 56, 56, 96)        55392

max_pooling2d_6 (MaxPooling  (None, 28, 28, 96)        0
2D)

conv2d_7 (Conv2D)            (None, 28, 28, 96)        83040

max_pooling2d_7 (MaxPooling  (None, 14, 14, 96)        0
2D)

flatten_1 (Flatten)          (None, 18816)             0

dense_2 (Dense)              (None, 512)               9634304

activation_1 (Activation)    (None, 512)               0

dense_3 (Dense)              (None, 26)                13338

=================================================================
Total params: 9,805,466
Trainable params: 9,805,466
Non-trainable params: 0
_________________________________________________________________
```

Fig: 3.2.4: Model summary using CNN

The model contains following layers

1. Convolution with 32 3x3 filters and ReLU activation function
2. Max Pooling
3. Convolution with 64 3x3 filters and ReLU activation function
4. Max Pooling
5. Convolution with 96 3x3 filters and ReLU activation function
6. Max Pooling
7. Convolution with 96 3x3 filters and ReLU activation function
8. Max Pooling
9. Flattening
10. Fully connected
11. ReLU
12. Fully connected + Softmax



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenet_1.00_224 (Functio  (None, 7, 7, 1024)        3228864
nal)

global_average_pooling2d (G  (None, 1024)              0
lobalAveragePooling2D)

dense (Dense)               (None, 64)                 65600

batch_normalization (BatchN  (None, 64)                256
ormalization)

dropout (Dropout)           (None, 64)                 0

dense_1 (Dense)             (None, 26)                 1690

=================================================================
Total params: 3,296,410
Trainable params: 3,274,394
Non-trainable params: 22,016
_________________________________________________________________
```

Fig: 3.2.5: Model summary using MobileNet

A base model is created at first using MobileNet with include_top=False for feature extraction. A 2-dimensional Global Average Pooling operation is next added, which entails averaging the feature maps from the previous convolutional layer to provide a single value for each feature map. This can then be fed into a fully connected layer. The output of a preceding layer is then normalised by adding Batch Normalization, which involves removing the batch mean and dividing by the batch standard deviation. After that, learnable parameters are used to scale and translate the normalised data. To avoid overfitting, the dropout layer is included. It happens when a model is too good at classifying training data, becoming overly specialised and unable to generalise to new, untried data. For picture categorization, a fully linked layer is inserted last.

CHAPTER - 4

RESULTS AND DISCUSSION

While building the model, we used thirty percent of the samples for testing and seventy percent of the samples for training. In all, we tested 26 classes using 5445 photos, and we trained 12705 images. We used a batch size of 20 and 5 epochs for both models. Therefore, each epoch has 636 iterations. We used the categorical cross-entropy loss function. Adam(optimizer) is used to lower the losses by adjusting the neural network's weights and learning rate. The accuracy of both models during training and evaluation is shown below.

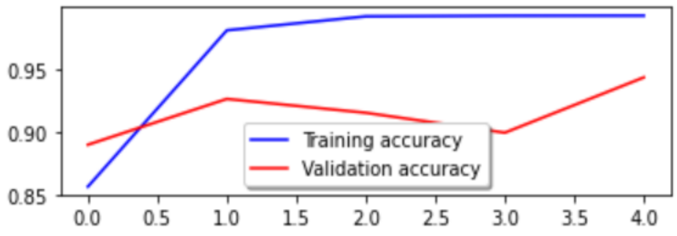


Fig: 4.1: Accuracy for CNN

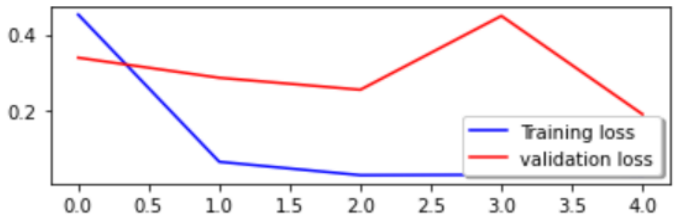


Fig: 4.2: Loss for CNN

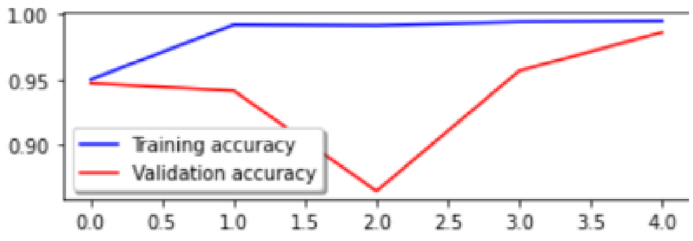


Fig: 4.3: Accuracy for MobileNet

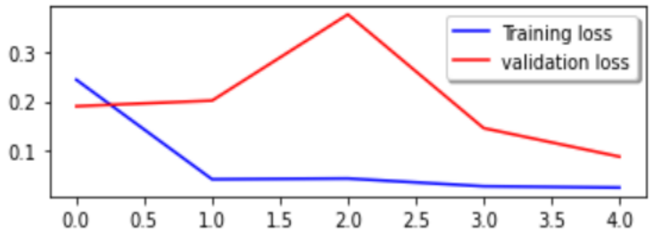


Fig: 4.4: Loss for MobileNet

As the saved model of MobileNet's model provides more accuracy(98 percent) than cnn(94 percent), we used it for live prediction.. During image uploading, the recognised sign is converted into both text and audio. An object for HandDetector is made for live prediction so that when the camera is opened, it will track the hand. The image was later cropped in order to find the ROI inside the specified limits. This hand that was identified is regarded as a test image. For live prediction on this test picture, we loaded our saved model and applied it.

Home page of Sign Language looks like below

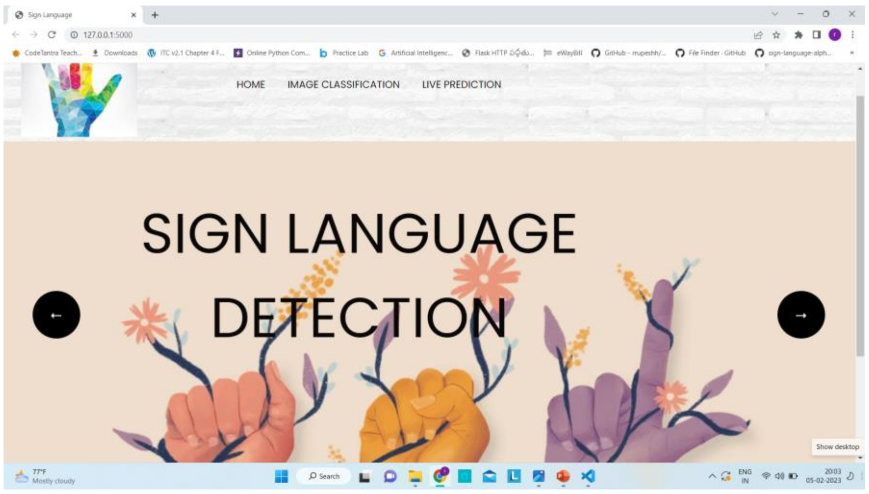


Fig: 4.5: Home Page

When we click on Image Classification, the following page will appear

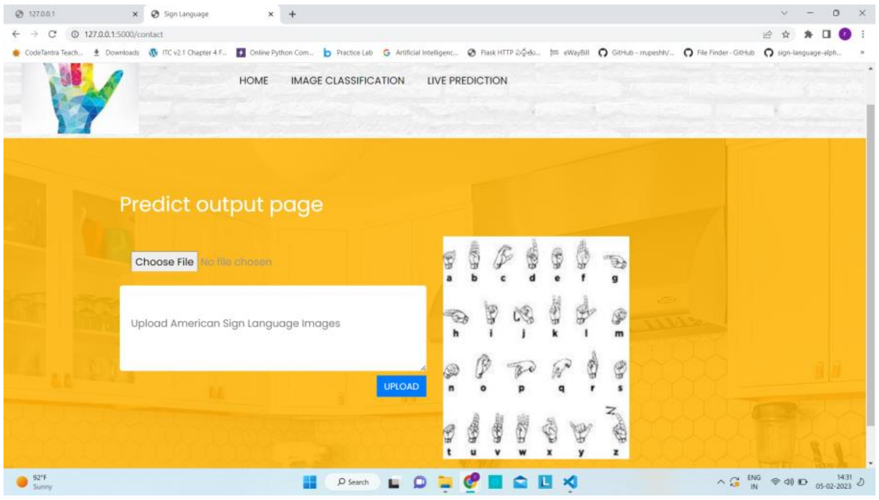


Fig: 4.6: Image uploading page

We have to choose the file and have to upload the image

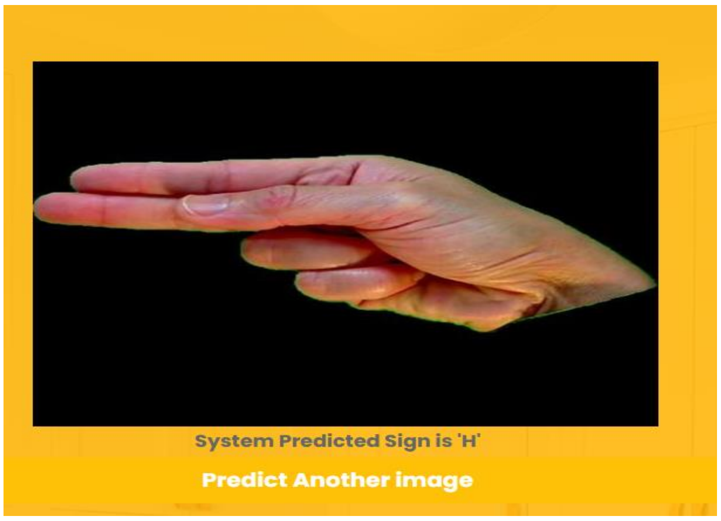


Fig: 4.7: Prediction of 'H' during image uploading

When we go for live prediction, the following window will appear

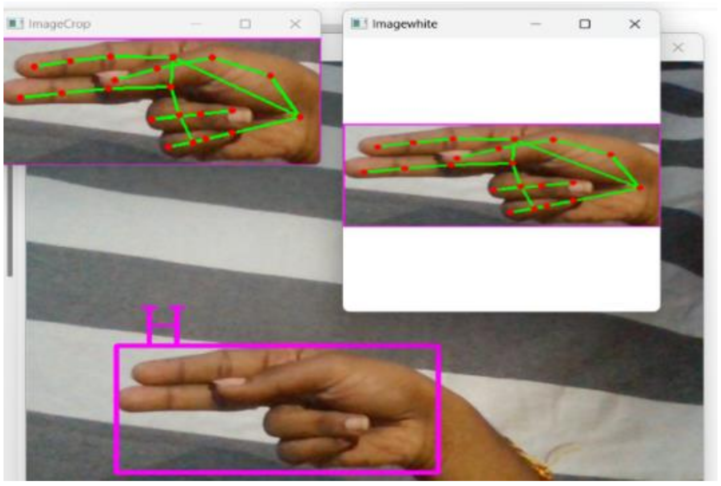


Fig: 4.8: Live prediction of 'H'

CHAPTER - 5
CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

Through the process of the whole project, we learned and found information about convolutional neural networks and image processing that we can use to classify images in the future. We used various built-in modules for our requirements. Based on the above results, we have observed that the pretrained model MobileNet of CNN is working more effectively than the CNN that we have created by adding layers to the sequential model with the help of keras module. We can identify the accuracy and loss of training and validation data for each epoch during the training phase. Finally, the model predicts the gesture, which is then shown on the screen through the provided graphical user interface. We come to the conclusion that using excellent lighting and a better camera for good performance can improve the prediction.

5.2 FUTURE SCOPE

The package was designed in such a way that future modifications can be done easily. This application avoids the mind work and the problems concern with it. Well, we have worked hard to provide a improved website together. Now, users can use it for ASL prediction, and it will be very helpful if it is improved a lot and adds additional features in the future.

BIBLIOGRAPHY

[1] D. Abdulla, S. Abdulla, R. Manaf and A. H. Jarndal, "Design and implementation of a sign-to-speech/text system for deaf and dumb people," 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, United Arab Emirates, 2016, pp. 1-4, doi: 10.1109/ICEDSA.2016.7818467.

[2] "Sign language recognition using sensor gloves," Proceedings of the 9th International Conference on Neural Information Processing, 2002, ICONIP '02., Singapore, 2002, pp. 2204-2206 vol.5, doi: 10.1109/ICONIP.2002.1201884. [3] T. Y. Ceiang and M. J. Hsiao, “Carry-select adder using single ripple carry adder,” *Electron. Lett.*, vol. 34, no. 22, pp. 2101–2103, Oct. 1998.

[3] Kaslay, Siddharth, Tejal Kesarkar, and Kanchan Shinde. "ASL Gesture Recognition using Various Feature Extraction Techniques and SVM." International Research Journal of Engineering and Technology (IRJET), vol. 7, no. 6, June 2020, pp. 3956. ISSN: 2395-0056.

[4] Apoorva M A, Harshitha M S, Chaitra S, Akshitha V, and Rajath A N. "An Efficient and Robust System for Hand Gesture Recognition and Interpretation." International Journal of Engineering Research & Technology (IJERT), vol. 6, no. 7, 2017, pp. 398-402. [5] J. M. Rabaey, *Digtal Integrated Circuits— A Design Perspective* Upper Saddle River, NJ: Prentice-Hall, 2001.

[5] Nagashree, R. N., Stafford Michahial, Aishwarya G. N., Beebi Hajira Azeez, Jayalakshmi M. R., and R Krupa Rani. "Hand Gesture Recognition using Support Vector Machine." The International Journal Of Engineering And Science (IJES), vol. 4, no. 6, June 2015, pp. 42-46. ISSN (e): 2319 – 1813, ISSN (p): 2319 – 1805.

[6] A. Saxena, D.K. Jain, and A. Singhal. "Sign Language Recognition Using Principal Component Analysis." Proceedings of the 2014 International Conference on Advances in Electronics, Computers and Communications (ICAECC), pp. 1-4, 2014.

[7] "Sign language identification," 2016 3rd International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, pp. 422-428, doi: 10.1109/RAIT.2016.7507939, by A. Kumar, K. Thankachan, and M. M. Dominic.

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru

Department of Computer Science and Engineering

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions., component, or software to meet the desired needs.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1 : Design, develop, test and maintain reliable software systems and intelligent systems.

PSO2 : Design and develop web sites, web apps and mobile apps.

PROJECT PROFORMA

Classification of Project	Application	Product	Research	Review
	√			

Note: Tick Appropriate category

Project Outcomes	
Course Outcome (CO1)	Identify and analyze the problem statement using prior technical knowledge in the domain of interest.
Course Outcome (CO2)	Design and develop engineering solutions to complex problems by employing systematic approach.
Course Outcome (CO3)	Examine ethical, environmental, legal and security issues during project implementation.
Course Outcome (CO4)	Prepare and present technical reports by utilizing different visualization tools and evaluation metrics.

Mapping Table

CS1537 : MAIN PROJECT															
Course Outcomes	Program Outcomes and Program Specific Outcome														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12		PSO 1	PSO 2
CO1	3	3	1					2	2	2				1	1
CO2	3	3	3	3	3			2	2	2		1		3	3
CO3	2	2	3	2	2	3	3	3	2	2	2			3	
CO4	2		1		3				3	3	2	2		2	2

Note: Map each project outcomes with POs and PSOs with either 1 or 2 or 3 based on level of mapping as follows:

1-Slightly (Low) mapped 2-Moderately (Medium) mapped 3-Substantially (High) mapped