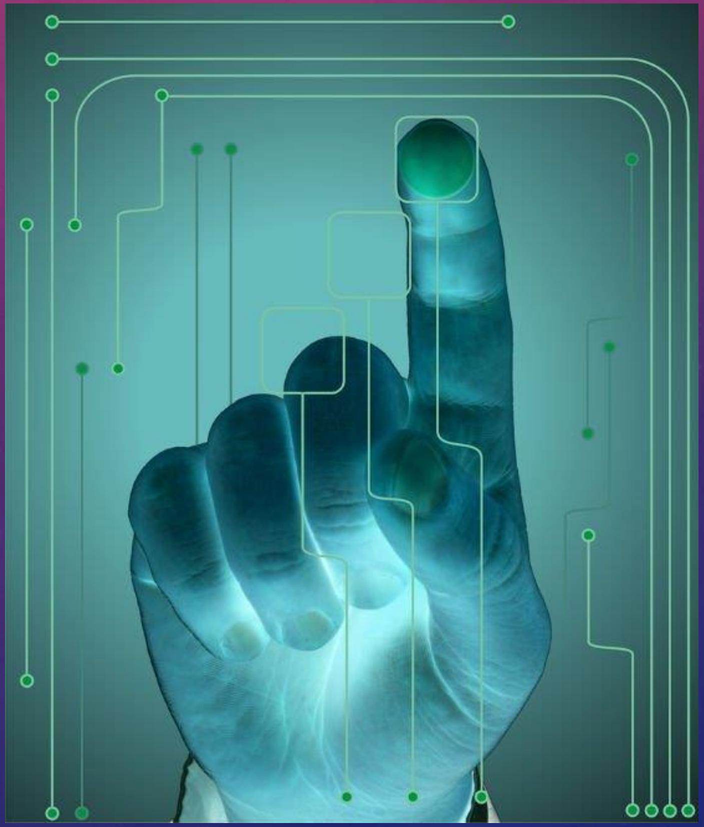
HAND GESTURE RECOGNITION



UNDER THE GUIDANCE OF

Mr. V Ramakrishna Sajja

Asst.Professor

BY BATCH B-9

181FA04027-KP BHASKAR

181FA04465-T NIKHIL TEJA

181FA04488-U VASU

CONTENTS

- Abstract
- Introduction
- Literature Review
- Proposed Methods
- Dataset
- Implemented Methods
- Experimental Results
- Software Requirements
- Conclusion
- References

ABSTRACT

Hand gestures are a non-verbal mode of communication that we witness in everyday life while interacting with one another. A gesture is described as a physical movement of the hands, fingers, arms, and other elements of the human body that allows people to communicate meaningful information to one another. The data gloves method and the vision-based approach are two methodologies for human–computer interactions. The vision-based technique was tested in the following trials, which included hand gesture detection and classification. A Hand gestures are one of the most reasonable methods to create a user-friendly and adaptable interface between gadgets and people. In Human Computer Interaction (HCI) systems, applications such as virtual object manipulation, gaming, and gesture detection are employed.

Hand gestures are the most effective communications tool and the most commonly employed idea in a gesture recognition system. Hand gestures may be recognized using one of two techniques: posture (a static hand form ratio without hand movements) or gesture (a dynamic hand motion with or without hand motions). Using any form of camera will detect any form of hand motion; however, various cameras will provide varying resolution quality. Most finger movement may be detected by two-dimensional cameras in a continuous surface known as 2D. In our project, we will be using Deep Learning models and certain pretrained models to recognize dynamic gestures

INTRODUCTION

Sign language is a language that employs signs made by moving the hands combined with facial expressions and postures of the body. It is one of several communication options used by people who are deaf or hard-of-hearing. Gesture language identification is one of the areas being explored to help the deaf integrate into the community and has high applicability. The essential aim of building hand gesture recognition system is to create a natural interaction between human and computer where the recognized gestures can be used for controlling a robot or conveying meaningful information. Specifically, we use the convolutional neural network (CNN) to recognize gestures and makes it attainable to identify relatively complex gestures.

LITERATURE REVIEW

The design of any gesture recognition system essentially involves the following three aspects:

(1) data acquisition and pre-processing;

(2) data representation or feature extraction and

(3) classification or decision-making

S.NO	Title	Journal	Dataset	Techniques	Performance
1	Hand gesture recognition using a neural network shape fitting technique	Journal of Image and Graphics	100 Training images, chrominance components of skin color 50 images	Artificial Neural Network (ANN) based on a form fitting methodology.	94.05% Accuracy
2	Static hand gesture recognition using artificial neural network	Journal of Image and Graphics	94 Persons, 20 Hand Gestures, 3 Backgrounds	ANN characteristics used, pixel changes through cross sections, boundaries, and scalar descriptions such as aspect ratio and edge ratio	98% Accuracy
3	Hand gesture recognition using haar-like features and a stochastic context-free grammar	IEEE Xplore	1712 Postures	AdaBoost, Context Free Grammar, Haar like features	97.5%, 98%, 97.1% for palm, fist, little finger

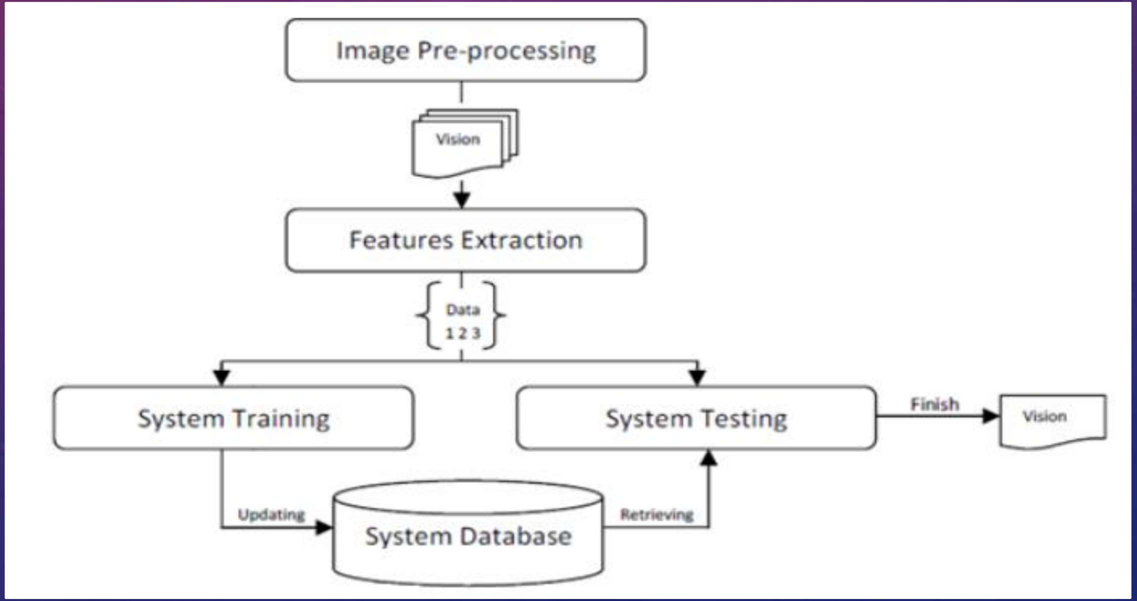
S.NO	Title	Journal	Dataset	Techniques	Performance
4	Static Hand Gesture Recognition using Convolutional Neural Network with Data Augmentation	IEEE Xplore	8000 Training samples, 1600 testing samples	Convolutional Neural Networks (CNN), Stochastic Gradient Descent as an optimizer function	92.87% without augmenting, 97.12% with augmenting
5	Hand Gesture Recognition for Human Computer Interaction	Arxiv	19,852 Samples, 16 Gestures, 200 for testing	Human Computer Interaction, OpenCV	20-30% for large input images, 99.8% for small size of input image

PROPOSED METHODS

The proposed model architecture can be divided into two different modules named feature extraction, model training & evaluation

Feature extraction: During this phase, most significant features of gesture images using various pre-trained models will be extracted

Model training and Evaluation: During this phase, we train the model with pre-trained features and hybrid features for comparative studies. We apply trained model on test data, evaluate it's performance.



In our project, we will be using Deep Learning models such as Convolutional Neural Networks (CNN) and certain pretrained models such as VGG-16, Inception V3, and OpenCV Functions to recognize dynamic gestures

Computer Vision for Hand Gesture Recognition

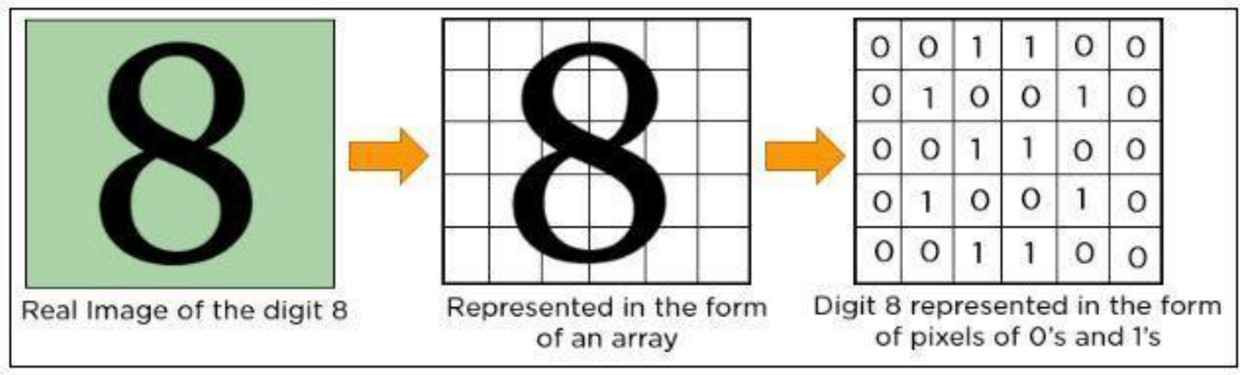
OpenCv is a widely used tool in computer vision. It is a computer vision library for real-time applications, written in C and C++, which works with the Windows, Linux and Mac platforms. It is freely available as opensource software.

A digital image is generally understood as a discrete number of light intensities captured by a device such as a camera and organized into a two-dimentional matrix of picture elements or pixels, each of which may be represented by number and all of which may be stored in a particular file format (such as jpg or gif)

We will be using CNN which performs both Feature Extraction and Classification followed by Transfer Learning scheme and some popular pre-training Models that we use for Feature extraction. Finally, Deep Neural Network Architecture for Hand Gesture Recognition.

Convolutional Neural Network

A convolutional neural network is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. It’s also known as a ConvNet. A convolutional neural network is used to detect and classify objects in an image. In CNN, every image is represented in the form of an array of pixel values



Real Image of the digit 8

Represented in the form of an array

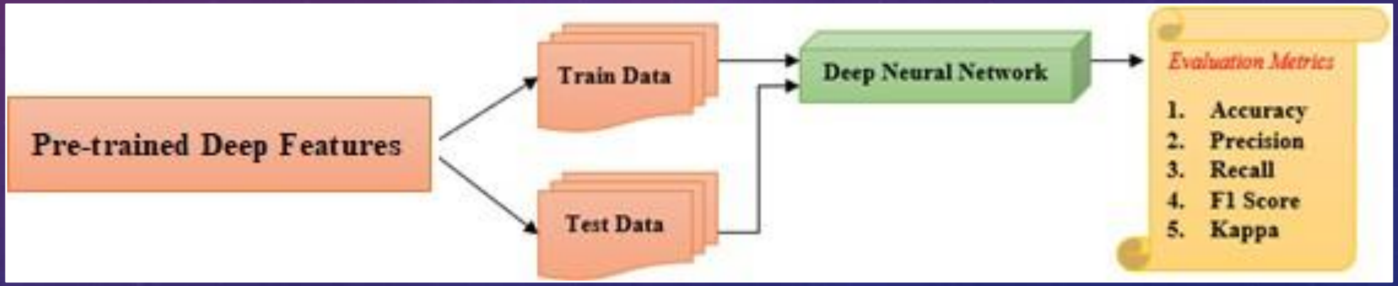
0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	1	0	0

Digit 8 represented in the form of pixels of 0’s and 1’s

Transfer Learning

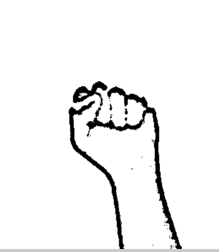
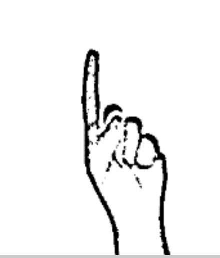
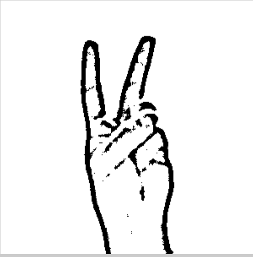
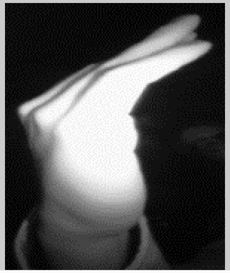
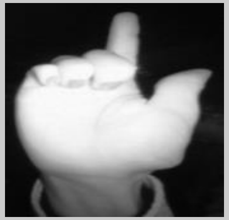
In computer vision, transfer learning is usually expressed using **pre-trained models**. A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem like the one that we want to solve. Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature (e.g. VGG, Inception)

VGG 16 — VGG16 is a pre-trained model trained on 14 million image dataset belonging to 1000 different classes in ILSVR (ImageNet) challenge.



DATASET

Source	Size	Type
Kaggle	Nearly 14000 samples	Near IR images under variety of imaging conditions

None	One	Two	C	I
				

IMPLEMENTATION USING CNN

We have imported all the necessary modules and layers from keras. The modules that we have imported are CV2, PyQt5, Numpy, Keras, Matplotlib, Tensorflow. We have implemented CNN by implementing sequential model. The sequential model creates an empty stack to add any number of convolutional layers we need. We have implemented 8 Convolutional layers and 6 Max Pooling Layers and Fully connected layer and a dropout layer to help out incase of overfitting and finally an output layer consists of softmax function and number of classes as parameters.

The input size of an image is 256*256*1 as the image is Gray scale image and thus we need only one channel, we have used ReLu as an activation function in all the hidden layers except output layer. The Max Pooling layer is used to reduce the size of the image so that we can have less number of computations and speedup the training process. The Dropout Layer is used inorder reduce the chances of overfitting.

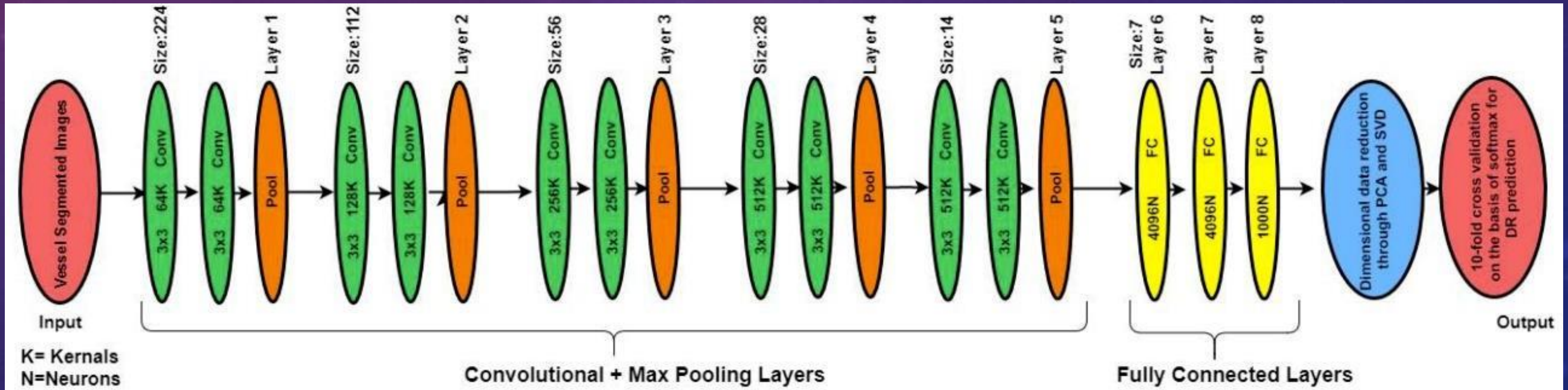
We have used Adam as an Optimizer and Categorical_Crossentropy as a loss function and Accuracy as a metrics to compile the designed CNN Model.

DATA AUGMENTATION FOR CNN

We have used Data Augmentation for CNN Model as the data which we have has limited number of datasets but these data cannot be sufficient for training and detecting the features from the gestures, if we have performed data augmentation to the data then we can have different variations of data that can easily fit the model and easily train the model to obtain better features. We have used callbacks to monitor the validation loss and it will be helpful to save the model if there were constant results for over a certain period of epochs, which we call it as patience. If the patience is set to 10 then it will monitor the validation loss and if there is no changes in the loss value for continuous 10 epochs then it will stop the training process and saves the model to disk. This helps us saving lot of computations and memory.

IMPLEMENTATION OF VGG-16 (PRE-TRAINED MODEL)

Her we have used VGG-16 as a Pre-Trained Model and imported all the necessary modules of VGG-16 from Keras.applications. The parameters for the VGG-16 are image size, weights, include_top. The image size for normal VGG-16 is 224*224*3 but we have only Gray Scale images so we have resized them to 256*256*1 and for a normal VGG-16 Model the weights are assigned from the imagenet but we have different classes in our dataset to we have given to None and include_top attribute is set to False as we don’t need the output layers from VGG-16 and we want to change those layers according to our problem statement. We are not even training all the layers of the VGG-16 and we have set the layers.trainable to False and we will train only few layers which are required and set the remaining layers to freeze.

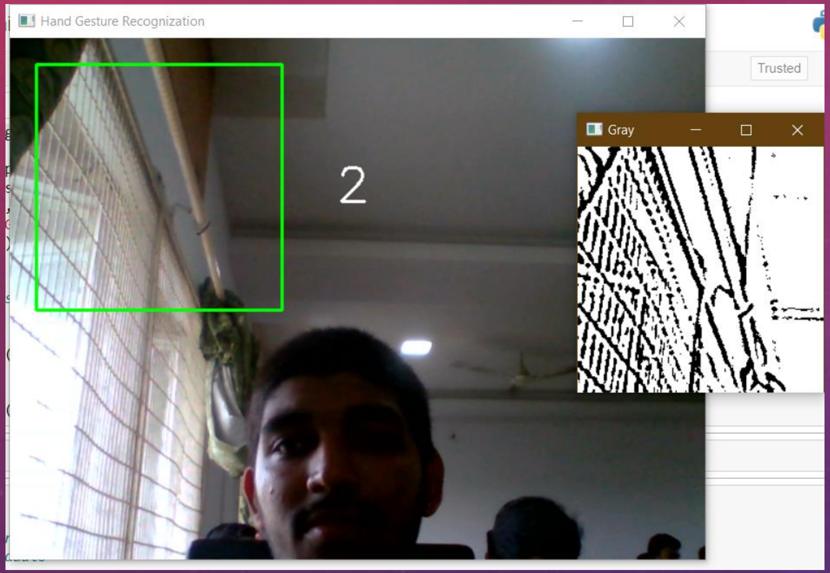


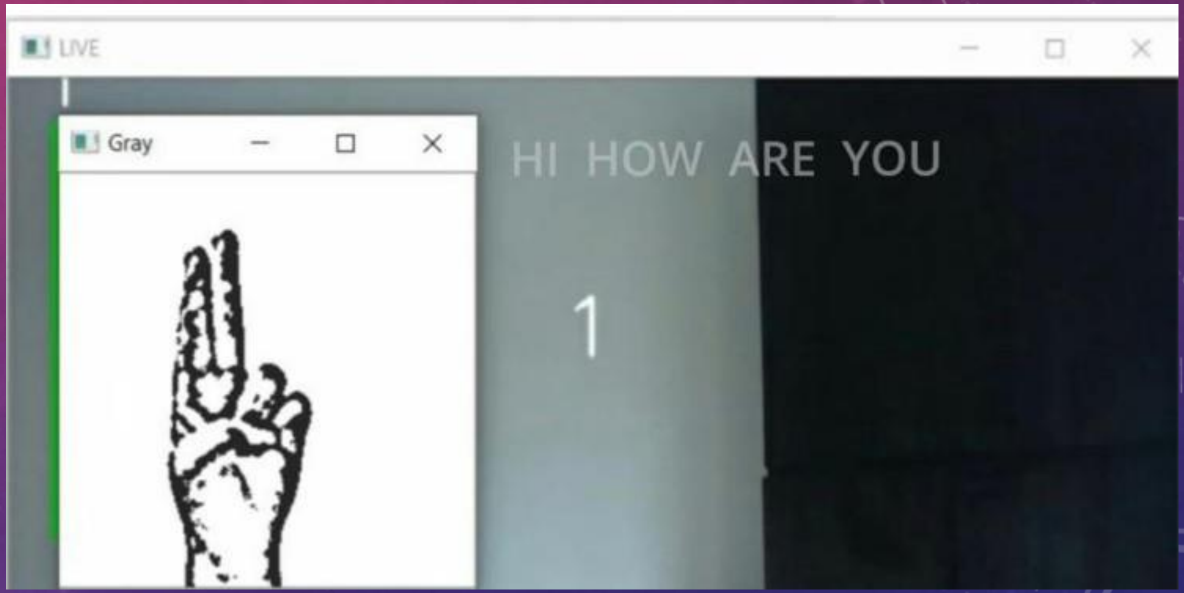
IMPLEMENTATION OF OPENCV AND GUI FUNCTIONS

Here we have imported all the necessary modules from CV2. The VideoCapture() is used to start the camera to trace the hand and draw the rectangle over the detected range. We have converted the Colored image to Gray scale image using cvtcolor() function and then we have applied different Blurring and Thresholding functions to preprocess the image to identify the gestures. Here we have used findContours function to identify the contours and to connect those contours. The ConvexHull function is used to identify the hull in the gesture and the defects in the convex Hull are known as convexity Defects that are used to classify the type of gesture. We have used some mathematical functions to identify the angle between the defects.

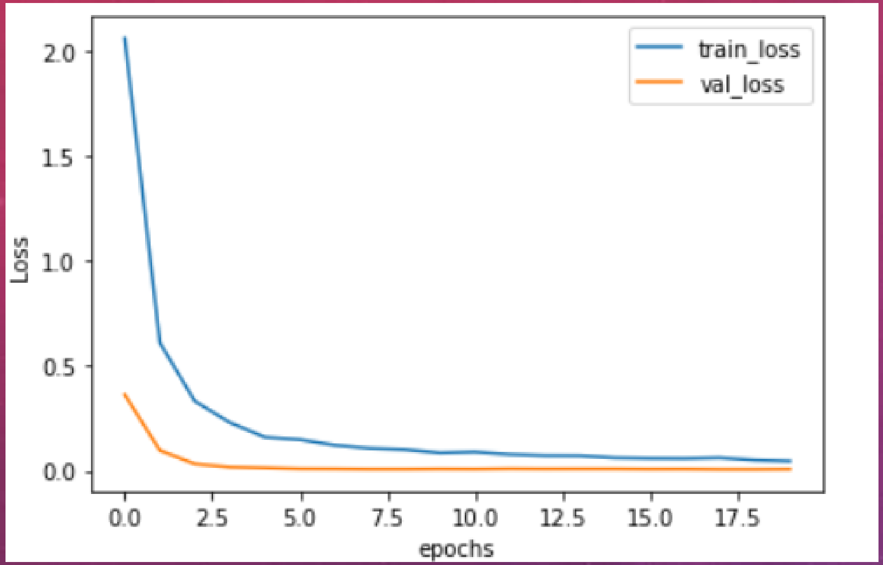
Here we have loaded the model which was trained and using those model weights we are predicting the images that are given as input to the classify function. To classify the image we have used model.predict() fuction.

EXPERIMENTAL RESULTS

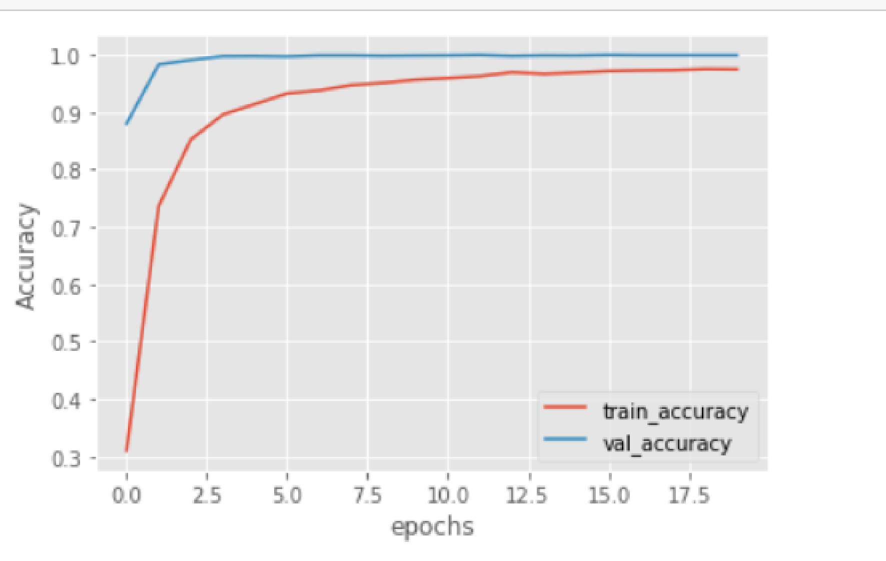




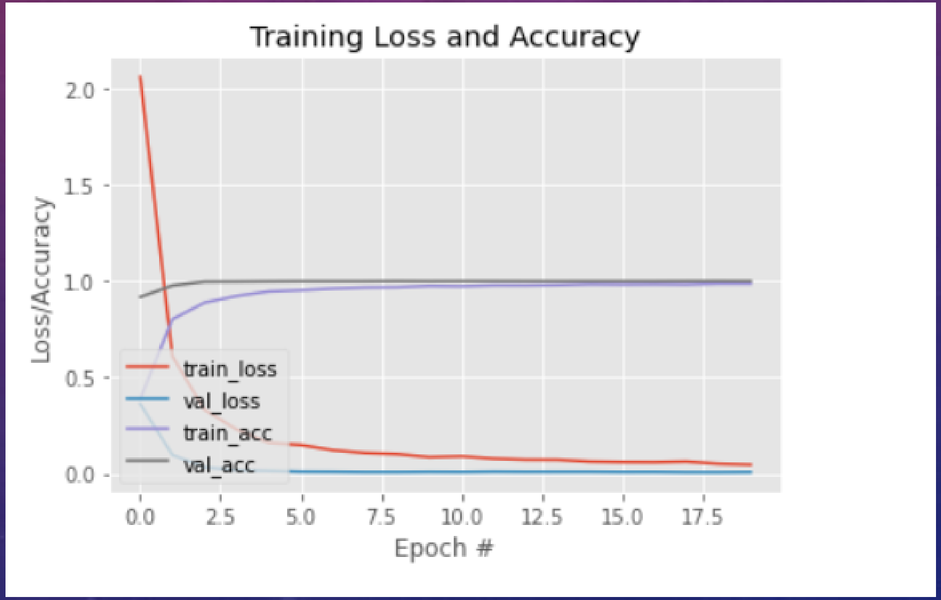
PLOTS



train_loss
val_loss
Loss
2.0
1.5
1.0
0.5
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
epochs



Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
train_accuracy
val_accuracy
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
epochs



Training Loss and Accuracy
Loss/Accuracy
2.0
1.5
1.0
0.5
0.0
train_loss
val_loss
train_acc
val_acc
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch #

SOFTWARE REQUIREMENTS

- Operating System : Windows 10
- Programming Language : Python 3.8
- Deep Learning Framework : TensorFlow 1.14 & Keras 2.3
- GUI Framework : PyQt5
- Supporting Libraries : Pandas, NumPy, matplotlib, sklearn

CONCLUSION

The objective of this work is to develop a Hand Gesture Recognition model from the given distinct gestures. Major features of the proposed model are simple and robust. The model we designed extracts the deep features from pre- trained models like VGG16 and CNN to represent the more prominent features from the Gesture images and these are used for training a neural network. To avoid over-fitting of the model we introduced dropout at the ending of the neural network. The power of deep learning and simplicity of the models make the model more robust with minimum number of misclassifications.

REFERENCES

1. E. Stergiopoulou and N. Papamarkos, “Hand gesture recognition using a neural network shape fitting technique,” Engineering Applications of Artificial Intelligence, vol. 22, no. 8, pp. 1141–1158, 2009
2. Pei Xu, “A Real-time Hand Gesture Recognition and Human-Computer Interaction System”.
3. T.-N. Nguyen, H.-H. Huynh, and J. Meunier, “Static hand gesture recognition using artificial neural network,” Journal of Image and Graphics, vol. 1, no. 1, pp. 34–38, 2013.
4. Q. Chen, N. D. Georganas, and E. M. Petriu, “Hand gesture recognition using haar-like features and a stochastic context-free grammar,” IEEE transactions on instrumentation and measurement, vol. 57, no. 8, pp. 1562–1571, 2008.
5. Md. Zahirul Islam, Mohammad Shahadat Hossain, Raihan Ul Islam, Karl Andersson, “Static Hand Gesture Recognition using Convolutional Neural Network with Data Augmentation”

THANK YOU