GESTURE RECOGNITION USING PYTORCH

Catherine Mariana Philips

TABLE OF CONTENTS

SYNOPSIS

To recognize the gestures and actions performed by various users over a real-time stream. Actions performed will be collected as images, and Deep Learning models will be used to train the collected data. The model will be tested over a real-time stream to identify the gestures.

INTRODUCTION

- We implement various deep learning models to recognize gestures and actions performed by multiple users over a real-time camera stream. Initially, gestures performed by users are collected as JPG image files. These collected images are loaded as input to three convolutional neural networks. Based of few metrics, the appropriate model is saved and tested over a real-time camera stream to recognize the gestures.
- The packages used in this project include
  1. os- for directory operations
  2. time- for time-based image capturing
  3. PyTorch(torch)- for tensor computation
  4. OpenCV(cv2)- for image processing and computer vision tasks
  5. Torchvision- for common image transformations
  6. torch.nn- help in creating and training neural networks

LITERATURE REVIEW

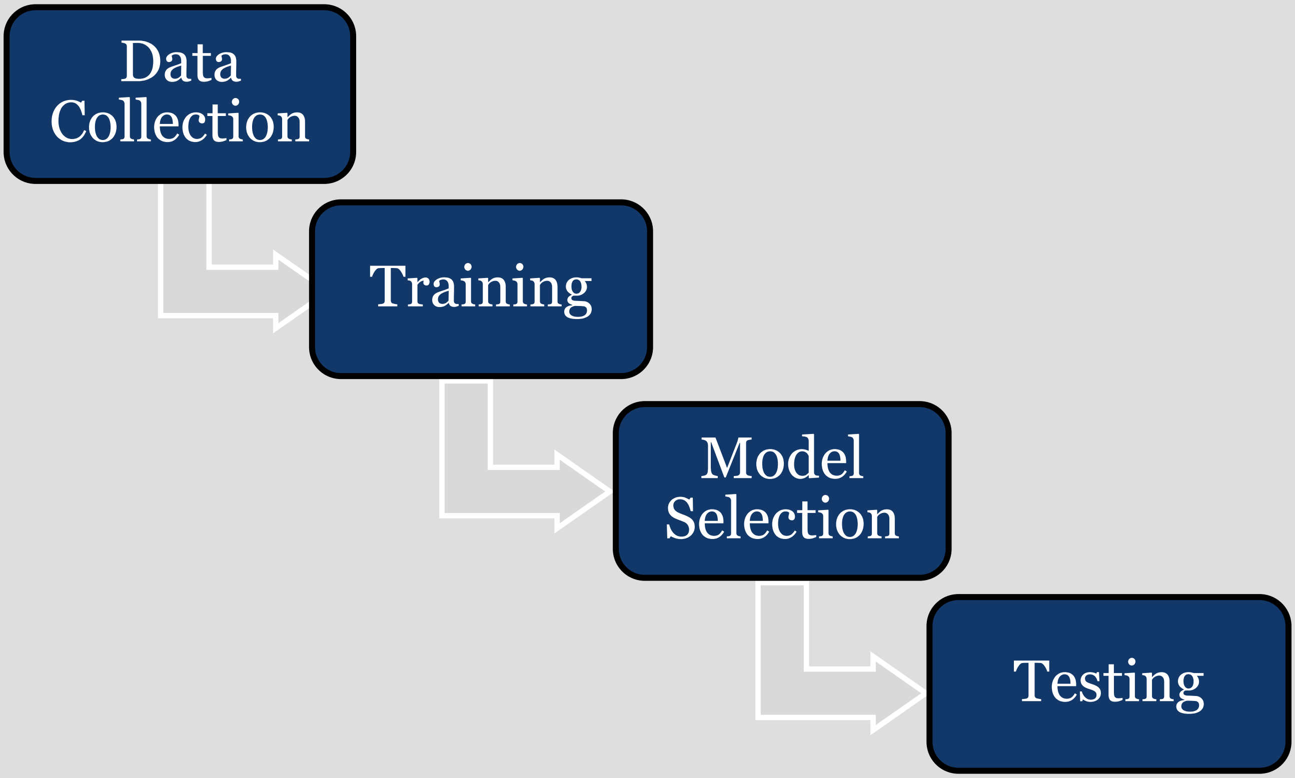
- In order to offer create chances to design intuitive interactions with computing machines, the authors Soeb Hussain ,Rupal Saxena, Xie Han, Jameel Ahmed Khan, and Hyunchul Shin researched an idea of the automatic interpretation of gestures based on computer vision

- Nicholas Renotte did a project on Object detection where he first captured the images using a time-based image capturing method. The images were labelled using a labelimg package. The tensorflow ZOO model was trained. The model was then used in two ways; implementing the model in a web application and using the slide view a microscope to detect the organism.

- Research on a Look Based Media Player with Hand Gesture Recognition was done by Saritha Niranjanrai in 2021. The idea is based on creating a media player that pauses itself when the user is not looking at it. When the user glances at the player, it resumes. The camera is used for this. Creation of an advanced music player that automatically plays and pauses videos based on facial recognition, and moves forward and backward by recognising hand gestures is being thought. The system monitors whether the user is looking at the screen or not using Haar-cascade Classifier.

- Controlling other functions of a media player is done using Convolutional Neural Networks.

PROCEDURE FOLLOWED

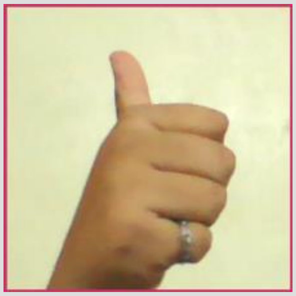


1)DATA COLLECTION

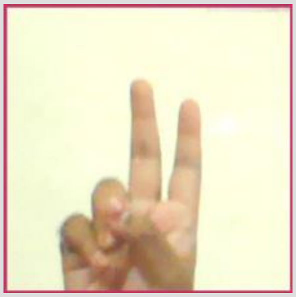
- We intend to capture frames of a video as images.
- Using the code, the webcam is accessed, and a rectangular frame, which will be used as a region of interest(roi).
- The gestures and actions performed inside the roi are captured and saved in the respective gesture folders that are in the train and validate folders.
- A total of 200 images are captured for the training set and 100 images for the validation set. Each image is of the size 224X224.
- The images are captured in front of a plain background.

- The training and validation sets consist of images for the four gestures 'play', 'pause', 'stop', and 'mute'.
- Additionally, images with different objects are captured as 'none'.

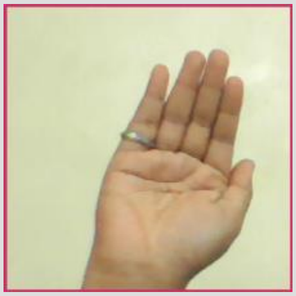
The following images are captured.



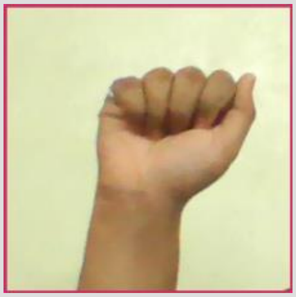
1:PLAY



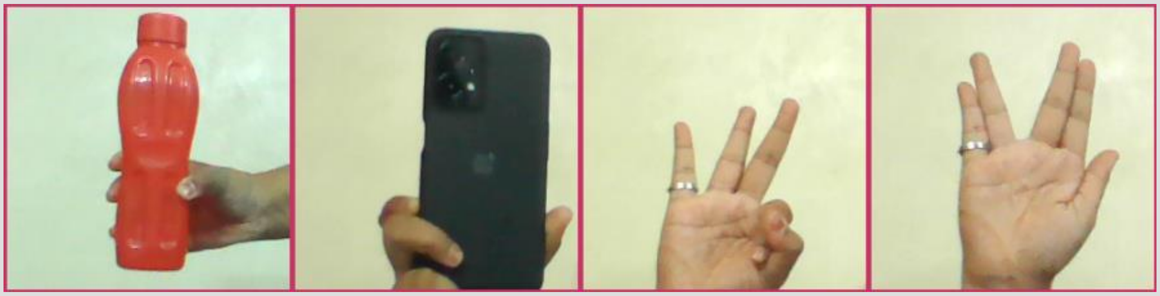
2:PAUSE



3:STOP



4:MUTE



5:NONE

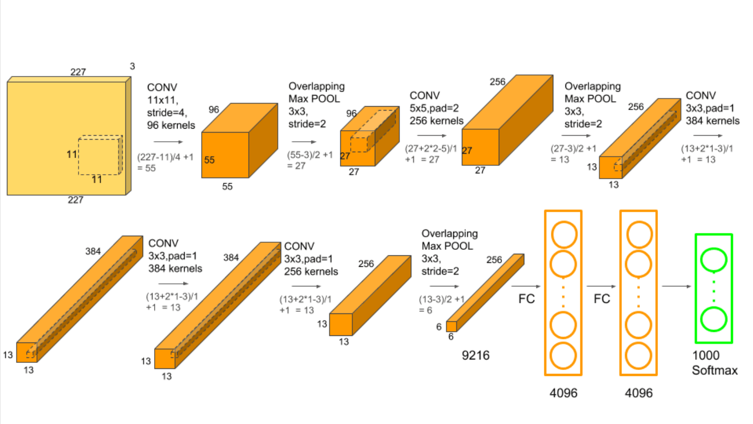
2)TRAINING

Training Method

- For both the training and validation sets, the images are sent as batches of 32.
- The required image transformations are used to normalize the data and convert them into tensors.
- Three CNN models namely Resnet18,Alexnet, and Googlenet are used and the images are given as an input to these models and trained over 5 to 6 epochs
- For each model, loss and accuracy are found for bit the train and validation set.
- The three models are saved and stored as a .pth file

About the models

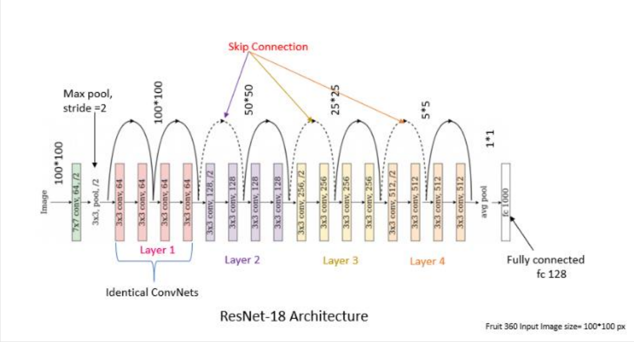
a) Alexnet



- When it comes to pre-trained models in the field of computer vision, Alexnet stands out as a top architecture.
- Compared to Lenet-5, the depth of the network was enhanced in this model.
- There are eight learnable layers in the Alexnet.
- Relu activation is used in each of the five levels of the model, with the exception of the output layer, which uses max pooling followed by three fully connected layers.
- The Imagenet dataset is used to train the model.
- There are almost a thousand classes and nearly 14 million photos in the Imagenet collection.

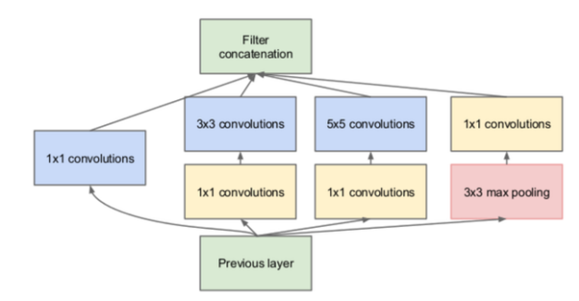
b) Resnet18

- ResNet-18 is a convolutional neural network that has 18 layers.
- The ImageNet database contains a pretrained version of the network that has been trained on more than a million images.
- The pretrained network can classify photos into 1000 different object categories, including several animals, a keyboard, a mouse, and a pencil.
- The network has therefore acquired rich feature representations for a variety of images.
- The network accepts images with a resolution of 224 by 224.



c) Googlenet

- Google's research team proposed Google Net, also known as Inception V1.
- In comparison to earlier state-of-the-art architectures like AlexNet and ZF-Net, Google's architecture is quite distinctive.
- It employs a variety of techniques, including global average pooling and 1-1 convolution, to build deeper architecture.
- There are 22 layers in the architecture as a whole.
- The architecture was created with consideration for computational effectiveness. the concept that even with limited processing resources, the architecture can be used on individual devices.



3)MODEL SELECTION

Training and Validation Accuracy

After the models are trained, we get the following training and validation accuracies.

MODEL NAME	TRAIN ACCURACY	VALIDATION ACCURACY
Alexnet	0.9270	0.9939
Resnet18	0.9790	0. 9959
Googlenet	0.9880	1.0000

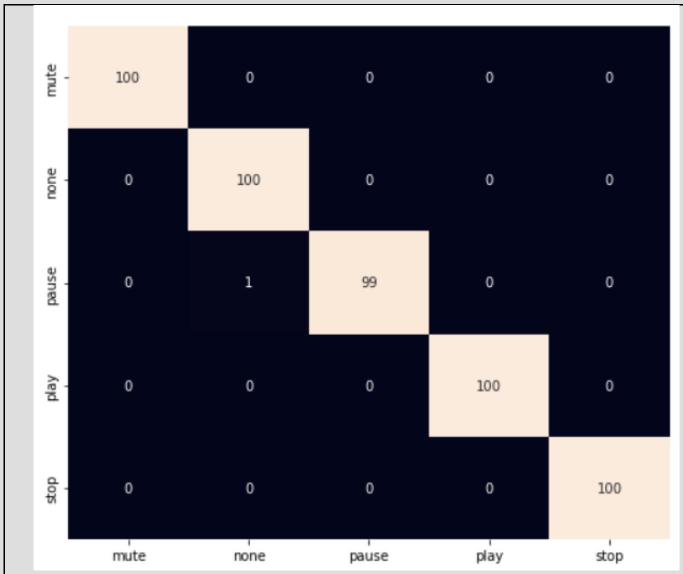
- From the above table, it can be noted that googlenet has a validation accuracy of 1. This could mean, there exists a sort of overfitting.
- For Resnet18, the validation accuracy is 0.9939 and the difference between the train and validation accuracy is small.
- For Alexnet, the validation accuracy is 0.9959 and the difference between the train and validation accuracy is very minute.
- We choose Alexnet and Resnet18 and further investigate which is the best model to be used.

Further Study

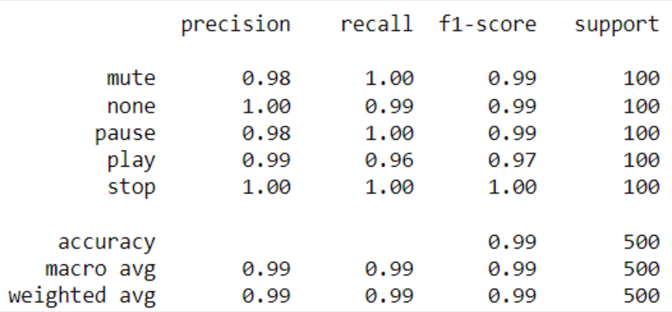
- The class wise validation accuracy is found for both the models.
- A confusion matrix is plotted to check for misclassified samples.
- A classification report is also obtained.

ALEXNET
Accuracy of mute: 100%
Accuracy of none: 99%
Accuracy of pause: 100%
Accuracy of play: 99%
Accuracy of stop: 100%

Class wise validation accuracy



CONFUSION MATRIX

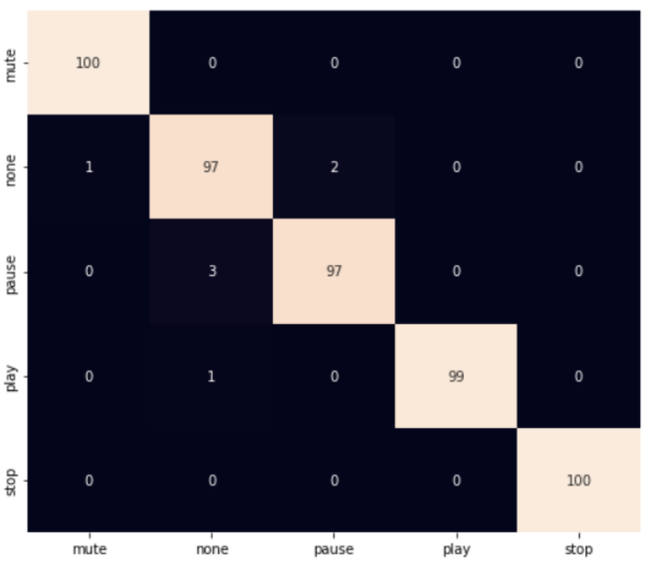


```
              precision    recall  f1-score   support

        mute       0.98      1.00      0.99       100
        none       1.00      0.99      0.99       100
       pause       0.98      1.00      0.99       100
        play       0.99      0.96      0.97       100
        stop       1.00      1.00      1.00       100

    accuracy                           0.99       500
   macro avg       0.99      0.99      0.99       500
weighted avg       0.99      0.99      0.99       500
```

CLASSIFICATION REPORT

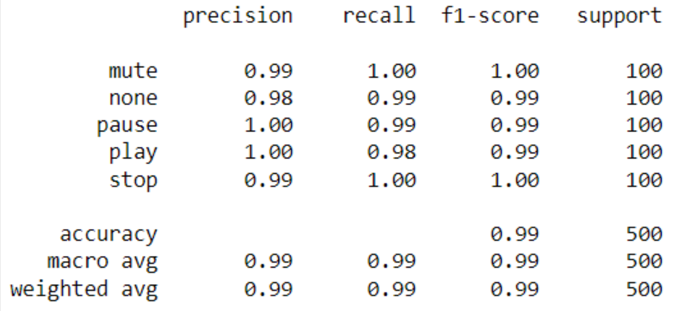
- From the confusion matrix, it can be noted that the gestures mute, none, play, and stop have been classified perfectly while pause has one misclassified sample.
- From the classification report, we can understand the various factors such as precision, recall, f1-score, and support.
- Precision shows the precise percentage of correct predictions.
- Recall tells the labels correctly labelled.
- The f1-score tells the percentage of labels correctly labelled.
- Support denotes the number of actual occurrences of the class in the dataset.

RESNET18
Accuracy of mute: 100%
Accuracy of none: 98%
Accuracy of pause: 99%
Accuracy of play: 98%
Accuracy of stop: 100%



	mute	none	pause	play	stop
mute	100	0	0	0	0
none	1	97	2	0	0
pause	0	3	97	0	0
play	0	1	0	99	0
stop	0	0	0	0	100

CONFUSION MATRIX



```
              precision    recall  f1-score   support

        mute       0.99      1.00      1.00       100
        none       0.98      0.99      0.99       100
       pause       1.00      0.99      0.99       100
        play       1.00      0.98      0.99       100
        stop       0.99      1.00      1.00       100

    accuracy                           0.99       500
   macro avg       0.99      0.99      0.99       500
weighted avg       0.99      0.99      0.99       500
```

CLASSIFICATION REPORT

- From the confusion matrix, it can be noted that the gestures mute, and stop have been classified perfectly while none, and pause have three misclassified samples, and play has one misclassified sample.

OBSERVATIONS	
Alexnet	**Resnet18**
Validation accuracy:0.9959	Validation accuracy:0.9939
Only one sample has been misclassified	A total of 7 samples have been misclassified

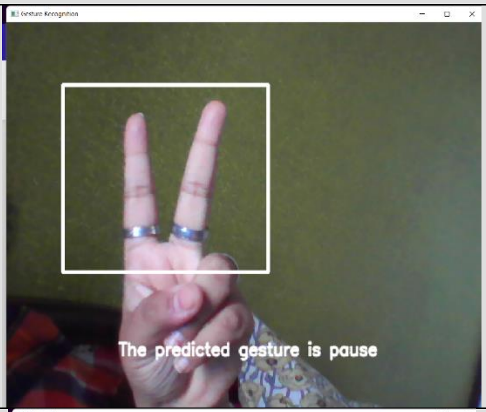
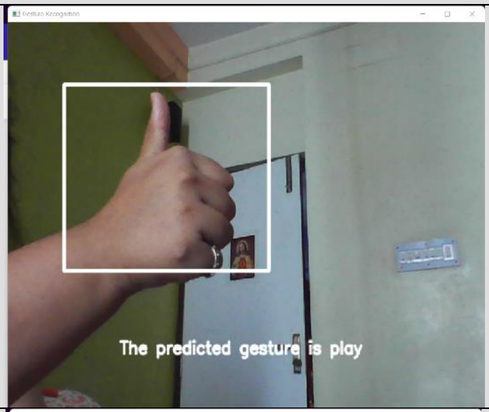
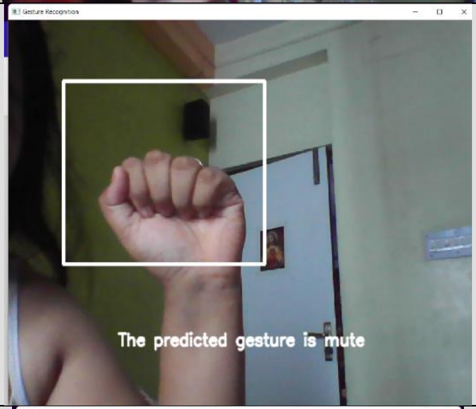
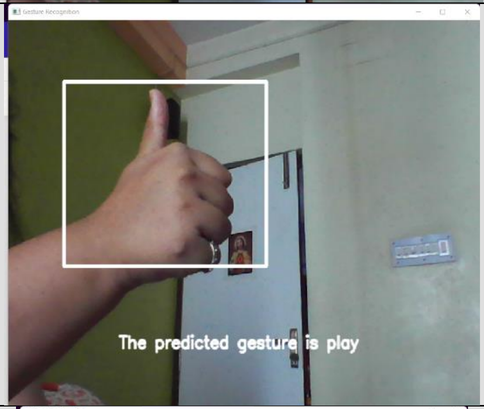
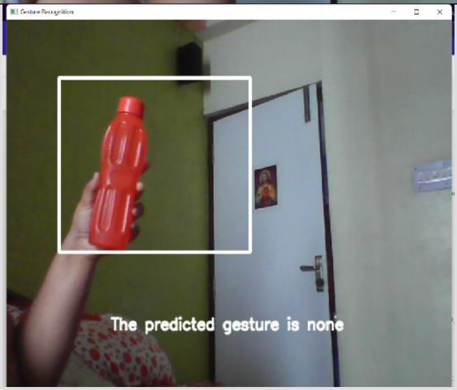
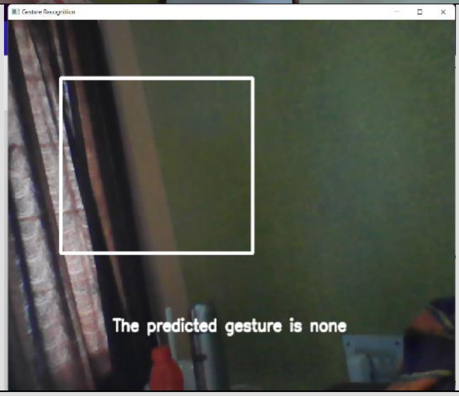
Taking into account the accuracy and the number of misclassified samples, we can understand that the "**Alexnet**" model has a better accuracy and lower sample misclassification.

Hence, we **select the Alexnet model** to perform testing.

4)TESTING

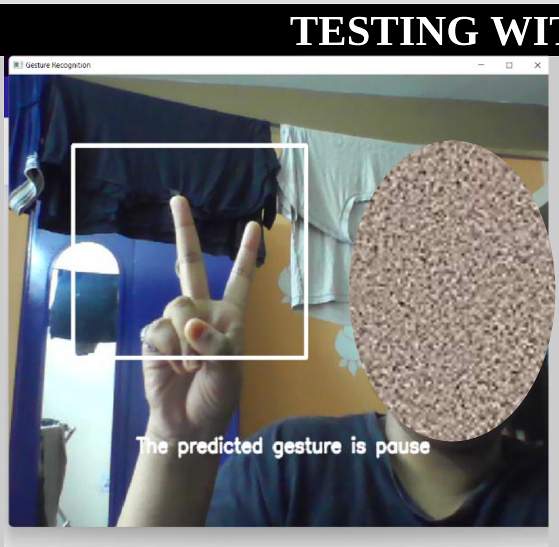
- The stored Alexnet model is loaded.
- A webcam stream with a rectangular region of interest is opened, which is similar to the image capturing process.
- The model recognizes the gestures performed in the frame and displays the name of the gesture on the camera stream window.

Initially, testing is done to check whether the model identifies the gestures correctly.

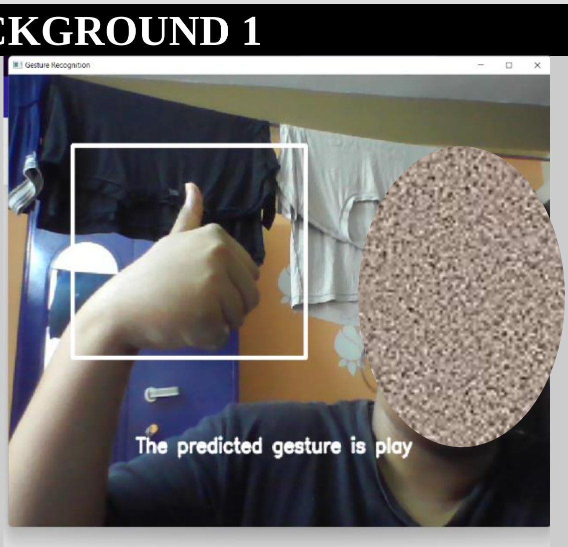
 The predicted gesture is pause	 The predicted gesture is play
 The predicted gesture is mute	 The predicted gesture is play
 The predicted gesture is none	 The predicted gesture is none

Testing has been done in 4 ways; 3 with different backgrounds and 1 with a different user.
The different testing done are shown below.

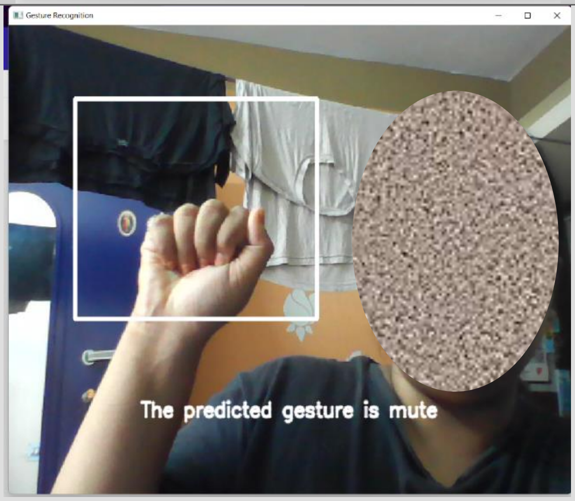
TESTING WITH BACKGROUND 1



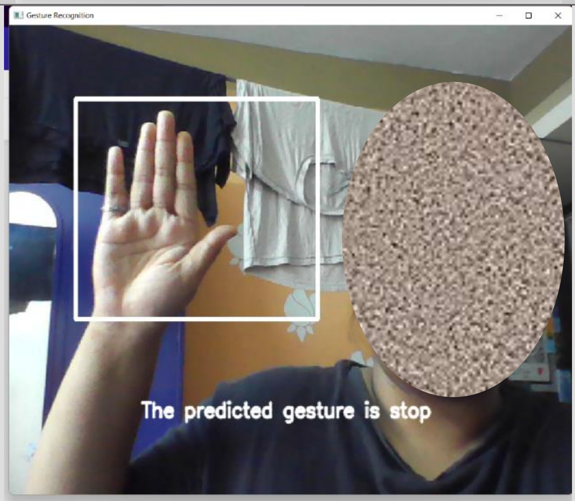
The predicted gesture is pause



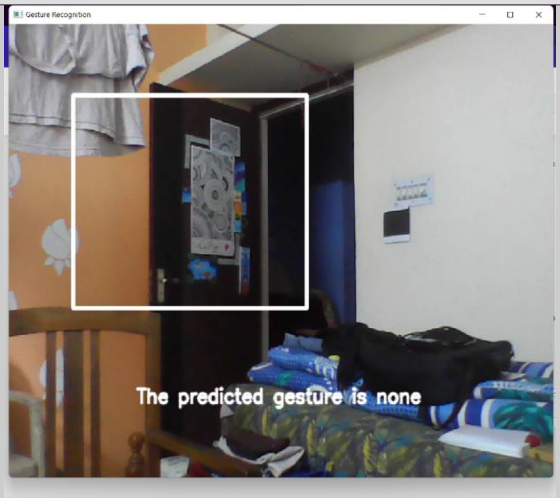
The predicted gesture is play



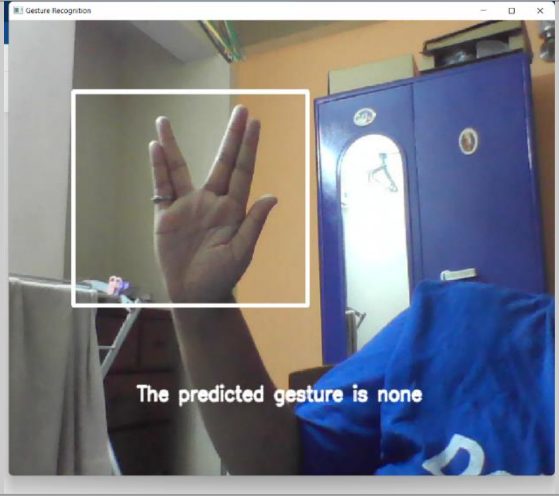
The predicted gesture is mute



The predicted gesture is stop

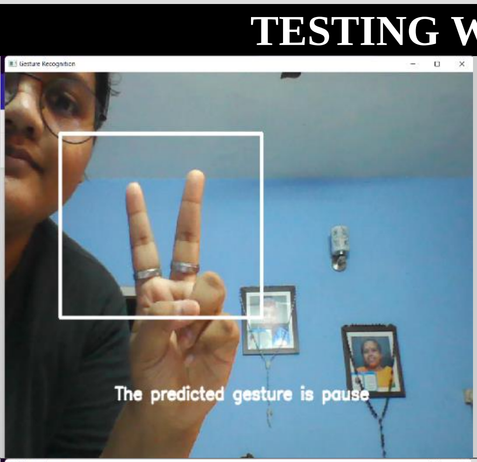


The predicted gesture is none

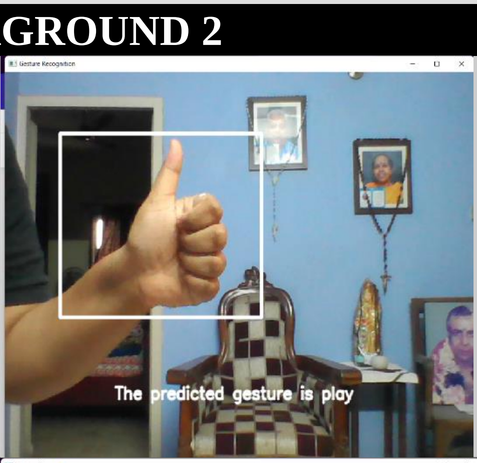


The predicted gesture is none

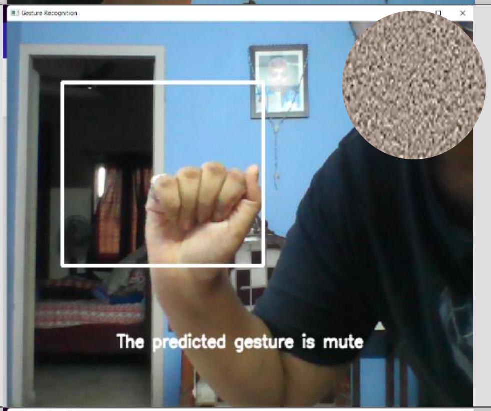
TESTING WITH BACKGROUND 2



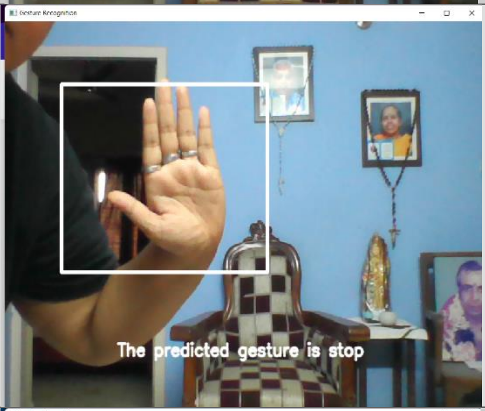
The predicted gesture is pause



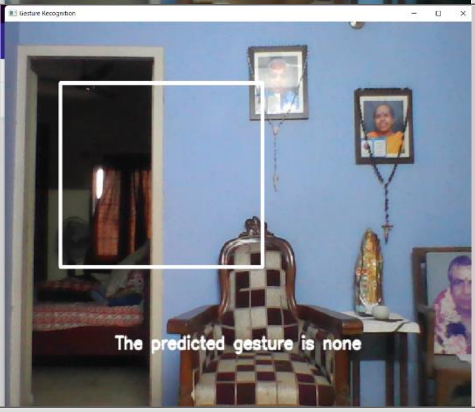
The predicted gesture is play



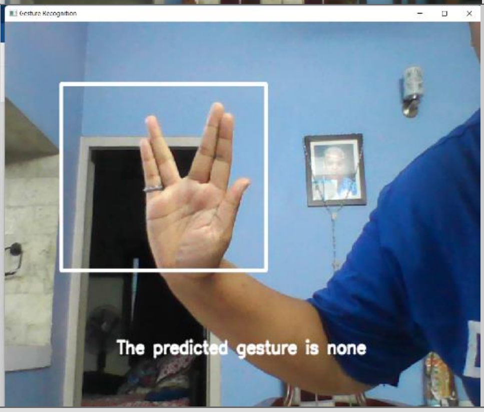
The predicted gesture is mute



The predicted gesture is stop

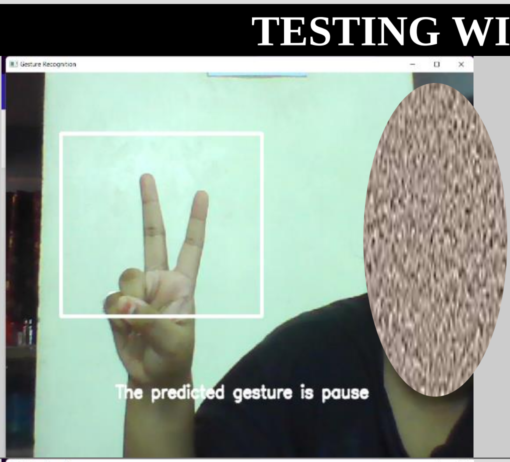


The predicted gesture is none

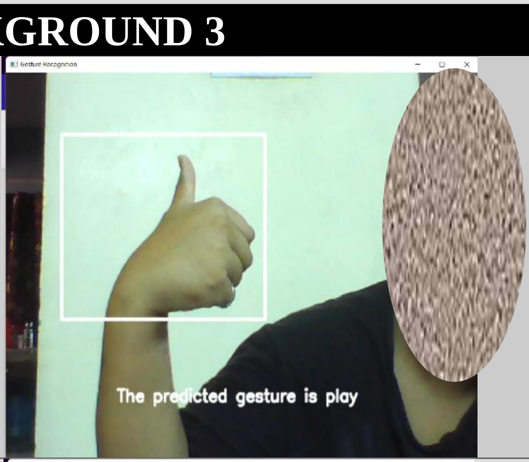


The predicted gesture is none

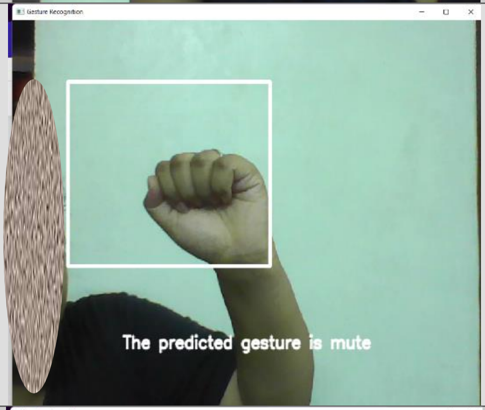
TESTING WITH BACKGROUND 3



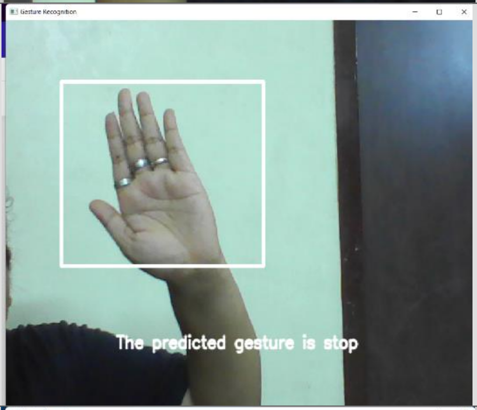
The predicted gesture is pause



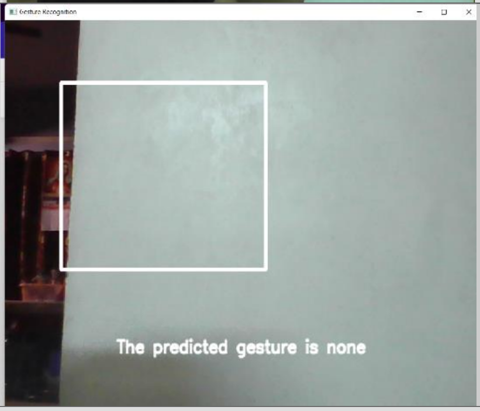
The predicted gesture is play



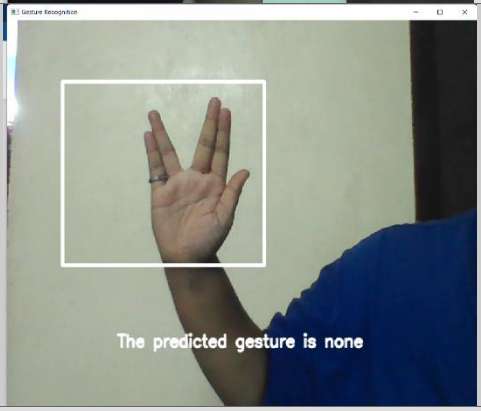
The predicted gesture is mute



The predicted gesture is stop

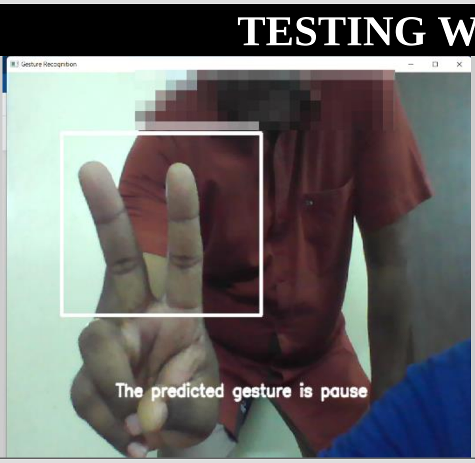


The predicted gesture is none

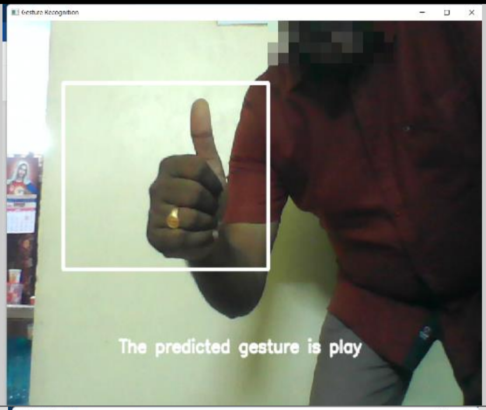


The predicted gesture is none

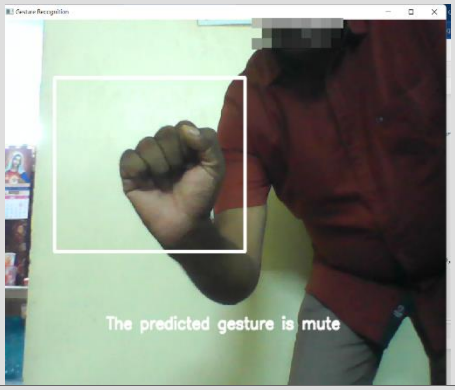
TESTING WITH DIFFERENT USER



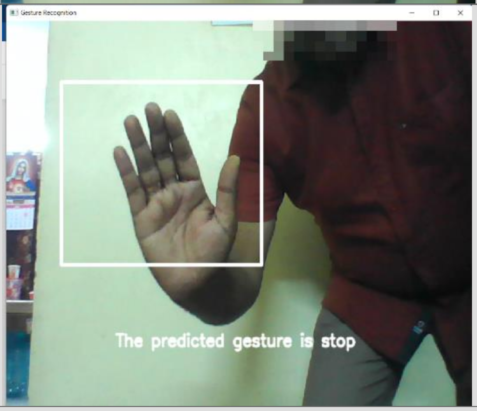
The predicted gesture is pause



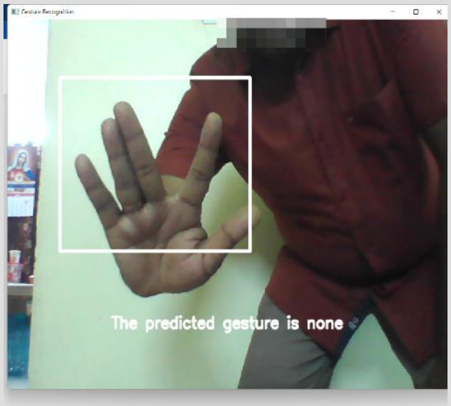
The predicted gesture is play



The predicted gesture is mute



The predicted gesture is stop



The predicted gesture is none

CONCLUSION

- Gesture recognition using PyTorch started with collecting 200 training and 100 validation images.
- The collected images undergo image transformation including normalization and conversion to tensors.
- These tensors are then given as input of batch size 32 to three models namely ‘Resnet18’, ‘Alexnet’, and ‘Googlenet’.
- Training and validation accuracies are found and compared for the three models.
- Confusion matrix and classification report are used in further study of model selection.
- After the study, it was found that Alexnet gave a better accuracy and lesser misclassified samples and so we use save it for testing.
- Alexnet model is loaded and testing is done on a live camera-stream.
- Gestures are performed inside the rectangular area inside the webcam stream.
- The gestures are identified and the name is displayed.

FUTURE SCOPE

- One of the future scopes of gesture recognition is the control and manipulation of a video player or a music player based on the gestures performed.
- Gesture recognition can also be used in the field of medicine for recognizing and treating life-threatening diseases like heart attacks or strokes, where advanced robotics systems with gesture recognition can be installed in homes or hospitals.
- It can also be used for an immersive experience while playing video games.
- Providing distinct and equally non-burdensome services to the differently abled and handicapped is one of the largest difficulties facing society today. Even while there are special provisions in place all across the world, there is still more that may be done to make all lives equal. For individuals who are less fortunate than most of us, gesture recognition technology can eliminate a lot of manual effort and significantly simplify their lives.

REFERENCES

i. https://youtu.be/yqkISICHH-U

ii. https://github.com/nicknochnack/GestureRecognition

iii. https://github.com/nicknochnack/RealTimeObjectDetection

iv. https://github.com/kevinam99/capturing-images-from-webcam-using-opencv-python

v. https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

vi. https://www.learnpytorch.io/06_pytorch_transfer_learning/

vii. https://github.com/rasbt/deeplearning-models/blob/master/pytorch-lightning_ipynb/cnn/cnn-alexnet-cifar10.ipynb

viii. https://pytorch.org/vision/stable/models.html

ix. https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765?gi=84e224a436ec

x. https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neural-networks-a-step-by-step-guide/

xi. https://becominghuman.ai/deep-learning-hand-gesture-recognition-b265f4e6cf02?gi=46f7c764bea4

xii. https://bura.brunel.ac.uk/bitstream/2438/20923/1/FulltextThesis.pdf