**A Comparative Study on the CNN Models and Indian Sign Language To Speech System**

**Gokul Krishna R a, Abishek S b and Alan Jeremiah Immanuel c**

**a,b,c** *Students, Vellore Institute of Technology, Chennai, Tamil Nadu, India*

*e-mail: alan.j.immanuel@gmail.com*

**Abstract—**Sign Languages are the system of communication-based on signs designed to help the deaf and dumb people to communicate. The signs include hand shapes or gestures made in a particular location, facial expressions and in some cases hand combinations. There exists no universally accepted Sign Language, but there exist around 140 known sign languages right now. These different sign languages differ in their signs. The alphabets (A-Z) are expressed in a different way for each sign language, such as the American Sign language, Indian Sign language, The British Sign Language and so on. All these sign languages help the impaired people to communicate with each other. Hence people need to learn sign language to communicate using signs. This limits the community audience to only include people who know sign language and communicating with the rest is still a block in the communicating world. Many Sign Language recognition systems using CNN models, ResNet50 and many as such using various image processing and feature detection techniques have been employed to predict or identify the signs. In our paper, the focus is to bridge the gap between the sign language speaking community and the non-sign language speaking community by helping them communicate. We aim to build a model to classify and recognise sign language and then form words by sequencing the recognized letters/numbers from the signs and then convert the recognized text to speech as well as captioning the text to help everyone who has no knowledge of sign language to communicate.

***Keywords:*** Indian Sign Language, Mediapipe, CNN, OpenCV, HSV, gTTS

**1. INTRODUCTION**

Sign language is a system of communication using visual gestures and signs, as used by deaf and dumb people. There are various categories in sign language like ISL (Indian Sign Language), ASL (American Sign Language), BSL (British Sign Language) and etc... But none of the sign languages is universal or international. A person should learn sign language to understand and communicate the language; this becomes complicated when a person who has an inability to speak or hear wants to tell something to a person or group of persons as most of them are not aware of the sign language. This paper has been proposed with the aim of implementing different models on the dataset and attaining maximum accuracy in the conversion of sign language to speech.

Deaf and dumb people communicate with each other using sign language, sign language is the most common form of communication, people who do not understand sign language face difficulty in understanding it, so a sign recognition system that can convert the signs to English or other spoken languages can be very useful for the society.

Sign languages are the native languages of the Deaf community and provide full access to communication. Although sign languages are used primarily by people who are deaf, they can also be used by others, such as people who can hear but can’t speak. Some children with autism spectrum disorder (ASD) struggle to develop verbal communication. Learning sign language can be a helpful communication tool for some children with ASD. So, sign language recognition systems can help people communicate with each other easily.

In sign language the singer (deaf/dumb person) uses his hand and facial expressions to communicate, in sign language the signs vary similar to each other, so the system should be capable of recognizing accurately. The speed of the communication changes from user to user, so the system should be capable of adapting. Many sign language recognition systems use special gloves to detect, but carrying gloves everywhere is not feasible and it is also not affordable.so This leads us to use vision-based sign recognition systems. Vision-based systems have their own drawbacks, the data set should be huge and versatile.

**2. RELATED WORKS**

Sign language recognition is a widely and most worked on problem, because of the social impact it has on disabled people. Many sign language recognition methods using various models such as CNN, RNN and many preprocessing methods on the datasets have been tried on to improve the efficiency of the system. Here, Kshitij Bantupalli et al [1]used an already existing CNN model named Inception which will extract the features from the video stream. Then using the SoftMax and pool layer of the CNN model, the features are fed into LSTM for temporal features from the video. Outputs of the CNN model and RNN model were compared. The Accuracy dropped when faces were in the image. Also, the skin tone of the hand affected the accuracy. Different skin tones gave different results.

In [2] Kumar A et al proposed a novel architecture to improve sign language detection. This improved model would be able to distinguish between static and dynamic (J, Z)gestures based on the change incog in subsequent frames and eliminate face using viola-jones face detection, uses Zernike moments as feature descriptor for static gestures and curve feature vector for dynamic gestures. These feature vectors are used for training and classification using an SVM. The proposed model was quite successful in classifying static gestures with 93% and dynamic gestures with 100 % accuracy.

The authors in [3] proposed a novel vision-based approach to detect sign language. It uses skin colour segmentation. static gestures and dynamic gestures are differentiated based on the change in cog. The model uses Zernike moments for static signs, motion trajectory for signs with significant hand movement and shape-based recognition for signs with low hand movement. The feature vectors are then fed into a SVM classifier for training and detection of the gestures. Skin color segmentation using YCbCr color space gave better results on a uniformed background. Trajectory based feature vectors improved the recognition of dynamic features.

In [4] The authors proposed a sign language recognition model using the Convolution Neural Network Model and found the better model using three different polling techniques, in which the image is captured using the front camera or selfie mode.the authors extracted key points effectively from a 2D front camera to make the application mobile friendly.The model had an average recognition rate of 92.88%.

Hore, S. et al in [5] trained the input images in a Neural Network Model, But the inputs were also fed into an optimization algorithm: Genetic algorithm, Particle Swarm Optimization and Evolutionary Algorithm.In the Genetic algorithm, the solution set undergoes a fitness test, such as the RMSE is calculated using the function for each solution. In the PSO algorithm, the solution set is assigned a fitness value, the fitness value is calculated for each iteration. In comparison The NN approach was 93.64% accurate which improved to be 96.7% using the MLP-FFN approach.

In [6] Cui, R et al proposed an architecture with recurrent convolutional neural networks of more learning capacity to achieve state-of-the-art performance on continuous SL recognition, without importing extra supervisory information.The model utilizes the iterative optimization process for training the data to improve the recognition. The proposed deep neural architecture consists of a deep CNN followed by temporal operations for representation learning, and Bi-LSTMs for sequence learning.The model with color image modal shows a 63.70% top-1 accuracy and a 86.37% top5 accuracy.

In [7] the authors proposed a glove based working device which helps in converting sign language (ASL) to speech. The sensor output is processed in Arduino Nano for text acquisition as the output displayed on the LCD. In addition the text is sent via the Bluetooth module to mobile phones / computers. In addition that data is converted from text to speech to speech conversion software. Also the sensors were stitched on the net, the fabric started to tear as a result of which the outputs were not accurate.

In [8] the Beena, M.V. et al proposed a real time sign language to speech conversion using CNN model .The model uses histogram of the hand to adjust the lighting and skin complexion of the hand, this image is then fed into a CNN model for training. The sign language translator had an accuracy of 95 %.

Abiyev, R.H., et al in [9] proposed a recognition system that uses the concept of image processing and Convolutional neural network to improve the integrity and Flexibility of the system. In gesture creation the hand coordinates needed to be set. While setting the coordinates of hand gesture, it is converted to binary image for minimizing the background disturbance. As they have trained and tested with their own dataset, the accuracy rate of the application is approximately 85%.

**3**. **EXPERIMENTAL SETUP**

**3.1. HARDWARE REQUIREMENTS**

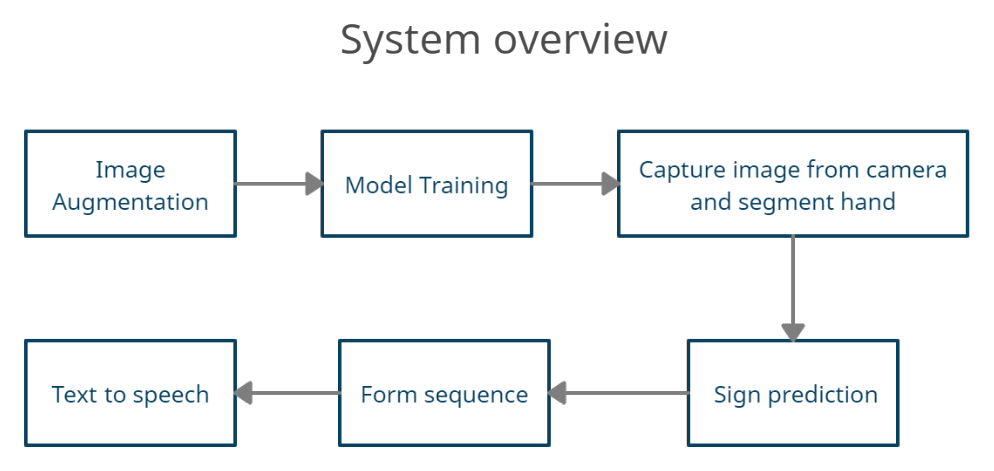
For the experiment, the computing system is composed of 1 x single core hyper threaded Xeon Processors @2.3Ghz i.e(1 core, 2 threads) with a maximum speed of 3.2Ghz, and 1xTesla K80 , compute 3.7, having 2496 CUDA cores , 12GB GDDR5 VRAM. This experiment involves the usage of heterogeneous architectures. The system has 12.6 GB RAM available with 33 GB of disk space available. webcam with 640 x 480 resolution at least.

**3.2. SOFTWARE REQUIREMENTS**

For this model we need the following softwares:

* Google colab - A service offered by google to train ML models easily
* Any code editor that can support python
* Tensorflow 2.0 - open source python library for ML
* OpenCV - open source python library aimed for computer vision.
* Mediapipe - cross platform framework for building machine learning

**4. SYSTEM DESIGN**



**Fig 1.** High level View of the system

The overall system design is shown in Fig 1.

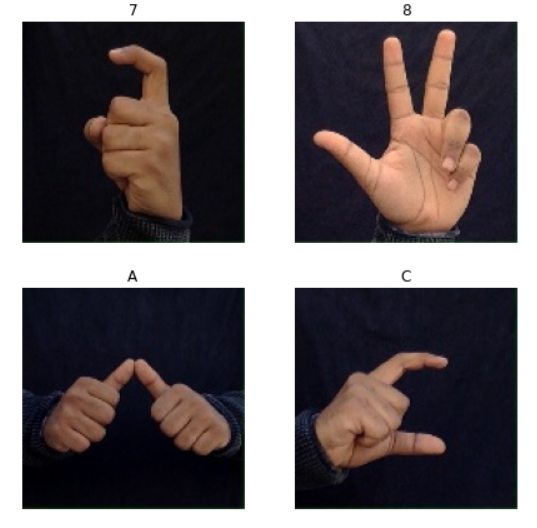
**4.1. DATASET**

Many datasets for sign language systems exist. This is because just like how different languages exist and are spoken in different regions and states, different sign languages also do exist as they vary in the signs as seen in Fig 2.1, where the letter A is compared. There are around 140 known sign languages today. The most frequently seen sign language dataset includes the American Sign Language dataset, but no universal sign language exists.

**Fig 2.1.** Comparison of Sign Languages (ASL on the left and ISL on the right)

For the given experiment, the Indian Sign Language image dataset was used. The dataset contained 44400 images of 37 classes, where 26 classes were alphabets (A-Z) ,9 classes were numbers(0-9) and the remaining two classes were Space and Delete. The dataset was obtained from the Kaggle website.



**Fig 2.2.** Indian Sign Language dataset

All the images were used for our models. The images were resized to 224 x 224 pixels before being fed into the models. The dataset was split into training and testing data. 35,520 images (80%) were used to train the model and 8880 images (20%) were used for testing. The input data can be seen in Fig 2.2.

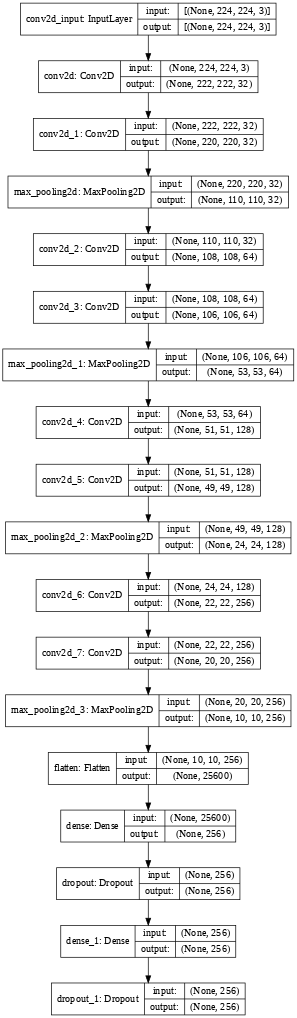
**4.2. MODELS**

The Indian sign language dataset will be trained and tested using the following CNN models for image classification using transfer learning method and compare the results.

**4.2.1. BASELINE MODEL**

CNN model is a deep neural network that is used to analyze images.It captures the spatial and temporal dependencies of the image. CNN model comprises various layers. Initial layers learn basic feature detection filters like edges, corners. The middle layer learns higher order features such as shapes and parts of object filters that detect parts of objects. The last layer is where the Model classifies the image ,this layer consists of dense layers for final output.

In this paper we propose a base CNN model which will be compared to other existing CNN models for image classification. The model consists of a total 18 layers.which includes 4 sets of 2 convolution layers followed by a max pooling layer,and flatten and 2 dense layers followed by dropout layers. All the layers have relu activation function only the dense layer has softmax as activation function for the classification. The architecture of our model is shown in Fig 3.

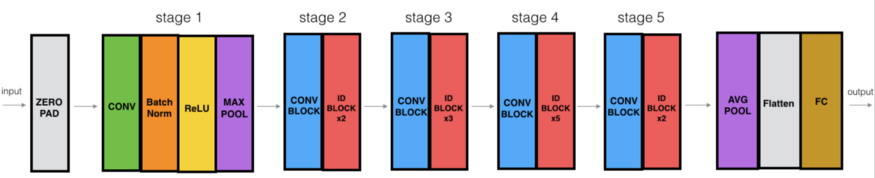


**Fig 3.** Base CNN Architecture

**4.2.2. RESNET50 MODEL**

ResNet-50 is a convolutional neural network that is 50 layers deep where we can load a pre-trained version of the network trained on more than a million images from the Image Net database.The pre-trained network can classify the images into 1000 categories, such as keyboard, mouse, pencil, and many animals.There are research papers based on resnet 50 model for transfer learning in many classification problems.

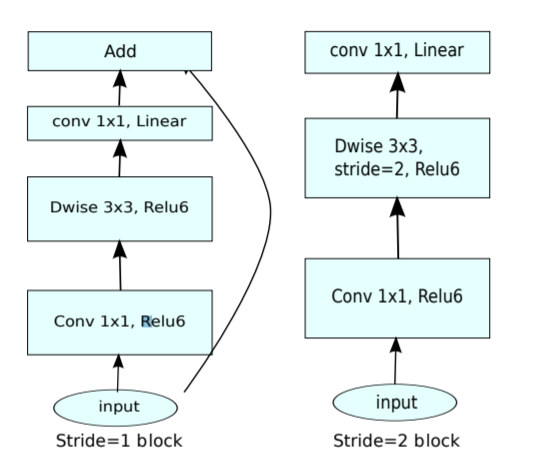
The Resnet model introduces skip connection which helps in the vanishing gradient problem and it is not easy to overfit.The model also gives a time advantage.Resnet 50 architecture is shown in Fig 4.



**Fig 4.** Resnet50 Architecture

**4.2.3. MOBILE**

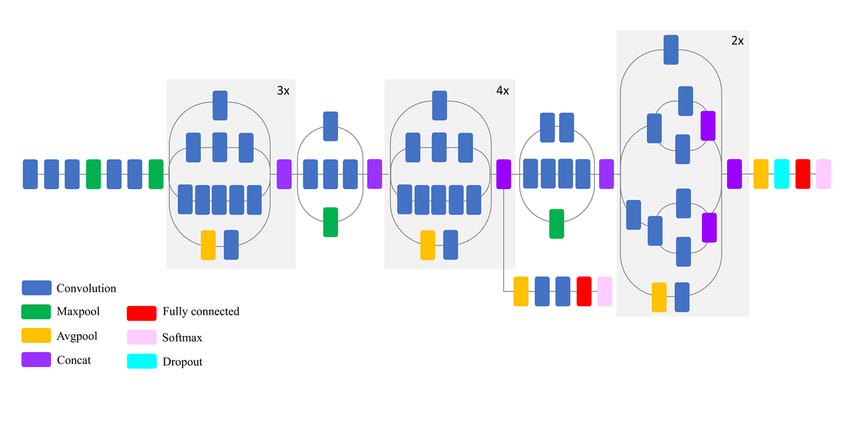
Mobilenet v2 is a CNN model that has a total of 53 layers.The model is part of a general purpose neural network developed by google for image classification.The Mobilenet v2 specifically built for mobile devices to perform segmentation and image classification.The architecture uses bottleneck depth-separable convolution with residuals as basic building block and it consists of 32 filters in the initial convolution layer followed by 19 residual bottleneck layers. This model uses linear bottlenecks between layers and shortcut connections between layers. The architecture of Mobilenet\_v2 is shown in Fig 5.



**Fig 5.** Mobilenet V2 Architecture

**4.2.4. INCEPTION\_V3 MODEL**

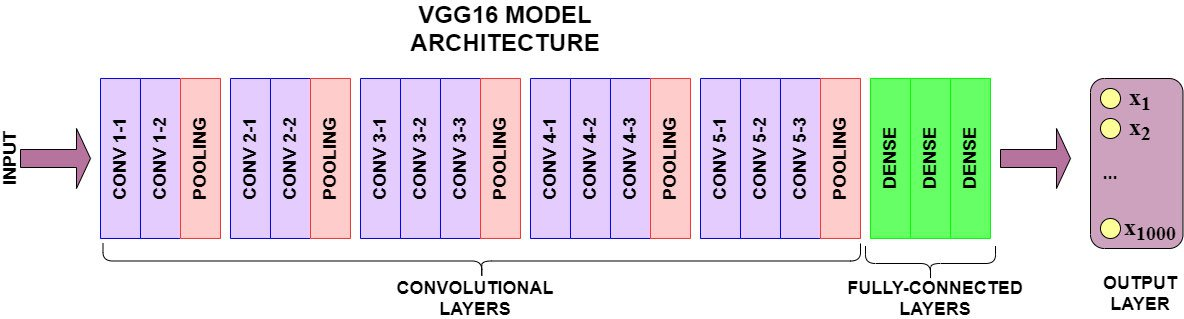
Inception v3 is a deep learning convolutional neural network architecture designed by Google. It was trained using a dataset of 1000 classes from the original ImageNet dataset which was trained with over 1 million training images.In comparison the VGGnet, inception networks have proven that it is computationally efficient both in terms of number of parameters generated by the network and the economical cost incurred.Inception v3 when augmented with an auxiliary classifier, factorisations of convolutions, RMSProp and Label smoothing can achieve the lowest error rates possible. The architecture of Inception v3 is shown in Fig 6.



**Fig 6.** Inception v3 Architecture

**4.2.5. VGG\_16 MODEL**

VGG 16 is a cnn architecture named after the visual geometry group from oxford who developed it and is 16 layer deep. The model loads a set of pre-trained weights on ImageNet. The VGG 16 is a CNN model with deeper architecture. This model is a vertical model and is named 16, because it stacks 16 layers as seen in Fig 7. The convolution layers are stacked on one another successively followed by a pooling layer and this repeats a few more times. After each pooling layer, the filter channels are altered, but the kernel remains the same. The entire model consists of only conv and pool layers. The architecture is shown in Fig 7.



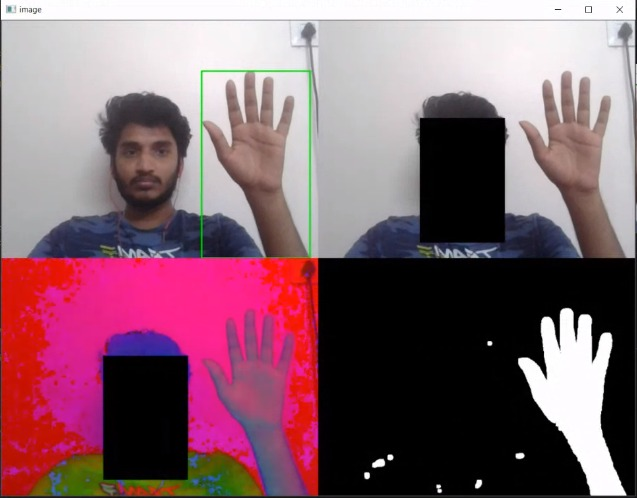
**Fig 7.** VGG-16 Architecture

**4.3. HAND SEGMENTATION**

In this module we have three different approaches for hand segmentation. Hand segmentation is fairly a difficult task as most of the existing hand segmentation algorithms perform a lot of image processing to identify the hand region or use ML models to detect the hand. The approaches we have implemented are given below.

**4.3.1. FACE REMOVAL AND HSV CONVERSION**

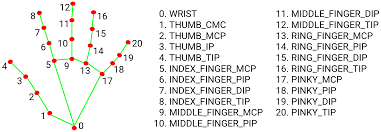
This approach is fairly simple. Hue ,saturation, value of human color lie within a certain range irrespective of age, gender or race [10][11]. As we are using Skin color segmentation to detect hands ,we have removed the face region from the image as the face has the same color as the hand .The first step would be to remove the face region ,using Haar cascade classifiers we can detect the face easily,then we mask the face region so only the hand is most prominent. Now converting the image to Hsv format and applying a mask will segment the hand region. The face removal and hsv conversion can be seen in Fig 8.



**Fig 8.** Hand Segmentation using HSV conversion

**4.3.2. MEDIAPIPE**

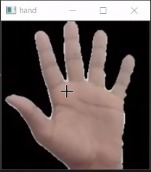
Mediapipe is a cross platform ML framework built by google to detect face ,hands posture etc. The mediapipe model can be used to find 21 different feature points from a hand image as seen in Fig 9. We use these points which are detected by mediapipe to draw a bounding box around the hand. As there is no image processing involved the detection and segmentation is very fast and accurate.



**Fig 9.** The 21 feature points detected by mediapipe

**4.3.3. FIXED BOUNDING BOX**

The fixed bounding box method is the most common and easy way of segmenting. In this approach the user has to place his hand in the fixed bounding box that will present on the screen. When the user places his hand in the bounding box ,we can segment the hand region as we know the location of the bouncing box present in the image. This can be seen in Fig 10.



**Fig 10.** Hand segmentation using fixed box method

When compared with three approaches the mediapipe method was the fastest and gave accurate results. So in our sign language to speech system we used the medipipe approach.

**4.4. SEQUENCE FORMATION**

When the segmented hand region is fed into the model ,The model predicts the sign ,we check if this sign is repeated for N consecutive frames. N can be adjusted based on the user speed. If it is repeated for N consecutive frames, we add the detected sign to the sequence string. When the model detects the space sign the system considers that the sequence string is a completed word. The Sequence window is shown in Fig 11.



**Fig 11.** The Sequence window

**4.5. TEXT TO SPEECH**

Text to speech has become easy with the help of APIs ,in our system we use Google's Text to speech API with the help of the python library gTTS.when ever the sequence formation module detects a complete word , The text to speech module converts the word into speech.

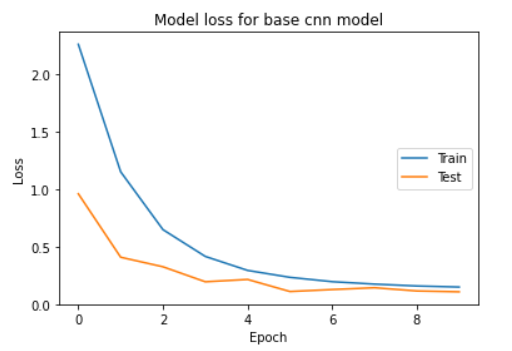
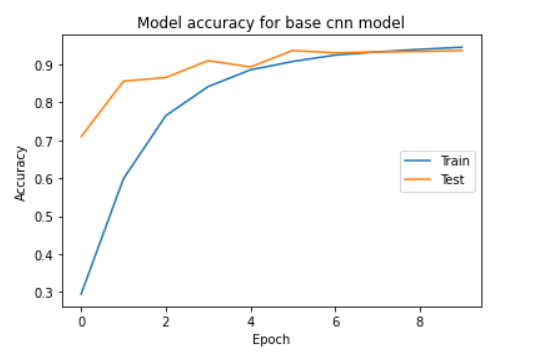
**5. IMPLEMENTATION OF SYSTEM**

**5.1. PREPROCESSING THE IMAGES FOR TRAINING**

For the experiment the ISL was used, the dataset contained 44400 images of 37 classes. To maintain consistency, all the images were captured with the same background. The original image size was 200 x 200 pixels. The images were then resized to 224 x 224 pixels to train the model. We performed image augmentation to generate more image data, so that the model does not overfit and make the model more robust while predicting on unseen images. The final dataset were then split in training and testing data. The training set included 35520 images (80%) and the testing 8880 images (20%)

**5.2. TRAINING THE MODEL**

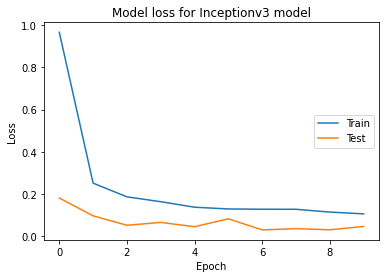
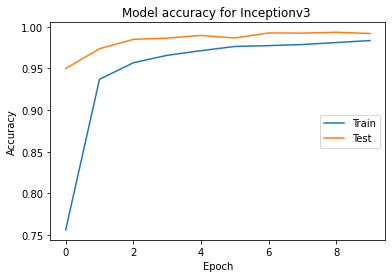
The training dataset is then fed into the base CNN model. The base CNN model consists of a total 18 layers.which includes 4 sets of 2 convolution layers followed by a max pooling layer,and flatten and 2 dense layers followed by dropout layers. All the layers have relu activation function only the dense layer has softmax as activation function for the classification. The model uses RMSProp as the optimizer and categorical loss function. The model is trained with a learning rate of 0.0001 and with 10 epochs, the fit parameters are maintained as constant for all the models. we can see that the model's accuracy and loss flattens from 8 - 10 epochs as seen in Fig 12. Increasing the epochs will result in overfitting.



**Fig 12.** The base CNN model’s accuracy and loss graph

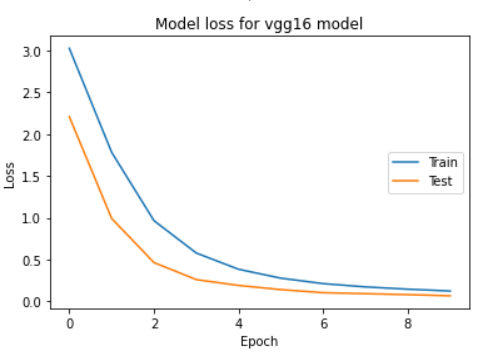
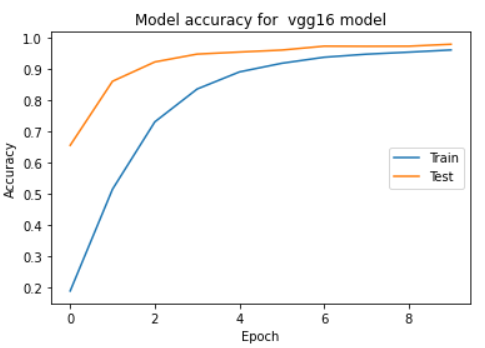
As the base model achieves its highest accuracy at 10 epochs.we keep the number of epochs fixed and compare the results by training the existing CNN models, Inception v3, VGG16, Resnet 50, Mobilenet V2 to classify the sign language. Even the optimizer, loss function and learning rate are kept constant.

In the Inception v3 model we downloaded the pretrained weights of the model with the help of keras.we removed the top layer of the model so that we can add our own top layer which will classify hand signs. We added a flatten layer followed by 2 dense layers, having a dropout of 25% and 50% respectively . The dropout layer helped in preventing overfitting.last we add a dense layer with softmax as the activation function which will classify the images. As mentioned earlier , we train the model with 10 epochs . The training accuracy and loss is shown in Fig 13.



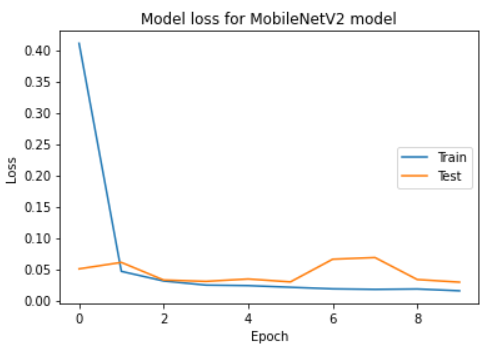
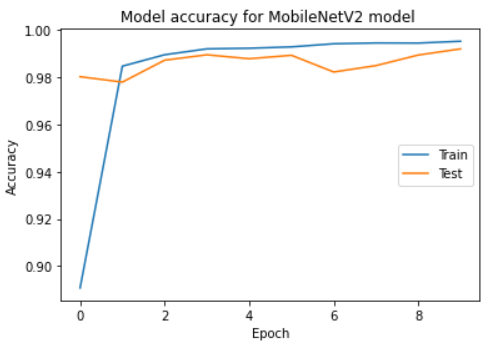
**Fig 13.** The Inception V3 model’s accuracy and loss graph

In VGG 16 model the pretrained weights were downloaded,we remove the top layer of the model to add our own top layer which consists of a flatten layer followed by 2 dense layers , with a dropout of 25% and 50% respectively.The last layer is a dense layer with softmax as the activation function The model is trained with 10 epochs, the training accuracy and loss is shown in Fig 14.



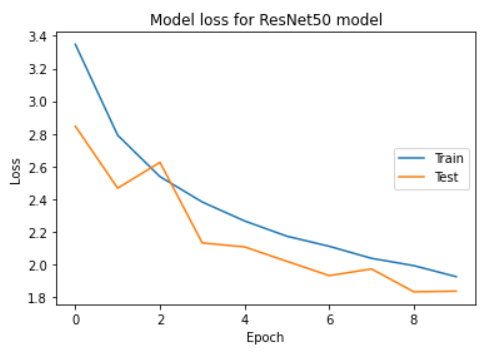
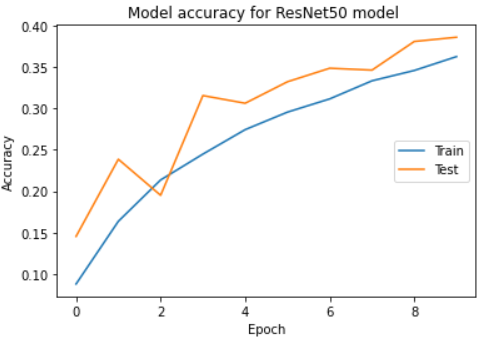
**Fig 14**. The VGG-16 model’s accuracy and loss graph

Pretrained weights of Mobilenet V2 are downloaded with the help of keras,we removed the top layer of the model to add our own top layer that can classify hand signs. We added a Global Pooling layer to the last layer of the model to reduce the dimensions and then we added a flatten layer followed by 2 dense layers , with a dropout of 25% and 50% respectively. The model accuracy and loss is shown in Fig 15.



**Fig 15.** The Mobilenet V2 model’s accuracy and loss graph

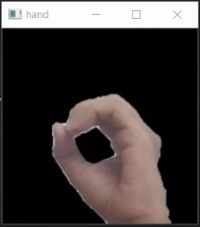
In the Resnet 50 model we downloaded the pretrained weights ,we removed the top layer of the model to add our own top layer that can classify hand signs.We added a Global Pooling layer to the last layer of the model to reduce the dimensions and then we added a flatten layer followed by 2 dense layers , with a dropout of 25% and 50% respectively. The model did not perform well with the dataset within the 10 epochs range .Accuracy and loss of Resnet 50 model is shown in Fig 16.



**Fig 16.** Resnet 50 model’s accuracy and loss graph

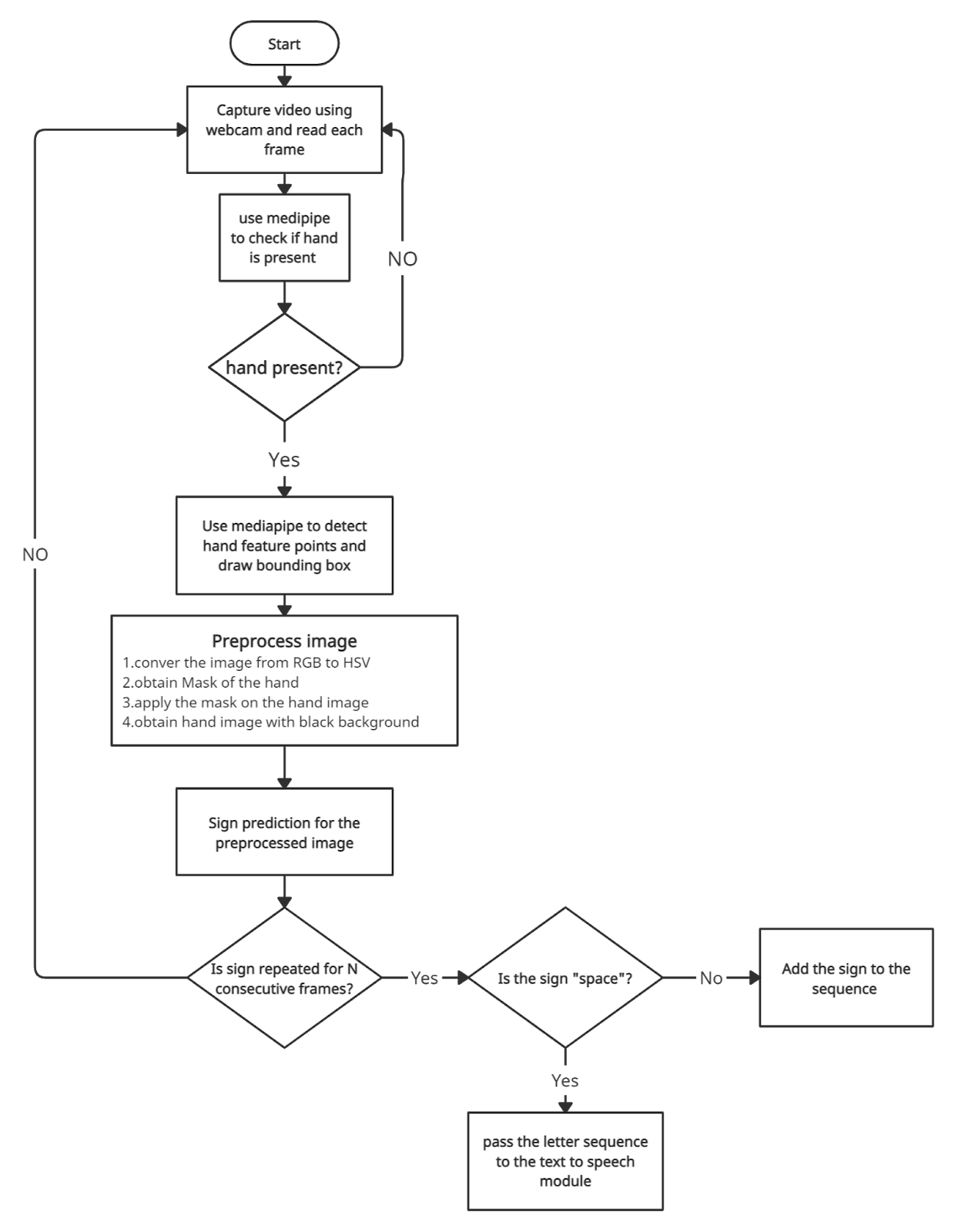
**5.3. HAND SEGMENTATION**

In this module we capture live video from the video,as video is just fast moving frames we read individual frames from the video.for each of the frame we check if the hand is present using Mediapipe library.If hand is present in the frame we use Mediapipe to identify feature points,based on feature points we can draw bounding box around the hand. The bounding box is cropped to get only the hand .Now we convert the cropped image into HSV color space to obtain a mask with only the hand shape . This mask is applied on the cropped image to remove the background in the cropped image. performing all these preprocessing operations results in a segmented hand with black background which can be seen in Fig 17.



**Fig 17 .** Hand Segmentation using mediapipe

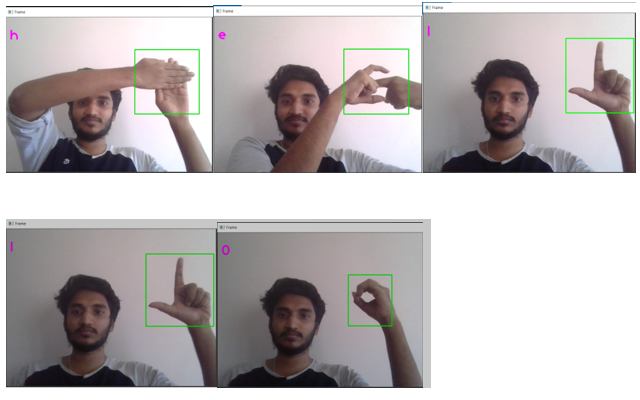
Flow diagram of the sign recognition is shown in Fig 18



**Fig 18.** Flow diagram of sign recognition

**5.4. SIGN PREDICTION**

Now the hand image will be resized to 224 x 224 pixels before passing it to our model for prediction. As we are using softmax layer as activation function in the model,the model returns a numpy array of size 37 i.e. number of classes in the dataset. The array consists of probability values of the class the model thinks the sign belongs to.we take the sign with the highest probability as the predicted sign. Now we check if this same sign is repeated for N consecutive frames. N can be adjusted based on the user’s speed. For any other sign other than delete and space the sign is added to the sequence word. If the “Delete” sign is recognized then we delete the last sign from the sequence. If the detected sign is “Space “ then the system considers that a word was completed in the sequence as seen in Fig 18.The completed word is sent to the text to speech module.



**Fig 19.** Sign recognition for “hello”

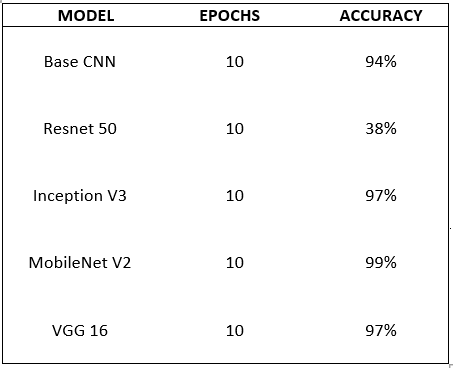
**5.4. TEXT TO SPEECH**

This module uses Google's Text to speech API with the help of the python library gTTS. The api takes strings as input . Whenever space is detected the sign prediction module sends the word to the text to speech module. This module converts the word to speech easily and also the time taken is very low (Real time).

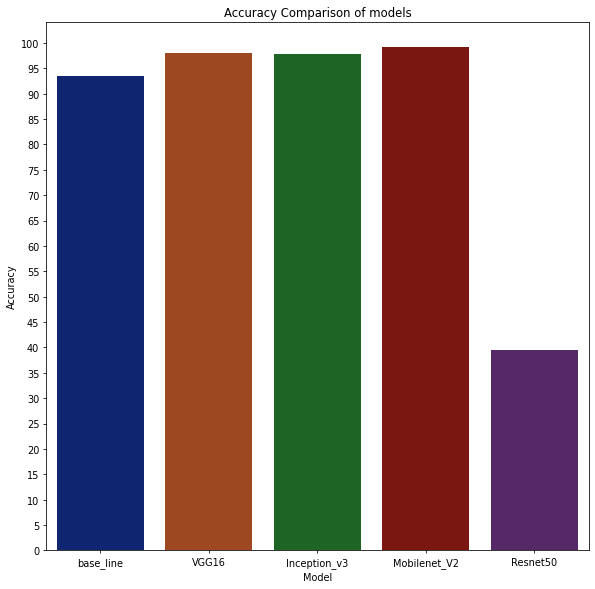
**6. RESULTS AND DISCUSSION**

The ISL dataset was trained and tested on 5 models with the same parameters. The Accuracy of the base CNN model was 94%, and the Inception V3 97%, MobileNet V2 99% and the VGG -16 97%. The Resnet50 had the lowest accuracy of 38%. The MobileNet V2 produced the highest accuracy among all the other models as seen in Fig 19.

**Table 1 : Comparison of Model Accuracy**

****

From table 1 we can see the accuracy of all the models at 10 epochs.

****

**Fig 19.** Boxplot of model Accuracy

In the hand segmentation module, when compared with three approaches the mediapipe method was the fastest and gave accurate results. So in our sign language to speech system we used the medipipe approach. This was because in the fixed box method, the box was static and does not support dynamic hand locations and also it makes the system not easily usable. The Hsv method captures the whole arm, and the prediction accuracy drops due to the involvement of the arm region, as the dataset contains only the palm region.

**7. CONCLUSION AND FUTURE SCOPE**

The system for sign language recognition to speech has been developed using CNN which has stacked layers of convolution and dense layers. For the system Indian sign language dataset was used . The base CNN model gave an accuracy of 94% . While testing we compared 3 different hand segmentation techniques and chose the best method i.e. segmentation using midipipe. The model had higher accuracy during training and validation phase, but when tested for real time recognition using webcam the model had an accuracy of 73%. The reason for this is due to the hand color difference of the dataset and color of our hands. This can be eliminated using color transformation. Also using pretrained models can help in increasing the accuracy of the system for live recognition. The future scope of this paper would be to introduce algorithms and methods that could increase the accuracy during the live recognition. The colour space transformation is one of the methods for the improvement of the system. Also, training with video datasets could help in creating real time sign language recognition that can recognise the location, facial expression and body posture.

**8**. **REFERENCES**

[1] Bantupalli, K. and Xie, Y., 2018, December. American sign language recognition using deep learning and computer vision. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 4896-4899). IEEE.

[2] Kumar, A., Thankachan, K. and Dominic, M.M., 2016, March. Sign language recognition. In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)* (pp. 422-428). IEEE.

[3] Athira, P.K., Sruthi, C.J. and Lijiya, A., 2019. A signer independent sign language recognition with co-articulation elimination from live videos: an indian scenario. *Journal of King Saud University-Computer and Information Sciences*.

[4] Rao, G.A., Syamala, K., Kishore, P.V.V. and Sastry, A.S.C.S., 2018, January. Deep convolutional neural networks for sign language recognition. In *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)* (pp. 194-197). IEEE.

[5] Hore, S., Chatterjee, S., Santhi, V., Dey, N., Ashour, A.S., Balas, V.E. and Shi, F., 2017. Indian sign language recognition using optimized neural networks. In *Information technology and intelligent transportation systems* (pp. 553-563). Springer, Cham.

[6] Cui, R., Liu, H. and Zhang, C., 2019. A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, *21*(7), pp.1880-1891.

[7] Karthiga, E., Fathimal, M.S., Gowthami, N., Madura, P. and Babu, M.P.G., 2018. Sign Language to Speech Using Arduino Uno. In *National Conference on Biomedical Signal and Image Processing, Hosur, INDIA*.

[8] Beena, M.V., Namboodiri, M.A. and Dean, P.G., 2017. Automatic sign language finger spelling using convolution neural network: analysis. *Int J Pure Appl Math*, *117*(20), pp.9-15.

[9] Abiyev, R.H., Arslan, M. and Idoko, J.B., 2020. Sign Language Translation Using Deep Convolutional Neural Networks. *KSII Transactions on Internet & Information Systems*, *14*(2).

[10] Mokhtar M. Hasan & Pramod K. Misra, HSV Brightness Factor Matching for Gesture Recognition System. International Journal of Image Processing (IJIP), Volume (4): Issue (5)

[11] Xingyan. Li. "Vision Based Gesture Recognition System with High Accuracy". Department ofComputer Science, The University of Tennessee, Knoxville, TN 37996-3450, 2005

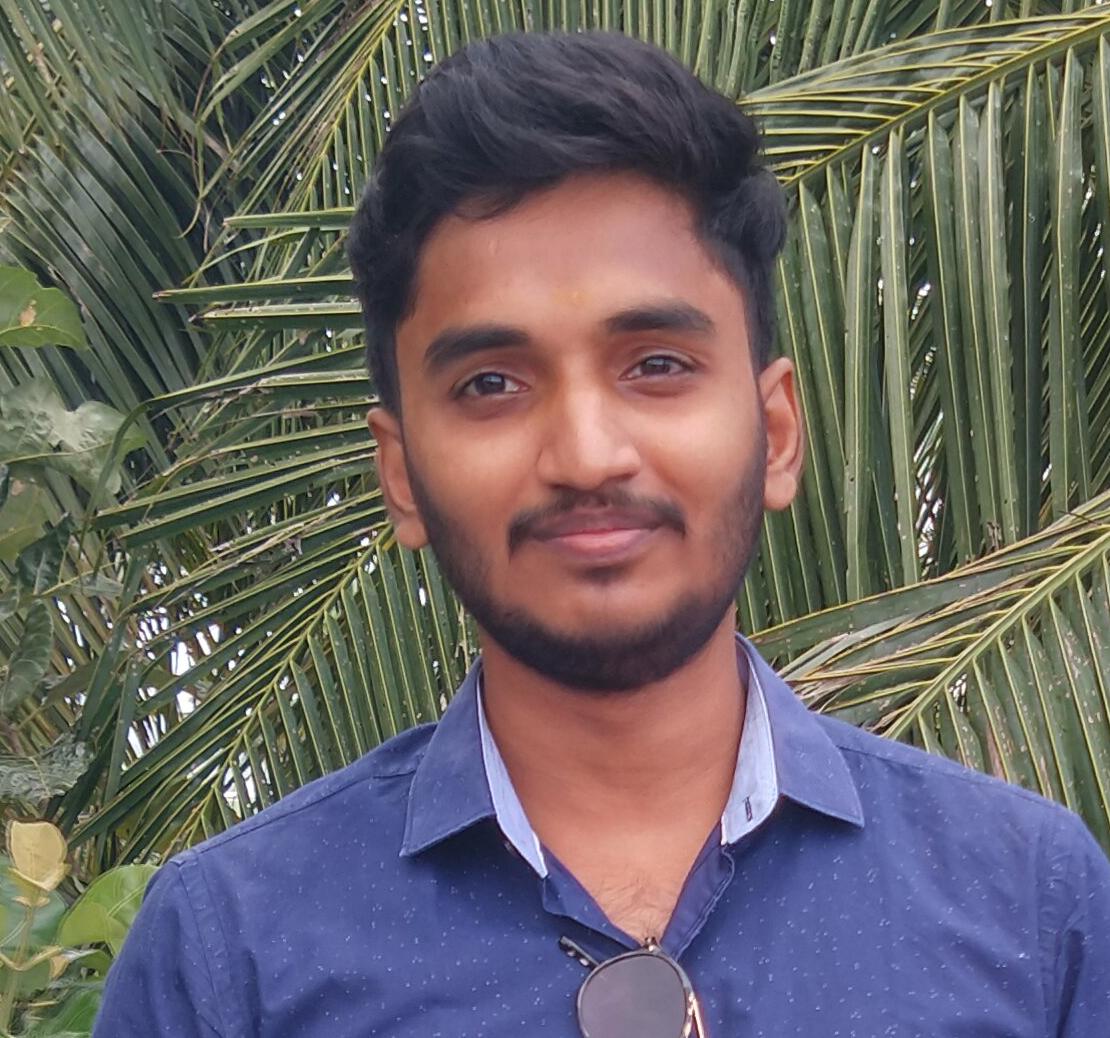
**Author's note**



Alan Jeremiah Immanuel, a undergraduate student in Vellore Institute of Technology is currently pursuing his Bachelor of Technology in Computer Science. His interests are in Computer Vision, Sports Analysis, Machine Learning, Artificial Intelligence and Video Editing.

E-mail: [alan.j.immanuel@gmail.com](mailto:alan.j.immanuel@gmail.com)

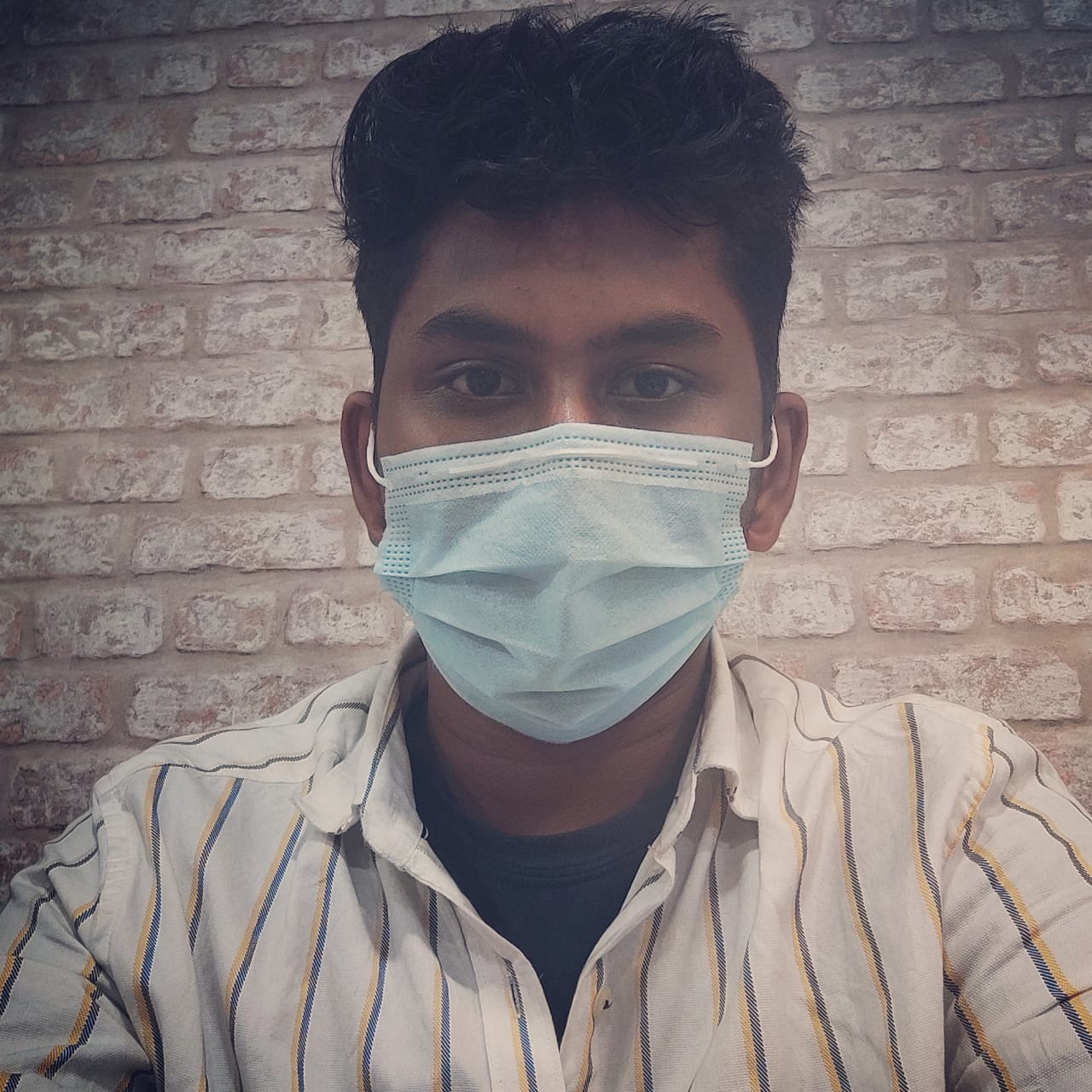
Mobile no: +91 96770 15392



Gokul krishna R is an undergraduate student pursuing Bachelor of Technology in computer science in Vellore Institute of Technology. His field of interests are Computer Vision,Image processing,Machine learning and Augmented Reality.

E-mail:[gokulkrishna.r.2000@gmail.com](mailto:gokulkrishna.r.2000@gmail.com)

Mobile no: +91 6379205857



Abishek S is an undergraduate student pursuing Bachelor of Technology in computer science in Vellore Institute of Technology. His field of interests are Computer Vision, Data Science, Parallel Computing and Data Visualisations.

E-mail: [abishekreddy22@gmail.com](mailto:abishekreddy22@gmail.com)

Mobile no: +91 73973 42830