Hand Gesture Based Control Interface for Smartphones

Shubham Pandey and Abir Karmakar

November 28, 2021

Hand Gesture based Control Interface for Smart phones

Shubham Pandey[1]*, Abir Karmakar[1] *

[1]*Department of Computer Science and Engineering, Amity University, Noida, Uttar Pradesh, India*

**Corresponding author. Email: shubhampandeysp001@gmail.com; abirkarmakar13@gmail.com*

ABSTRACT: This work provides various description of different methods of improving hand gesture recognition algorithms. A vision-based Hand Gesture Recognition system had been useful to recognize hand gesture in air. The work realizes the segmentation of hand gestures by establishing the skin color model and AdaBoost classifier based on CNN according to the particularity of skin color for hand gestures, as well as the denaturation of hand gestures with one frame of video being cut for analysis. In this regard, the human hand is segmentd from the complicated background, the real-time hand gesture tracking is also realized by CamShift algorithm. To extract the features of air gesture we used statistical technique which is Principal Component Analysis and Machine Learning Algorithms such as Convolutional Neural Network and Adaboost for Object and Motion Detections. The idea here is to recognize gestures without the need to connect to a computer in which a database is located to perform training process. With this system, all steps can be done with optimum accuracy and the results were much improved. The work comprised of image acquisition, image processing, and after application of required machine learning algorithms our accuracy was close to 88.26%.

1 INTRODUCTION

Hand gestures are natural and intuitive communication way for the human being to interact with his environment. Gestures can have different meanings according to the language or culture. They can also be a way to interact with machines. The subject of our research concerns the design and development of computer vision methods for recognizing hand gestures by a mobile device.

With the development of interaction between human and machine, the interaction between computer and human is becoming more and more frequent. Among them, hand gestures are commonly used in this aspect. Since there are various hand gestures and enriched information contained in them, recognition of hand gesture has been greatly used in many fields, such as UAV, somatosensory game, sign language recognition and so on [8][2]. In this regard, it is of great significance to study on hand gesture recognition. Advancement in the mobile hardware makes it possible to utilize computer vision software's on mobile platform. Hand gesture is a powerful, natural means of communication between human beings. Now it can be a promising way to interact with computers, where gesture recognition is the core of such technique. The hand is the primary operating tool for humans. Therefore, its location, orientation and articulation in space is vital for many potential applications, for instance, object handover in robotics, learning from demonstration, sign language and gesture recognition, and using the hand as an input device for man-machine interaction, we focus on appearance model-based method. There have been a number of research efforts on appearance-based method in recent years. Freeman and Weissman recognized gestures for television control using normalized correlation [3]. This technique is efficient but may be sensitive to different users, deformations of the pose and changes in scale, and background. Cui and Weng proposed a hand tracking and sign recognition method using appearance-based method [4]. Although its accuracy was satisfactory, the performance was far from real-time. Just et al introduce modified census transform into hand gesture classification. For the purpose of classifying each gesture respectively, their method obtains fairly good results. While the performance in recognition experiments was not so satisfactory and the recognition result of different gesture great disparity. Elastic graphs were applied to represent hands in different hand gestures in Triesch's work with local jets of Gabor filters. It locates hands without separate segmentation mechanism and the classifier is learned from a small set of image samples, so the generalization is very limited. Hand gesture recognition has the advantage of being applicable to a range of applications, such as handling presentations, controlling drones, and more. Hand-motion-gesture interaction puts forwards a few restrictions on the surrounding environment. For example, voice commands are prone to error in noisy environments, but motion gestures can be performed as long as the hands of users are free. Finally, motion-based hand gestures can be designed in three-dimensional space. Compared to surface gestures, there remains larger design space for a variety of interactive tasks. Hand gesture recognition is achieved using two main kinds of sensors: contact sensors and non-contact sensors. For the interaction with mobile currently we rely on the touch input but this limits the expressiveness of the input. Here we review the literature that expands the input space from touchscreen to the areas around mobile and to help deaf and dumb persons to communicate with each other. A human hand is very tough for a machine to detect correctly. We need to recognize the edges, lines and angles within the palm and the fingers. We are going to use CNN algorithm for training the machine for different positive and negative images. Then we will use Ada boost to enhance the weights assigned to each object. Then we will use Non – Maximum Suppression Algorithm to detect multiple hands in a single frame. AndroSpell uses android camera to grab the image and further preprocessing feature extraction and classification is done on device. For Classification machine learning algorithm such as decision tree and neural network are used.

LITERATURE REVIEW

Shrey et al., (2007) [3] in their research paper worked upon Facial Emotion Recognition using Convolution Neural Network. The accuracy was obtained by using the dataset and the approach of selecting the data, processing the data, training the data, and selecting the appropriate model, which gave good accuracy and remained suitable to the problem, and extracting the features from the given image in order to make a list to assist the model, using activation and generalisation for tagging.

Guojun et al., (2018, May). *[4]* in their research paper worked upon Emotion Recognition Using Deep Neural Network with Vectorized Facial Features. With proper modelling, emotions were recognized using methods like computer vision and DNN. The proposed vectorized facial feature was used to train a DNN for emotion recognition in this paper. With 'vectorized facial landmarker approach', a set of experiments were carried out to evaluate the efficiency of this suggested method providing 84.33% accuracy.

Ariel et al., (2018, September). *[5]* in their research paper worked upon Deep Learning for Emotion Recognition in Faces. In this paper various techniques have been used to perform emotion recognition from faces such as Hidden Markov Models, State Vector Machines, and neural networks. To classify photos into seven emotions, two CNN architectures with automatic feature extraction and representation were used, followed by fully linked softmax layers. The first architecture deals with reducing the number of deep learning layers providing an accuracy rate of 96.93 % and the second splits the input images horizontally into two streams based on eye and mouth positions providing an accuracy rate of 86.73 %.

Ninad et al., (2020) *[6]* in their research paper worked upon Facial emotion recognition using convolutional neural networks. In this paper, a technique called facial emotion recognition using convolutional neural networks (FERC) was proposed. FERC is based on two-part convolutional neural network (CNN): The first part concentrates on removing the background from the image while the second one deals with facial feature vector extraction. 96% accuracy was obtained, using a Expressional Vector (EV) of length 24 values.

Mostafa K. et al., (2014) *[7]* in their research paper worked upon Fully Automated Recognition of Spontaneous Facial Expressions in Videos Using Random Forest Classifiers. The design and implementation of an automated comprehensive facial expression detection and classification framework are discussed in this study. The technique that was used in this paper was Random Forest classifiers. For database AFEW accuracy was 4.53% For JAFFE-CK database accuracy was 54.05% For CK – CK database accuracy was 90.26%.

Li Zhang, et al., (2013). *[8]* in their research paper worked upon Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot. NN-based upper and lower facial Action Units analysers and a NN-based facial emotion recogniser techniques were implemented to detect emotions from real-time posed affective facial expressions. For database C-K Accuracy came out to be 75.83%.

Zisheng, et al., (2009, October). *[9]* in their research paper worked upon Facial-component-based Bag of Words and PHOG Descriptor for Facial Expression Recognition. In this paper a framework of facial appearance and shape information extraction for FER was proposed. Multi-class SVM classifiers with RBF kernels were used to classify the six basic facial expressions. For shape extraction, PHOG (Pyramid Histogram of Orientated Gradient) descriptors were computed on the 4 facial component regions to obtain the spatial distribution of edges. Lastly the Accuracy came out to be 96.33% using the Cohn-Kanade database.

De Silva, et al., (2003, December). [10] in their research paper worked upon Real-time facial feature extraction and emotion recognition the main objective of the paper is the real-time implementation of a facial emotion recognition system. The proposed method uses edge counting and image-correlation optical flow techniques to calculate the local motion vectors of facial feature. After performing facial feature extraction, motion analysis techniques are used to determine the relative spatial motion of the facial feature displacements, and the motion vectors are sent to a neural network module for evaluation. For database CMU Accuracy came out to be 93.3 %.

Douglas, et al., (2012). *[11]* in their research paper worked upon "FACIAL AND EXPRESSION RECOGNITION" which was aimed at helping visually impaired people the main problem they encountered during the whole process was face recognition. They went on with another method know as Local binary patterns (LBP) authors were successfully able to implement this method but in practice they found it lacking it was fast, but it was not as accurate as they wanted it to be. Technique used was OpenCV and Principal Components Analysis and Database used was CMU-Pittsburgh AU-*Coded Face Expression Image* Database. For this database Accuracy came out to be 92%

*Caifeng et al., (*2009) *[12]* in their paper evaluated facial representation based on statistical local features, Local Binary Patterns, for person-independent facial expression recognition. They observed in their experiments that LBP features perform stably and robustly over a useful range of low resolutions of face images and yield promising performance in compressed low-resolution video sequences captured in real-world environments. Highest Accuracy achieved was 94% performance using a three-layer neural networks.

Hamit Soyel et al., (2007, August) *[13]* in their research paper, proposed a novel approach for facial expression analysis and recognition. Objective was Facial Expression Recognition Using 3D Facial Feature Distances. The proposed approach relies on the distance vectors retrieved from 3D distribution of facial feature points to classify universal facial expressions. Facial expressions such as anger, sadness, surprise, joy, disgust, fear and neutral are successfully recognized with an average recognition rate of 91.3%. It was found that the highest recognition rate reaches to 98.3% in the recognition of surprise.

Lei Xu, et al., (2018, December). *[14]* in their research paper "Face Expression Recognition Based on Convolutional Neural Network" used CNN model for facial expression recognition. In this method it pre-processes the facial expression images after that for feature extraction it used some trainable convolution kernels, the largest pooling layer is used for dimensionality reduction after that it seven types of facial expressions are recognized with a classifier known as "SoftMax "also the dataset used by authors are "FER2013" to verify their model. For database FER2013 Accuracy came out to be 69%.

Author & Year	**Proposed Technique**	**Objective**	**Observation**
• Modi, S., & Bohara, M. H. (2021, May).	introduce a (FER) framework based on (CNN) and Deep Learning Strategies	analyze Facial Emotion Recognition using Convolutional Neural Networks.	IN model has a better accuracy than the transfer learning model after 35 epochs which is "0.8251".
• Yang, G., Ortoneda, J. S. Y., & Saniie, J. (2018, May).	rforming Vectorization and Normalization using Radboud Faces Database (RaFD)	analyse facial emotions using Computer Vision and Deep Neural Network	IN model can predict emotions with 84.33% accuracy
• Ruiz-Garcia, A., Elshaw, M., Altahhan, A., & Palade, V. (2018, September).	e emotion recognition from faces using CNN in this paper used the Karolinska directed Emotional faces database (KDEF)	explore 2 architectures for Convolutional Neural Networks (CNN) to achieve deep learning classification of emotional states	st CNN architecture provides accuracy rate of 96.93% and second provides an accuracy rate of 86.73%.
• Mehendale, N. (2020)	RC technique used tasets used were - CohnKanade expression, Caltech faces, CMU and NIST.	cial Emotion Recognition using convolutional neural networks	% accuracy was obtained, using a Expressional Vector (EV) of length 24 values.
• Abd El Meguid, M. K., & Levine, M. D. (2014)	chnique used: Random Forest classifiers tabase used: AFEW, JAFFE-CK, CK-CK	tomated Recognition of Spontaneous Facial Expressions in Videos Using Random Forest Classifiers	r database AFEW accuracy was 4.53% For JAFFE-CK database accuracy was 54.05% r CK – CK database accuracy was 90.26%
• Zhang, L., Jiang, M., Farid, D., & Hossain, M. A. (2013)	chnique used: NN based Facial emotion recogniser	elligent facial emotion recognition and semantic-based topic detection for a humanoid robot	r database C-K Accuracy came out to be 75.83%.
• Li, Z., Imai, J. I., & Kaneko, M. (2009, October).	chnique used: PHOG (Pyramid Histogram of Oriented Histogram) descriptors. tabase used: AFEW, JAFFE-CK, CK-CK	cial-component-based Bag of Words and PHOG Descriptor for Facial Expression Recognition	r database C-K Accuracy came out to be 96.33 %.
• De Silva, L. C., & Hui, S. C. (2003, December).	chnique used: Edge counting and image-correlation optical flow tabase used: Camegie Mellon University (CMU)	al-time Facial Feature Extraction and Emotion Recognition	r database CMU Accuracy came out to be 93.3 %

• Astler, D., Chau, H., Hsu, K., Hua, A., Kannan, A., Lei, L.& Zhang, X. (2012).	chnique used: OpenCV and Principal Components Analysis tabase used: CMU-Pittsburgh AU-Coded Face Expression Image Database	cial and Expression Recognition for the blind using Computer Vision	r this database Accuracy came out to be 92%
• Shan, C., Gong, S., & McOwan, P. W. (2009)	chnique used: Support Vector Machine (SVM), Linear Discriminant Analysis (LDA) and the linear programming tabase used: MMI and the JAFFE database	cial expression recognition based on Local Binary Patterns	ghest Accuracy achieved was 94% performance using a three-layer neural networks
• Soyel, H., & Demirel, H. (2007, August).	chnique used: To use distance measures extracted from 3D face vectors to determine the facial expressions. tabase used: BU-3DFE	cial Expression Recognition Using 3D Facial Feature Distances	r database BU-3DFE, Average recognition rate of 91.3%.
• Xu, L., Fei, M., Zhou, W., & Yang, A. (2018, December).	chnique used: Deep Learning using SoftMax Classifier tabase used: Kaggle's FER2013	ce Expression Recognition Based on Convolutional Neural Network	r database FER2013 Accuracy came out to be 69%

METHODOLOGY

The whole project work is divided into 4 phases that are as follows: -

1) Data Preprocessing
2) Model training
3) Testing process
4) Control interface for smartphones

Data Preprocessing

The selection of the dataset was the first step for our project and also it was important for the accuracy of the project. We started looking for the appropriate dataset for our project. We then tried different datasets which were used previously in various research papers, then we came across MNIST image dataset which has 34,627 images. Each image represent a single 28x28 pixel image with grayscale values between 0-255. After that we need to handle the missing data into the datasets. If some of the data in our dataset is missing, it could be a significant challenge for our machine learning model. Hence it is necessary to manage lost values present in the dataset. This can be accomplished by two ways: by deleting the row or by calculating the mean.

Model Training

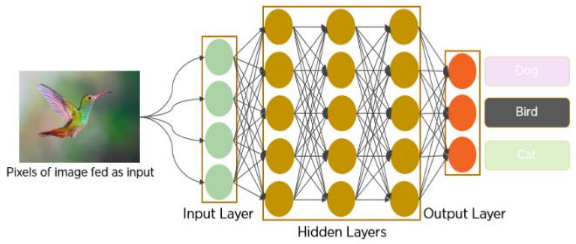
The dataset is originally from the American Sign Language letter database of hand gestures. This dataset has 26 classes of letters (0-25) from A to Z. It's in CSV format which represents labels and values in single rows. Then we load the csv pixels from the file (dataset) in the data frame. Then we apply pre-processing procedures like resizing, reshaping, converting to grayscale and standardize. Then use vectorization by transforming images to NumPy arrays and panda data frames.

Testing Process

1. Implementation for CNN

Convolutional Neural Networks (ConvNet/CNN) are a type of deep neural network that is often used to analyze visual information. They are recognized as change-in-variant or space in-variant artificial neural networks that scan the hidden layers and contain translation in variance characteristics. The filters in most image processing systems are often developed by an engineer using heuristics. CNNs can learn which filter properties are most significant. Because we don't require as many parameters, we save a lot of time and trial and error labor.

The primary distinction between a CNN and a traditional neural network is that CNNs employ convolutions to perform the arithmetic behind the scenes. Convolution is used as an alternative of matrix multiplication in at least one layer of the CNN. Convolutions take two input functions and output one.

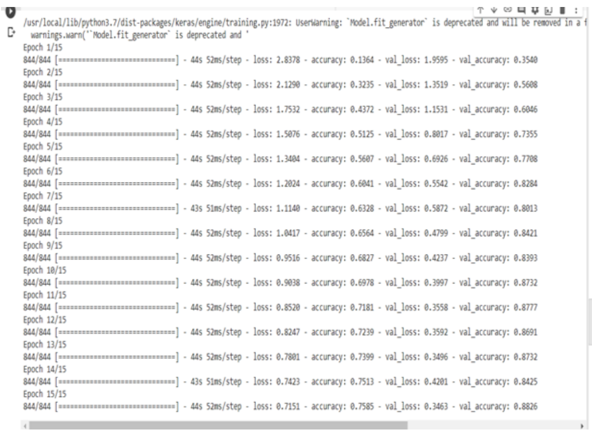


The above figure depicts how the CNN structure works. So, initially, the input picture is supplied into the CNN as preprocessed data, and the image is then passed through many layers. Convolution is the initial layer in which features are extracted from the input picture using mathematical operations. It uses an image filter and a kernel as inputs. After that, the pooling layer will assist in lowering the amount of parameters when the image is too huge, and we may utilise max pooling, average pooling, or sum pooling to do this.

In this model, max pooling was applied. The image is repeated for the convolution layer, where features are retrieved using various ways. The suggested model makes use of ReLU, Leaky ReLU, and LeakyReLU. The Fully Connected Layer (FC) is the final phase in the CNN architecture/structure, in which we flatten our matrix into a vector and input it into a fully connected layer, similar to a neural network.

We were able to successfully implement the CNN model which was proposed by the author in one of the research papers. We divided our datasets in two parts, first one was for training the model which we were developing and second one was for testing the model. We kept the numbers as 27,455 for training and 7,172 for testing out of 34,627 from our MNIST image Dataset. Below is the output for the same.

As author trained his model on 15 epoch ("The number of epochs is a hyperparameter that specifies how many times the learning algorithm will process the full training dataset.") We also did the same and trained our model for 15 epoch. At the end when our model was fully trained, we got an accuracy of 88.26%. Below is the output clearly showing the accuracy of the trained model.

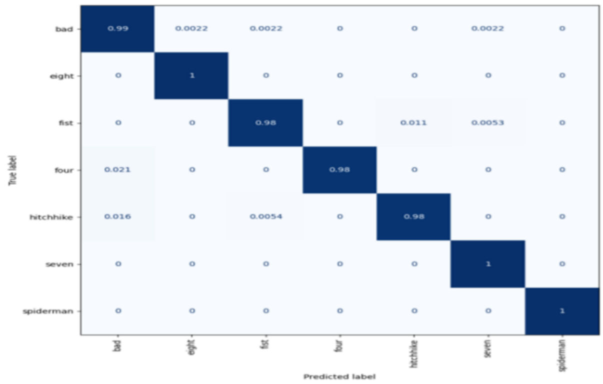


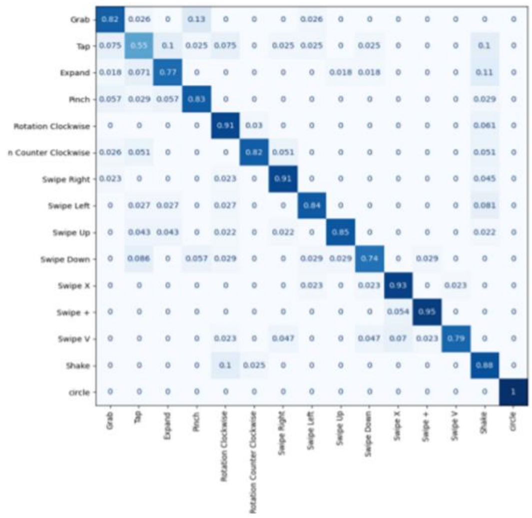
```
/usr/local/lib/python3.7/dist-packages/keras/engine/training.py:1972: UserWarning: `Model.fit_generator` is deprecated and will be removed in a f
  warnings.warn('`Model.fit_generator` is deprecated and '
Epoch 1/15
844/844 [==============================] - 44s 52ms/step - loss: 2.8378 - accuracy: 0.1364 - val_loss: 1.9595 - val_accuracy: 0.3540
Epoch 2/15
844/844 [==============================] - 44s 52ms/step - loss: 2.1290 - accuracy: 0.3235 - val_loss: 1.3519 - val_accuracy: 0.5608
Epoch 3/15
844/844 [==============================] - 44s 52ms/step - loss: 1.7532 - accuracy: 0.4372 - val_loss: 1.1531 - val_accuracy: 0.6046
Epoch 4/15
844/844 [==============================] - 44s 52ms/step - loss: 1.5076 - accuracy: 0.5125 - val_loss: 0.8017 - val_accuracy: 0.7355
Epoch 5/15
844/844 [==============================] - 44s 52ms/step - loss: 1.3404 - accuracy: 0.5607 - val_loss: 0.6926 - val_accuracy: 0.7708
Epoch 6/15
844/844 [==============================] - 44s 52ms/step - loss: 1.2024 - accuracy: 0.6041 - val_loss: 0.5542 - val_accuracy: 0.8284
Epoch 7/15
844/844 [==============================] - 43s 51ms/step - loss: 1.1140 - accuracy: 0.6328 - val_loss: 0.5872 - val_accuracy: 0.8013
Epoch 8/15
844/844 [==============================] - 44s 52ms/step - loss: 1.0417 - accuracy: 0.6564 - val_loss: 0.4799 - val_accuracy: 0.8421
Epoch 9/15
844/844 [==============================] - 44s 52ms/step - loss: 0.9516 - accuracy: 0.6827 - val_loss: 0.4237 - val_accuracy: 0.8393
Epoch 10/15
844/844 [==============================] - 44s 52ms/step - loss: 0.9038 - accuracy: 0.6978 - val_loss: 0.3997 - val_accuracy: 0.8732
Epoch 11/15
844/844 [==============================] - 44s 52ms/step - loss: 0.8520 - accuracy: 0.7181 - val_loss: 0.3558 - val_accuracy: 0.8777
Epoch 12/15
844/844 [==============================] - 44s 52ms/step - loss: 0.8247 - accuracy: 0.7239 - val_loss: 0.3592 - val_accuracy: 0.8691
Epoch 13/15
844/844 [==============================] - 44s 52ms/step - loss: 0.7801 - accuracy: 0.7399 - val_loss: 0.3496 - val_accuracy: 0.8732
Epoch 14/15
844/844 [==============================] - 43s 51ms/step - loss: 0.7423 - accuracy: 0.7513 - val_loss: 0.4201 - val_accuracy: 0.8425
Epoch 15/15
844/844 [==============================] - 44s 52ms/step - loss: 0.7151 - accuracy: 0.7585 - val_loss: 0.3463 - val_accuracy: 0.8826
```

2. Implementation using Mediapipe

Real-time, simultaneous notion of human pose, face landmarks and hand tracking on cell devices can allow a ramification of impactful packages, inclusive of fitness and recreation evaluation, gesture manage and signal language popularity, augmented reality consequences and extra. MediaPipe, an open-supply framework designed particularly for complex belief pipelines leveraging increased inference eg.,GPU or CPUs already offer speed and accuracy. Combining all of them in real-time right into a semantically constant give up-to-cease answer is a uniquely tough hassle requiring simultaneous inference of multiple, structured neural networks.
The MediaPipe Holistic pipeline integrates separate fashions for pose, face and hand components, every of which are optimized for their particular area. However, due to their special specializations, the input to 1 aspect is not properly-perfect for the others. But if one had been to crop the hand and face areas from that picture to bypass to their respective models, the photograph decision could be too low for accurate articulation.

We drew the confusion matrix to display the performance visualization table for the algorithm. It is also known as the error matrix. Below is the confusion matrix for both hand gestures, static and dynamic. In static gestures, we tried numbers like one, two, etc., and some poses like spiderman pose(index and little finger up) and for dynamic gestures, we tried swipe up, swipe down, index and middle finger closing, etc.
We successfully implemented the algorithm in PyCharm and are able to detect hand gestures. We allotted numbers for each important point in the hand which helped us to get good results in detecting dynamic hand gestures. Below is the implementation part for these gestures within the computer frame using the inbuilt webcam. We also made controls for volume of the computer by using index and thumb distance. We framed the initial distance of index and thumb as current volume then, decreasing the distance resulted in decreasing the volume of the computer and vice-versa.





Control interface for smartphones
Currently our model is able to detect hand with great accuracy and also some functions are working pretty good. Next thing we are going to do is to expand the model to do more tasks with good accuracy and no latency. We will define every function a user needs for a control interface. Then we will make a framework for smartphones. Whatever a user is doing with the touch interface will be there in the hand gesture interface.

Conclusion

We initiated our project by reading research papers. After going through each paper thoroughly we began gathering information like which algorithms did the authors use and what was the accuracy of the proposed algorithms. After studying 14 research papers we completed the Literature Review and started our implementation part. Our entire project workflow was divided into 4 phases that starting with Data Preprocessing, Model Training, Testing process and then lastly designing control interface for smartphones. After acquiring the data and training the model we researched on various techniques through reading research papers and at last we came to the conclusion that we would be implementing our model using two approaches that is CNN and using Mediapipe. We were able to successfully implement CNN model and at the end when our model was fully trained, we got an accuracy of 88.26%. Like CNN only we were also able to detect hand gestures using Mediapipe.

The main purpose of our project is to make a non-touch control interface using hand gestures. For this we are continuing the mediapipe approach for our project as it is providing greater accuracy and no latency. We have also tried some of the functionalities a user needs for a control interface like controlling the volume and swiping screen. We will provide every functions for a control interface for smartphones.

Future work

We will extend our model to a control interface which could be able to detect hand gestures and do the tasks based on that. For this, we will design a framework that could provide good accuracy and no latency for smartphones as it might be the future of our technical era. The user will be able to do every task he/she is currently doing with the use of touch based control interface like playing games, volume control, typing. etc.

References

[1] Perveen, N., Ahmad, N., Khan, M. A. Q. B., Khalid, R., & Qadri, S. (2016). Facial expression recognition through machine learning. *International Journal of Scientific & Technology Research*, *5*(03).

[2] Kodhai, E., Pooveswari, A., Sharmila, P., & Ramiya, N. (2020, July). Literature Review on Emotion Recognition System. In *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)* (pp. 1-4). IEEE.

[3] Modi, S., & Bohara, M. H. (2021, May). Facial Emotion Recognition using Convolution Neural Network. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1339-1344). IEEE.

[4] Yang, G., Ortoneda, J. S. Y., & Saniie, J. (2018, May). Emotion Recognition Using Deep Neural Network with Vectorized Facial Features. In *2018 IEEE International Conference on Electro/Information Technology (EIT)* (pp. 0318-0322). IEEE.

[5] Ruiz-Garcia, A., Elshaw, M., Altahhan, A., & Palade, V. (2018). A hybrid deep learning neural approach for emotion recognition from facial expressions for socially assistive robots. *Neural Computing and Applications*, *29*(7), 359-373.

[6] Mehendale, N. (2020). Facial emotion recognition using convolutional neural networks (FERC). *SN Applied Sciences*, *2*(3), 1-8.

[7] Abd El Meguid, M. K., & Levine, M. D. (2014). Fully automated recognition of spontaneous facial expressions in videos using random forest classifiers. *IEEE Transactions on Affective Computing*, *5*(2), 141-154.

[8] Zhang, L., Jiang, M., Farid, D., & Hossain, M. A. (2013). Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot. *Expert Systems with Applications*, *40*(13), 5160-5168.

[9] Li, Z., Imai, J. I., & Kaneko, M. (2009, October). Facial-component-based bag of words and phog descriptor for facial expression recognition. In *2009 IEEE International Conference on Systems, Man and Cybernetics* (pp. 1353-1358). IEEE.

[10] De Silva, L. C., & Hui, S. C. (2003, December). Real-time facial feature extraction and emotion recognition. In *Fourth international conference on information, communications and signal processing, 2003 and the fourth pacific rim conference on multimedia. Proceedings of the 2003 Joint* (Vol. 3, pp. 1310-1314). IEEE.

[11] Astler, D., Chau, H., Hsu, K., Hua, A., Kannan, A., Lei, L., ... & Zhang, X. (2012). *Facial and expression recognition for the blind using computer vision* (Doctoral dissertation).

[12] Shan, C., Gong, S., & McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and vision Computing*, *27*(6), 803-816.

[13] Soyel, H., Yurtkan, K., Demirel, H., & McOwan, P. W. (2016). Brain MR image denoising for Rician noise using intrinsic geometrical information. In *Information Sciences and Systems 2015* (pp. 275-284). Springer, Cham.

[14] Xu, L., Fei, M., Zhou, W., & Yang, A. (2018, December). Face expression recognition based on convolutional neural network. In 2018 Australian & New Zealand Control Conference (ANZCC) (pp. 115-118). IEEE.