International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-9, July 2020

Vision-Based Hand Gesture Recognition Techniques using Smartphones

Sachin Devangan, Omkar Joshi, Shanu Jaiswal, Apratim Gholap, Netra Lokhande

Abstract: In the past few years, the computational performance of smartphone devices has seen tremendous growth. Due to which the smartphone has become a suitable platform for various computer-vision based applications which earlier was not possible. In this paper, we study various methods through which we can achieve computer vision-based hand gesture recognition natively on smartphones. If smartphones can support hand gesture recognition it can provide a new way to interact with mobile devices and overcome the hurdles of voice and touch-based user interface improving the user experience at the same time also supports other gesture-based applications. The techniques we study are mainly vision-based since camera module is present on most of the smartphones and it does not require other additional sensors or other hardware. We have compared the various methods available based on algorithms used and corresponding accuracy.

Keywords: Hand gestures, Hand Gesture Recognition (HGR), Smartphone, interface, user experience, Sign Language.

I. INTRODUCTION

In recent years smartphones have become one of the most essential mobile computing devices due to the wide range of functionality and features encompassed by them. Statistics reveal that about 2.7 billion people own smartphone in 2019 and is destined to increase in upcoming years i.e. Every third person in the world owns a smartphone. As a result of this increased demand for smartphones has pushed smartphone manufacturers to inculcate new features, add more hardware, and invest heavily in R&D which has resulted in a new era of smartphones. Today smartphone's system-on-chip (SOC) computing power has risen to such levels that it could challenge even desktop computers released not so long ago. There have been vast improvements in smartphone hardware such as Camera modules, CPU, GPU, Display, Better sensors, and the list goes on.

Revised Manuscript Received on July 30, 2020.
***** Correspondence Author

Sachin Devangan*, Department of Information Technology, MIT College of Engineering Pune, India.

Omkar Joshi, Department of Information Technology, MIT World Peace University, Kothrud, Pune, India.

Shanu Jaiswal, Department of Information Technology, MIT World Peace University, Kothrud, Pune, India.

Apratim Gholap, Department of Electrical Engineering, MIT World Peace University, Kothrud, Pune, India.

Dr Netra Lokhande, Associate Professor, Department of Electrical Engineering, MIT World Peace University, Kothrud, Pune, India.

Smartphone manufactures have realized that Artificial Intelligence is way forward and have started adding dedicated AI cores, NPUs, GPUs, Dedicated signal processing units (DSP) to enable smooth AI and deep-learning-based computations. The addition of these A.I centric components to smartphones SOC has led to the drastic improvement in computer vision tasks such as image classification, image enhancement, object tracking, image preprocessing which are fundamental to implement fast and efficient vision-based HGR algorithms. Smartphones make use of touch screen display and voice as the primary mode of Human-computer interaction which involves human voice-based commands or using fingers or any other pointing device to interact with touch screen display. Using touch as a means of HCI comes with its own limitations. Smartphone screen size is small due to which performing touch-based gestures also get restrained. There is also a limitation of physical contact which might prove difficult in some situations like in case the hands are wet or the smartphone is out of reach. Voice-based interaction becomes difficult when the background environment is noisy and natural language processing (NLP) is still not very accurate when it comes to various regional-based dialects. Hand gestures based HCI overcomes the obstacles faced by these methods and can drastically change how we interact with smartphones and provide a better end-user experience.

Hand gesture recognition (HGR) also acts as a tool to identify hand gestures of speech-impaired people which can be further translated into voice so that normal people can easily understand. The main barrier to this application is the portability and cost of hardware. Vision-based HGR implemented on smartphones overcomes both of these hurdles. As smartphones are readily available and we don't need any additional hardware as smartphones consist of cameras, necessary computing resources, microphones for voice are available on a smartphone.

In this paper, we are going to review various techniques to implement vision based HGR using a smartphone and study their limitations and scope of improvements

II. LITERATURE REVIEW OF VISION-BASED HGR FOR SMARTPHONE

Vision-based HGR on smartphones involves capturing images or video using the camera of a smartphone as the first step. Then acquired data can be processed in two ways: the smartphone itself or on the cloud. Both methods have their own merits and demerits.

1. Processing data locally on the smartphone:

In this method, we process acquired data locally on the smartphone SOC itself.

We need to make use of modified HGR techniques so that they are suitable for running effectively with smartphones related constraints such as limited computing power, low power use as they use a battery, compatibility with Mobile CPU architecture (ARM, ARM64, x86), etc. Processing data locally allows us to avoid issues such as network problems, Requirement of internet, latency, reduced cost of application, etc.

2. Processing of data on cloud:

In this technique, data is processed on cloud instead of smartphone SOC. Processing of data on cloud has allowed us to run computationally heavy algorithms, have access to virtually unlimited resources due to scalability property of cloud, low power usage as all the resource-heavy computation is performed on the cloud instead of the device itself. But when we make use of the cloud for performing the computation, we face certain difficulties such as the requirement of having an active internet connection, latency issues, difficulty to resolve errors, increased complexity due to the addition of the cloud layer.

In this literature review, we are going to focus on processing data locally on smartphone SOC. Both the methods consist of the same steps for processing of data which are:

Image Frame Acquisition, Hand Tracking, Feature Extraction, Classification, and Output Gesture illustrated in below figure 1.

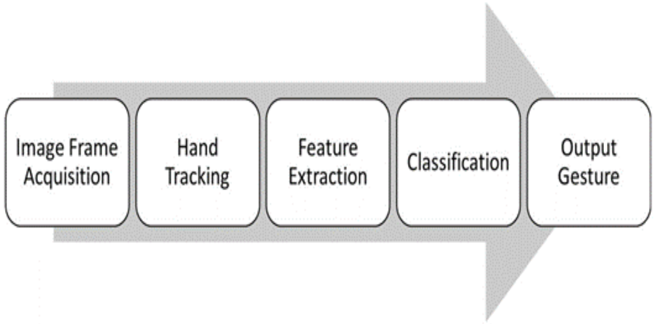


Fig 1: Block diagram for hand gesture recognition system

2.1 Image preprocessing:

The first step for a hand gesture recognition system using a smartphone device is Image preprocessing.

Pre-processing is performed on images at the lowest abstraction level, with both input and output intensity images. Such representations are the same as the original data collected by the sensor or a camera, with an image of intensity normally represented by an image function values matrix. Pre-processing is aimed at improving the image data which suppresses distortions or enhances some image function

Pre-processing is used to hand gesture recognition to capture and interpret gestures or signals, thus gesture recognition is one of the most complex and difficult tasks in the processing of images. The captured images are subject to the process of image processing. The processing steps consist of image acquisition, pre-processing, segmentation, extraction of features and finally classifiers. Coupled with machine learning methods, a higher image processing accuracy is achieved

In [10] Aishwarya Danoji developed an application 'MonVoix' for hand gesture recognition for visually challenged people. The mechanism behind this includes a smartphone camera detecting an ASL gesture and translating image into text using image processing techniques to recognize images, removing the need for a human translator. Image processing has been done for various alphabets and numbers. The images are captured using a smartphone camera and the images were altered from RGB to Grayscale to get region of interest (ROI). Further using Gaussian blurring and edge detection the gesture was detected and noises were removed. The histogram is taken of the transformed image. Further Fast Fourier Transform (FFT) was applied to give a number of pixels present for the frequencies present in the image. The captured image gave 91.7% accuracy when compared to the training image.

In [11] Hemapriya T. presented a few pre-processing techniques for coin image recognition and compared them. They were mainly Gaussian filter, Wiener filter, median filter and mean filter. Upon comparison she came to a result that Wiener filter was the best pre-processing technique which reduced noises and blurs from the photos captured by digital cameras. Wiener filter was used to make a target region via a noisy process which makes use of a linear time invariant (LTI) filtering. According to the author, the images need to be enhanced to extract valuable pixels which leads to the final pixel being evaluated to remove noise from the image.

In [12] S.Perumal proposes various image pre-processing and filtering techniques to avoid unwanted noise and improve image quality of lung scan images. The author proposed the following techniques in an experiment to determine the most effective way- Wiener filter, Median filter, Gaussian low and high pass filter. The original image (227x222) were divided into 3 images of 128*128,256*256,512*512 images and the methods were tested on these images. The results showed that Gaussian low and high pass filter showed clear images, Wiener filter showed dark images and median filter showed bright images. Upon observation the author states that the median filter showed the best pixel result. Median filter operates on a rectangle region and adjusts image size when filtering. The filter output is a single value which replaces the current pixel value which was x & y

In [13] Mahesh M. proposed a method BRIEF (Binary Robust Independent Element Features) which compared gestures based on reducing the CPU time which worked as a sign language translator application for mobile devices. For this method, the pre-processing step played a major part. A couple of pre-processing steps were followed to get a more accurate result. Skin Detection was done using RGB, YCbCr, HSI methods. Once a method confirmed a pixel as a skin color pixel, it was flagged, and the flag variable was incremented. When all the 3 methods confirmed the skin color pixel, the corresponding pixel was chosen as skin color pixel. Different alphabets and numbers were tested and were tested against the given dataset and gave values 1 or 0 if the images matched with those values from dataset. When these methods were implemented on an android device and the accuracy was calculated, it turned out to be 70 percent accurate for static hand gesture, it could not process dynamic hand gesture because video processing was needed

2.2 Hand Tracking

After the image is captured using a smartphone camera the next step is preprocessing the image. This step is crucial to improve the performance and accuracy of our HGR model. The next step after image preprocessing is Hand tracking also known and hand segmentation or hand detection. In this step, we try to crop the hand from the background noise. Various different methods are proposed for hand tracking.

Recently Google released an open-sourced AI algorithm that is capable of recognizing hand shapes and motion in real-time using MediaPipe [2]. It is a cross-platform framework that identifies 21 3D key points of hand from a single frame of an image. This method supports real-time hand detection and allows scaling to multiple hands on a mobile phone. It makes use of three AI models namely BlazePalm, hand landmark model, and gesture recognizer.

In research [3] H.Lahiani et al. propose to combine HOG and LBP features and classify them with the help of the AdaBoost classifier. Independently HOG and LBP feature-based classifiers that have their own advantages but by cascading the features we are able to combine both their unique abilities such as energy-saving characteristics of HOG and faster computation supported by LBP and also the resulting algorithm achieves better accuracy as compared to individually using HOG and LBP.

In research [4] Chun Yu et al. propose a novel approach to detect hand gestures using the front camera of smartphones with the help of stereo vision by placing a right-angle prism on the front camera. This technique allows capturing hand gesture right above the screen surface

In research [5] Twinkle Sharma et.al propose using OpenCV library algorithms to detect hand gestures on mobile devices. OpenCV library has been extensively used for desktop-oriented hand gesture recognition due to its rich variety of 2500 machine learning algorithms. Earlier OpenCV oriented implementation was avoided on mobile devices because these algorithms are computationally very costly but with increasing computation power on mobile devices, they too can take advantage of using the OpenCV library. This proposed method makes use of the native android camera. It consists of three stages matrix allocation, frame processing, and gesture extraction and assimilation stage.

In research, Bryan G. Dadiz et.al [14] solved the problem of slow hand gesture tracking by using the KLT algorithm to track hand gestures by providing it with the limited features vector extracted using the OpenCV function instead of tracking all pixels in the resulting image and within a given foreground object.

2.3 Features Extraction:

The initiation of the process of feature extraction is done on a set of measured data to build values which are non-recurrent, detailed and aid the steps in generalizing and subsequent learning processes of machine learning, image processing and pattern recognition. Dimensionality reduction is linked to feature extraction. Reduction of features is performed on an input data that is too large to be processed by an algorithm, to generate reduced sets of features. This can be termed as feature extraction. These extracted features have necessary information from the entire input dataset, which can be utilized to do the required tasks on a representational data rather than the complete data.

The palm was derived in this work on the basis of mathematical color models [12]. On the foundation of the trained sample, a model in RGB-H-CbCr colored ranges was prepared. The hand possibility object was then thresholded. After anatomical closure, the tag of the related elements was performed to obtain the core of the binary object and the two different dimensions of the pixels. Recognition of the hand position was done by Gabor Filter.

Contour Extraction and Polygon Approximation were the processes used by Raimundo [26] to achieve feature extraction. Suzuki and Be [23] had described a technique to accomplish this. In this, contour points are calculated for each element on the image. This could however be noisy if the quality of image is low. Polygonal approximation is performed to solve the problem as elaborated in Ramer [24]. This process iteratively discards the points that are above the epsilon distance from the mean contour. Hence, the threshold epsilon value must be such that the polygon created must be similar to the initial contour, representing subtle simplifications.

The procedure followed by C. Jian [25] was focused on smartphones where they proposed that barely any functionality could be removed in order to develop better identification accuracy. Elevated-dimensional interfaces designed to direct knowledge dismissal and raise conceptual entanglement. Frequent features of the hand trajectory include various characteristics of place, inclination of direction and velocity.

They adopt the direction angle as the succession feature. If there were numerous points placed interminably less than the average value, they have been defined as dotted regions. The original dotted region and the final event of the dotted region were established as the origin and the conclusion of the trajectory.

When the gap from the last center of mass point the preceding juncture is well below the normal location, the most recent center of the mass area shall never be considered for a route point. Ultimately, the path angle in each movement where the center of mass was known to be the core of the point. The final stage of the course was then attained. Then the advancement of the route function was developed in the end.

In [14] Byyan G. Dadiz proposed using Kanade-Lucas-Tomasi (KLT) methodology for exploring a simpler method to manage commands across the smartphone by using visual recognition techniques. Then the KLT algorithm made use of feature extract points to uncover fixed hand movements. It is also referred to as Go-Motion. Use the KLT function to monitor the algorithm. It gets the framework to track hand and finger movements using almost any smartphone's front camera. Further, using OpenCV and android NDK, languages like C and C++ were implemented into smartphones for better functioning. Upon testing the mean of functioning criteria was set to 4.2. The criteria were accuracy, functionality, usability, portability, efficiency, maintainability.

The total mean of those criteria came out to be 4.41 which means that using KLT shows a very satisfactory result

2.4 Classification

Recently the trend has shifted towards the use of convolution neural network for classification tasks as CNN allows to process Real-Time data, dynamic gesture recognition, process video, requires no tedious feature extraction but CNN have their own caveats they require a large number of resources to train a model for these tasks but there has been the advent of CNN algorithm which are optimized to use lesser resources and can be implemented on mobile platform In research [9] Andrew G Howard et.al proposes MobileNets for mobile and embedded devices which are built using depth wise separable convolutions which is extremely efficient as compared to standard convolution. It uses an additional layer called 1*1 convolution or pointwise convolution that gives it the ability to create new features. MobileNet consists of 3*3 depth wise separable convolutions which uses 8-9 times fewer computation resources than standard computation by the relatively small tradeoff of accuracy. MobileNets further utilize two parameters namely Width multiplier (α) and Resolution Multiplier(ρ). Width multiplier is used to achieve a smaller model, it does it by thinning the network uniformly at each layer. Resolution Multiplier reduces computational complexity by multiplying parameter ρ to the input image and the internal representation of every layer is subsequently reduced. These two global parameters help to achieve a tradeoff between latency and accuracy.

In research [6] have made use of pre-trained model based on SqueezeNet to classify American Sign Language(ASL) Hand Gesture to the corresponding alphabet the primary objective of squeeze net is to reduce the parameters required to train the model, in turn, reducing the memory footprint which allows executing these models efficiently on mobile platforms. The model used here is fed an input image which is first processed by dividing every pixel by 255 and thereby resizing the image to 244*244 (bottleneck approach)

CNN usually requires heavy computation resources and memory resources for storing large datasets both of which are limited on a smartphone which makes it difficult for implementing CNN on smartphones. There are also cases when we have smaller size datasets available. Training CNN over this small size dataset can lead to an increased risk of overfitting to tackle all of these problems. There are various modified CNN based classification techniques suitable for smartphones.

In research [7] Zhi lu et.al proposed a new framework for One-shot Learning hand gesture Recognition (OSLHGR) which is based on modified Inflated 3D ConvNets(I3D) [8] for smartphones. OSLHGR modified I3D to reduce storage consumption and improve processing speed. OSLHGR uses modified lightweight I3D for the extraction of spatiotemporal (consist of both time and space) features. On this basis, Multimodal feature fusion based on canonical correlation analysis was performed to gain useful knowledge that can be transferred and utilized to initialize the target network trained on the acquired dataset to reduce the risk of overfitting and fasten the convergence of the network. A voting mechanism based on multiple nearest neighbor classifiers was used to further improve classification results

III. APPLICATIONS FOR HAND GESTURE RECOGNITION

3.1 Sign Language:

Sign Language is a form of non-verbal communication in which a person makes a certain touch to communicate. Keeping in mind the similarities of the human fingerprint and the thumb, the hand recognition is carried on the basis of the acquisition of other artifacts such as familiarity, Center of mass centroid, finger position, thumb in the raised position or the fingers of the wrists. There are many suggested programs to see sign language benefit using a variety of sign languages. For example, Setiawardhana et al. [25] The recognized Indonesian Sign Language program. In [20] S. Swamy et al. develop a program that recognizes Indian Sign Language. The program proposed by [21] Prasuhn et al.is was used to recognize American Sign Language.

3.2 Automated Homes

Nowadays home use is affecting the value. Gestational awareness refers to observing the movements of the human body such as hands, face, etc. Most of the electronic components focus on a hands-free base. Due to the use of gestures using things such as changing a legacy to use touch becomes much easier. This touch can also work on changing room lights or turning on electronic items such as TV, AC, Refrigerator etc. [22] P. N. Arathi et al. respected home automation applications.

3.3 Medical systems and assistive technology

These types of hand gestures can be used in the medical field as while physical activity can be helpful and efficient. These gestures can be helpful such as a robot arm can be used to help with the operations, and they can be accessed with the help of gestures. [23] Hongyang Zhao et al. used to monitor the impact in the Medical field

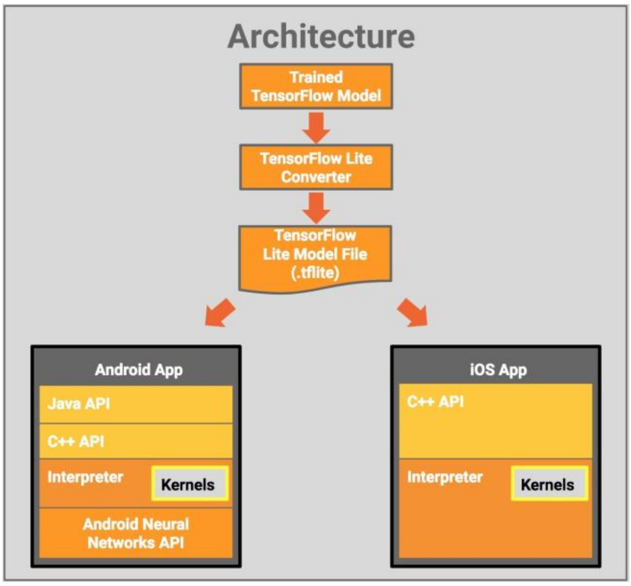
3.4 Gaming:

In a video game design, touch is used by game commands instead of pressing buttons on the keyboard or moving the mouse. In these communications, unintentional and continuous movement gestures should be set up to provide the user with a more natural display. In [13] Mahesh B. Mariappan et al. developed a program called “PicoLife” that creates a virtual reality game where 3D characters are controlled by gestures on an Android smartphone

3.5 Driving

Touch level when driving is also high as this gesture display can be used to assist the driver while parking or also while driving. Since most cars have cruise control on them. Body touch can be used to activate cruise control. Also, touch can be used to control the vehicle environment as a control. [24] Pavlo Molchanov et al. is used to monitor Drivers hands while driving.

IV. PROPOSED METHODOLOGY



The proposed target is to create a hand gesture recognition system, which can be used on smartphones. The basic idea is to run a hand gesture recognition model using TensorFlow Lite on Android / IOS devices.
The proposed methodology follows this step: -
1. Implementing the necessary datasets on the TensorFlow framework using CNN (Convolutional Neural Network) algorithm. TensorFlow is a Python friendly open source library/framework for numerical computations that makes machine learning easier.

2. But for our scope, we need to make it work on a smartphone system as the TensorFlow Lite converter can optimize the model to be compatible with smartphone specifications as a fully trained neural network can be extremely heavy for a mobile phone. For this, we will use ONNX, which will convert the TensorFlow model to TensorFlow Lite, which the smartphone systems can support.

3. Setting up the Skeleton of the App using Flutter will be our next step (The reason for choosing Flutter is that Flutter and TensorFlow are both developed by Google, so the Compatibility is extremely well as well as it enables cross-platform development for iOS and Android)

4. After the conversion of the model to TensorFlow Lite, we will add the model to the App.

5. The Next Step would be to initialize a TensorFlow Lite interpreter, after the setup of our TensorFlow Lite interpreter we will be using the TensorFlow Lite Support Library to simplify the image pre-processing.

6. Now, we will test run the app on an actual device or in the Android Studio Emulator (Additionally, we can accelerate inference with GPU delegate)

7. The final step would be the deployment of the app on an actual smartphone

V. CONCLUSION

After reviewing various different vision-based hand gesture recognition techniques we have come to the conclusion that implementing hand gesture recognition based techniques is feasible with results showing higher accuracies when dataset was trained with CNN algorithm and the provided method could be improved by developing an application on a smartphone which would help the users in day to day life. One of the applications in day to day life extend to a sign language translator for visually impaired people and another application being VR based applications for smartphones where each gesture would relate to an action being performed in the virtual environment.

REFERENCES

1. "•Number of mobile phone users worldwide 2015-2020 | Statista." [Online]. Available: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/. [Accessed: 20-Dec-2019].
2. "Google AI Blog: On-Device, Real-Time Hand Tracking with MediaPipe." [Online]. Available: https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html. [Accessed: 17-Dec-2019].
3. H. Lahiani and M. Neji, "Hand gesture recognition method based on HOG-LBP features for mobile devices," *Procedia Comput. Sci.*, vol. 126, pp. 254–263, 2018.
4. C. Yu *et al.*, "Handsee: Enabling Full Hand Interaction on Smartphones with Front Camera-based Stereo Vision," *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 1–13, 2019.
5. T. Sharma, S. Kumar, N. Yadav, K. Sharma, and P. Bhardwaj, "Air-swipe gesture recognition using OpenCV in Android devices," *2017 Int. Conf. Algorithms, Methodol. Model. Appl. Emerg. Technol. ICAMMAET 2017*, vol. 2017-January, pp. 1–6, 2017.
6. N. Kasukurthi, B. Rokad, S. Bidani, and D. A. Dennisan, "American Sign Language Alphabet Recognition using Deep Learning," 2019.
7. Z. Lu, S. Qin, L. Li, D. Zhang, K. Xu, and Z. Hu, "One-Shot Learning Hand Gesture Recognition Based on Lightweight 3D Convolutional Neural Networks for Portable Applications on Mobile Systems," *IEEE Access*, vol. 7, pp. 131732–131748, 2019.
8. "Exercise Classification with Machine Learning (Part II)." [Online]. Available: https://towardsdatascience.com/exercise-classification-with-machine-learning-part-ii-d60d1928f31d. [Accessed: 28-Dec-2019].
9. A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017.
10. Kamat, R., Danoji, A., Dhage, A., Puranik, P., & Sengupta, S. (2016). MonVoix-An Android Application for the acoustically challenged people. Journal of Communications Technology, Electronics, and Computer Science, 8(November), 24. https://doi.org/10.22385/jctecs.v8i0.123
11. Hemapriya, T., K. S. Archana, and T. Anupriya. "Comparison of preprocessing techniques for coin recognition using image processing methods." *International Journal of Engineering & Technology* 7.2.21 (2018): 351-354.
12. Perumal, S., and T. Velmurugan. "Preprocessing by contrast enhancement techniques for medical images." *International Journal of Pure and Applied Mathematics* 118.18 (2018): 3681-3688.
13. Mahesh, M., Jayaprakash, A., & Geetha, M. (2017). Sign language translator for mobile platforms. 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017-January 1176–1181.
14. Dadiz, B. G., Abrasia, J. M. B., & Jimenez, J. L. (2017). Go-Mo (Go-Motion): An android mobile application detecting motion gestures for generating basic mobile phone commands utilizing KLT algorithm. 2017 IEEE 2nd International Conference on Signal and Image Processing, ICSIP 2017, 2017-January 30–34.
15. Haseeb, M. A. A., & Parasuraman, R. (2018). Wisture: Touchless Hand Gesture Classification in Unmodified Smartphones Using Wi-Fi Signals. IEEE Sensors Journal, 1–1.
16. Lahiani, H., & Neji, M. (2018). Hand gesture recognition method based on HOG-LBP features for mobile devices. Procedia Computer Science, 126, 254–263. doi: 10.1016/j.procs.2018.07.259

17. Núñez, J. C., Cabido, R., Pantrigo, J. J., Montemayor, A. S., & Vélez, J. F. (2018). Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition. Pattern Recognition, 76, 80–94. doi: 10.1016/j.patcog.2017.10.033
18. Devineau, G., Moutarde, F., Xi, W., & Yang, J. (2018). Deep Learning for Hand Gesture Recognition on Skeletal Data. 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). doi:10.1109/fg.2018.00025
19. Alani, A. A., Cosma, G., Taherkhani, A., & McGinnity, T..(2018). Hand gesture recognition using an adapted convolutional neural network with data augmentation. 2018 4th International Conference on Information Management (ICIM).
20. Swamy, S., Chethan, M.P., Karnataka, M.: Indian sign language interpreter with android implementation. Int. J. Comput. Appl. 97(13), 36–41 (2014).
21. Prasuhn, L., Oyamada, Y., Mochizuki, Y., Ishikawa, H.: A HOG-based hand gesture recognition system on a mobile device. In: IEEE International Conference on Image Processing (ICIP), pp. 3973–3977 (2014).
22. P. N. Arathi; S. Arthika ; S. Ponmithra ; K. Srinivasan ; V. Rukkumani. : Gesture based home automation system. In: 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2).
23. S. Suzuki and K. Be, “Topological structural analysis of digitized binary images by border following,” Computer Vision, Graphics, and Image Processing, vol. 30, no. 1, pp. 32–46, 1985.
24. U. Ramer, “An iterative procedure for the polygonal approximation of plane curves,” Computer Graphics and Image Processing, vol. 1, no. 3, pp. 244–256, 1972
25. C. Jian, M. Yang and M. Zhang “Mobile terminal trajectory recognition based on improved LSTM model”.
26. Raimundo F. Pinto Jr, Carlos D. B. Borges, Antoˆnio M. A. Almeida and Ia´lis C. Paula Jr”Static Hand Gesture Recognition Based on Convolutional Neural Networks”.

AUTHORS PROFILE

Sachin Devangan is currently pursuing his Final year B.E. in department of Information Technology in MIT College of Engineering Pune. His area of interest is Machine Learning, Artificial Intelligence. He has a Total of 2 research paper published in his domain.

Omkar Joshi is pursuing BE in Information Technology final year at MIT College of Engineering, Pune. Passionate about IoT and ML.

Shanu Jaiswal is currently pursuing BE in Information Technology final year at MIT College of Engineering, Pune. Passionate about ML and data science

Apratim Gholap is pursuing BE-IT final year at MIT College of Engineering, Kothrud, Pune. Interested in Computer Vision, Edge Computing, Deep Neural Network and Natural Language Processing.

Dr Netra Lokhande is currently an Associate Professor at School of Electrical Engineering, MIT World Peace University, Kothrud, Pune, India. She has an Academic experience of 23 years. Her areas of research are Image Processing, Power Systems, Power Electronics, IOT. She has published 32 papers in total