iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VI **Month of publication:** June 2020

DOI: http://doi.org/10.22214/ijraset.2020.6211

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

Hand Gesture Recognition for Emoji Prediction

Jatin Gupta[1], Shreya Goel[2], Sanyam Varshney[3], Ms. Shafali Dhall[4], Ms. Neha Gupta[5]
[1, 2, 3]*Student,* [4,5]*Assistant Professor, Department of Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi*

Abstract: ***Emojis are ideograms and smileys visual symbols that are used widely in wireless communication. They present rich novel possibilities of representation and interaction as a new modality. They exist in various genres, including hand gestures, human faces, figures, and signs. Hand gestures, which are the most common and intuitive non-verbal means of communication when we are using a computer, and related work, has recently sparked an interest. Hands often appear in images, videos, and their appearances and pose can give important clues about what people are doing. A combination of hand gestures and emojis can communicate and express the message very conveniently. Considering the positive outcomes of image recognition from specific deep learning methods, we suggest an emoji predictor in real-time. This project consists of a hand gesture recognition method and emoji generator using filters to detect hands and Convolutional Neural Network (CNN) for training the model. Here, a database is being created of hand gestures to train the system. The prediction will be focused on the hand movements and capture the changes by preparing for different hand movement positions to the highest degree of precision.***

Keywords: Deep Learning, Convolutional Neural Network (CNN), Hand Detection, Gesture Recognition, Morphological Operation, Contours, Gaussian Blur, Emoji Prediction.

I. INTRODUCTION

Deep learning in Machine Learning is, at present, a hot subject. Deep learning is good at recognizing images, detecting objects, translating natural languages, and predicting trends. It uses several layers to slowly remove functionality at a higher level from the raw data. For example, lower layers may identify edges in image processing, whereas upper layers may identify factors that are crucial to a particular human person, such as digit numbers or symbols or faces. Human beings can clearly understand body language and sign language. The research field of hand recognition and gesture recognition seeks to identify these gestures, which usually include particular stance and movement of the body's hands, arms, eyes, or even skeletal joints [1]. With the advancement of computation and the facility of exposure to emerging technology, it has become easy to understand human expressions and to anticipate emoji that correspond to the gesture. The methodology of activity identification is split traditionally into two groups: dynamic or static [2]. The Static hand gestures method only enables the computation of a single object as the input data of the classification model. The dynamic movements involve image sequencing encryption. Several methods and techniques focused on unsupervised and supervised learning are considered in the literature. Many sources, including neural networks [3], deep convolutional neural network [4], and support for vector machine algorithms [5] - [8], are available.

Emoji use has become another form of social communication, which is essential considering that, for example, chat apps may enhance an integrated framework. They have been an integral part of human life to express our thoughts and emotions. As we see an increased use of emojis, the way a human expresses his feeling in a different body and sign language, mostly using hand gestures, where arises a need for an instant generator of emojis based on gestures [9].

In this work, we used image bases of 11 gestures and filters to detect hands and to utilize convolutional neural networks (CNNs) for categorizing. With the suggested approach, we have shown that the incredible results are obtained with underlying architectures of deep convolutional neural networks for the classification of static gestures [10]. We will provide a brief overview of the methods we used, our suggested approach, and the experiments we performed. The final parts of this work demonstrate the outcomes we obtained and our conclusions and insights for future research.

II. RELATED WORK

The gesture recognition and Emoji prediction built in this project involve a series of steps: mainly hand detection and removal of context, gesture recognition, emoji prediction, and control of device behavior. A Gesture recognition method using convolutional neural networks includes the primary step as taking an input image. It is taken from the web camera, which is an RGB image consisting of 3 channels with intensity levels ranging from 0 to 255 and consists of many objects with hand images [11], [12]. Then, Hand detection and background removal are indispensable to gesture recognition. There are several approaches proposed for conducting hand detection. Most of them are based on shape [11], [13] - [14], colour [12], and Haar features [7], [15].

These methods work well if we impose certain limitations on the environment for the detector to find out the hand. Both of these approaches, however, have their drawbacks in realistic situations where the context environment can be cluttered, alterable, and unpredictable. For example, color-based strategies can become useless when people wear gloves, or when the background color or combination is too close to the face. We must segment the hand region from the context so that the gesture recognition algorithm can work properly. The hand segment is retrieved from the background by using the background subtraction method [8], [12] - [13], [16]. Hand gesture detection algorithms include neural networks, support vector machines [6] - [8], and Adaptive Boosting (AdaBoost), which are based on various machine learning approaches. Among these methods, AdaBoost-based hand-region detectors have a Haar-like characteristic to fabricate the detector vigorous [7], [15]. Color segmentation using an MLP network, morphological erosion, and closure operations for the Image processing stage, is used to remove background image noise [14] - [15]. The convolutional network achieves better results on images pre-filtered by a Gabor filter [17]. The Kalman filter can also be used to estimate the position of the hand based upon which mouse cursor is controlled in a stable and efficient manner [18]. SIFT filters could also be used to recognize the hand through intrigue concentrates and the creation of a vector-like the descriptor [19]. Filters like SIFT provide several image highlights which do not rely on various factors, such as object scalability and rotation [8]. Not only is the precision of gesture recognition, but even the efficiency of noise suppression defines the intensity of the gesture-based engagement process. The noise here is not necessarily caused by the cluttered background, camera blurring, or some other external factors but is often caused by the gesture of the hand itself and is inescapable.

III. CONCEPTS USED

A. *Morphological Operation*

An order of operations that processes an image based on its shapes. Morphological operations apply structural elements to an input image and form an output image [20]. The use of morphological filters is a standard method for extracting components of the image and helps to represent and define the forms. Those filters are used to remove or isolate resources from segmentation during image processing, minimizing noise, as seen in Figure 1. Two basic operations that describe them are erosion and dilation; other essential operations are open and close [20]. First of all, we had to make sure the structure of the hand was right. Erosion destroys foreground target boundaries, which are used to decrease an image 's attributes. Erosion is followed by Dilation. Erosion removes white noise but shrinks the object [21]. So, it is dilated afterward. The noise from the background is removed due to which the object area increases. Closing is obtained by the dilation of an image followed by an erosion, while vice versa happens in Opening.

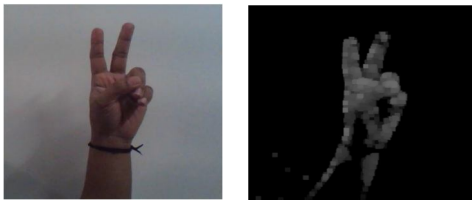


Figure 1: Morphological operation on detected image

B. *Contour Extraction*

Contours can be easily described as a curve of the same color or intensity at all consecutive points (along the border). It is a tool that helps in analysis and object detection and recognition. We can define the contours of objects in the image. Firstly, the way of doing is to evaluate all points on its contour of every element throughout the image [20]. The resultant contour shape might still, however, be noisy when the image quality is low. A polygon alignment technique may be used to solve these problems. This method again removes the contour points, which means the distance from the contour line is higher than an absolute epsilon value (the maximum distance between the original curve and the simplified curve) [22]. Finding a contour is like finding a white object from a black background. The object to be found should be white, and the background should be black, which is clearly shown in Figure 2.

Figure 2: Contour Extraction on detected image

C. Convolutional Neural Networks

CNNs, or Convolutional neural networks, are widely used to classify images, recognize objects, and detect them [23]. Its structure can be summarized in Figure 3 [23] - [24], which indicates three types of layers: convolution, grouping, and pooling used in our model. The CNN architecture has to be specified by an application and is typically defined by the number of alternate players involved in pooling and convolution, the total count of neurons present in every layer, and the choice of the mechanism of activation [24] - [25].

The input of CNN is a picture defined by some color information for both image processing and classification. In the perception layer, every Neuron is linked to, during CNN segmentation, training, and recognition, a kernel interface customized with an image of the input [26]. The concentrations of every related Neuron are the kernel of such a decision. This step produces a number of N images, which is for every one of the N neurons. These new images can have a negative meaning due to retaliation. To solve this problem, modified linear units (ReLUs) are used to replace negatives with zero values. This layer's performance is termed a function diagram.

It is common to place the pooling layer after the convolution layer [24]. This is important because the pool feature reduces map mobility and reduces network training time. At the end of the calling and pooling structure, a multi-layer neural perceptron network executes categorizations based on the characterized maps described in previous layers [14]. CNN is a popular methodology for extensive research learning, considering the vast number of layers. The architecture automatically extracts different image characteristics such as corners, circles, lines, and textures. The layer is further optimized by removing the properties. It should be noted that, during the training of CNN, the kernel filter value, which is added to the homogeneous layer, is the result of the function of backpropagation.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 46, 46, 32)	832
max_pooling2d_1 (MaxPooling2	(None, 23, 23, 32)	0
conv2d_2 (Conv2D)	(None, 19, 19, 64)	51264
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 64)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_1 (Dense)	(None, 1024)	1049600
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 12)	12300

Total params: 1,113,996
Trainable params: 1,113,996
Non-trainable params: 0

None

Figure 3: Structure of CNN Network

IV. METHODOLOGY

This section discusses the techniques for processing images and for classifying data used in this research. Data must be extracted to perform classifier training. Besides, it is essential to delete features only from the areas of interest when processing images [7]. We are using the techniques for segmentation, filters, and morphological manipulations to enhance the necessary design details [27]. When using traditional neural networks, you don't need to eliminate the feature map from the image, because the image is its own input for such a network interface. A successful pre-processing step will distinguish significant image features from distortion for such consideration.

A. Dataset

In this project, we choose to train our models that make it identifiable for all the eleven emojis, and a comparison of the target (output) and training images can be seen in Figures 4 and 5. There are 1,200 corresponding image specific training images to every eleven emoji that maintain the stability of our training setup [25]. After the training, we use a web camera to capture the hand gestures in actual environments, to process images within the pixel format, and to simulate in near real-time.

Figure 4: Contour Images

Figure 5: Emoji Images

B. Overview

The outline of the emoji prediction and hand gesture recognition is described in Figure 6. Firstly, the hand is detected by using the subtraction process of background [8], [16], and the effect is converted into an image of contour. The gestures are then identified to make the Emoji prediction easier. The corresponding Emoji is predicted and represented in the frame.

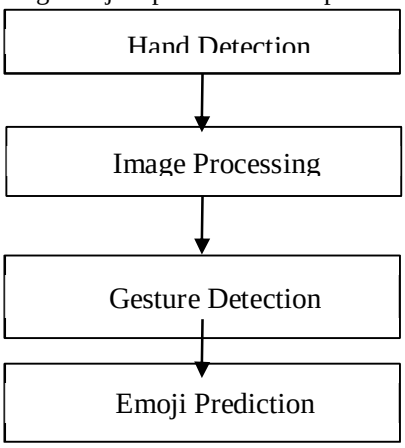


Figure 6: The overview of the proposed method

C. Hand Detection

We use a camera to monitor user-created real-time gestures. The user can make any hand movements that the camera will record and forward for processing further. The documented photos for the prediction of emoji corresponding to a particular gesture are taken in and are then compared with the dataset. The original images for the identification of hand gestures within the project can be seen in Figure 7. All the hand images are captured under similar environmental conditions. The background of all the images is kept identical. The image of the detected hand is resized to 350*350.

Figure 7: Detected Gesture Images

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

D. Image Processing

ImageDataGenerator specifying a horizontal flip was used to obtain another copy of the dataset. The first step we had to perform was to consolidate the size of the input images from the web camera plug-in. And we shrunk the image to 50 * 50 from 350 * 350 to match the scale of the dataset.

Furthermore, we filtered our images by skin color spectrum to transform them into an HSV image followed by a black-white mask to match our dataset's input configuration; the monochrome images trained models [28]. To boost our system's precision, we then blurred the images with Gaussian Blur to reach our final input functionality. After obtaining the consistent shapes of hand movements, we extract the input image by using these morphological operations that is by dilating the image to delete the background noises and get the fairly right-hand gesture accompanied by closing action that helps in filling the tiny spaces inside the foreground objects or small black spots on the hand object [14]. This flow is shown in figure 8.

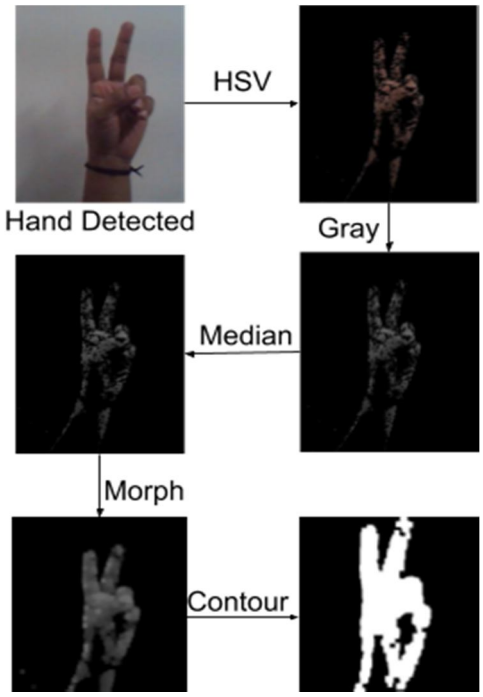


Figure 8: Flow of Image Processing

E. Gesture Detection

Our model uses convolutional neural network architecture. It contains convolutional layers, layers of ReLu, and layers of max-pooling. For our model [25], we have two collections of convolutional layers, ReLu, and max-pooling layers [24]. After that, we placed a completely connected layer, a SoftMax, and a ReLu layer, to predict the correct label [19].

The CNN classifier feeds on the processed, binary images in the gesture recognition stage, where the contour of the hand is oriented and modified to a fixed size, and then produces a probabilistic result. One of the advantages of using CNNs for grading images is that you don't have to extract features manually. CNN itself removes and learns all of the functions. Such attributes often lead to better results for classification when there is no effectual way of extracting features. The CNN used in this section consists of two concentrated layers: one with a maximum pool and two fully joined layers [24]. It employs a linear unit (ReLU) that is addressed as an activation. [24]. It uses a linear unit (ReLU), which is rectified as activation. The threshold, resizing, and figuring the middle of the hand image is done in pre-processing steps, and thus, during the CNN learning process, a certain degree of invariant comparison, scale, and the translation are added [29]. The use of max-pooling layers often makes characteristics acquired by the CNN classifier to some extent rotation-invariant. A clear framework means that the process of identification can be done in real-time.

Figure 9: Detecting the Gestures

F. Emoji Prediction

A value is assigned while predicting a gesture used to blend the image to predict the corresponding Emoji. Images are given different weights, so it provides a feeling of transparency or blending. The relevant Emoji is being displayed, as shown in figure 10.

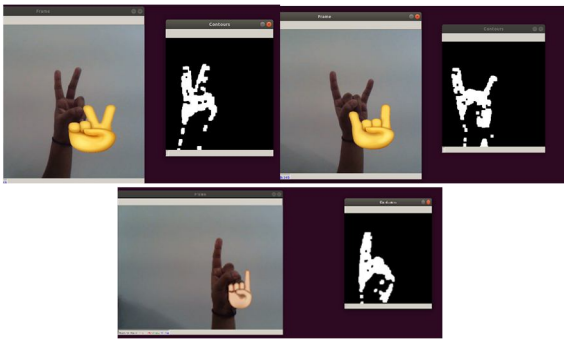


Figure 10: Prediction of Emoji

V. RESULT

We implemented our model to the dataset of the input hand gestures. We have got a really significant outcome:

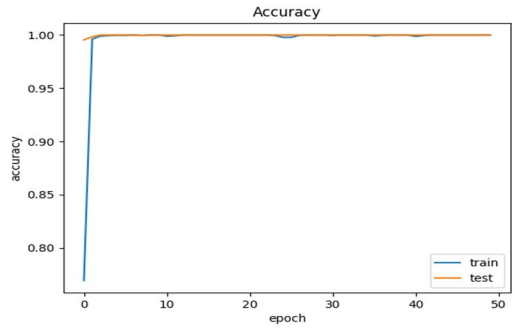


Figure 11: Accuracy vs Epoch

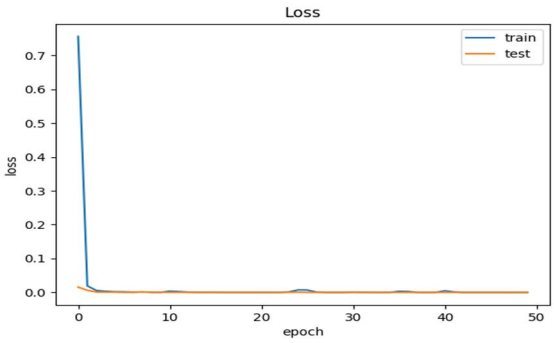


Figure 12: Loss vs Epoch

Both the training accuracy and validation accuracy are similar to 1 (0.996, etc.), and all training losses converge to 0 after completion of the training cycle. For our training model, the learning curves are as follows. We can ultimately see how the behavior converges very quickly in the two graphs specified above in Figure 11 and 12. Within the first few epochs, accuracy is increasingly rising to around 1 in 50 epochs. The deficit reduces to 0, at the same moment. Yet, the model's output is not congruent when we evaluate our model in actual environments instead of using static hand motion images. The first possibility is that owing to the influence of light, the range of skin colors does not reach all areas of the hand. The second possibility is that in the training images, while capturing the hand, we have to put the hand at the exact fixed location in the window as in the training hand position. Sometimes, the hand gestures are falsely labeled due to some environmental conditions. Besides that, our model typically determines the correct marks when we adjust the position of our hand movements.

VI. CONCLUSION

For this project, we have powerful test results that are close to 1.0. Deep learning is excellent at pictorial recognition, and this is seen in the result. There are some aspects we need to do to enhance our model when it is still at one accuracy. For example, in our data collection, applying pre-trained image segmentation mechanisms with a lighting threshold will remarkably minimize the unnecessary noise. Next, a more substantial selection of lighting tools and picture quality should be checked. Secondly, from successive frames, we can use numerous pictures as input. The largest of the output tags will be used as the final feature output tag from labeling several inputs. Finally, we would like to add further databases for various hands to ensure that our layout can be tailored to specific configurations of hand.

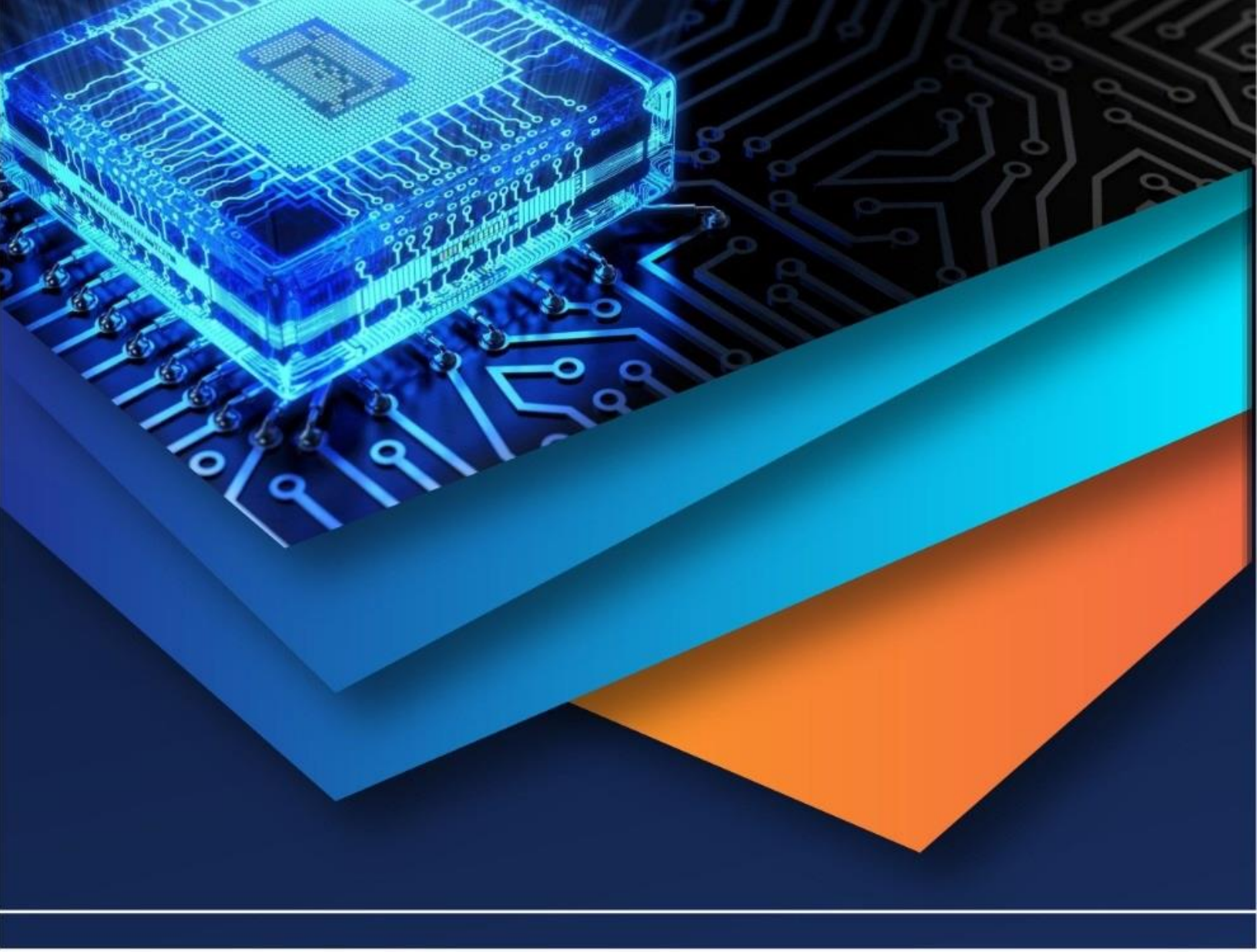
VII. FUTURE SCOPE

Real-Time Management recognition can be used in many applications depending on the requirement, such as for medical applications primarily for people with physical disabilities and for commercial purposes ranging from consumer shops to applications held at home [7]. The essential modes of interaction i.e., hand gestures and emoji Prediction, can be used to control the robot or for offices and household applications. It can be used as a technique for man-machine interaction, which is based on gesture recognition using OpenCV technology, which provides primary data structures for image processing [6]. We can also research robust classifiers for dynamic gestures and develop a gesture-based human-computer interaction system with complex motion recognition support [30].

REFERENCES

[1] Ahlawat, S., Batra, V., Banerjee, S., Saha, J., & Garg, A. K., “Hand gesture recognition using convolutional neural network”, Lecture Notes in Networks and Systems, 2019.
[2] Pisharady, P. K., & Saerbeck, M., “Recent methods and databases in vision-based hand gesture recognition A review”, Computer Vision and Image Understanding, 2015.
[3] Nguyen, T.-N., Huynh, H.-H., & Meunier, J., “Static Hand Gesture Recognition Using Principal Component Analysis Combined with Artificial Neural Network”, Journal of Automation and Control Engineering, 2015.
[4] Oyedotun, O. K., & Khashman, A., “Deep learning in vision-based static hand gesture recognition”, Neural Computing and Applications, 2017.
[5] Huang D.-Y., Hu, W.-C., & Chang, S.-H, “Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination,” Expert Systems with Applications, 2011.
[6] Trigueiros, P., Ribeiro, F., & Reis, L. P., "Generic System for Human-Computer Gesture Interaction", IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2014.
[7] Hsieh, C.-C., & Liou, D.-H. "Novel Haar features for real-time hand gesture recognition using SVM", Journal of Real-Time Image Processing, 2015.
[8] Dardas, N. H., & Georganas, N. D., "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques", IEEE Transactions on Instrumentation and Measurement, 2011.
[9] Cappallo, S., Svetlichnaya, S., Garrigues, P., Mensink, T., Snoek, C.G.M., “The New Modality: Emoji Challenges in Prediction, Anticipation, and Retrieval”, IEEE Transactions on Multimedia, 2019.
[10] Hasan, H., & Abdul-Kareem, S., "Human-computer interaction using vision-based hand gesture recognition systems: a survey", Neural Computing and Applications, 2014.
[11] Rahman, S. U., Afroze, Z., Tareq, M., “Hand Gesture Recognition Techniques for Human Computer Interaction Using OpenCV”, International Journal of Scientific and Research Publications, 2014.
[12] Srinivasa H S, Manasa, Tyapi, Laxmi, Suresha H S, “Online Hand Gesture Recognition by Using OpenCV”, JETIR, 2015.
[13] Chen, Z. H., Kim, J. T., Liang, J., Zhang, J., & Yuan, Y. B., “Real-Time Hand Gesture Recognition Using Finger Segmentation”, Scientific World Journal, 2014.
[14] Pinto, R. F., Borges, C. D. B., Almeida, A. M. A. and Paula, I. C., “Static Hand Gesture Recognition Based on Convolutional Neural Networks”, Journal of Electrical and Computer Engineering, 2019.
[15] Gaurav, R. M., & Kadbe, P. K., “Real time Finger Tracking and Contour Detection for Gesture Recognition using OpenCV”, International Conference on Industrial Instrumentation and Control (ICIC), 2015.
[16] Xian, Z., Yeo, J., & Mall, S., “Hand Recognition and Gesture Control Using a Laptop Web-Camera”, Stanford University, 2015.
[17] Núñez Fernández, D., & Kwolek, B., “Hand Posture Recognition Using Convolutional Neural Network”, Lecture Notes in Computer Science, 2018.
[18] Xu, P., “A Real-time Hand Gesture Recognition and Human-Computer Interaction System”, Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, 2017.
[19] Kiruthika, U., Mohan, M., & Abraham, N., “Hand Gesture Recognition for Emoji and Text Prediction”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019.
[20] Nordin, N., “Hand Gesture Recognition using Curvature”, International Journal on Future Revolution in Computer Science & Communication Engineering, 2018.
[21] Ganapathyraju, S., "Hand Gesture Recognition Using Convexity Hull Defects to Control an Industrial Robot", 3rd International Conference on Instrumentation Control and Automation (ICA), 2013.

[22] Yao, Y., & Fu, Y., "Contour model-based hand-gesture recognition using the Kinect sensor", IEEE Transactions on Circuits and Systems for Video Technology, 2014.
[23] Köpüklü, O., Gunduz, A., Kose, N., & Rigoll, G., "Real-time hand gesture detection and classification using convolutional neural networks", 14th IEEE International Conference on Automatic Face and Gesture Recognition, 2019.
[24] Nagi, J., Ducatelle, F., Caro, G. A., Ciresan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, F., Gambardella, L. M., "Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition", IEEE International Conference on Signal and Image Processing Applications, ICSIPA, 2011.
[25] Zhan, F., "Hand gesture recognition with convolution neural networks", IEEE 20th International Conference on Information Reuse and Integration for Data Science, IRI, 2019.
[26] Avraam, M., "Static Gesture Recognition Combining Graph and Appearance Features", International Journal of Advanced Research in Artificial Intelligence, 2014.
[27] Shaikh, S., Gupta, R., Shaikh, I., & Borade, J., "Hand Gesture Recognition Using Open CV", International Journal of Advanced Research in Computer and Communication Engineering, 2016.
[28] Correa, M., Ruiz-del-Solar, J., Verschae, R., Lee-Ferng, J., & Castillo, N., "Real-time hand gesture recognition for human robot interaction", Lecture Notes in Computer Science, 2010.
[29] Barros, P., Magg, S., Weber, C., & Wermter, S., "A multichannel convolutional neural network for hand posture recognition", Lecture Notes in Computer Science, 24th Int. Conf. on Artificial Neural Networks (ICANN), Springer, 2014.
[30] Abhishek, B., Krishi, K., Meghana, M., Daaniyaal, M., & Anupama, H. S., "Hand Gesture Recognition using Machine Learning Algorithms", International Journal of Recent Technology and Engineering, 2019.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY