Real Time Contextual Based Gesture Recognition System Using Deep Learning

[1] Karthik Vas S, [2] Anisha B.S

[1] Post Graduate student, Software Engineering, RV College of Engineering,
[2] Assistant Professor, Department of Information Science and Engineering, RV College of Engineering.

Abstract: **Nowadays Computer plays a crucial role in daily activities and used among various fields in society. The Interactions between a computer and human is done by basic input devices like keyboard, mouse etc. Hand Gestures can be one of the best and effective medium for human-computer interactions which help in interactions between the both easier. Convolution Neural Network (CNN) which is basically used for image classification and image representation. Also, CNN can learn non-linear relationships among set of images, considering this we propose a gesture recognition system using CNN. Face recognition software's are used in various fields like user detection, automobile control, biometrics and security authentication and enhancements. This project focuses on developing a model to control a system using gestures, while simultaneously monitoring continuously for user and their profile using face recognition to avoid unauthorized access. An effective gesture recognition is developed and applied in conjunction with face recognition system to perform a mapped function in real time.**

Index Terms **– Facial Recognition, Gesture Recognition, Background Subtraction, Convolutional Neural Network.**

I. Introduction

Face recognition is a crucial component of computer vision and is used in a variety of human-computer interface applications. Face recognition is a non-invasive technology that can be used for real-time recognition and is a human-centric means of identification, even though fingerprints and iris scans achieve higher precision. Defense, recognition, computer games, surveillance, automobiles, human-machine interaction, and robotics are only a few of the areas where such a device might be used. Defense, identification, video games, security, cars, human-machine interaction, and robotics are only a few possible applications. A new face detection algorithm was implemented in that is based on skin color detection while compensating for lighting conditions, then detection using the model's elliptical skin. Centered on the orientation of the face, the ellipse vote, and the eyes/mouth maps, a face score is calculated for each validated eye-mouth triangle, followed by the Hough transform to extract the best-suited ellipse. It has been discussed the possibility of using a facial recognition technology in a real-time world. They use Neural Networks to identify faces and perform recognition tasks with high precision. Deep learning-based function extractors are used to boost the results even further. As seen in a CNN-based function extractor, the accuracy is very high. Gesture perception is another essential aspect of machine vision and human-computer interaction. Gesture identification is difficult because it necessitates the extraction of hand regions and the monitoring of their orientation. Using hand signals to manipulate a video player using neural networks is one of the most recent developments in this field. It involves extracting the hand area using skin color recognition, then extracting the hand shape's features, and finally classifying the gesture using a neural network. The hand region centroid was successfully

identified, followed by area extraction, which included the formation of convex hulls, and finally Kalman filter monitoring.

II. Related Work

Hand motion identification has been the subject of several studies, and some notable research in this area is discussed. A hand motion recognition method based on the form fitting technique has been developed using an artificial neural network (ANN). After filtering, a color segmentation technique on the YbrCr color space was used to detect the hand in this method. The hand morphology then resembled the form of the hand. An ANN was used to extract the outline of hands and finger orientation characteristics. Using this approach, they were able to achieve an accuracy of 94.05 percent.

To detect motions, a computational approach based on haar-like features was suggested. The AdaBoost algorithm was used to learn the model in this scheme. The project was divided totally into two sections.

A stochastic context-free grammar was used to detect movements at a higher level. Postures were found at a lower degree. According to the grammar, a terminal string was created for each input. Each rule's likelihood was determined, and the rule with the highest probability for the given string was chosen as the winner. The input gesture was returned as the gesture correlated with this law.

Then there's the digital feature engineering, which isn't as time-consuming as it sounds and isn't as skewed as it sounds. Furthermore, automatic feature engineering will catch almost all of the functionality. CNN can draw useful functionality from structured data. As a result, a shift to digital function engineering was introduced, and deep learning, or CNN, emerged. Another method for detecting gestures using CNN that is stable under five invariants has been proposed: size, rotation, translation, light, noise, and context. The dataset was Peruvian Sign Language (LSP). On the LSP Dataset, they were 96.20 percent accurate.

A facial recognition algorithm is typically divided into the following practical modules: a face image detector locates human faces in a regular image against a basic or complex context, and a face recognizer decides who this individual is. A feature extractor transforms the pixels of the facial image into a useful vector representation, and a pattern recognizer searches the database for the best match to the incoming face image.

III. System Architecture

The overall working of the software basically would include 2 modules integrated together.

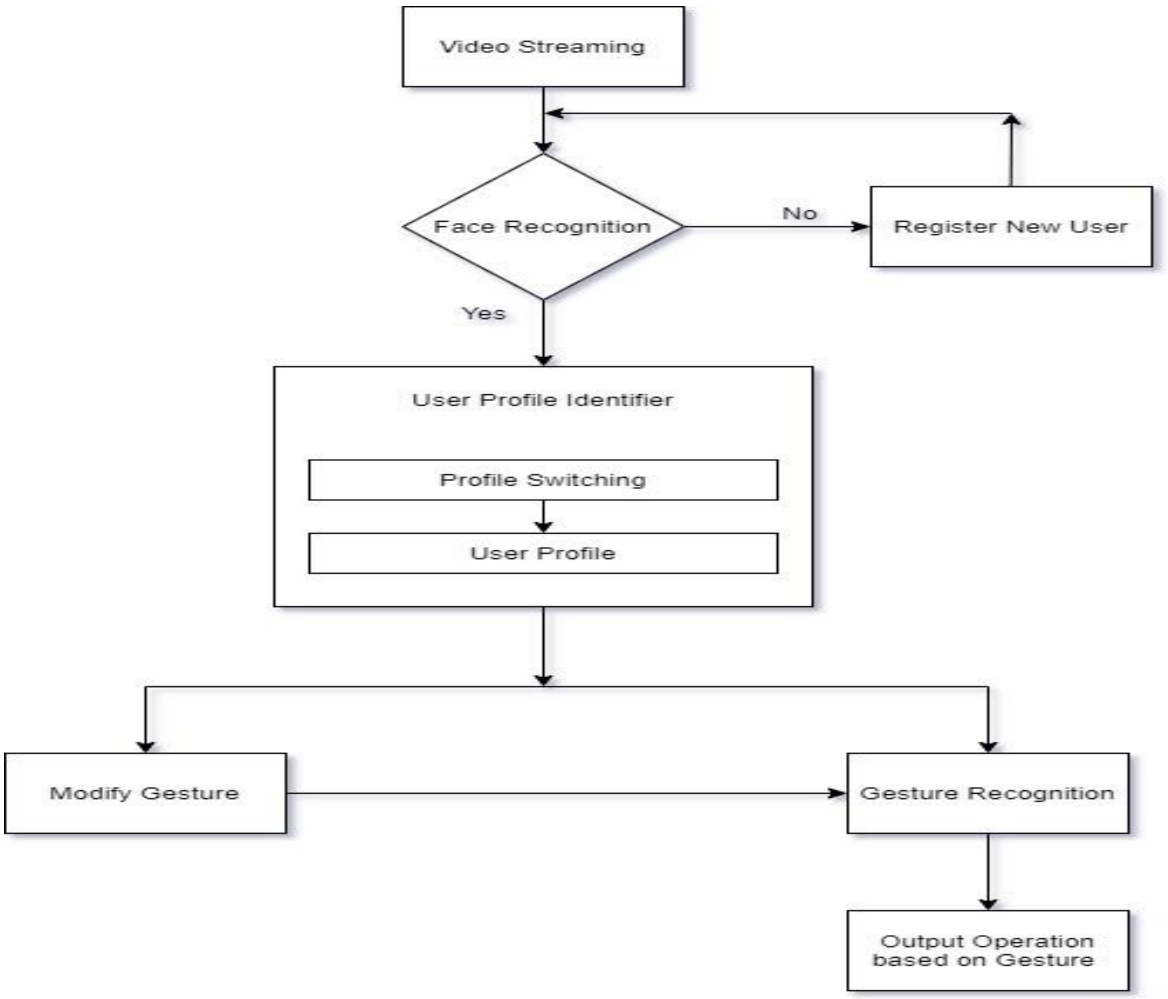

Fig 1. System Architecture

These 2 modules of the software determine the output operation based on the gesture made by the user of the application based on his stored profile in the database.

The software starts with the video frame popping up where both the Face recognition and gesture recognition modules run simultaneously on a single video frame to determine the output of the gesture made, where the corresponding application would pop up as a result of it.

Face recognition looks for the user in the video frame by tracking the face portion of the video stream, while gesture recognition looks for the user's hand gesture with the highest priority by profile switching, which is achieved by face recognition.

After the user's face is recognized, the module runs the profile swapping module, which gives the user precedence. When the person appears in the video frame, the user's profile is immediately shifted, and the associated profile gesture database is active profile. This also gives the user the option of changing the motions.

IV. Design

1. Input and Training Data

A webcam was used to capture the images required to train and test the model. In front of the webcam, ten System Framework members made the motions. The input pictures are believed to include only one hand, with movements made with the right hand, the palm facing the camera, and the hand approximately vertical. If the context is simple and the contrast on the hand is strong, the recognition process would be less complicated and more effective. As a result, it's thought that the photographs' backgrounds were simpler and more uniform.

2. Pre-Processing

To minimize computational complexity and improve performance, a simple pre-processing was applied to the dataset. To begin, the background of the images was extracted using Z. ZivKovic's background subtraction process. Background subtraction is mostly dependent on the K-gaussian distribution, which chooses the required gaussian distribution for each pixel and allows for greater adaptability to changing scenes due to changes in lighting. Just the representation of the hand remains after the backdrop has been removed. The files were then translated to grayscale. Since grayscale pictures only have one color channel, CNN would have a simpler time learning them. After that, morphological erosion was used. After that, the noise was reduced using a median filter. Noise reduction is often desired in signal processing. The photographs were then resized to 50x50 pixels in order to be fed to CNN. This experiment has used a dataset called "Hand Gesture Recognition Database" in addition to our self-developed dataset. Other objects from these photographs were omitted by choosing the larger object, in this case the side.

3. Convolution Neural Network Configuration

Two convolution layers, two max pooling layers, two entirely linked layers, and an output layer make up the CNN that was used in this study to identify hand gestures. To avoid over-fitting, the network has three dropout performances. With a kernel size of 3x3, the first convolution layer has 64 separate filters. Rectified Linear Unit is the activation mechanism used in this layer (ReLU). ReLU was used to implement non-linearity, and it was shown that it outperforms other activation functions like tanh and sigmoid. We must define the input size since it is an input layer. The stride has been adjusted to its default value. The input form is 50x50x1, implying that a grayscale image of 50x50 pixels should be given to this network. This layer creates the function maps and sends them to the next. The CNN then has a max pooling layer with a 2x2 pool size that takes the full value from a 2x2 slot. As the pooling layer takes out the highest value and discards the remainder, the spatial scale of the representation shrinks. Since it only picks the most appropriate features, this layer aids the network's understanding of the images. Another convolution layer follows, with 64 separate filters and a kernel size of 3x3 and default stride. In this sheet, ReLU was used as the activation mechanism once more.

Following this layer is another max pooling layer with a pooling size of 2x2. To avoid the model from over-fitting, the first dropout layer was applied, which randomly discards 25% of the total neurons. This layer's output is transferred to the flatten layer. The flattening layer receives the output from the

previous layers and converts it to a vector from a two-dimensional matrix. This layer enables the completely linked layers to process the data that has been collected up to this stage. The next layer is the first totally connected layer, which has 256 nodes and was enabled using ReLU. To avoid overfitting, the layer is preceded by a dropout layer that removes 25% of the neurons. The second fully connected layer has 256 nodes and uses ReLU as an activation layer to obtain the vector generated by the first fully connected layer. To avoid overfitting, the layer is preceded by a dropout layer that removes 25% of the neurons. The output layer has ten nodes, one for each of the hand gesture groups. The SoftMax function is used as an activation function in this layer, and it outputs a probabilistic value for each of the groups. After that, the model is built using the Stochastic Gradient Descent (SGD) function with a learning rate of 0.001. Since the model is compiled with more than two classes, the categorical cross-entropy function was used to assess failure. Finally, error and accuracy measurements were defined in order to keep track of the assessment process. This configuration was selected after experimenting with different node and layer combinations.

Table 1 CNN Configuration

Model Content	Details
First Convolution Layer	64 filters of size 3x3, ReLU, input size 50x50
First Max Pooling Layer	Pooling Size 2x2
Second Convolution Layer	64 filters of size 3x3, ReLU
Second Max Pooling layer	Pooling size 2x2
Dropout Layer	Excludes 25% neurons randomly
First Fully connected Layer	256 nodes, ReLU
Dropout Layer	Excludes 25% neurons randomly
Second Fully Connected Layer	256 nodes, ReLU
Dropout Layer	Excludes 25% neurons randomly
Output Layer	6 nodes for 6 classes, SoftMax
Optimization Function	Stochastic Gradient Descent (SGD)

V. Results

Real Time Contextual Based Gesture Recognition System is successfully developed and is deployed on Windows platform. This software helps in opening up the application as when the gesture and face is detected by the software. This software helps most of the illiterate individuals to accomplish simple tasks. This software will also help medically ill individuals to accomplish the tasks on the computer.

With the help of simple User interface, we can help the individual to start the software with just a click of a start button, which is an easy task for any kind of Individual.

Impact

Real Time Contextual Based Gesture Recognition System has a major impact on Windows users as the software is developed for just windows at the present, which works with the integration of face recognition and gesture recognition into a single system.
When the software is run, a video frame appears and where both the face and gesture regions would be working simultaneously as a single software and after which corresponding profiles would be active and recognizes the gestures made by the individuals. Later, depending on the active profile after facial recognition, the individual's expression would be recognized and a subsequent move would be executed, resulting in an application being launched.

Future Developments

Real Time Contextual Based Gesture Recognition System is developed for opening the application as of now, this would be taken further by initiating the further steps towards working on the how the gestures may interact inside of specific application, we can also work on to mapping the hot keys of one specific application and help to achieve the tasks by performing the actions inside of that application in specific.
This software can be further implemented on Smart devices like Android TV, Android smartphones and can even be taken further to Internet of Things devices which can be useful to map the functioning of devices to gestures. This software can even be implemented in the field of Robotics which can be useful to develop any responses to any kind of gesture in future. As overall this concept has vast futuristic scope on development and can be amended based on the requirement at that point of time.

VI. Conclusion

Real Time Contextual Based Gesture Recognition System is the software developed for the ease of Human interactions between humans and computer which involves both Face recognition using KNN algorithm and Gesture recognition using CNN algorithms which would ultimately help in supporting the windows system for identifying the user and activating user profile and corresponding gestures would be executed to opening of the application as when those gestures is recognized by the system.
This software provides an opportunity for any kind of individual to have hands on experience by using the software on their personal devices. This project helps the individual and does provide the futuristic ideologies on many coexisting domains in the industry.

Acknowledgement

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. Several personalities, in their own capacities have helped me in carrying out this project work. I would like to take this opportunity to thank them all.
I would like to express my sincere gratitude to my guide, Prof. Anisha B S, Assistant Professor, Department of Information Science & Engineering, RV COLLEGE OF ENGINEERING, Bengaluru for her valuable guidance, expert review and her encouragement in choosing this domain and her constant support throughout the project.
I would also like to express my sincere gratitude to Dr. B. M. Sagar, Head of the Department, Information Science & Engineering, RV COLLEGE OF ENGINEERING, Bengaluru for his valuable suggestions, support and regular source of encouragement and assistance throughout this project.

I would also like to express my sincere gratitude to Dr G.S Mamatha, Prof & Associate Dean, Information Science & Engineering, RV COLLEGE OF ENGINEERING, Bengaluru for his valuable suggestions, support and regular source of encouragement and assistance throughout this project.

I would also like to thank Dr. K. N. Subramanya, Principal, RV COLLEGE OF ENGINEERING, Bengaluru, for his moral support towards completing my project work.

I thank my Family, and all the Faculty members of Department of Information Science & Engineering for their constant support and encouragement.

Lastly, I would like to thank my peers, team mates and friends who gave me their valuable suggestions in the improvement of my project.

References

[1] A. D. Wilson and A. F. Bobick, "Learning visual behavior for gesture analysis," in Proceedings of International Symposium on Computer VisionISCV. IEEE, 1995, pp. 229–234.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, p. 436, 2015.

[3] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in Shape, contour and grouping in computervision. Springer, 1999, pp. 319–345.

[4] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," Engineering Applications of Artificial Intelligence, vol. 22, no. 8, pp. 1141–1158, 2009.

[5] T.-N. Nguyen, H.-H. Huynh, and J. Meunier, "Static hand gesture recognition using artificial neural network," Journal of Image and Graphics, vol. 1, no. 1, pp. 34–38, 2013

[6] Q. Chen, N. D. Georganas, and E. M. Petriu, "Hand gesture recognition using haar-like features and a stochastic context-free grammar," IEEE transactions on instrumentation and measurement, vol. 57, no. 8, pp. 1562–1571, 2008.

[7] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 1–7.