Controlling Media Player with Hand Gestures using Convolutional Neural Network

Gayathri Devi Nagalapuram
Department of Computer Science and Engineering
Dayananda Sagar University
Bengaluru, India
gayathri1462@gmail.com

Roopashree S
Department of Computer Science and Engineering (AI&ML) and (Cybersecurity),
JAIN (Deemed-to-be University)
Bengaluru, India
roopashaily@gmail.com

Varshashree D
Department of Computer Science and Engineering
Dayananda Sagar University
Bengaluru, India
varshashree73@gmail.com

Dheeraj D
Department of Computer Science and Engineering
Dayananda Sagar University
Bengaluru, India
dheerajdass044@gmail.com

Donal Jovian Nazareth
Department of Computer Science and Engineering
Dayananda Sagar University
Bengaluru, India
djovian.n@gmail.com

2021 IEEE Mysore Sub Section International Conference (MysuruCon) | 978-1-6654-3888-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/MYSURUCON52639.2021.9641567

***Abstract*—In today's world, everyone opts for fast interaction with complex systems that ensure a quick response. Thus, with increasing improvement in technology, response time and ease of operations are the concerns. Here is where human-computer interaction comes into play. This interaction is unrestricted and challenges the used devices such as the keyboard and mouse for input. Gesture recognition has been gaining much attention. Gestures are instinctive and are frequently used in day-to-day interactions. Therefore, communicating using gestures with computers creates a whole new standard of interaction. In this project, with the help of computer vision and deep learning techniques, user hand movements (gestures) are used in real-time to control the media player. In this project, seven gestures are defined to control the media players using hand gestures. The proposed web application enables the user to use their local device camera to identify their gesture and execute the control over the media player and similar applications (without any additional hardware). It increases efficiency and makes interaction effortless by letting the user control his/her laptop/desktop from a distance.**

Keywords— Convolution Neural Network, Deep Learning, Hand Gesture Recognition, Media Player Control, OpenCV, Streamlit

I. INTRODUCTION

Along with the evolution of society, technology has been going through a series of revolutions. Computers have been growing and advancing significantly over the decades since they originated. This creates a demand for many technology-related fields [1]. One such field is Human-Computer Interaction (HCI).

Human-Computer Interaction (HCI) is the key to the success of interactive systems. It involves bringing together the understanding of human abilities and technical understanding of hardware and software technologies [2]. For example, earlier, if a person wanted to perform tasks such as writing a book or perform complex calculations, they carried them out without a computer. But now, to make the tasks easier, people choose computers rather than carrying them out manually. The computer changes the linear writing process into an effortless process. Several gadgets are available for human-computer interaction. Some of them are familiar tools and have flourished lately or are yet to emerge in the future. This paper addresses one of the emerging concepts i.e., Gesture Recognition [3].

Gesture recognition is an active research field that is a kind of non-cognitive computing interface for users that allows devices to catch and interpret human gestures as commands. The usage of gestures comes naturally to all. The study reveals that blind people also use gestures while speaking with others, as gestures easily represent ideas and actions. This motivates the idea of integrating gestures to interact with computer devices and carry out the tasks easily [4]. It has many applications including virtual reality environment control, sign language translation, robot remote control, musical creation, school kids interaction, drowsiness detection of drivers, and also activity monitoring of elderly or disabled people [5]. In recent years, the HCI method has been evolving rapidly [6]. Traditional input devices such as a mouse, keyboard, and joystick are being replaced by these methods by simplifying the interaction with interfaces.

Gesture recognition technology will cause interaction to be more natural. It minimizes the hardware needed to use any system which increases the efficiency of the system in terms of power and speed. This interaction is easily learnable and adaptable. Although there has been a development in this field, problems like slow and expensive computation persist. Devices like glove-based systems with sensors are used

978-0-7381-4662-1/21/$31.00 ©2021 IEEE

to acknowledge various challenges. But such devices are expensive and gloves had to be worn all the time to make use of them. Another problem in gesture recognition is to deal with noise found in the region of interest and image background. These problems are solved using neural networks by implementing color segmentation and morphological operations. This being a major step proves the efficiency of using neural networks for gesture recognition.

Technology has been using gestures to control various areas of work in many fields, including entertainment, automation, health and medical, electronics, and academics [7]. In healthcare, gestures can provide the sterility needed to prevent any infection from spreading, by using them to interact with medical equipment, control visualizations, and distribute resources too. The entertainment field has been a very propitious and rewarding arena for innovations. End users are always keen to try new paradigms of interfaces. The usage of gestures will provide a sense of naturalness which will improve the user experience. Further, gestures are used in disaster relief, human-robot interaction, and crisis management [9].

In this paper is about controlling the media player with hand gestures using CNN. Some of the use cases include watching a movie or having to change the attention to another direction by looking or moving away from the computer, which leads to missing an interesting clip of the movie/cooking tutorial/workout videos. For all the above distractions, the user has to repeatedly pause or play and rewind the video, which eventually leads to the usage of dirty or sweaty hands to control the player [8].

The solution to all the above scenarios is the proposed system. A web application that uses the device camera to give users a touch-free and remote-free control over any media player application without the requirement of any special hardware. The only information required is the mapping of gestures to controls. This application is very productive, makes life easier and comfortable by controlling the computer device from a distance.

The next section is the review based on related recent papers and propose a logical explanation for the method used for implementation. The last sections provide the result analysis regarding the work done, conclusion, and approaches for future scope.

II. Related Works

Yashas J, Shivakumar G [10] presents the literature survey performed on Hand Gesture Recognition. Possible methods such as sensors worn on hands and camera is discussed with respect to data acquisition. Cameras provide a better advantage by not limiting the physical movement of the hand but bring in new challenges like region of interest, background noise and lighting. These challenges are solved to an extent by devices like the Microsoft Kinect. Learning algorithms like Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Deep Learning. The usage of CNN is challenged by Adapted Convolutional Neural Networks (ADCNN) as the latter uses data augmentation to analyze data in better way to allow for better classification. The paper concludes by stating that data augmentation is a field yet to be explored in detail.

Sharma P and Sharma N [11] focuses on a proposed methodology for recognizing posture and gestures. Feature extraction of input images is done using PCA and Singular Value Decomposition (SVD). The SVD extracts the silent features of input images to reduce dimensions of the images and prepare it for training the model. Feed-Forward Neural Network is used to train the obtained features and classify the gestures and postures. The proposed system was limited to work under uniform backgrounds and could only recognize limited gestures.

Hakim NL et. al., [12] discuss the usage of 3DCNN model for dynamic hand gesture recognition. Data collected was a combination from RGB and depth cameras to allow a better input for deep learning. As the recognition has to work for dynamic gestures, the need for spatio-temporal features rises. This was addressed by combining the 3DCNN with Long Short-Term Memory (LSTM) model. Finite State Machine (FSM) eased the model's working and improved the accuracy as it narrowed the search of gesture classification on the model into a compact one.

Cardenas EJ and Chavez GC [13] present an approach for dynamic gesture recognition by combining information of temporal and pose from CNN descriptors and cumulative magnitudes histogram. Human poses are detected from RGB and Depth data, then the extraction of spatio-temporal features is done. The approach of Cao et. al., [18] is used to estimate the skeleton data, if not already available. Two CNN descriptors are used to process hand and body images at the local level followed by the construction of cumulative magnitudes histograms. The extracted features from these methods are then sent for classification to a SVM classifier which provided efficient results.

Adithya V and Rajesh R [14] proposes an automatic hand posture recognition using CNN with deep parallel architectures. It aims at avoiding the tedious feature extraction stage by making use of the hierarchical architecture of CNN which automates the feature extraction by studying the high-level abstractions in images. The model proposed reduces the computational time and attains a great accuracy.

Munir Oudah et. al., [15] reviewed various techniques for hand gesture recognition based on computer vision, from glove-based attached sensors to deep learning models. The basic idea is to detect skin color by using threshold values for a color space without the usage of any hardware. Image subtraction and foreground and background segmentation algorithms are used to segment Region of Interest (ROI). They further discussed the motion-based detection system using frame difference subtraction, skeleton-based recognition of hand, depth, and 3D model-based recognition to interact with virtual

systems, and deep learning-based recognition using neural networks for hand classification. This paper showed both the advantages and disadvantages of the above methods for different circumstances and conditions.

Adam Ahmed Qaid MOHAMMED et. al., [16] in their paper proposed hand detection to the gesture recognition system. The hand detection is done using a RetinaNet based detector that extracts the hand regions. CNN followed this for recognition of the hand gesture. They ran the proposed architecture on different available datasets like the Oxford 5-signers for hand detection, and the LaRED, the TinyHands for gesture recognition. The results are compared with conventionally used methods such as the Recurrent Neural Network(RNN) to find out that the lightweight CNN proposed gave the best values for the evaluation metrics. But the CNN had other complexities in cluttered surroundings and with people in the background, which was to be worked on in the future. This paper gives an obvious conclusion that CNN works better than previously used algorithms with gesture recognition.

Raimundo F. Pinto Jr. et. al.,[17] exhibits various methodologies to achieve gesture recognition using neural networks. The paper focuses on image processing before loading it to the CNN. Images are segmented based on color using Multilayer Perceptron (MLP). Contour extraction and morphological filters are used to extract the necessary components from images. Polygonal approximation removes background noise, making the images ready to be fed to the CNN. Further on experimenting with various CNN models, a highest accuracy of 96.83% is obtained to demonstrate the effectiveness of the techniques used.

III. PROPOSED METHODS

The proposed gesture detection system for media player control is broadly divided into two stages. The first stage performs the detection of hand gestures and the keyboard controls are integrated with each gesture in the second stage. The gesture recognition is implemented using computer vision and deep learning techniques with a custom-built dataset. A new dataset is created that contains 7 gestures as shown in Fig. 1. These raw images from the dataset are converted to black and white images as shown in Fig. 2.

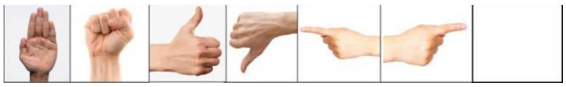

Fig. 1. Raw Data collected for the dataset

Fig. 2. Gesture data after preprocessing

The CNN model is trained by feeding the image frames to three convolution layers with a ReLU activation function. Pooling layers are added to perform max pooling which is then flattened and added to dense layers as shown in Fig 3.

In the proposed system, the gestures are predicted by the CNN model. The preprocessing and re-sizing of the image are performed before passing it into the model. The trained CNN model is used to predict one of the seven gestures. Each gesture is connected to an individual keyboard control which will further control any media player playing on the system whenever a gesture is detected.

A. Proposed Workflow

The system is divided into five modules. Fig. 4 shows the overview of the system workflow, emphasizing the gestures and their corresponding controls.

Fig. 3. Proposed CNN model

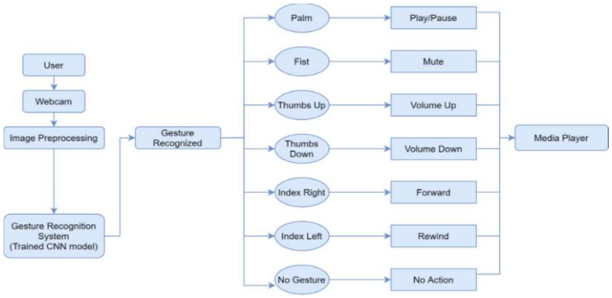

Fig. 4. The System Design Workflow

1) Image Acquisition and Pre-processing

When the user performs hand gestures in front of the webcam. The image frames are collected from the live video using the OpenCV. These images are converted into black and white images as shown in Fig. 5, to improve accuracy in predicting gestures and then stored in respective directories.

In this project, gestures are collected from 3 different people. The dataset contains 150 images for each gesture. A directory structure is created to store images. Two modes, train, and test are provided for user convenience. The user is given an option to choose either one of the modes. The images collected in train mode are used to train the model and the test mode images are used to test the model accuracy. The images are stored in a specific directory based on the user input.

While the camera is on, two frames are displayed on the screen and the user can capture images frame-by-frame using the read function and the mirror image is simulated. The user has to place the hand in the Region Of Interest (ROI) i.e., the bounding box, and perform the gestures. The frames are extracted from ROI and resized to 120x120x1. The count of the number of images in each directory is printed onto the

screen. The count is increased every time user captures an image by pressing 0 to 6 number keys on number keys on the keyboard and the images are saved to their respective class directory.

The preprocessed image can be seen in the small frame while capturing and these images will be stored in the dataset. The user can exit after data collection by pressing the escape key on Keyboard.

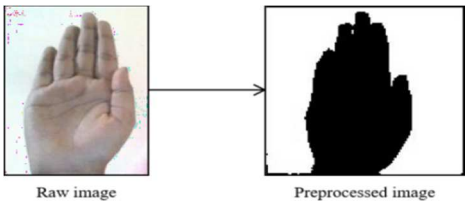

Fig. 5. Data Preprocessing

2) Feature Extraction

Import Keras models and hidden layers required to build convolutional networks. The CNN model is built using a hidden input layer followed by two convolution layers. An activation operation called ReLU and a pooling layer called MaxPooling are added after every convolution layer. A flattening layer is added along with two fully connected layers, one with RELU activation and another with softmax activation which is used for classification. Fig. 6 shows the architecture of the CNN model used for feature extraction and classification of the gestures.

Lastly, the compilation of the CNN model is performed by implementing adam algorithm as an optimizer, categorical_crossentropy as a loss function method to find error and accuracy as the performance metrics to evaluate the model.

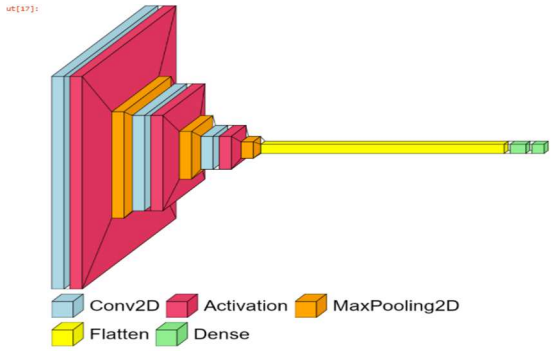

Fig. 6. The architecture of the CNN model

3) Train the model

Next, ImageDataGenerator class is used to generate batches of images to train and validate the model. The fit function is used to train the model with a fixed number of epochs. After training, the trained model is saved in JSON format and the weights are saved directly from the model using the save_weights function.

4) Media control using predicted Hand gestures

The trained model in JSON format and the model weights are loaded to predict the hand gestures. The PyAutoGUI which is used for Keyboard key integration with hand gestures and Streamlit which is used to create a user interface are also imported.

Three web pages are created using the streamlit web framework. The web pages are designed using the in-built streamlit functions and HTML templates. The first page is an about page with a brief introduction about the project. The second page contains the video demo of the project. The third page is the demo page which is used to predict the hand gestures to control the media player.

When the user clicks the start button on the web page, the web camera starts running. The user can perform the hand gestures in the Region Of Interest and the trained CNN model will predict the gesture. Each hand gesture is integrated with a Keyboard key using PyAutoGUI with help of conditional if-else statements to call predicted gestures. Each gesture is mapped with a keyboard key control and a label. The number of presses is assigned as 1, so every time a gesture is predicted the integrated control function is performed once. The user can exit the system by pressing the escape keyboard key. The video frame will display the gesture predicted and the action being performed whenever the user is using the system to control the media player.

The following conditional statements are used to perform the actions as shown in Table I.

TABLE I. GESTURES AND THEIR RESPECTIVE ACTIONS

Predicted Gesture	Action	Keyboard Keys
Palm	PLAY/PAUSE	Playpause
Fist	MUTE/ UNMUTE	Mute
Thumbs Up	VOLUME UP	Volumeup
Thumbs Down	VOLUME DOWN	Volumedown
Index Left	FORWARD	Prevtrack
Index Right	REWIND	Nexttrack
No Gesture	NO-ACTION	NIL

5) Web App and Deployment

A web application is created with an about page and demo page, along with a download page. The download page has a download button which when clicked directs the user to the GitHub repository of the project.

This web application is deployed on the streamlit.io sharing website by signing up with a GitHub account and selecting the git repository which contains the project source files. A website link is obtained after the successful deployment. Users can download the files from this link and use this proposed system on their devices.

B. Architecture

The outline of any gesture recognition system generally involves the following three aspects:

1. Data acquisition and pre-processing
2. Data representation or feature extraction
3. Classification or decision-making

The proposed system architecture can be seen in Fig. 7. It includes the three main aspects required in a gesture recognition system.

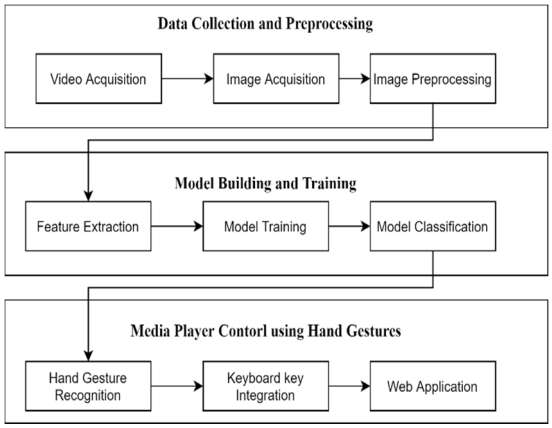

Fig. 7. Proposed architecture diagram.

1) Data Acquisition and Preprocessing Phase:

The image data input is captured from a webcam, in the form of a video. The video is further broken into frames. These frames are converted into black and white images using OpenCV and then stored in specific directories.

2) Model Building and Feature Extraction:

A CNN model is built using Keras libraries. ImageDataGenerator class is used to pre-process the images in the directories to extract only the needed features. Model is trained using train dataset images.

3) Classification and Prediction:

Compilation of the model is performed and test accuracy is evaluated. Keras libraries are used to perform save and load of the model and to classify the gestures to a particular class.

4) Integrating the Keyboard controls:

Integrate every gesture with control functions using the Pyautogui library. The control of the prediction is executed. Deploy a web app with all project files using streamlit sharing.

IV. Experimental Results

To test the proposed system in a real-world application, seven gestures are selected based on the convenient use of the applications and conducted several experiments to test the model's robustness. The custom dataset consisted of 150 images in each class for seven different classes, adding up to a total of 1050 images. The images are collected in proper lighting with a static background and no background noise as shown in Fig 8.

On performing real-time testing on the proposed system using different sets of people, good results are achieved under proper lighting with no background noise conditions. The system is tested with hand gestures from different people on three different CNN models as shown in Table II. By varying the number of convolutional and pooling layers in the model, three unique architectures are created to inspect the effectiveness and assess the best model based on accuracy.

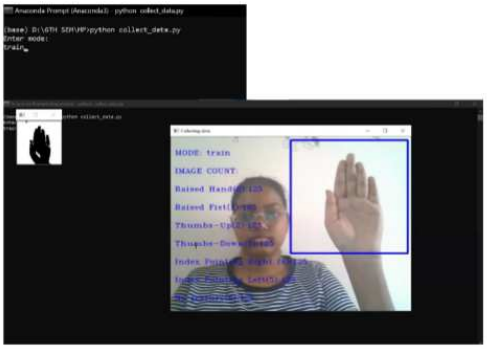

Fig. 8. Data collection for training the CNN model.

TABLE II. CNN Models Architecture

Depth	CNN Model 1	CNN Model 2	CNN Model 3
1	Convolutional (3x3)	Convolutional (5x5)	Convolutional (5x5)
2	Convolutional (3x3)	Max Pooling (2x2)	Max Pooling (2x2)
3	Max Pooling (2x2)	Convolutional (7x7)	Convolutional (7x7)
4	Convolutional (3x3)	Max Pooling (2x2)	Max Pooling (2x2)
5	Max Pooling (2x2)		Convolutional (5x5)
6			Max Pooling (9x9)

The training time and testing time obtained from each CNN model are shown in Table III.

TABLE III. Training and Testing Time

Time taken	Training (in milliseconds/step)	Testing (in seconds)
CNN Model 1	460	0.136736869812
CNN Model 2	694	0.0553321838378
CNN Model 3	828	0.1521420478820

The performance metrics of the 3 CNN models are compared and the system showed a highest of 98% accuracy for model 3 as shown in Table IV.

TABLE IV. CNN Models Evaluation Metrics

	CNN Model 1	CNN Model 2	CNN Model 3
Accuracy	0.9492	0.9746	0.9778
Precision	0.95	0.97	0.98
Recall	0.95	0.97	0.98
F1 Score	0.95	0.97	0.98

The accuracy of the proposed model is compared with different models as shown in Table V.

TABLE V. COMPARISON OF ACCURACY WITH PREVIOUS WORKS

Author	Method	Dataset	Accuracy
Pinto RF et. al., [17]	Deep Learning with CNN 4	Custom	96.83%
Mohammed AA et.al., [16]	Deep RetinaNet based lightweight CNN	Oxford	72.1%
		5-signers	97.9%
		EgoHands	93.1%
		Indian Classical Dance	85.5%
Sharma P, Sharma N [11]	Feed-Forward Neural Network with PCA and SVD	Custom Gesture	95.9%
		Custom Posture	90.3%
Hakim NL et. al., [12]	3DCNN+LSTM	Custom (Early Fusion + Depth + RGB hand only)	95.8%
Proposed Model	Deep Learning with CNN	Custom	97.78%

To perform real-time prediction to control the media player, the user has to click the start web camera button on the user interface and the OpenCV frames will be displayed to start the prediction as shown in Fig. 9.

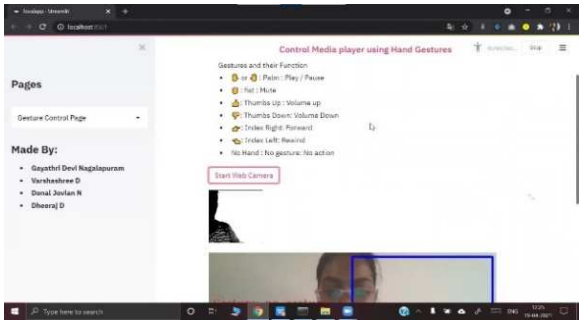

Fig. 9. The web camera turns on when the button is clicked.

The real-time working of the system when different gestures are shown in front of the webcam is displayed from Fig 10 to Fig 17. The images show how the media player is controlled using hand gestures. When the fist is shown the first time the volume is muted and when it is predicted again the volume is unmuted. Similarly, the media player is controlled with the other six gestures. It has given equal responses from other media players too.

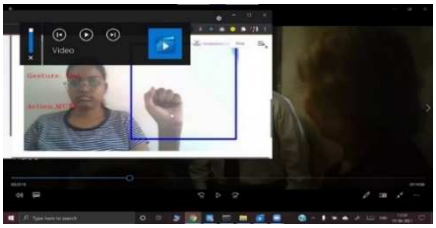

Fig. 10. Media player is muted when the fist is detected.

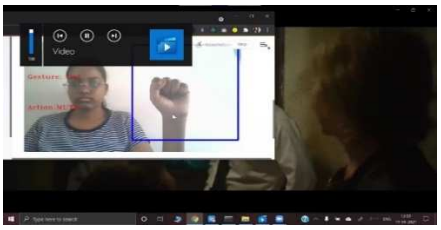

Fig. 11. Muted state is unmuted when the fist is detected.

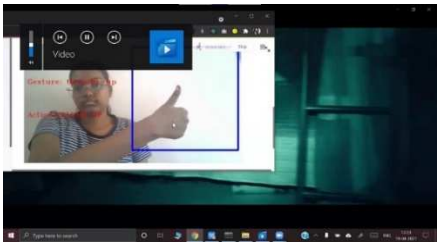

Fig. 12. Volume is increased when thumbs up are detected

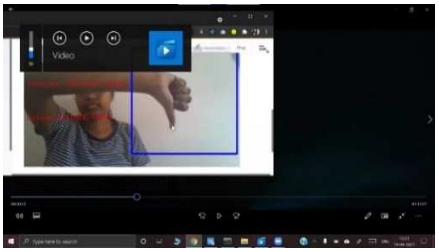

Fig. 13. Volume is decreased when thumbs down is detected

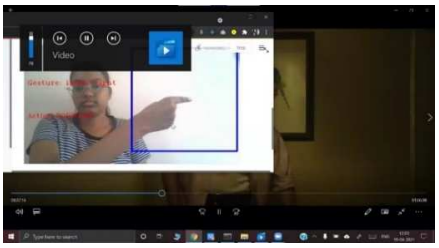

Fig. 14. Video is forwarded when index right is detected

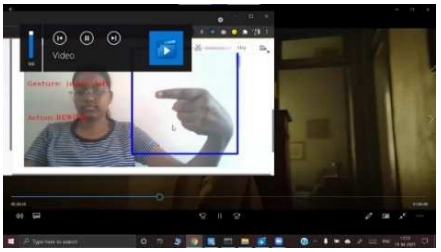

Fig. 15. Video is rewound when index left is detected

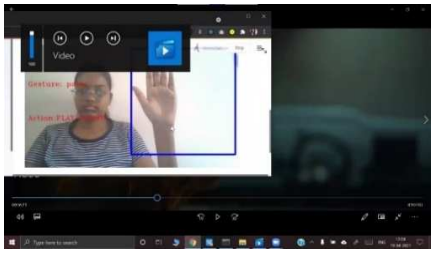

Fig. 16. Video is paused while playing when the palm is detected

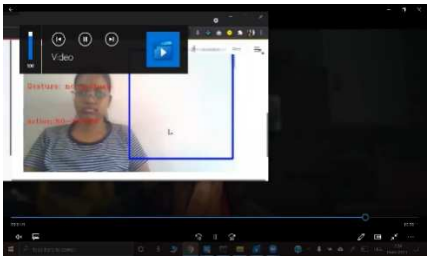

Fig. 17. Media player continues in its state when no gesture is detected

The project sources files are hosted in a web app using streamlit.io sharing which enables the system to be used from their respective devices. It also provides a link to download the source files. Fig.18 to Fig. 20 shows different pages of the web app deployed on streamlit.

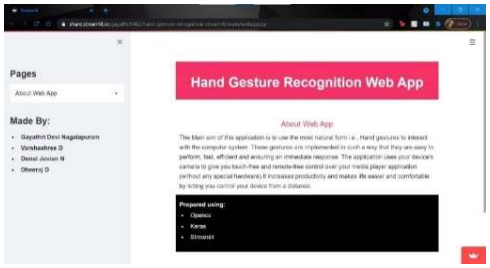

Fig. 18. About page of streamlit web application

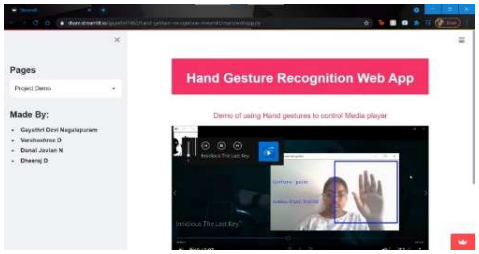

Fig. 19. Demo page of streamlit web application

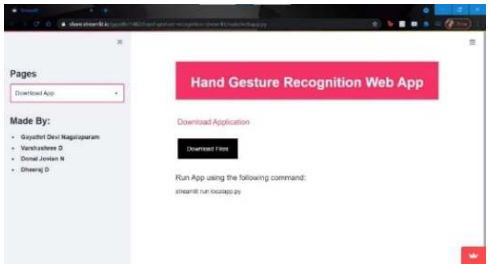

Fig. 20. Download page of streamlit web application

The proposed CNN model for the hand gesture recognition system acquired high accuracy. Fig. 21 shows the comparison of the accuracy of train and test sets for the third CNN model and Fig.22 shows the comparison of the loss of both train and test sets of the third model of CNN. The convergence of the graphs concludes that the model can be used for prediction and has no overfitting or underfitting issues.

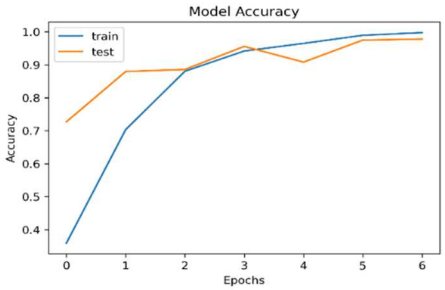

Fig. 21. Graph of variations in Accuracy with Epoch in CNN model.

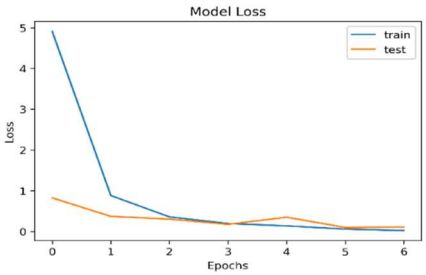

Fig. 22. Graph of variations in Loss with Epoch in CNN model.

The performance of the CNN model is also evaluated using different performance metrics such as precision, recall, and f1-score and displayed using the classification report. Fig. 23 shows the classification report of model 3.

The CNN model is implemented with a fully connected softmax layer as the final layer. Fig. 24 shows the best CNN model with 97.7% testing accuracy and misclass of only 2% on the custom dataset.

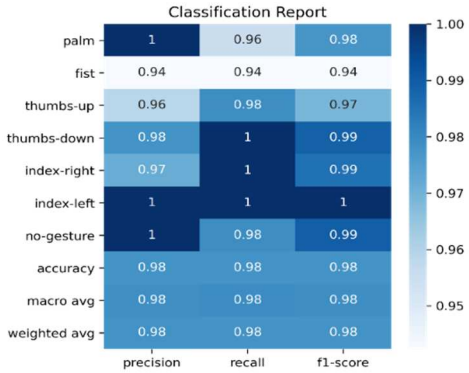

Fig. 23. Classification report of the CNN model

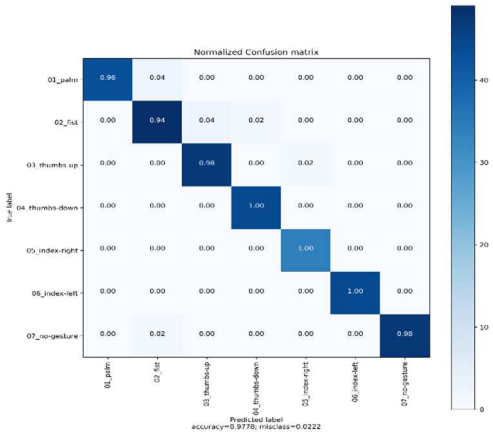

Fig. 24. Confusion matrix of the CNN model

V. CONCLUSIONS

This paper presents the work to control the media player using the hand gesture recognition system. The OpenCV techniques are used to capture the images, 2-Dimensional Convolutional Neural Network is used to extract features and predict the gestures, PyAutoGUI is used to control the keyboard keys whenever a gesture is integrated with it is predicted. A custom dataset of 7 gestures is collected to test the proposed model. This model is also tested with these gestures in real-time to examine the accuracy of the proposed system. The proposed CNN model achieved a high accuracy of 98%, providing a user-friendly, cost-effective approach to interaction with computer systems. Thus, the proposed system is a true real-time model with low to negligible latency. The future scope is to work on improving the gesture recognition capabilities in varied environments such as illumination levels. Also, the integration of more functions is proposed with new hand gestures to use with other applications such as typing in word documents and web browsers.

REFERENCES

[1] Zhuang H, Yang M, Cui Z, Zheng Q. A method for static hand gesture recognition based on non-negative matrix factorization and compressive sensing. IAENG International Journal of Computer Science. 2017 Mar 1;44(1):52-9.

[2] Paul PK, Kumar A, Ghosh M. Human-Computer Interaction and its Types: A Types. International Conference on Advancements in Computer Applications and Software Engineering (CASE 2012), At Chittorgarh, India. –2012 2012 Dec 21.

[3] Ritupriya G.Andurkar, Human–computer interaction. International Research Journal of Engineering and Technology, (ISSN: 2395-0072 (P), 2395-0056 (O)). 2015; 2(6):744-72.

[4] [4] Shanthakumar VA, Peng C, Hansberger J, Cao L, Meacham S, Blakely V. Design and evaluation of a hand gesture recognition approach for real-time interactions. Multimedia Tools and Applications. 2020 Feb 21:1-24.

[5] Bashir A, Malik F, Haider F, Ehatisham-ul-Haq M, Raheel A, Arsalan A. A smart sensor-based gesture recognition system for media player control. In2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) 2020 Jan 29 (pp. 1-6). IEEE

[6] Gope DC. Hand Gesture Interaction with Human-Computer. Global Journal of Computer Science and Technology. 2012 Feb 10.

[7] Zhao L. Gesture Control Technology: An investigation on the potential use in Higher Education. University of Birmingham, IT Innovation Centre: Birmingham, UK. 2016 Mar.

[8] Harshada Naroliya, Tanvi Desai, Shreeya Acharya, Varsha Sakpal Enhanced Look Based Media Player with Hand Gesture Recognition. International Research Journal of Engineering and Technology, (ISSN: 2395-0072 (P), 2395-0056 (O)).2018;5(3): 2032-35.

[9] Wachs JP, Kölsch M, Stern H, Edan Y. Vision-based hand-gesture applications. Communications of the ACM. 2011 Feb 1;54(2):60-71

[10] Yashas J, Shivakumar G. Hand Gesture Recognition: A Survey. In2019 International Conference on Applied Machine Learning (ICAML) 2019 May 25 (pp. 3-8). IEEE.

[11] Sharma P, Sharma N. Gesture Recognition System. In2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) 2019 Apr 18 (pp. 1-3). IEEE.

[12] Hakim NL, Shih TK, Kasthuri Arachchi SP, Aditya W, Chen YC, Lin CY. Dynamic hand gesture recognition using 3DCNN and LSTM with FSM context-aware model. Sensors. 2019 Jan;19(24):5429.

[13] Cardenas EJ, Chavez GC. Multimodal hand gesture recognition combining temporal and pose information based on CNN descriptors and histogram of cumulative magnitudes. Journal of Visual Communication and Image Representation. 2020 Aug 1; 71:102772.

[14] Adithya V, Rajesh R. A deep convolutional neural network approach for static hand gesture recognition. Procedia Computer Science. 2020 Jan 1;171:2353-61.

[15] Oudah M, Al-Naji A, Chahl J. Hand gesture recognition based on computer vision: a review of techniques. journal of Imaging. 2020 Aug;6(8):73.

[16] Mohammed AA, Lv J, Islam MD. A deep learning-based End-to-End composite system for hand detection and gesture recognition. Sensors. 2019 Jan;19(23):5282

[17] Pinto RF, Borges CD, Almeida A, Paula IC. Static hand gesture recognition based on convolutional neural networks. Journal of Electrical and Computer Engineering. 2019 Oct 10;2019

[18] Cao Z, Hidalgo G, Simon T, Wei SE, Sheikh Y. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. IEEE transactions on pattern analysis and machine intelligence. 2019 Jul 17;43(1):172-86.