Gesture Controlled Media Player

Supriya. Telsang, Dhanashree.S.Petare, Vaishnavi.Pawar, Swapnil Pawar, Jeyeshtha Pendharkar, Harshal Pendor, Vedant Pawar
Department of Engineering, Sciences and Humanities (DESH)
Vishwakarma Institute of Technology, Pune, 411037, Maharashtra, India

Abstract - Human-Computer Interaction has greatly changed in the modern era of sophisticated technologies. Thus leading to reduced task complexity and more focus on user involvement. This paper will Investigate into the conceptualization, development and execution of an advanced gesture-controlled media player. Among the major issues examined here include the technology stack that supports the gesture recognition algorithms as well as user interface (UI) design principles designed specifically to cater for the gestures in the system.

Keywords - Convolutional neural network (CNN module), Image acquisition, Open CV, Deep Learning, PyAutoGUI, Media player control.

I. INTRODUCTION

In recent years, the development of gesture-controlled systems has garnered significant attention in the field of human-computer interaction (HCI). These systems leverage natural hand gestures to provide an intuitive and immersive way to interact with digital devices, eliminating the need for traditional input methods like keyboards and mice. The aim of this project is to design and implement a gesture-controlled media player system, allowing users to manage their media playback through simple hand movements. This innovation seeks to enhance user experience by providing a more seamless and engaging way to interact with multimedia content.

A considerable amount of research has been conducted in the domain of gesture recognition and its applications. Early studies focused on the basics of gesture recognition algorithms and the hardware required for capturing gestures, such as cameras and sensors. More recent research has explored advanced machine learning techniques to improve the accuracy and responsiveness of these systems. Applications of gesture control have been extended to various fields, including gaming, virtual reality, and assistive technologies for individuals with disabilities. Despite these advancements, many existing systems still face challenges in terms of reliability, ease of use, and the ability to function effectively in diverse environments.

This project addresses several gaps that have been identified in previous research. Specifically, many existing gesture-controlled media players struggle with precision and consistency, particularly in varying lighting conditions and with different hand sizes and shapes. Additionally, user-friendly calibration and customization options are often lacking. To overcome these shortcomings, our project incorporates a robust gesture recognition algorithm that utilizes deep learning techniques to improve accuracy and adaptability. Moreover, we have implemented a user-friendly interface that allows for easy calibration and customization of gestures, ensuring a more personalized and efficient user experience. This novel approach not only enhances the functionality and reliability of gesture-controlled media players but also expands their accessibility and usability in real-world scenarios.

II. METHODOLOGY/EXPERIMENTAL

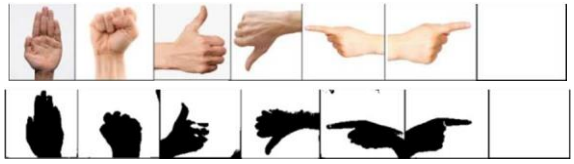
➢ Algorithm:

1. System Overview:

- The proposed gesture detection system for media player control is broadly divided into two stages.
- The first stage involves the detection of hand gestures.
- The second stage integrates the keyboard controls with each detected gesture.

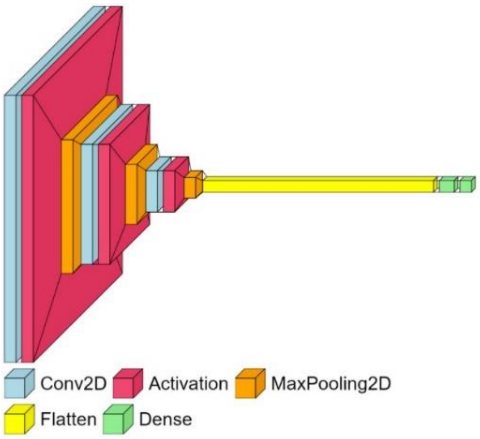
2. Gesture Recognition Implementation:

- The gesture recognition is implemented using computer vision and deep learning techniques.
- A custom-built dataset containing 7 gestures is created and used for the system.
- The raw images from the dataset are converted to black and white images.



3. CNN Model Architecture:

- The CNN model is trained using the image frames from the dataset.
- The model consists of three convolutional layers with ReLU activation functions.
- Pooling layers are added to perform max pooling, followed by flattening and dense layers.



4. Gesture Prediction and Control:

- The trained CNN model is used to predict one of the seven gestures.
- The input images are preprocessed and resized before being passed into the model.
- Each detected gesture is mapped to an individual keyboard control, which can be used to control the media player.

➢ Image Acquisition and Pre-processing:
- Image frames are captured from a live video feed using OpenCV when the user performs hand gestures in front of a webcam.
- The collected images are converted to black and white to enhance gesture prediction accuracy and then stored in designated directories.
- The dataset comprises gestures from three individuals, with 150 images captured for each gesture.
- During camera operation, two frames are displayed on the screen, allowing users to capture images frame by frame using the read function with a mirrored image simulation.
- Users position their hand within the Region of Interest (ROI) or bounding box and perform gestures for image extraction.
- Extracted frames from the ROI are resized to 120x120x1 dimensions.
- The image count in each directory is displayed on the screen, incrementing as users capture images by pressing number keys (0 to 6) on the keyboard, saving images to their respective class directories.
- Preprocessed images are previewed in a small frame during capture and stored in the dataset.
- Users can exit the data collection process by pressing the escape key on the keyboard.

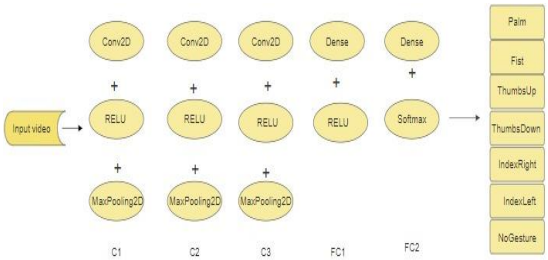


Fig. Proposed CNN model

➢ Feature Extraction:
The process involves importing Keras models and essential hidden layers to construct convolutional networks.
The CNN model is structured with a hidden input layer, followed by two convolutional layers.
Each convolution layer is supplemented with an activation function known as ReLU and a pooling layer known as MaxPooling.
A flattening layer is introduced, along with two fully connected layers: one utilizing ReLU activation and the other employing softmax activation for classification purposes.
The architecture of the CNN model employed for feature extraction and gesture classification is depicted in Fig.

➢ Model Compilation:

The CNN model is compiled by utilizing the adam algorithm as an optimizer, categorical_crossentropy as the loss function for error determination, and accuracy as the performance metric for model evaluation.

➢ Model Training:

The training process involves utilizing the Image Data Generator class to create batches of images for model training and validation.
The model is trained using the fit function with a specified number of epochs.
Upon completion of training, the trained model is serialized and saved in JSON format, while the weights are directly saved from the model using the save_weights function.

➢ Media Control with Predicted Hand Gestures:

- The trained model, stored in JSON format along with its weights, is loaded to predict hand gestures.

- Additionally, PyAutoGUI for keyboard key integration with gestures and Streamlit for creating a user interface are imported.
- Using the Streamlit web framework, three distinct web pages are generated. The first page serves as an introduction to the project, while the second page showcases a video demonstration of the project.
- The third page, known as the demo page, allows users to predict hand gestures to control the media player.
- Upon clicking the start button on the web page, the webcam initiates, enabling users to execute hand gestures within the Region Of Interest (ROI) for prediction by the trained CNN model.
- Each predicted gesture is linked to a specific keyboard key through PyAutoGUI, utilizing conditional if-else statements to trigger the associated action.
- Gesture-key mappings are established, with each gesture assigned a keyboard control and a corresponding label.
- Each gesture trigger is set to execute the integrated control function once, indicated by a single press assignment.
- Users can exit the system by pressing the escape key on the keyboard.
- The video frame displays the predicted gesture and the corresponding action whenever users interact with the system to control the media player.

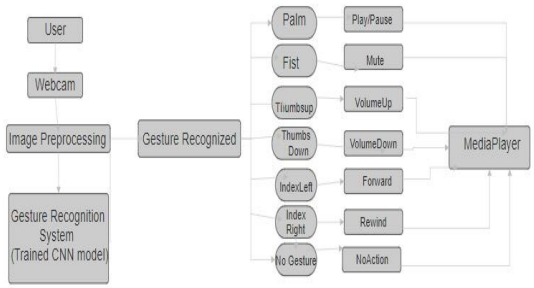


Fig. The System Design Workflow

TABLE I: GESTURES AND THEIR RESPECTIVE ACTIONS

Predicted Gesture	Action	Keyboard Keys
Palm	PLAY/PAUSE	Space
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

III. RESULTS AND DISCUSSIONS

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 118, 118, 32)	320
max_pooling2d_6 (MaxPooling2D)	(None, 59, 59, 32)	0
conv2d_7 (Conv2D)	(None, 57, 57, 64)	18,496
max_pooling2d_7 (MaxPooling2D)	(None, 28, 28, 64)	0
conv2d_8 (Conv2D)	(None, 26, 26, 128)	73,856
max_pooling2d_8 (MaxPooling2D)	(None, 13, 13, 128)	0
flatten_2 (Flatten)	(None, 21632)	0
dense_4 (Dense)	(None, 256)	5,538,048
dense_5 (Dense)	(None, 7)	1,799

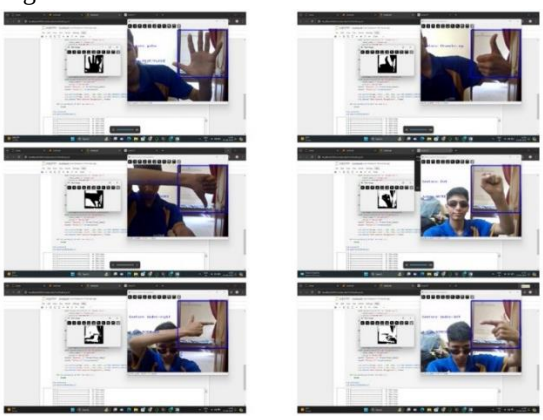
Total params: 16,897,559 (64.46 MB)
Trainable params: 5,632,519 (21.49 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 11,265,040 (42.97 MB)

TABLE II. EVALUATION TABLE

	Precision	Recall	F1-score	Support
dataset	0.99	1.00	1.00	112
images	0.00	0.00	0.00	1
accuracy			0.99	113
Macro-avg	0.50	0.50	0.50	113
Weighted-avg	0.98	0.99	0.99	113

Output Images:
Image 1:

Image 2:



IV. CONCLUSION

In this article, we present a project consisting of controlling the media player via hand-gesture recognition. It involves using opencv techniques to take pictures, 2D convolutional neural network (CNN) to analyze them and forecast respective gestures as well as pyautogui for pressing keys on the keyboard in response to any predicted gesture that is accompanied by one. To test out our model we created our own database containing seven different gestures. The cnn model being proposed has a 98% high accuracy and thus providing an affordable and friendly way in which people can interact with computers; so it is indeed a real time model that has minimal or no much delays hence latency that are reasonable. For the coming days ahead though not limited to light exposure levels; work should be done on enhancing gesture recognition ability for diverse settings, more functions can be added alongside different hand signs for use in various software programs.

REFERENCES

[1] Paul PK, Kumar A, Ghosh M. Types of Human-Computer Interaction and their types, International Conference on Advancements in Computer Applications and Software Engineering (CASE 2012), in Chittorgarh, India. 2012, Dec 21

[2] 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.

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

[4] Human-Computer Interaction through hand gestures was discussed by Gope DC in the year 2012 on February 10 in the Global Journal of Computer Science and Technology.

[5] Molina-Cantero, A. J., Guerrero-Curieses, I., Rodríguez-Ascariz, J. M., & García-Bermejo, J. R. (2018). A Comparison of Gesture Recognition Approaches for a Multimedia Player Control System. IEEE Access, 6, 35055-35066

[6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT press.

[7] Smedt, Q. D., Wannous, H., & Vandeborre, J. P. (2016). Gesture recognition using depth data for natural human-computer interaction. In 2016 International Conference on 3D Imaging (IC3D) (pp. 1-8). IEEE.

[8] In 2018, Harshada Naroliya et al. published an article on a new look-based media player in the International Research Journal of Engineering and Technology in which they incorporated hand gesture recognition.

[9] Sharma, P.; Sharma, N. Gesture Recognition System. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 18–19 April 2019 (pp. 1-3). IEEE.

[10] Pigou, L., Dieleman, S., Kindermans, P. J., & Schrauwen, B. (2015). Sign language recognition using convolutional neural networks. In European Conference on Computer Vision (pp. 572-578). Springer, Cham.

[11] 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

[12] Yashas and Shivakumar explore hand gesture recognition in their work “Survey”. This study was presented at The International Conference on Applied Machine Learning (ICAML) 2019 May 25 (pp. 3-8) by IEEE.