Author: Viet Thanh Le
PSU Computer Vision CS510 Final Project

# A Fully Customizable Face Gesture Recognition System With MediaPipe

## Abstract

This paper presents a fully customizable face gesture recognition system designed for a wide range of applications, including human-computer interaction, security, and entertainment. The proposed system leverages advanced deep learning models and an experimental user interface to allow end-users to define and train unique face gestures. Our experiments demonstrate the system's high accuracy and adaptability across various conditions. We provide a comprehensive evaluation of the system's performance and future development of existing methods.

*Keywords: Face Gesture Recognition, Customizable Gestures, Machine Learning, Human-Computer Interaction*

## 1. Introduction

Facial gestures are a natural and intuitive mode of communication for humans. With the increasing integration of technology into daily life, recognizing and interpreting these gestures has become a significant area of research. This paper introduces a customizable face gesture recognition system that enables users to define and train their unique gestures, enhancing the flexibility and application range of the technology. This paper acts as a proof of concept where with industrial settings, the same methodology can be applied to achieve different objectives that could be more meaningful in healthcare, human-computer interaction…

The primary contributions of this paper are:
- A novel customizable face gesture recognition system.
- A user-friendly interface for defining and training gestures.
- An extensive experimental evaluation demonstrating the system's accuracy and robustness.

## 2. Method

The customizable face gesture recognition system comprises several key components: facial feature extraction, gesture definition and training, and gesture recognition.

### 2.1 Facial Feature Extraction

I utilize the MediaPipe framework to recognize faces and detect facial landmarks. The system processes video input in real-time, analyzing and returning a canonical face

model with more than 450 facial landmarks including overall face, eyes, eyebrows, mouth contour, face tessellation grid and precise iris location.

## 2.2 Gesture Definition and Training

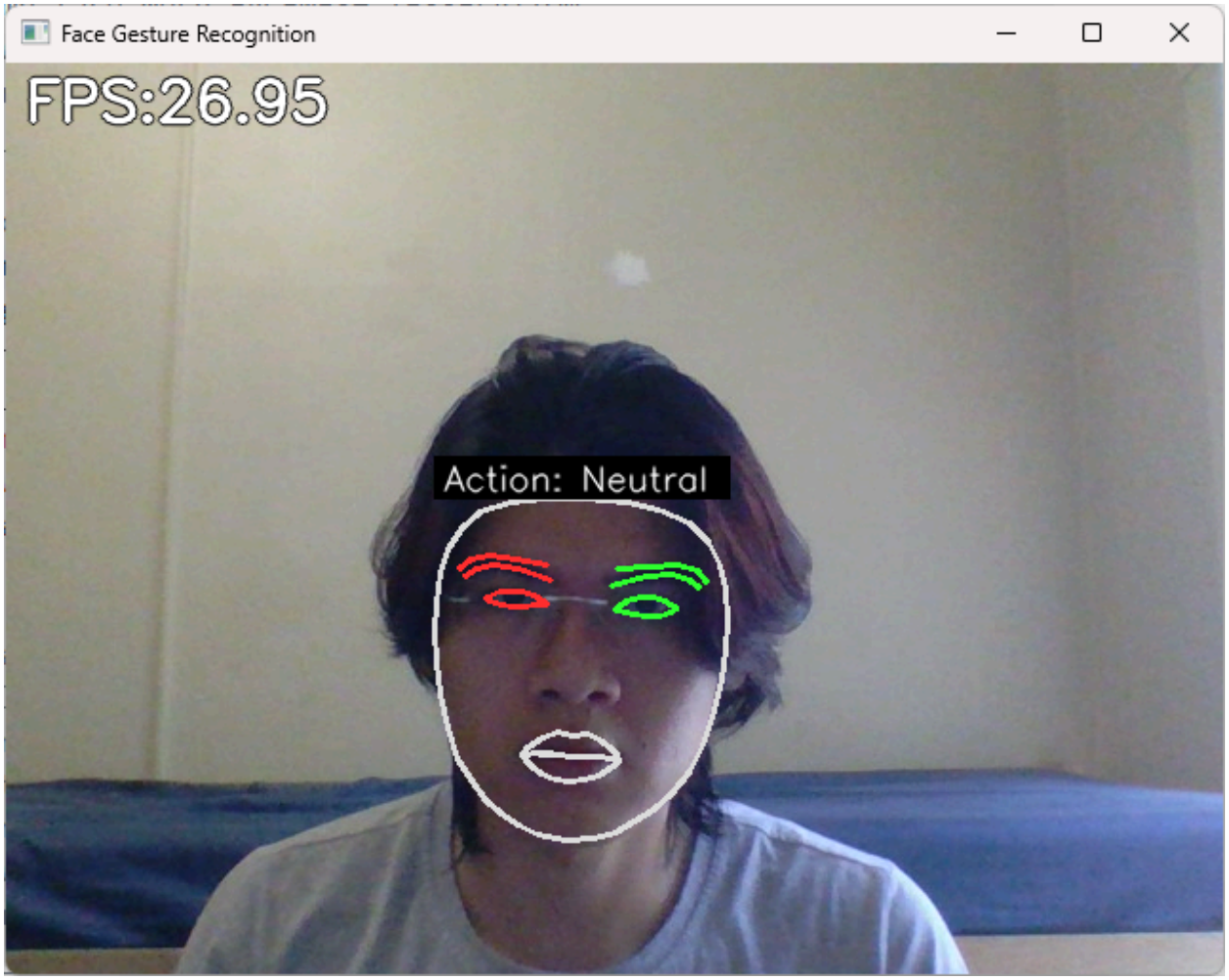
Inspired by the work of Kazuhito Takahashi in recognizing hand gestures and sign language recognition, I continue to venture into customizable facial gestures. Users can define gestures through an interface, performing desired gestures in front of the camera. The system captures these gestures and uses a supervised learning algorithm to train a deep learning model specific to each gesture.

## 2.3 Gesture Recognition

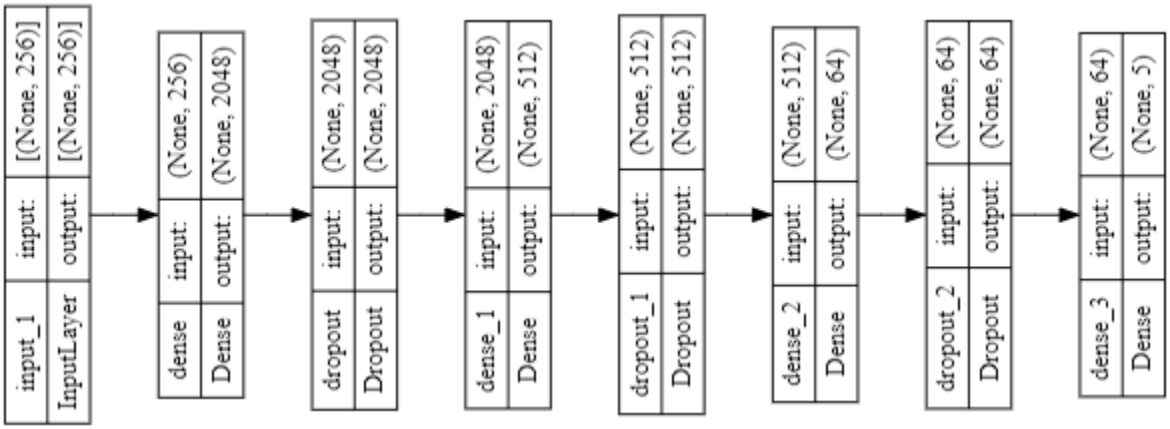
I define two separate types of facial gestures: static and dynamic gestures. Static gesture refers to a gesture that can be completed and recognized by a machine despite movements. Dynamic gesture refers to a gesture that could not be completed or recognized without previous facial states. Both types are usually hard to define and often are used interchangeably in some systems. To simplify the idea, I define some examples that are real labels in my system in the table below.

| Static gestures | Dynamic gestures |
|---|---|
| Left/Right wink | Head tilt left |
| Raise eyebrows | Head tilt right |
| Gasp | Nod |
| Neutral | Neutral |

- Static gesture: Users can create a snapshot of their gesture in real time, define its label and feed into the training pipeline to complete the definition of the new gesture. System only captures one frame and analyzes related facial landmarks (highlighted in the picture below)




The architecture of static gesture is:




*Source: Keras API extension*

The details of the input layer will be discussed in the next section. The architecture of this model is fairly simple, I add more hidden neuron layers to capture complex

patterns in the dataset and dropout layers to make the model more robust. The advantage of this model is that it is scalable if the number of labels grows linearly.

- Dynamic gesture: Users create a series of snapshots of their gesture in real time, repeat the same procedure with static gesture. System captures a series of consecutive frames (default to the highest fps of the current system, which is 32 in this paper) of the pre-selected facial landmarks (highlighted in the picture below relatively). The reason for this selected collection of points is to reduce the computational power needed to classify dynamic gestures but generally still capturing adequate information for models to learn.

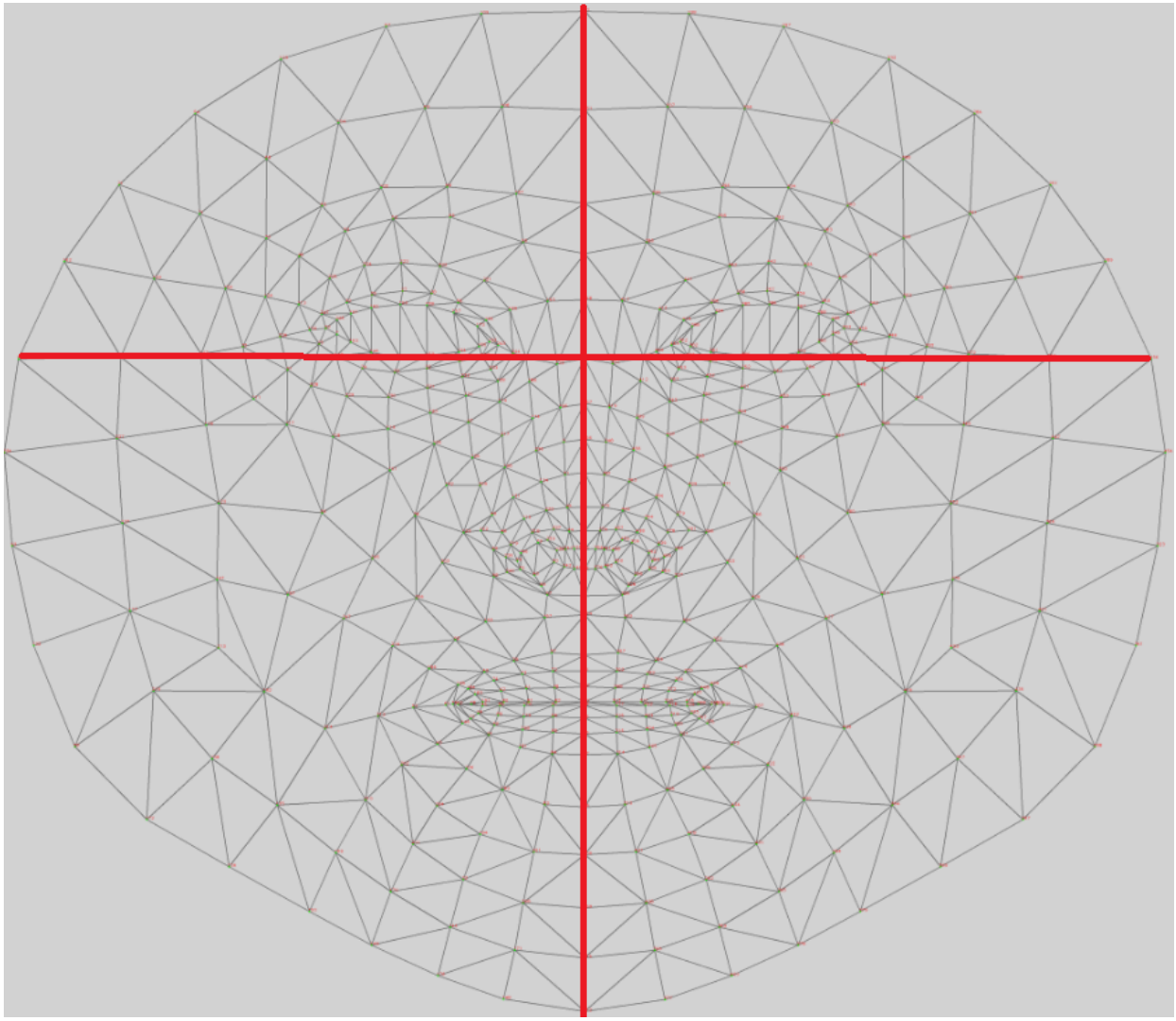
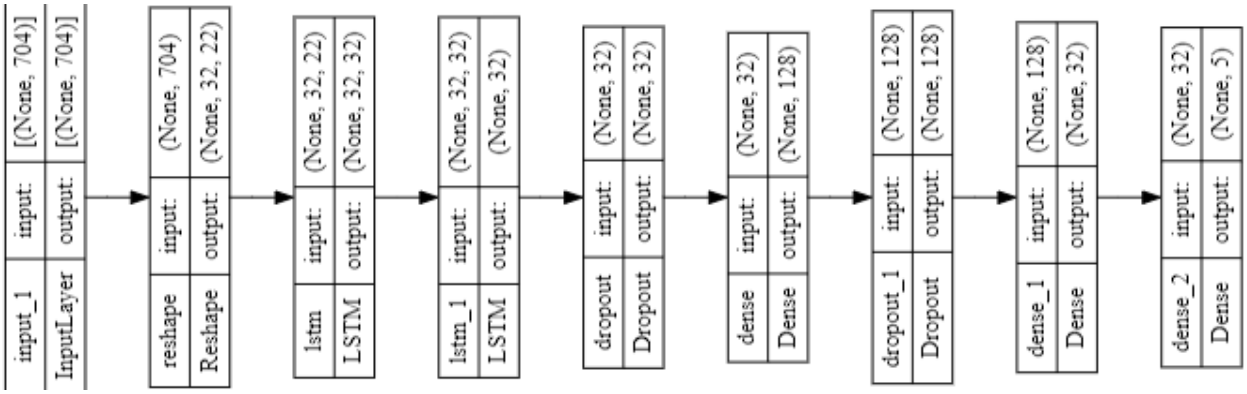


*Image source: Mediapipe*

The architecture of dynamic gesture is:




*Source: Keras API extension*

The architecture of this model utilizes the power of LSTM nodes where we try to understand what movements are key to decide what action that is. I also add some dropout layers to make the model more robust to noises in the dataset. In terms of implementation, we need to be cautious about the training performance since the dropout layers could have disabled key data points where it could significantly affect the model's performance.

## 3. Experiments

The experiments were designed to test the system's ability to capture real time landmarks, training time as well as model's complexity and finally model's accuracy, robustness and its potentials.

### 3.1 Experimental Setup

The first step for the system to work is to capture data points and label them. By modifying existing framework from Kazuhito Takahashi code, I was able to easily create two small experimental datasets for static and dynamic gestures. This step should not take too long since the model proves itself to be effective with a limited amount of 100 data points in each category and will be discussed in the following section. The datasets are then processed by two different models as mentioned above and being loaded back into the system to make inferences in real time.

### 3.2 Datasets

- Static gesture: The default facial contour from Mediapipe has 128 selected landmarks and since we collect the coordinates of them, for each data point, we have 256 features.

Dataset distribution:

| Left wink | Right wink | Raise eyebrows | Gasp | Neutral |
|---|---|---|---|---|
| 12.5% (59) | 14.7% (69) | 19% (89) | 22.8% (107) | 31% (146) |

Although the dataset is a bit skewed, it does, in some aspects, reflect that neutral state takes up most of the time when we do inferences and it does not affect our model's performance which will be illustrated in the next section.

- Dynamic gesture: As shown above, we select 11 landmark coordinates across the face to capture general movements of the face and keep the last 32 frames (around 1 second on the tested system) which results in over 700 features for one data point.

Dataset distribution:

| No gesture | Head tilt left | Head tilt right | Nod |
|---|---|---|---|
| 50% (298) | 17% (101) | 16.5% (99) | 16.5% (99) |

As mentioned above, the distribution of the dataset is not balanced due to its nature, reflecting inference time distribution. Since the data is collected manually, the data is limited and noisy.

### 3.3 Evaluation Metrics

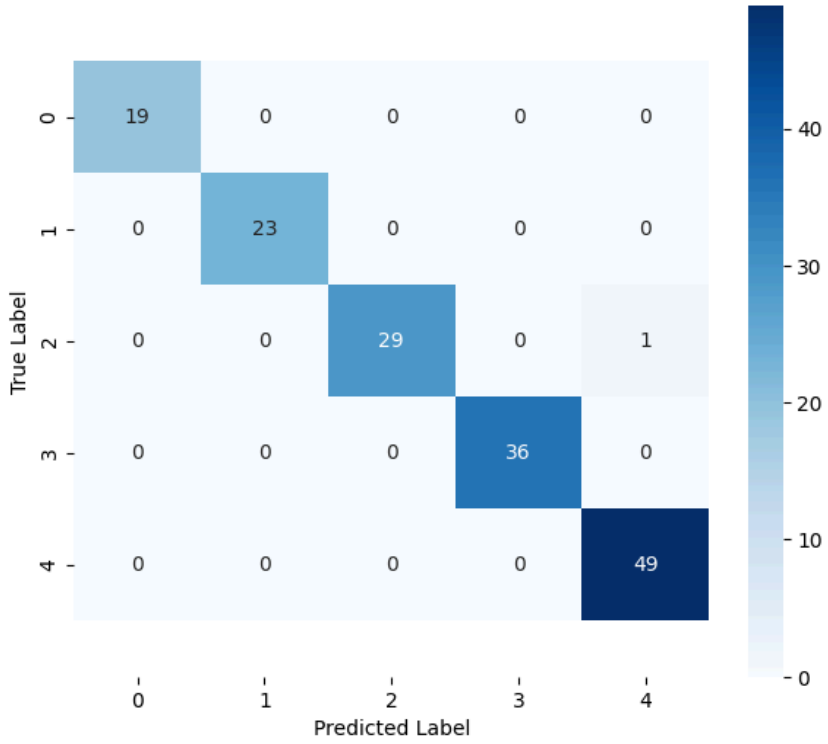
Both models use *adam* optimizer, *sparse category cross entropy* loss function and use *accuracy* as evaluation metric. The overall performance of the system was evaluated using accuracy, f1-score, precision and recall.

### 3.4 Results

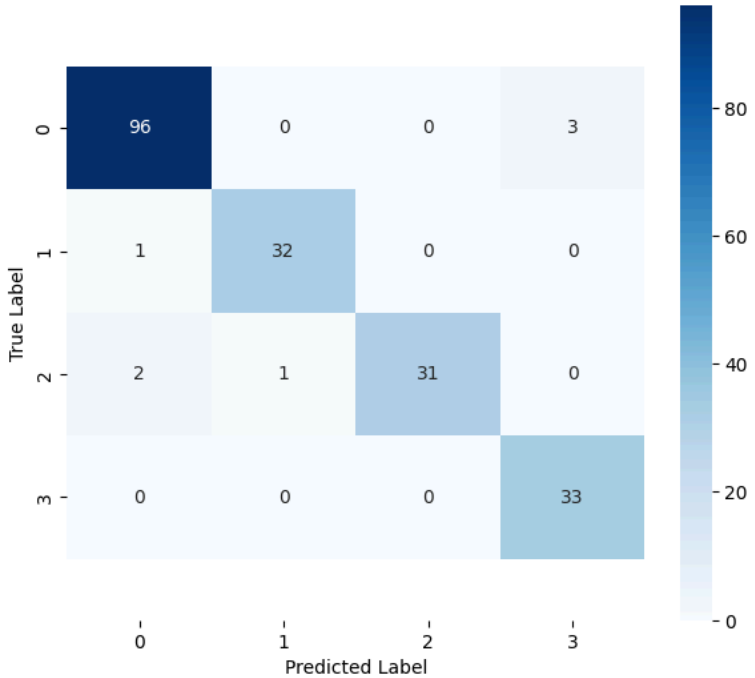
The experimental results indicate that our system achieves a high recognition accuracy of 99.4% and 96.5% in static and dynamic gestures respectively. The system's performance remained robust despite the shortage of data across different lighting conditions and backgrounds as long as a face is detected.

| | Accuracy | F1-score | Precision | Recall |
|---|---|---|---|---|
| Static gesture | 99.36% | 0.99 | 0.99 | 0.99 |
| Dynamic gesture | 96.48% | 0.96 | 0.96 | 0.96 |

As we can see in the table, with a minimal dataset, the model proves to have captured different face gestures with high accuracy.




Confusion matrix for static gestures on testing data




Confusion matrix for dynamic gestures on testing data

### 3.5 Time complexity

Overall, our models can make inferences in real time, the static gesture model is quantized and it does not take long to make predictions, almost having no effect on FPS. The dynamic gesture model on the other hand is not quantized and makes the system slower (it usually lowers FPS from 32 to 16) which is acceptable as it makes predictions using CPU and totally scalable using GPU.

### 3.6 Discussion & Future work

My findings suggest that the face gesture recognition system is feasible given an adequate amount of training data. My ultimate goal is to capture facial action units in real time and give predictions on what the person is expressing and this acts as a proof of concept since I am not equipped with the suitable resources. Another approach is to have generalized facial gesture recognition models and end users can calibrate accordingly. With that said, facial, hand, pose gestures recognition plays an important role in accessibility for people needing it.

## 4. Conclusion

This paper presented a fully customizable face gesture recognition system that empowers users to define and train their unique gestures. Our experiments demonstrate the system's high accuracy, robustness and the potential of this method.

## 5. References

*Github*, https://github.com/google-ai-edge/mediapipe/wiki/MediaPipe-Face-Mesh.

Accessed 6 June 2024.

*Github*, https://github.com/Kazuhito00/hand-gesture-recognition-using-mediapipe.

Accessed 6 June 2024.