Computer Vision Project

“Hand Gesture Recognition”

Submission by

Group no. 2

Team Member

Anurag Pandey [23PGAI0059]

Problem Statement:

A company XYZ Ltd intends to adopt facial recognition-based access authorisation for its employees. But to start-off they want to do a small POC for their Board Room, which is accessible to only a few senior leaders. So, the face recognition model should recognise & grant access to all these senior leaders who are authorised to access this Board Room and, also deny access to everyone else.

Dataset: Collected by Self.

GitHub link:

https://github.com/anuragdhirubhai/CV_Assignment_march

Solution Approach:

Here we assume camera & other hardware installation is already in place and our part is to train this Face Recognition ML model and ensure appropriate model performance. Model will take image as input and will predict the pre trained class level.

Following models and algorithm will be used in this.

1. Image Collection

2. Feature extraction using cv2

3.Classification — using Deep Learning models

5. Testing model on fresh image

Workload Division:

1. Anurag Pandey: Problem Statement, Data Collection, Feature extraction, Classification, Testing models on fresh image

Introduction

Hand gesture recognition is an area of computer vision that has been increasingly studied in recent years. It involves the use of machine learning algorithms to recognize and interpret hand movements and gestures. Hand gesture recognition technology has been widely used in various applications like virtual reality, human-computer interaction, gaming, and healthcare, etc. In this project, we aim to use deep learning techniques such as CNN for feature extraction and classification for hand gesture recognition. We will use a dataset of hand gesture images consisting of 6 classes for this project.

About Data

We will collect data for hand gesture recognition by capturing 400 images of our own hand gestures using a standard camera for alphabets and numbers based on American sign language. The images will be of the same size and follow a standard naming convention to generate labels. To ensure the dataset includes a variety of hand gestures, we will perform the gestures in different lighting conditions, backgrounds, and angles. This dataset will be used for training and testing our model for hand gesture recognition.

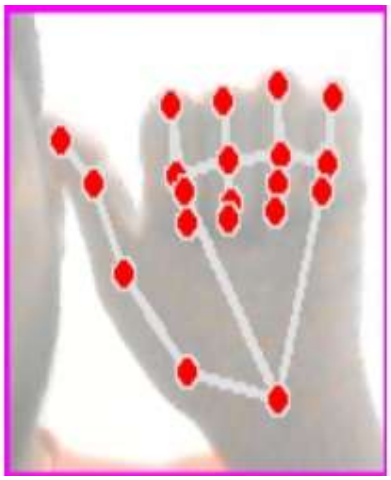
Methodology

For hand gesture recognition, the project methodology includes several steps, such as data collection, pre-processing, feature extraction, dimensionality reduction, and classification. In this project, we will use a standard camera to capture 400 images of each hand gesture, ensuring that all images are of the same size and follow a standard naming convention. To ensure the dataset's diversity, we will perform hand gestures in various lighting conditions, backgrounds, and angles. We will then pre-process the images and extract relevant features using DL techniques.

Pre-processing: We first resized all the images to a standard size. Resizing the images helps in reducing the computational complexity and also ensures that all the images have the same size.

Feature Extraction: This code is capturing the image from a video camera and detecting a hand using a hand detector. The detected hand is then cropped from the image and resized to a fixed size. The aspect ratio of the cropped image is maintained, and it is padded with white pixels to make it a square. The cropped and resized image is displayed on the

screen for the user to verify. The code also saves the image in a specified directory if the "c" key is pressed. This process can be repeated to collect a dataset of images for training a model for hand gesture recognition.



Label – A

Label - B

Dimensionality Reduction: Once the hand is located, the code creates a bounding box around the hand and crops the image based on the bounding box. Then it resizes the cropped image to a square image of a specified dimension using OpenCV's resize function. It also maintains the aspect ratio of the image by padding the image with zeros if necessary.

Classification: Image classification is a process where an algorithm identifies images by assigning labels based on their features. Transfer learning is a technique used in image classification that involves taking a pre-trained model and fine-tuning it to perform a specific task. Pre-trained models are deep neural networks that have learned various features of images, like edges, textures, and shapes, from a large dataset like ImageNet. To use transfer learning for image classification, we remove the pre-trained model's classification layer and add a new classification layer that is trained on our dataset. By fine-tuning the pre-trained model's weights, we can classify new images with high accuracy, even with small datasets. Transfer learning is useful in image classification because it allows us to leverage the knowledge learned from large datasets without the need for large amounts of data

Model Training: Due to system limitation I have used Transfer learning technique to train my model on a pretrained model available on Google Teachable Machine website to produce the output.

Model Testing: After training, tested the model on the real test image given in real time.

Testing model on fresh images:

To test the model using real-time video, I will be using a video camera to capture my gestures and feed it to the model for prediction. The model will analyse the frames from the camera feed in real-time and predict the label or gesture corresponding to the given input. This will be done by running the prediction code file on the frames captured by the video camera and displaying the predicted label or gesture in real-time. This approach will allow me to test the model's accuracy and performance in real-world scenarios and help me evaluate the effectiveness of the model for real-time gesture recognition applications.

Further Improvements:

For further improvements I will be adding gesture for all the alphabets from A to Z and numbers from 0 to 9, since the model is not performing extremely well, I will be fine tuning the data model and check for best performing model that can be applied.

Conclusion

The model is working fine but the labels are not being predicted clearly for every image so, there is still room for improvement in our model, and we can experiment with different techniques to improve its performance.