Control the Mountain-Car-v0 Gym env using webcam

Step 1: Study Mountain-Car-v0 Gym environment

This environment takes three actions 0,1 and 2.

Where 0 is move left, 1 is no movement and 2 is move right.

Step 2: Create Dataset

Number of gestures to control car will be three since there are 3 different actions in Mountain-Car-v0 Gym environment.

Control of mountain car will be done through webcam so; gestures must be very quick to change and webcam angle should always be the same with respect to the hand.

So, three distinct gestures are selected which can be change quickly and only facing front of hand to the webcam.

Fig. 1
Action = 1

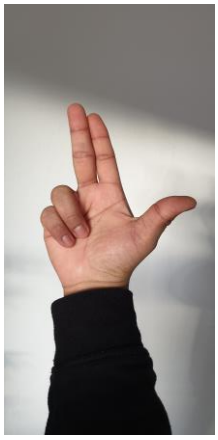

Fig. 2
Action = 2

Fig. 3
Action = 0

To create dataset, pictures are taken from both mobile phone and webcam. So that number of images can be increased and different quality and angles can be included in the dataset.

Dataset includes images where only gesture is present in the image i.e., Fig 1,2 and 3, and also where gesture and face/body are visible in image to create more generalized dataset.

Risk involved with this dataset are:
Dataset is small.
It includes images of one person only and very minimum background.
It might fail when predicting on image where other objects/body parts are present in image along with gesture.

Almost 20% of the images were used for validation.
Total Images = 159
Training images = 124
- Action 2 images = 44
- Action 1 images = 41
- Action 0 images = 39

Validation images = 35
- Action 2 images = 12
- Action 1 images = 12
- Action 0 images = 11

Step 3: Training Model

Due to small dataset, I preferred not to create custom model from scratch.

With small dataset transfer learning provides advantage and much better performance because it is already trained on very large and general enough dataset.

To select the suitable pre-trained model, I looked for research papers to find out which model works better with gesture detection. And I found out that VGG16 is compatible with gesture detection from this paper.

https://link.springer.com/chapter/10.1007/978-981-16-7118-0_11

When using transfer learning to train model Not every model will provide desired outcome with every dataset.

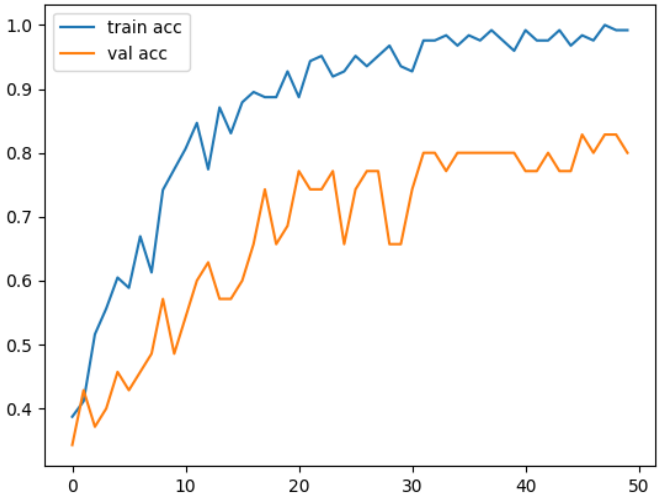

Fig. 4
Training Accuracy and Validation Accuracy

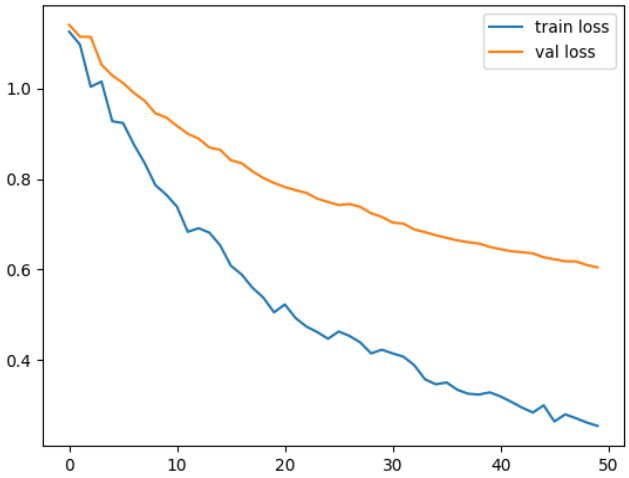

Fig. 5
Training Loss and Validation Loss

From Training Loss and Validation loss it is concluded that the training dataset might be unrepresentative, or the model is underfit. Because training loss is

decreasing continuously and has not become stable. The model can be improved using more generalized dataset and by increasing training time.

It is also concluded from Loss curves that model has trained at good learning rate and Accuracy of model has increased with respect to epochs and time.

Training and Validation Accuracy are promising enough thus it is concluded that the model is ready for testing.

Step 4: Model Testing

For this task the testing of model was done using webcam in real-time. Model was accurate when there was only gesture visible through webcam, but when face is also visible to webcam model made false predictions.

Model also gave false prediction when there was no gesture present in front of webcam.

These false predictions can be result of small dataset to train model and less training time. Model does not extract features precisely for gesture detection.

Model gave desired output when gestures were close enough to webcam and no other object was present in image.

Step 5: Model visual explanation using the Grad-CAM

I have chosen Grad-Cam to interpret model prediction because it produces a coarse localization map highlighting the important regions in the image for predicting the concept. Which helps to visualize exactly what model looks at in the image for particular class prediction.

Grad-CAM can be used to gain better understanding of a model by providing insight into model failure modes.

Fig. 6 shows what model looks at when detecting gesture for Action = 2.

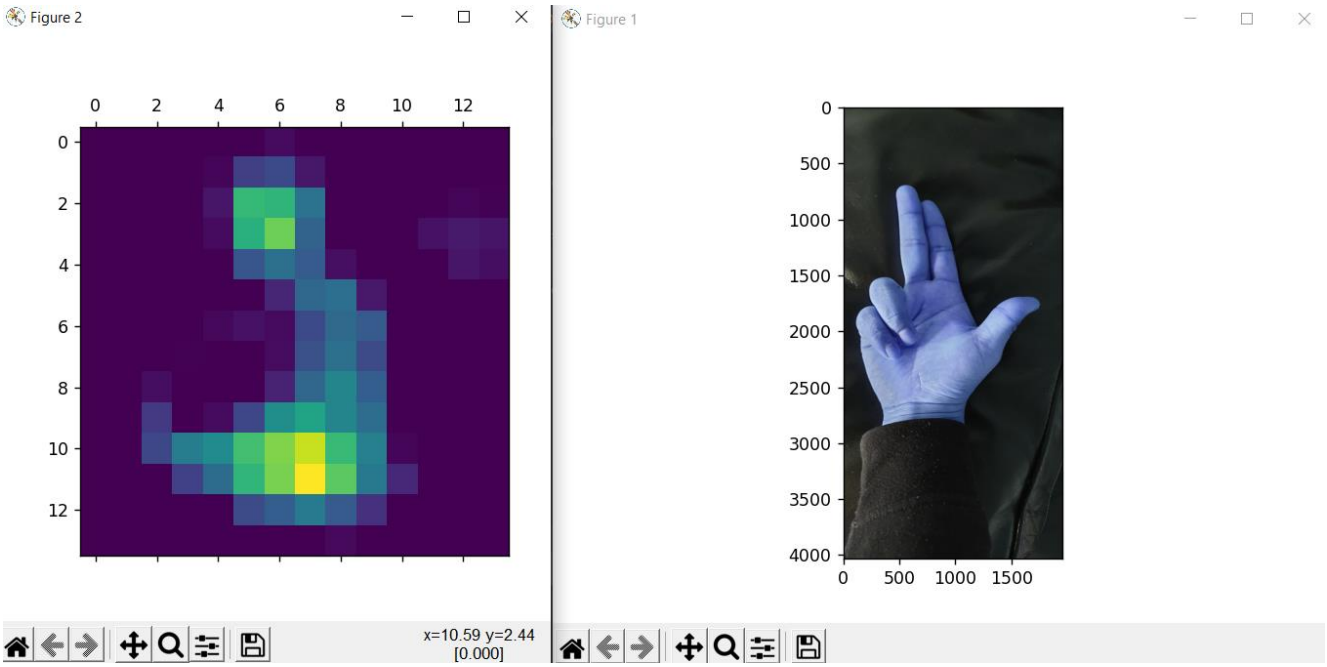

Fig. 6
Grad-CAM Vision

Several images belonging to all class were passed to Grad-CAM function and results were analyzed. From these results it was concluded that model has understanding of difference between all three gestures when there was only gesture present in the image, that's why model gave accurate predictions.

Step 6: Control Mountain Car

Predicted class by model were given to Mountain-Car-v0 Gym env as action.

Control of mountain car was success which can be seen in the video. Model predicted correctly and those predicted class value were given to environment as action and car showed changes in position and velocity according to the given action.

The goal was achieved in first try so there were no further approaches e.g., feature extraction.

Code Flow/ Explanation

1. To create dataset using webcam create_dataset.py python file was created.
    1.1. Create_dataset function takes two arguments as input. First is path where the images will be stored and second is number of images to be taken and save.
2. To train model detection_model.py python file was created.
    2.1. VGG16 model is downloaded without top (Prediction) layer with ImageNet weights.
    2.2. All the weights were frozen to stop them from training.
    2.3. New prediction layer is added with 3 output nodes.
    2.4. Model is created combining VGG16 model and new prediction layer and trained prediction layer.
    2.5. Model training and validation accuracy and loss were analyzed.
3. To test the model live_detection.py python file was created.
    3.1. It converts frames of webcam into images and feeds them as input image to the model. Model prediction were compared and analyzed.
4. To understand the model grad_cam.py python file was created.
    4.1. Grad-CAM function returns heatmap of what model looks at in the image.
5. To control mountain car contro_mountain_car.py python file was created.
    5.1. This is extension of live_detection.py python file. Which takes prediction class as input and feeds them to the Mountain-Car-v0 Gym environment as Action.