Extracting Average Temperature from Thermal Videos - Problem 1

Tanishq Joshi *(B19EE083),* and Suneel Saroj *(B19CSE087)*

Abstract

In this report, we have discussed the complete machine learning pipeline that we used while implementing the project. We have talked about all the pre-processing steps that are involved while implementing the pipeline. We have also discussed the results obtained by using the different models while extracting the average temperatures from the thermal videos.

I. Introduction

The basic goal of our project is to extract the average temperature of a person across a thermal video. It is very essential in the current scenario because in this covid situation, the temperature of a person can determine whether he has covid or not. Thus, our goal in this project is to develop an efficient machine learning pipeline that helps us to extract the average temperature of a person over a thermal video and further, it might help to give an indication as to whether a person has covid or not.

II. ABOUT THE DATASET

The data set given to us contains recorded thermal videos of people. In those thermal videos, we can see numeric values of temperature predicted by the thermal gun. As these values of temperature vary over the thermal video, our task is to compute the average temperature of the person across the duration of the video.

Now, we have a very limited quantity of these videos in the data set which is not enough to train models on. Thus, we use the MNIST data set to train different models to classify digits. The limitation is that MNIST data set contains hand-drawn numbers while we have to predict on digitized numbers. But, with good training, we can see that it is a good bet to use this MNIST data set for training models. This fact can later be seen as the models perform well on the digitized images even when trained on the MNIST data set.

III. PRE-PROCESSING AND MAKING THE PIPELINE

In this part we will discuss how we created the pipeline to extract the average temperature from a thermal video.

1. So, the first part is basically to train all the desired models on the MNIST dataset. In this project we have trained 4 models namely Random Forest, Support Vector Machine, Multi-Layer Perceptron and Convolutional Neural Network. The parameters and architecture of each model is optimized and we evaluate the performance of the model before finalizing. The finalized models are saved for further use.
2. The next step is to create an opencv VideoCapture object. This helps to read the video from its address and helps to easily access each of its frames.
3. So basically, each of the thermal videos either contains a red bounding box or a green one. If there are green bounding boxes in the thermal video, we have to compute the average temperature on the green box frames. Else, we compute the average temperature on the red box frames.
4. So, we create 2 arrays to store the temperatures computed in cases where there are red boxes or green boxes.
5. Now, we loop through each frame in the video. The first task is to find the outer blue coloured bounding box in the thermal video. For this, we first convert the pixels to their HSV values. After that we check if the H,S and V values for each pixel are lying in the range for blue color or not. Like that, we compute the four corners of the blue bounding box.
6. Now, after computing the outer blue bounding box, we have to compute whether there is a red or a green bounding box inside the blue bounding box. The method followed is the same. Here, we just have to compare with the H,S and V ranges for green and red color respectively.
7. The next task is to use these found locations of green and red bounding boxes to crop just the portion of the image that contains the digits. We try to minimize the noise.
8. Now, the next task it to extract the images of individual digits from the above cropped image. For this, we follow the below mentioned steps:

A. The first step is to read the cropped image and resize it to a height of 500 pixels.
B. Next, we convert the image to grayscale.
C. The next step is to apply Gaussian Blur to the images.
D. Now, we find the contours in the transformed image using the opencv library. We select those contours that have a larger area than a threshold area. These contours represent our digits. Thus, we crop these contours by drawing bounding boxes and store these images containing single digits.

9. Now, the next task is to use our trained models to predict on these extracted digits. For this task also there are some steps.
- First, add a padding of 5 black pixels to the image in all directions.
- Resize the image to size of 28 * 28 pixels
- Use the trained models to predict on the images of individual digits.
- Construct the temperature predicted from the predictions made by the model on individual digits.

- Now, append the predicted temperature on to the array storing the temperatures of the red boxes or green boxes respectively.
10. All these steps are repeated on all frames in the video.
11. Finally, if green boxes are present in the image, we return the average value of the temperatures on the green boxes. Else, we return the average temperature on the red boxes.

So, this is the pipeline and pre-processing steps that we follow right from reading a video to returning the computed average temperature of the person from the video.

IV. Models Used

We have trained a total of 4 models on the MNIST dataset. These include:

A. Support Vector Machine

This model uses an algorithm that tries to find the best hyperplane that divides the classes. It only works for binary classification problems. Thus, we have to use approaches like One Vs One Classification (OVO) and One Vs All Classification (OVA) for the multiclass classification problems. Again, we have tried to tune its parameters using the Grid Search CV technique on the validation data.

B. Random Forest

This classifier uses the ensembling technique of Bagging. We have implemented it using the sklearn library. We have also tuned its hyperparameters using the Grid Search CV method by using it on the validation data. Some of the parameters include the number of estimators. These are basically the number of weak Decision Tree Classifiers that Random Forest model trains and makes the average of predictions on. We optimized the results by using its value as 1000.

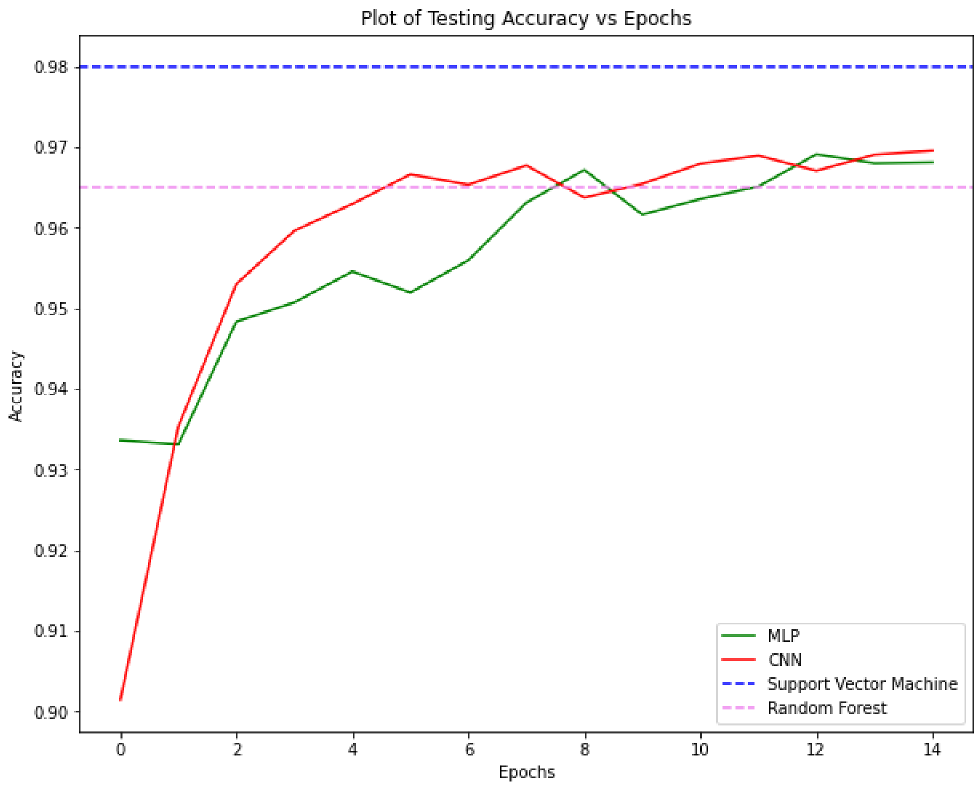
C. Convolutional Neural Network

It is a deep learning algorithm best suited for image classification tasks. It works by processing the image pixels by multiplying them with suitable matrices and using the information learned from these pixels to classify the image. We have optimized its architecture by experimenting different combinations. Its implementation was done using Tensorflow.

D. Multi-Layer Perceptron

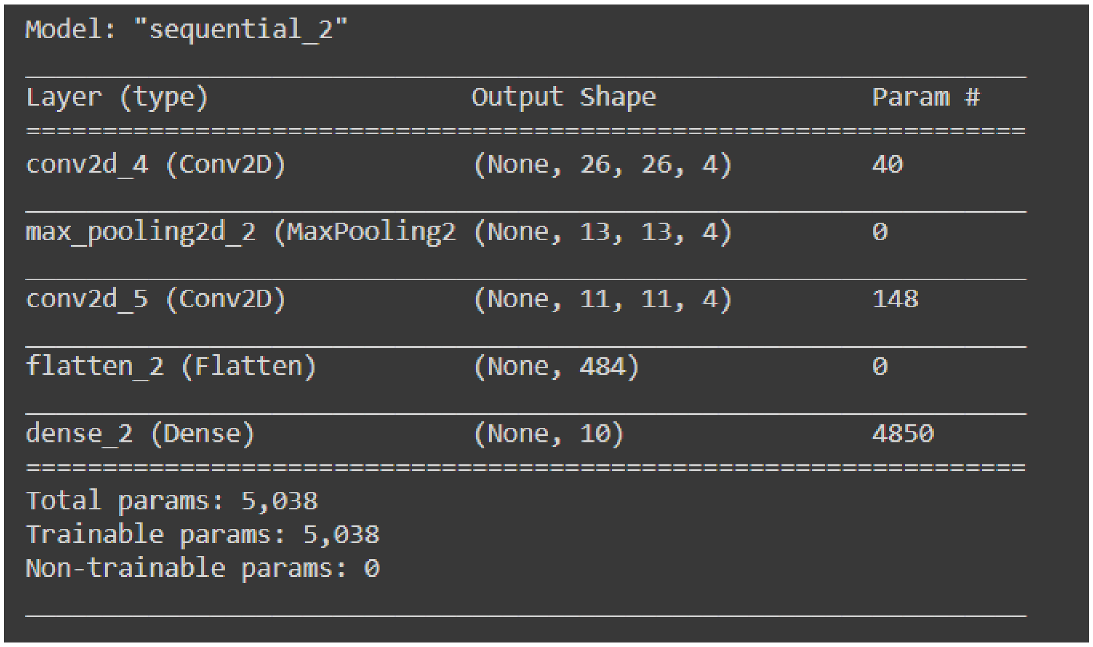
It is a model that combines different neurons together to form a model. We have implemented this model using the Tensorflow library. Again, in this model also, we have tried to tune its hyperparameters to get optimized results. The technique for optimization was similar to one used in the Convolutional Neural Network.

Comparing the performance of these Models



V. BEST MODEL

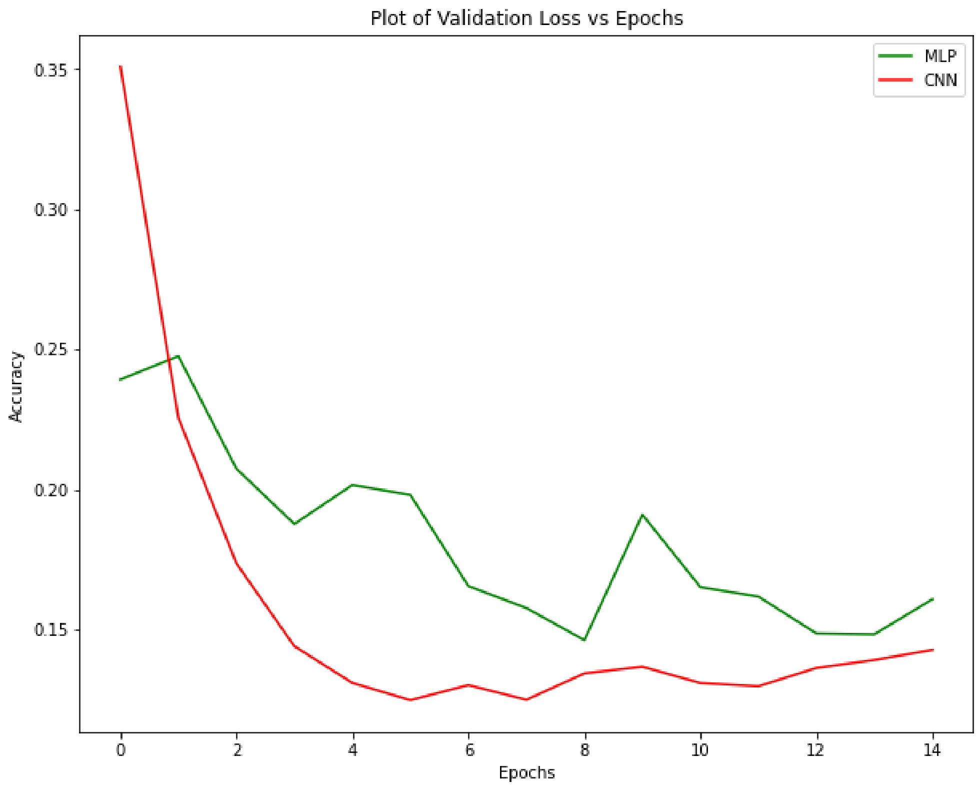
The best performing model was the CNN model with the following architecture:



```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 26, 26, 4)         40
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 13, 13, 4)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 11, 11, 4)         148
_________________________________________________________________
flatten_2 (Flatten)          (None, 484)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                4850
=================================================================
Total params: 5,038
Trainable params: 5,038
Non-trainable params: 0
_________________________________________________________________
```

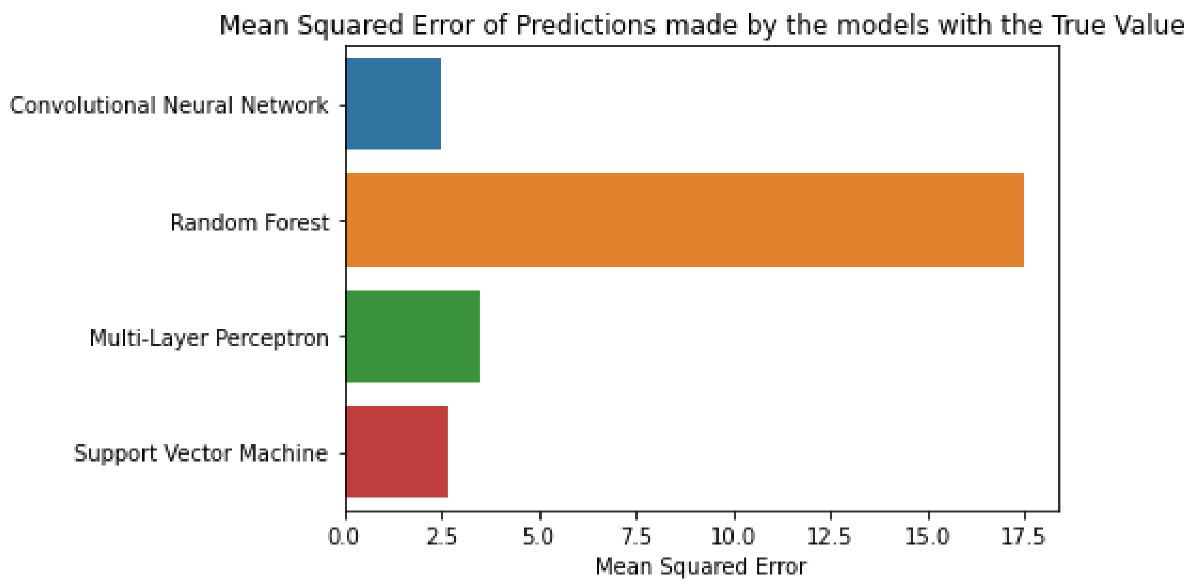
We trained the data on a variety of CNN models and finally concluded that this model was performing best among all the tried models.

VI. RESULTS AND CONCLUSIONS



As we can observe from plots 1 and 2, we can see that the Convolutional Neural Network Model and the Support Vector Machine model perform best on the MNIST model. Both show high testing and training accuracies.
They are followed in performance by the Multi-Layer Perceptron Model which also performs only slightly poorer than these models. Finally, comes the performance of the Random Forest Models which performs significantly bad than the other models.
From the second plot, we can see that the loss of the Convolutional Neural Network model decreases rather smoothly as compared to the Multi-Layer perceptron which forms a rather noisy plot. This fact is also reflected in their performances while predicting on the digits in the thermal videos.

We now compute the predictions of average temperatures computed by all the models on the thermal videos. Then, we compare the predictions with their true value obtained from the data sheet.
The evaluation metric used for the performance is the mean squared error. We compute the mean squared error for all the 4 models. The plot for the same is given as follows:



We can see that our hypothesis of the models that we created by seeing the performance of the models on MNIST data is indeed confirmed. We can see that the CNN model, which is considered one of the best models for image tasks indeed performs the best.
It is followed in performance by the Support Vector Machine model which also performs quite comparable to the CNN model.
As, we could see from the loss function of the Multi-Layer perceptron model as how noisy it was, we confirm the fact that it performs slighly poorer than the top 2 models.
Finally, as expected from the performance on the MNIST data, the Random Forest Model performs the worst among all the models.

VII. Contributions

The contributions made by all the members are all follows:

A. Research about the project, Existing solutions and Report Making

This part was undertaken by both the group members. Suneel looked out for existing solutions while Tanishq went through the research papers and tried to understand them. The report was contributed by both the team members.

B. Data Preprocessing

This part was undertaken by Tanishq.

C. Model Creation

1) *Random Forest:* This part was undertaken by Suneel and both took part in tuning the hyperparameters.
2) *Multi Layer Perceptron:* This part was undertaken by Suneel and both took part in tuning the hyperparameters.
3) *Support Vector Machine:* This part was undertaken by Tanishq and both took part in tuning the hyperparameters.
4) *Convolutional Neural Network:* This part was undertaken by Tanishq and both took part in tuning the hyperparameters.

VIII. References

- https://stackoverflow.com/questions/58663227/extracting-digits-from-image-with-python-and-opencv
- https://www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/