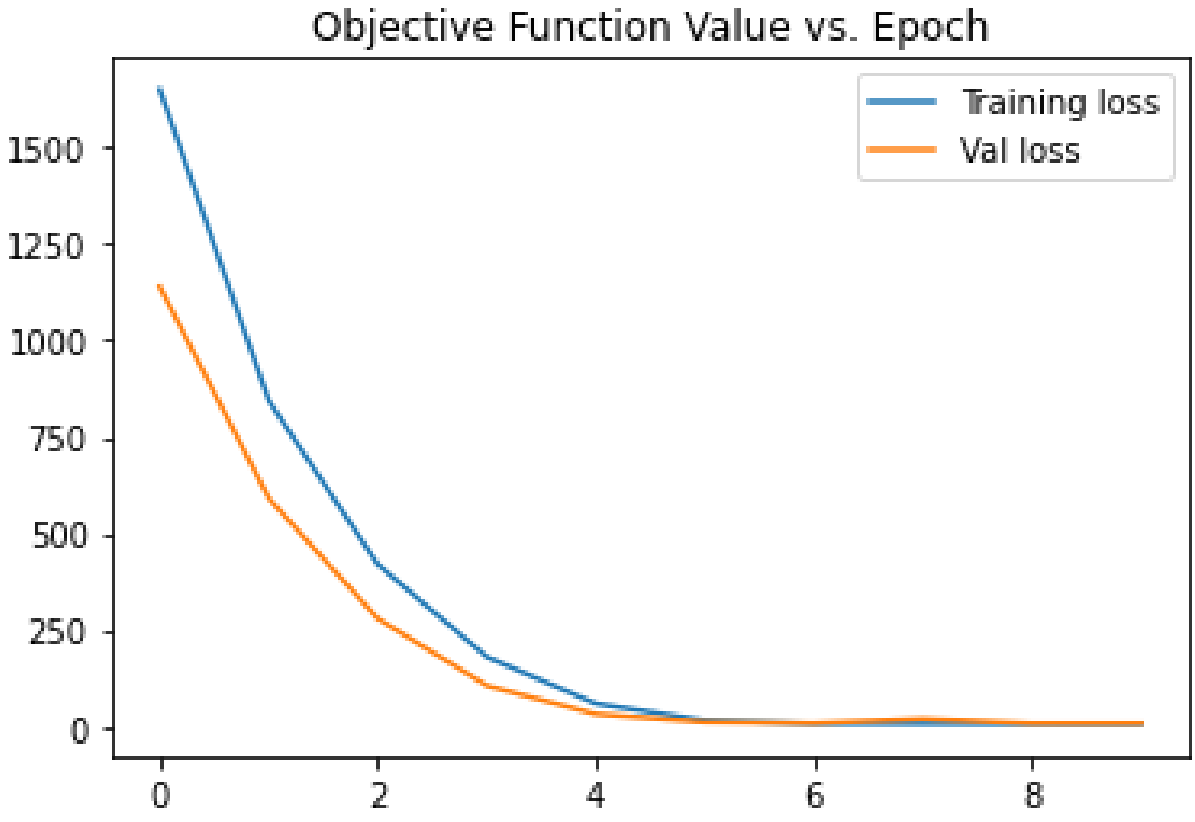
Facial Keypoint Detection Challenge Report

This report is based on the Kaggle Dataset: Facial Keypoints Detection

GitHub Code: https://github.com/AbhishekR3/FacialKeypoint

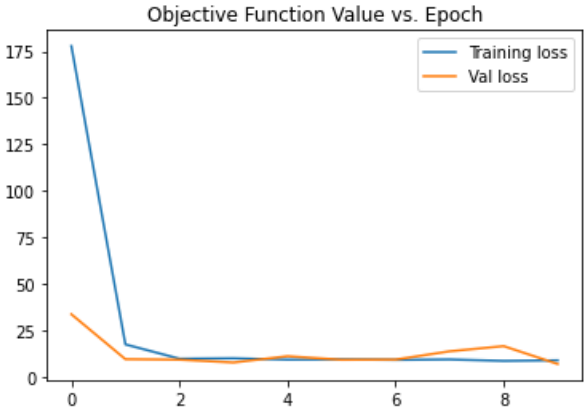
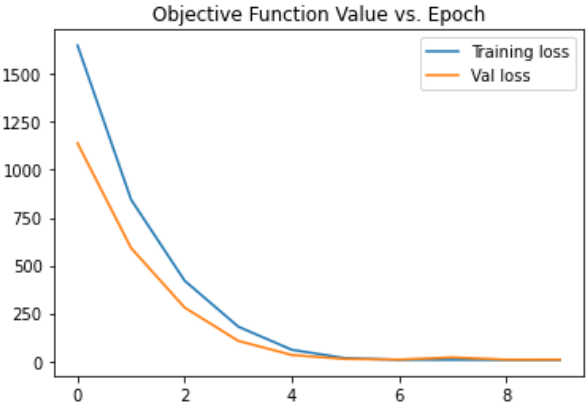
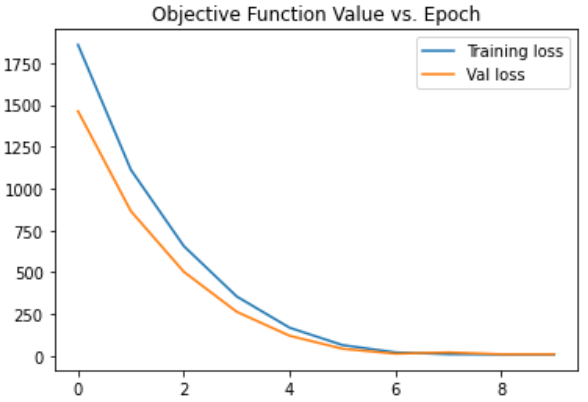
I used this as the base model, then changed specific variables to see how the results change.

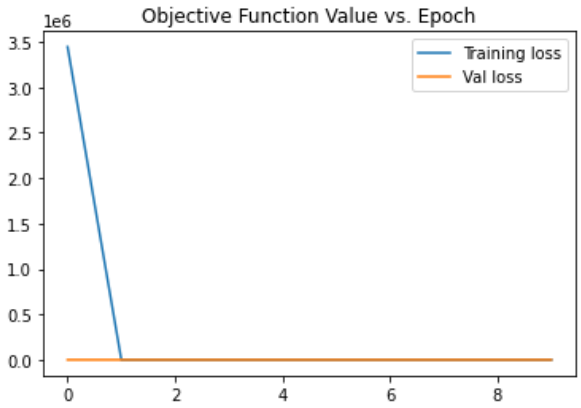
Pretrained Model: resnet18
Epochs: 10
Learning Rate: 0.0001
Optimization: RMSprop



Architectures

I used 4 different types of architectures: ResNet, GoogLeNet, VGG, and my own Convolutional Neural Network.

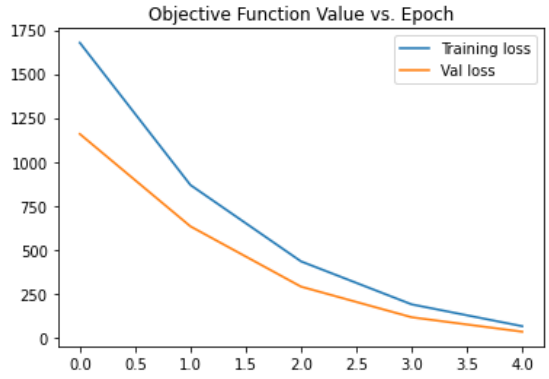
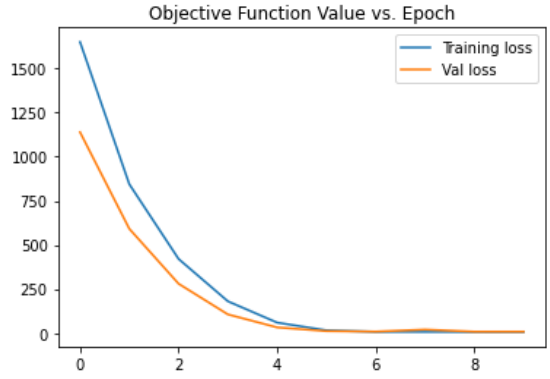
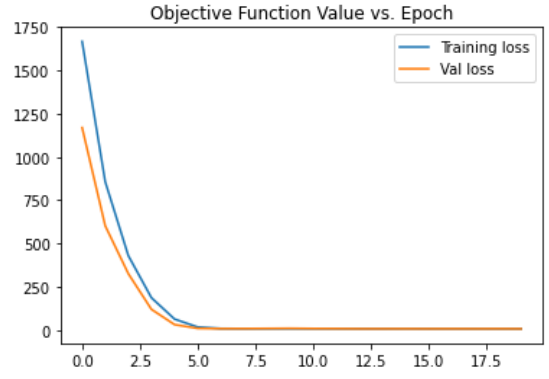
Model	Objective Function Plot
My Own CNN Model	 Validation Loss - 6.7418
ResNet18	 Validation Loss - 8.2262
GoogLeNet	 Validation Loss - 10.0786

VGG16	 Validation Loss - 43.4596

Based on this table and the ranking, it is assumed that my CNN model would be the best, but on further inspection of the graph, it can be seen that the validation loss was constantly fluctuating. The model kept fluctuating around 10, with 8,9,10th epochs being 13.5748, 16.3586, 6.7418 respectively. I am making an assumption that my CNN model is not accurate as shown, but ResNet18 had the best since it was constantly decreasing.

Epochs

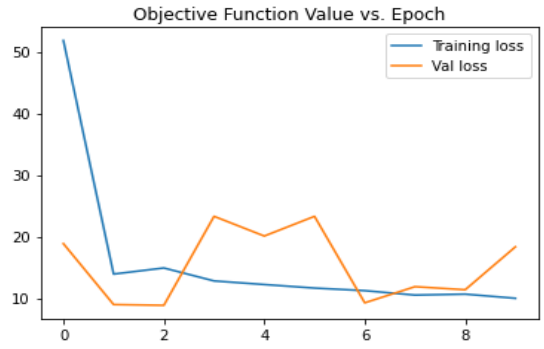
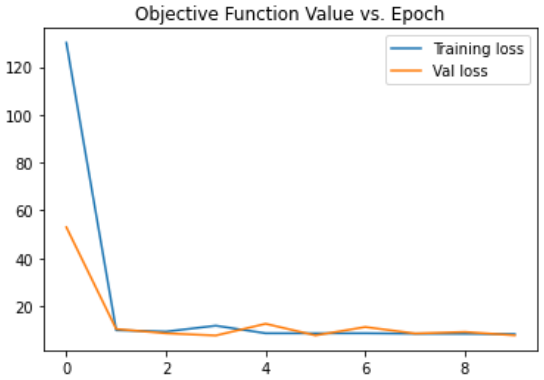
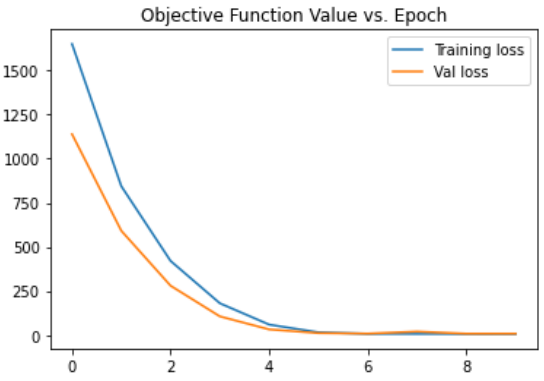
This was my reading difference between 5, 10, and 20 epochs of the base model.

Epochs	Objective Function Plot
5	 Validation Loss - 32.7423
10	 Validation Loss - 8.2262
20	 Validation Loss - 6.8144

Based on the graphs, 20 epochs had the lowest objective function. It seems after 7 epochs, the decrease in loss is minimal. 5 epochs still had a lot more training to do.

Learning Rate

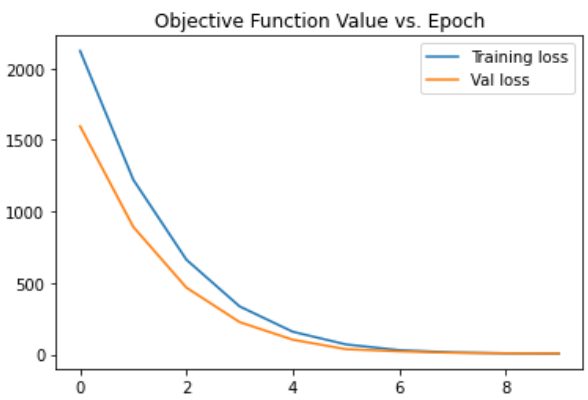
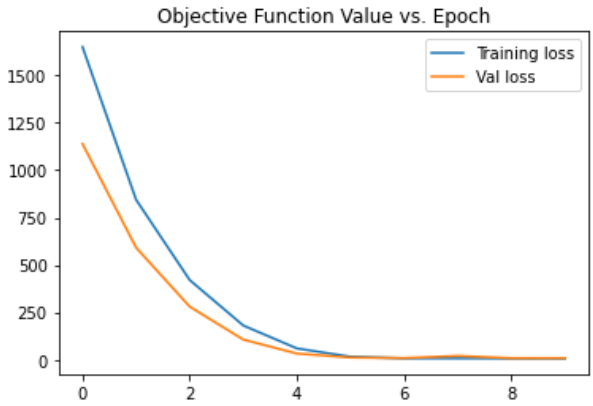
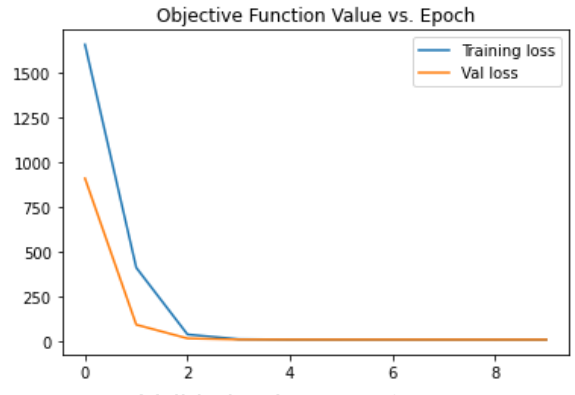
I used 0.001 as the learning rate. I also tried 0.01 and 0.0001 learning rate with the base model.

Learning Rate	Objective Function Plot
0.01	 Validation Loss - 18.3275
0.001	 Validation Loss - 7.7311
0.0001	 Validation Loss - 8.2262

0.001 was the best learning rate to predict for this model.
I believe there was overfitting in the 0.01 learning rate model, since the validation set kept fluctuating and was generally increasing.

Optimization Algorithms

I used 3 optimization algorithms: Adam, RMSprop, and SGD with the following results

Optimization Algorithm	Objective Function Plot
Adam	 Validation Loss - 7.9125
RMSprop	 Validation Loss - 8.2262
SGD	 Validation Loss - 7.3477

Overall the optimization algorithms performed similarly. SGD ended with the lowest objective function.

Data Augmentation

To further improve the model prediction, I used data augmentations using the Albumentations library. These are the following data augmentations I did:

- A.Rotate(limit = 25, p = 0.5) → 50% probability that the image will be rotated by any value between -25 and 25 degrees
- A.GaussNoise() → Applies gaussian noise to the image
- A.RandomBrightness() → Randomly changes brightness of image

I actually found out that Data Augmentation in this instance increased the objective value, so it made the prediction worse.

Kaggle

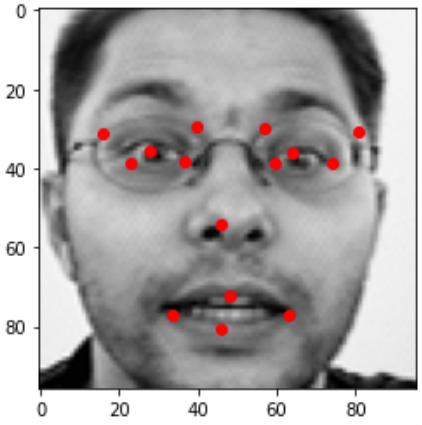
At the end of seeing the objective function on different types of models, I decided to submit this as my best score.

Pretrained Model: ResNet18
Epochs: 20
Learning Rate: 0.001
Optimization: SGD

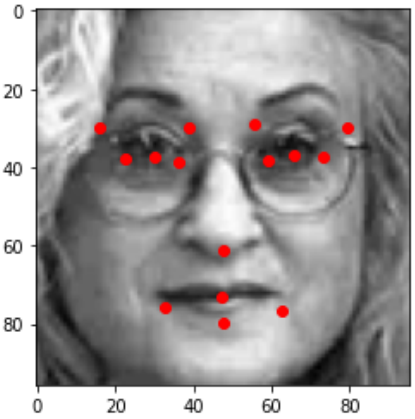
With these values set, I got a score of 28.93591

Prediction

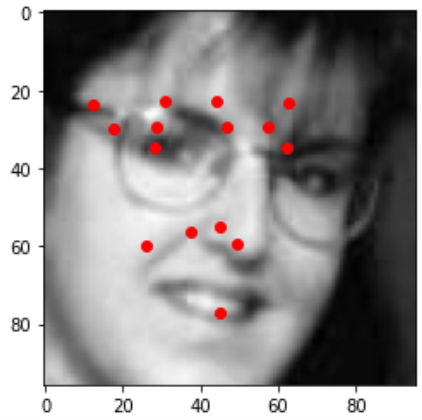
I used my best model to predict the face key points of various image and visualize them



I would say this is very close to most of the locations required. The face is facing straight and glasses didn't cover any eye parts thus, it was able to identify close to the actual location.



Another relatively close prediction. Again I believe the glasses and darker shade on the eyes threw the model off.



This isn't a good prediction. The face isn't straight and the person is wearing glasses, which is a reason it could have thrown the prediction off.