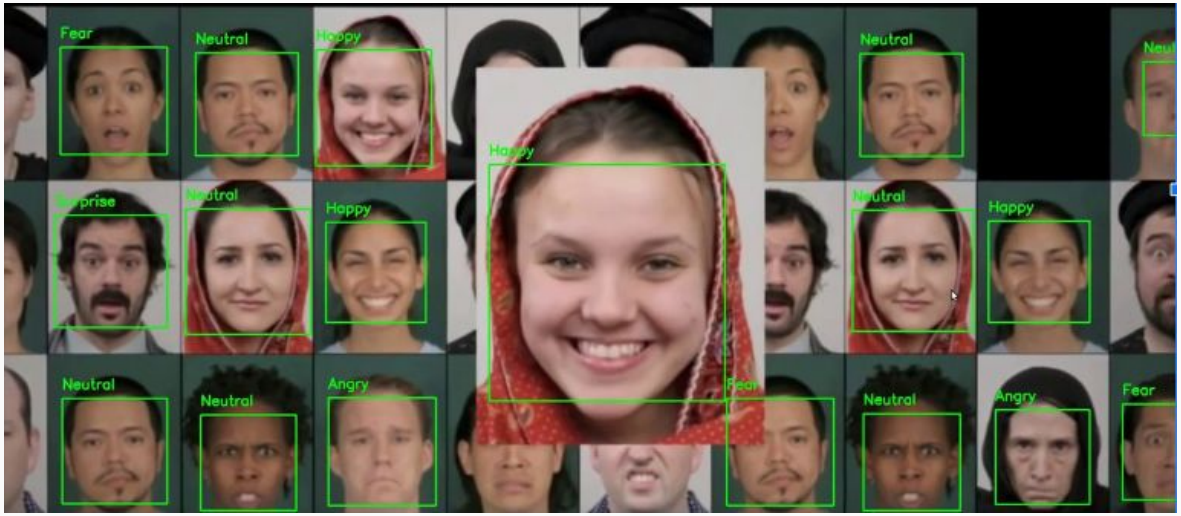
Facial Expression Recognition

By Rita Samir and Sohayla Mohammed

Overview

Problem

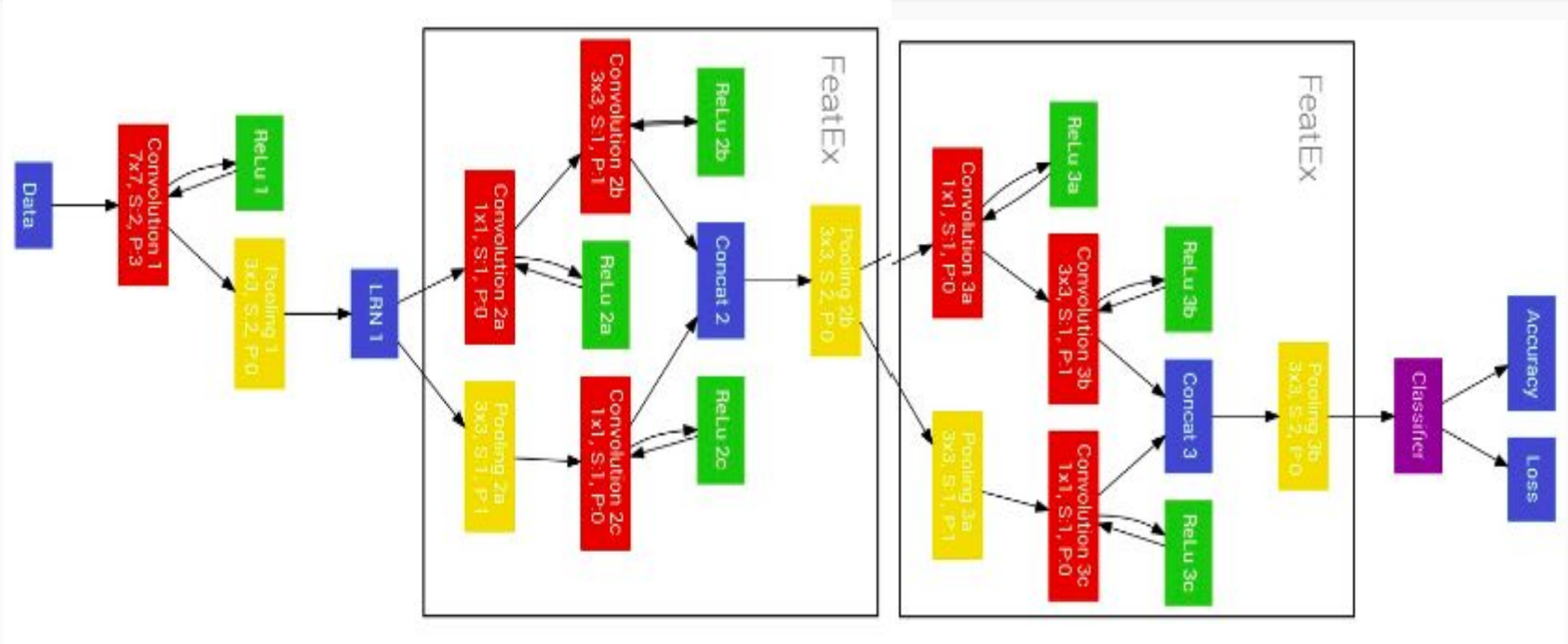
- **Facial expression recognition** software is a sentiment analysis tool and is able to automatically detect the six basic or universal expressions: happiness, sadness, anger, surprise, fear, and disgust.
- **Evaluation Metric :** Accuracy derived from confusion matrix



Datasets used

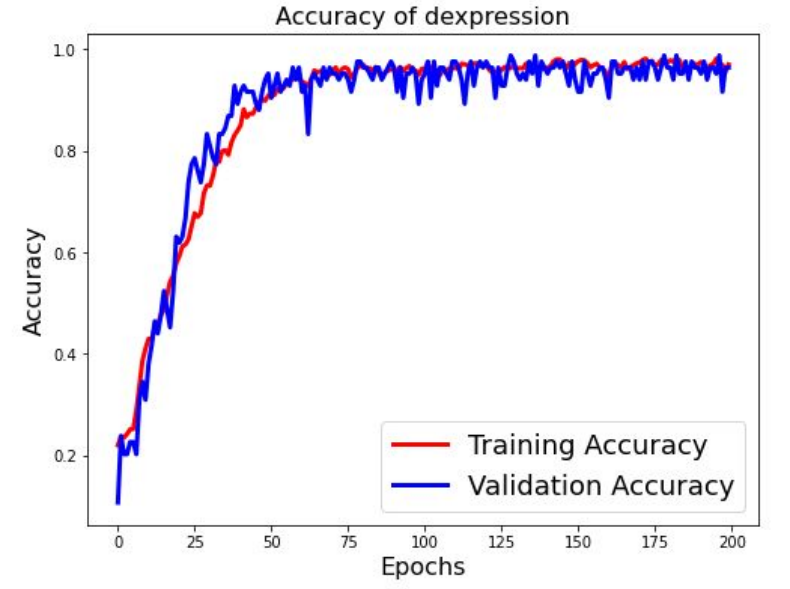
- Ck
  - 48x48 resolution.
  - Unique
  - Posed
  - 6 BEs & 1 Neutral.
- FER2013
  - 48x48 resolution.
  - Wild
  - Posed and spontaneous.
  - 6 BEs & 1 Neutral.

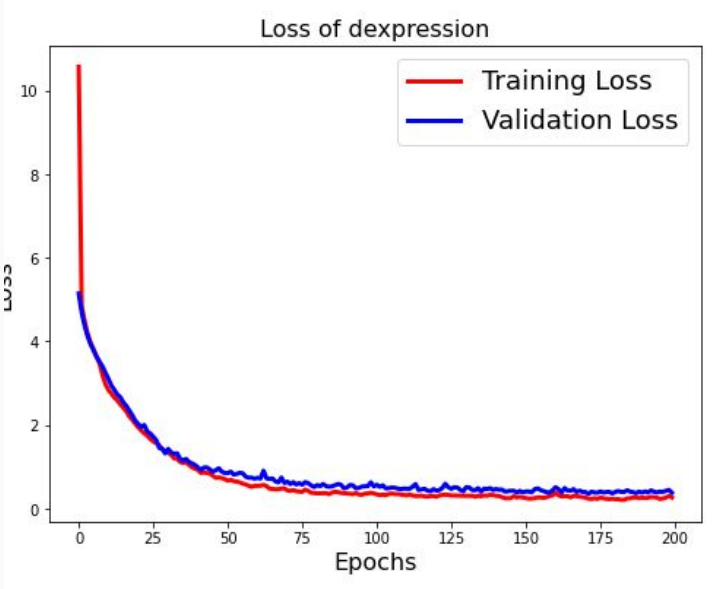
Dexpression model



Previous progress

- Reaching **97.29 %** accuracy with **proposed** model running over Ck dataset



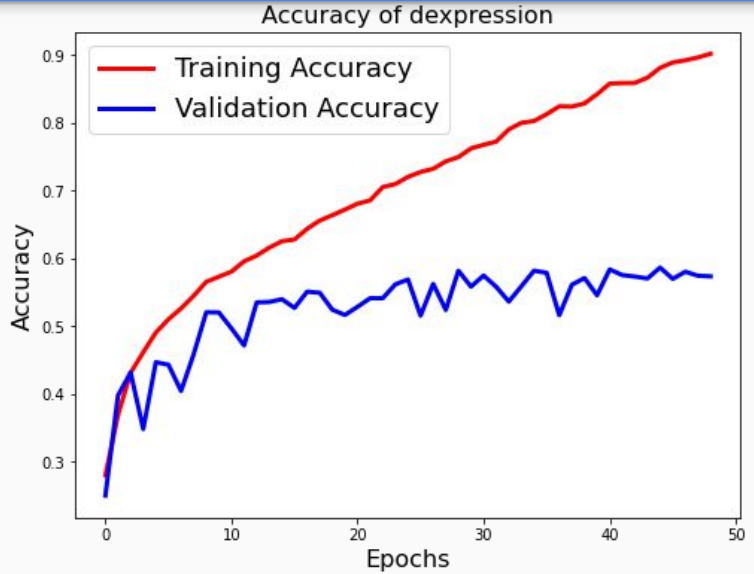


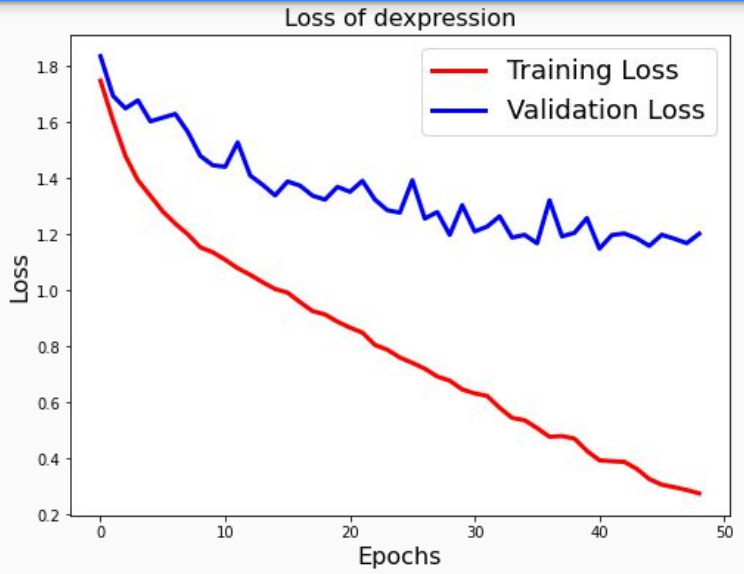
Current Progress

Running the FER2013 dataset on the proposed model

- Running the FER2013 dataset on the proposed model without any regularization.
- Running the FER2013 dataset on the proposed model with some regularization.
- Running the FER2013 dataset on the proposed model with regularization and data augmentation

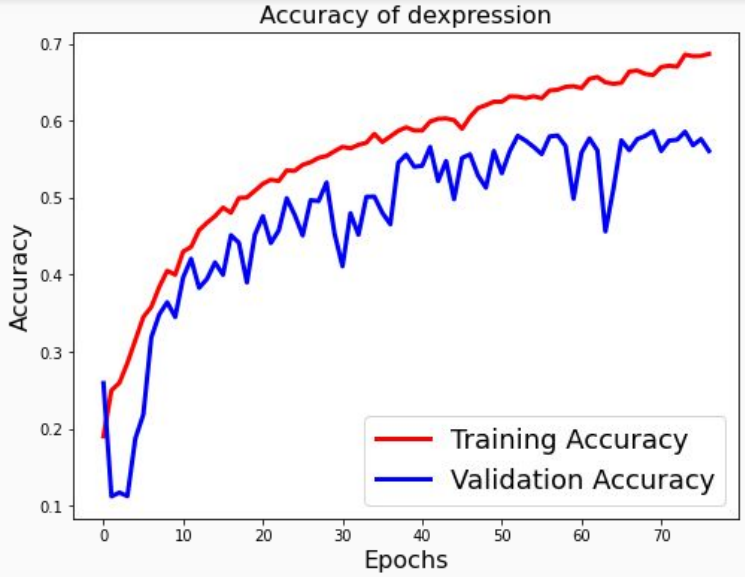
Running the FER2013 dataset on the proposed model without any regularization.

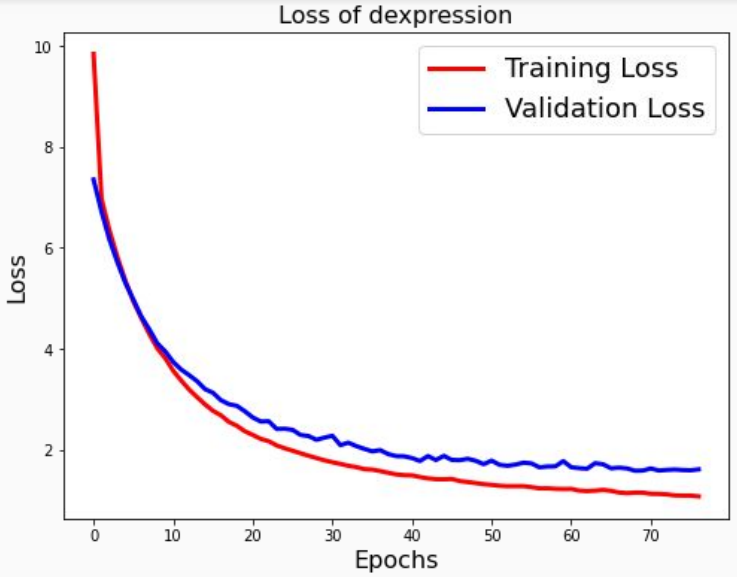





```
Testing Loss = 1.1369704008102417
Testing Accuracy = 0.5600445866584778
```

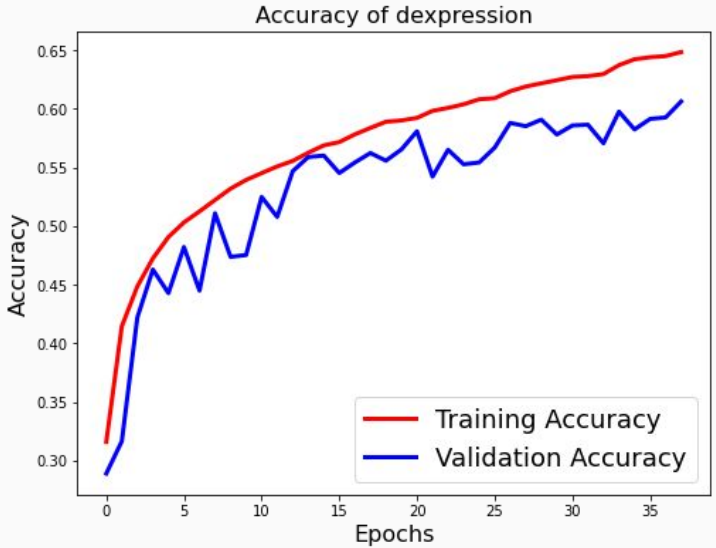
Running the FER2013 dataset on the proposed model with some regularization.

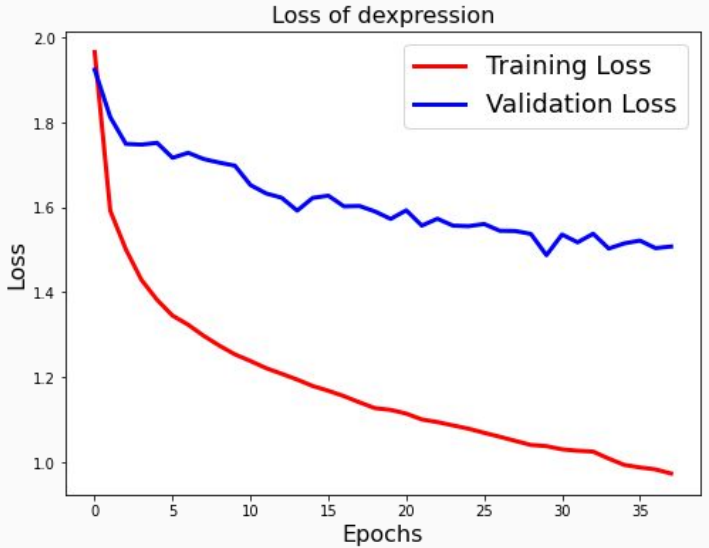





```
Testing Loss = 1.633882999420166
Testing Accuracy = 0.5553078651428223
```

Running the FER2013 dataset on the proposed model with regularization and data augmentation






```
Testing Loss = 1.1221622467041102
Testing Accuracy = 0.6219002604484558
```

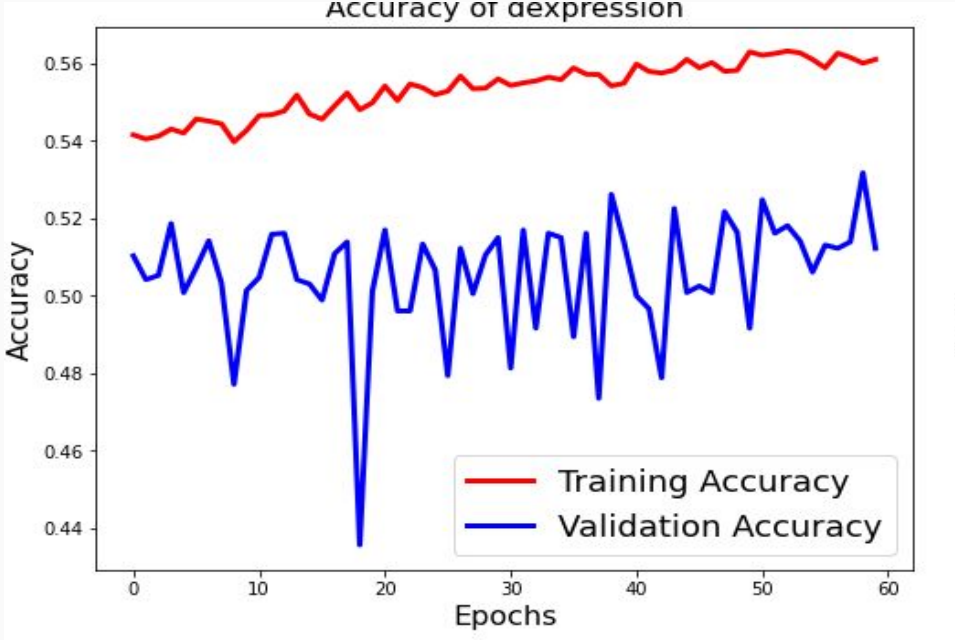
Ensemble CNN

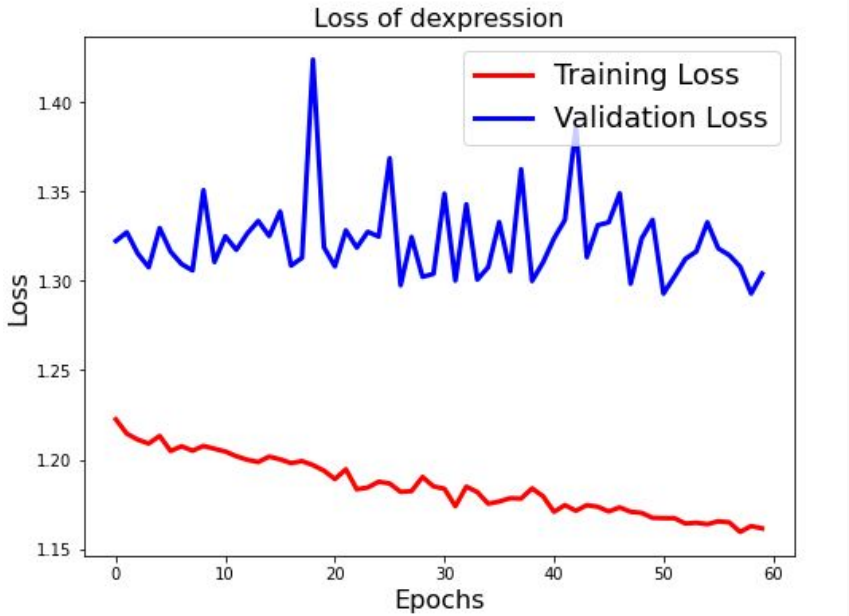
- Trying out different models to construct the ensemble.
- Tune each model individually to achieve best accuracy for it.
- Compare between different means to evaluate the accuracy of the ensemble as a whole.

Constructing the ensemble

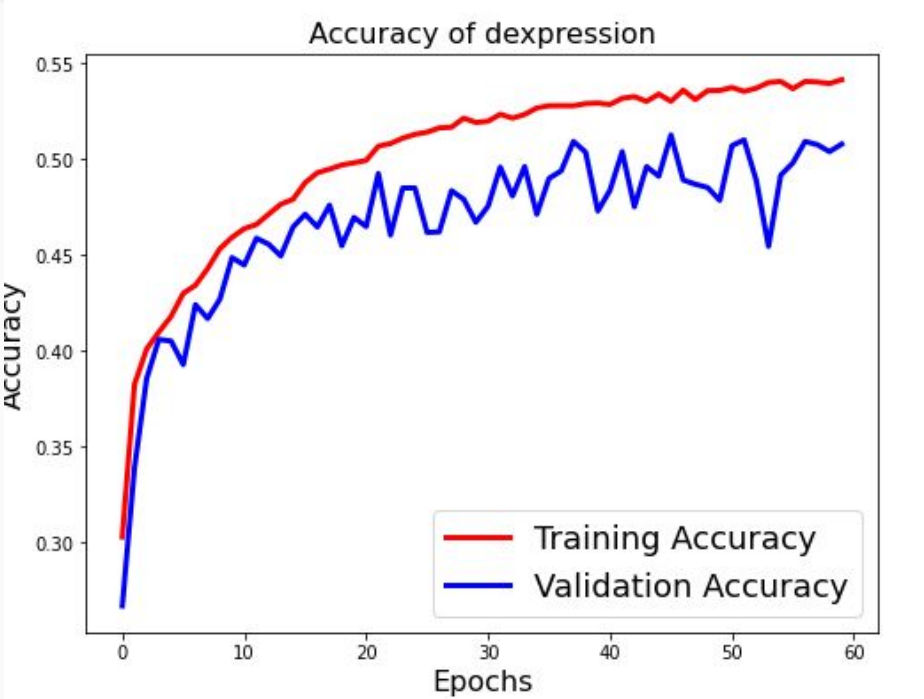
- Tried different models that perform somewhat well with the FER2013
- Models used in this milestone :
  - Original (Dexpression) CNN model.
  - Simplified version of the original model with only one feature extraction block.
  - 2 basic CNN models (basic feature extractions and classification blocks)
  - A ResNet model :
    - Used built in ResNet50 from keras, with top layer to preprocess gray scale images.

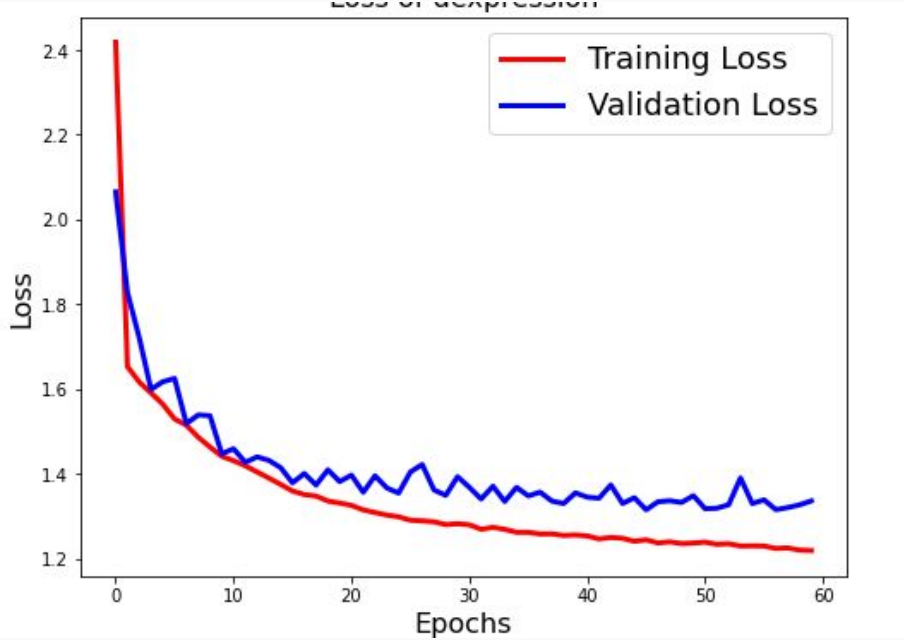
Original Dexpression



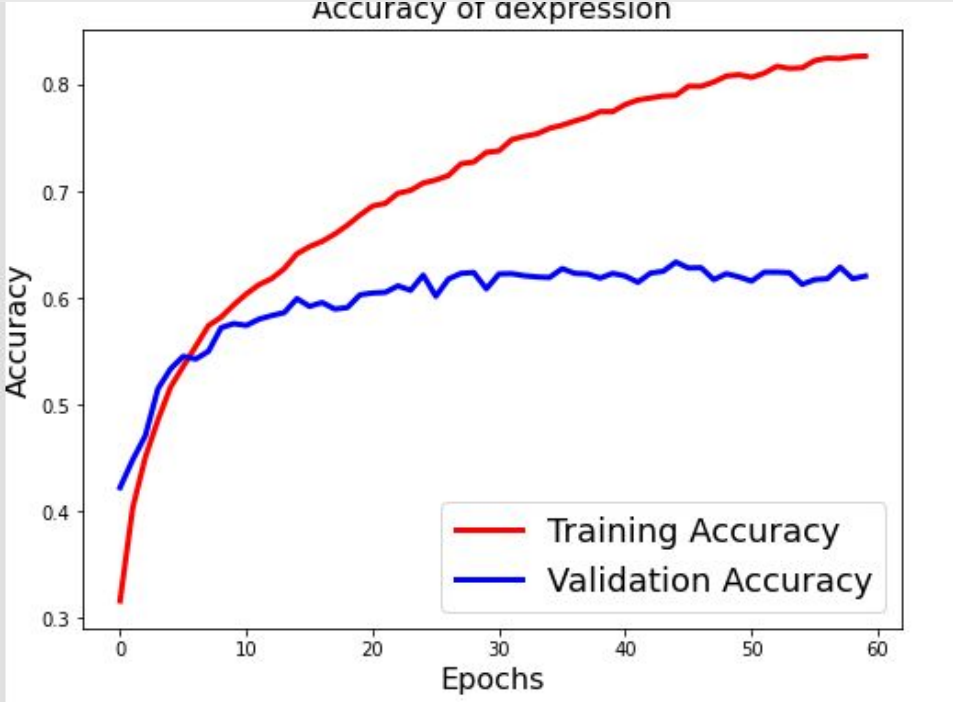


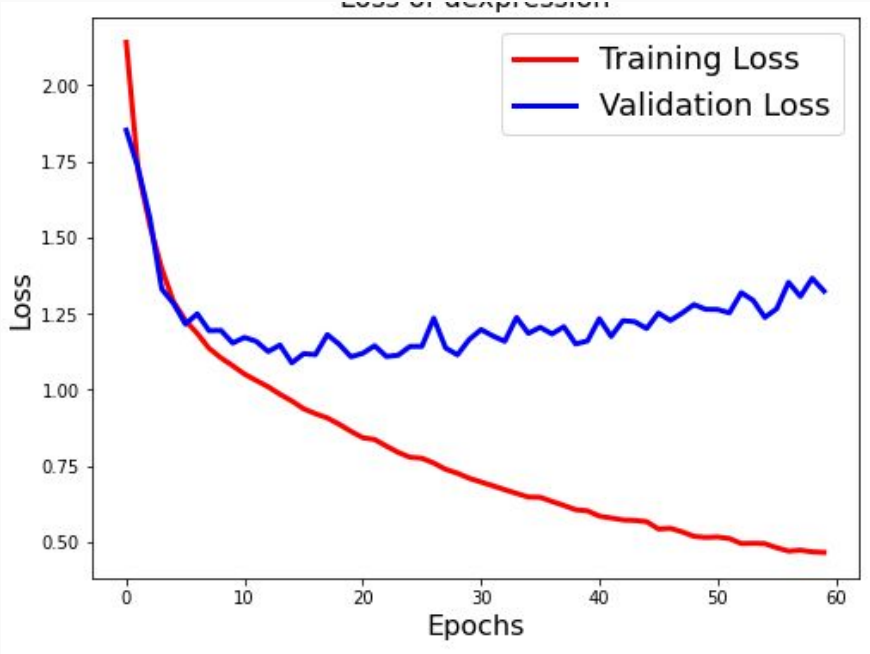
Simplified Dexpression



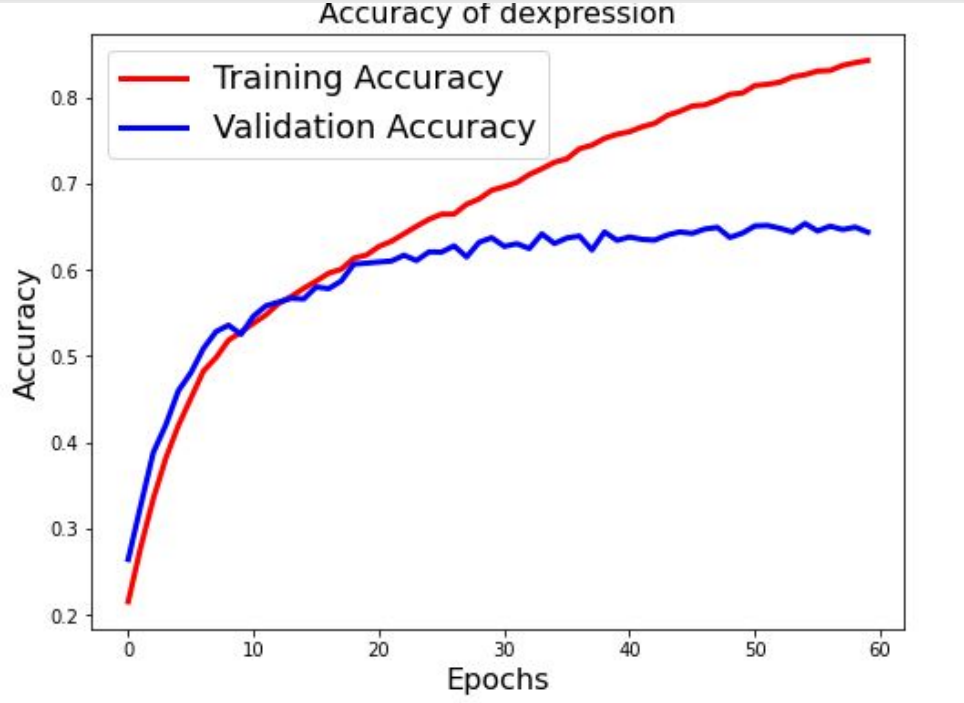


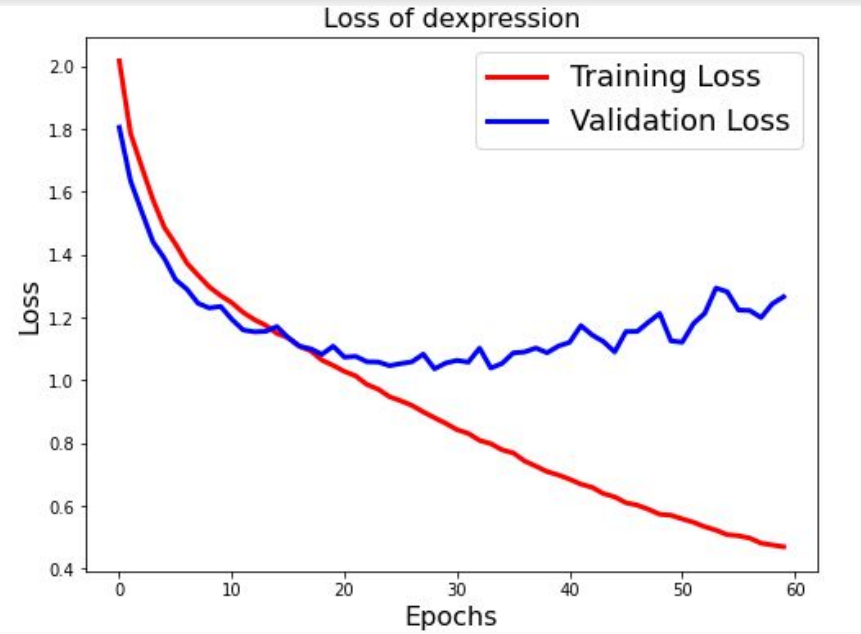
Basic CNN_1


Accuracy of dexpression
Accuracy
Epochs
Training Accuracy
Validation Accuracy


Loss
Epochs
Training Loss
Validation Loss

Basic CNN_2

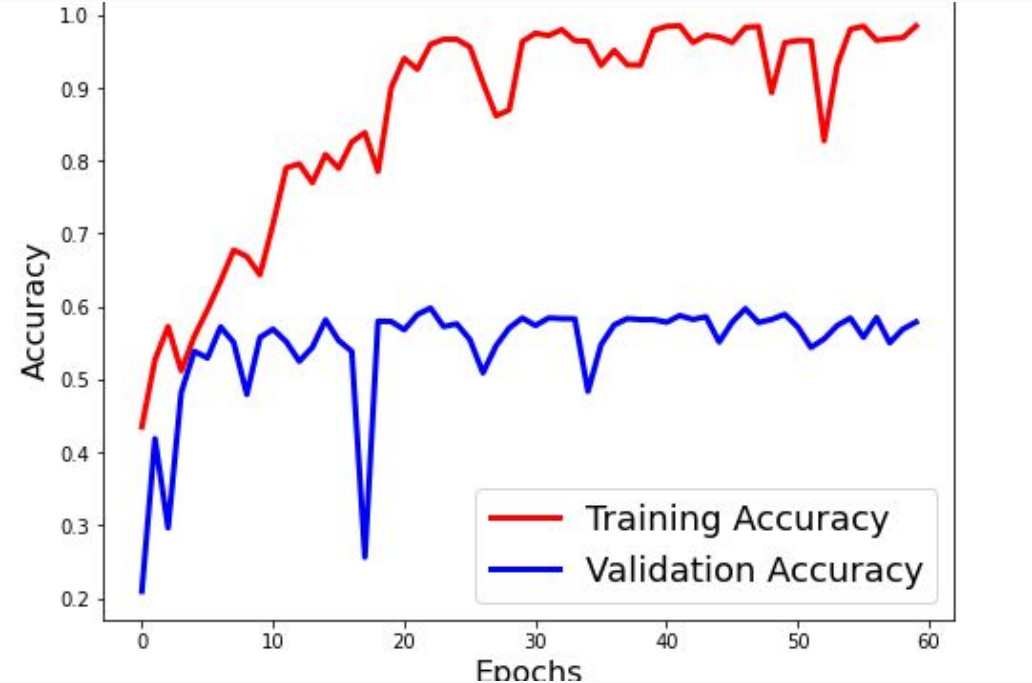

Accuracy of dexpression
Training Accuracy
Validation Accuracy
Accuracy
Epochs
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
10
20
30
40
50
60

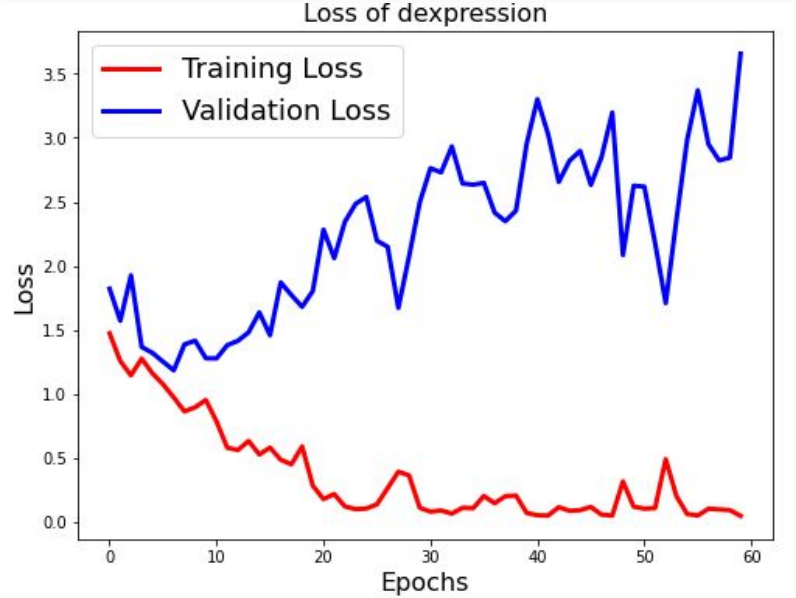

Loss of dexpression
Training Loss
Validation Loss
Loss
Epochs
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
0
10
20
30
40
50
60

ResNet Model

	Training Accuracy	Validation Accuracy	Testing Acc
Resnet50 (no freezing)	98.35	56.26	56.26
Freezing 7 Layers	0.9861	0.5946	0.56
Freezing 3 layers	0.8573	0.5116	0.52
Freezing 3 layers(lr = 0.001)	0.7997	0.5634	0.54
Freezing 20 layers	0.25	0.216	0.22

ResNet Model





ResNet Model

- Huge overfitting still occurs between the training accuracy and the testing set accuracy.
- Next steps would be trying to implement a ResNet model from scratch and not use the built in one.

Ensemble Evaluation

- Used three main techniques to evaluate the ensemble as a whole :
  - Majority voting.
  - Average (Sum).
  - Weighted average
    - Validation accuracy as weight.
    - Individual testing accuracy as weight.

Results

	Accuracy
Majority Voting	63.77%
Average	66.6%
Weighted Average (val acc)	67.12%
Weighted Average (test acc)	**67.26%**

Next steps

Next steps

- Applying preprocessing on the dataset (e.g Data Augmentation).
- Implement ResNet Model from scratch for the ensemble model.
- Tuning individual models to get highest accuracy achievable.

Thank you ^ ^.