DL Assignment 4 Report

Ayush Agarwal MT22095
Janak Kapuriya MT22032
Shubham Agarwal MT22124

Question 1

We have used the VGG16 model as a feature extractor. After that we have added 1 fully connected layer before the output layer.

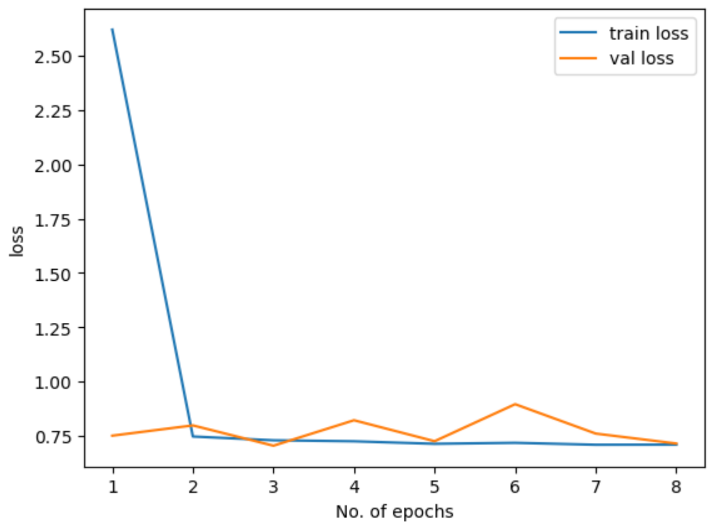
Hyperparameters

Loss = CrossEntropyLoss
num_epochs = 10
learning rate = 0.0005
optimizer = Adam

Assumption

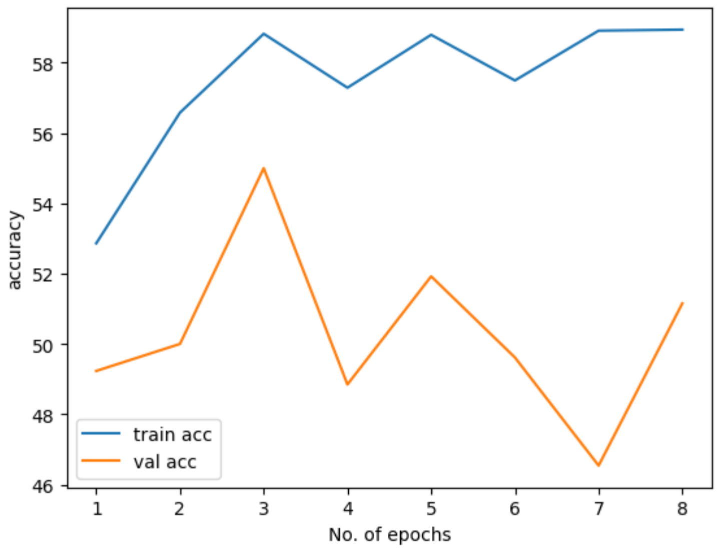
I have taken random stratified samples of 70 % from the train_jsonl file because of computation constraints.

Loss Plots



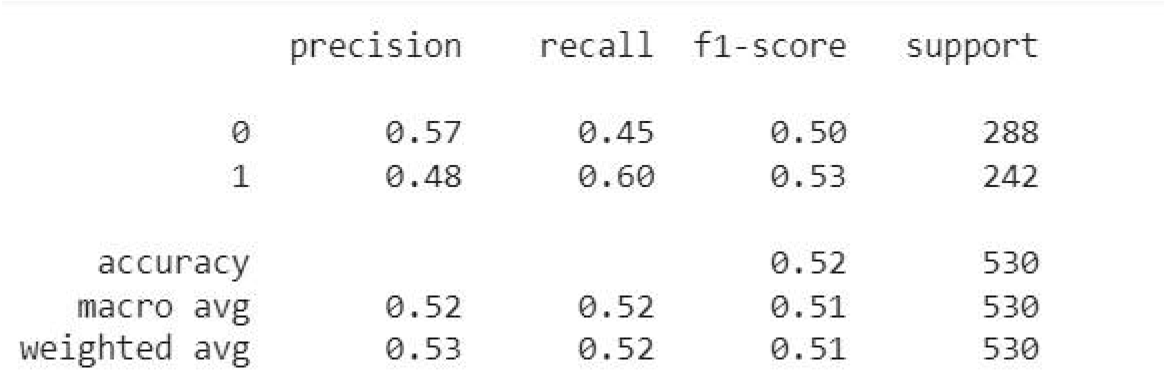
As we can see from the graph that train loss is decreasing as no. of epochs increased but validation loss is not decreasing and remained almost flat and not decreased further.

Accuracy Plot



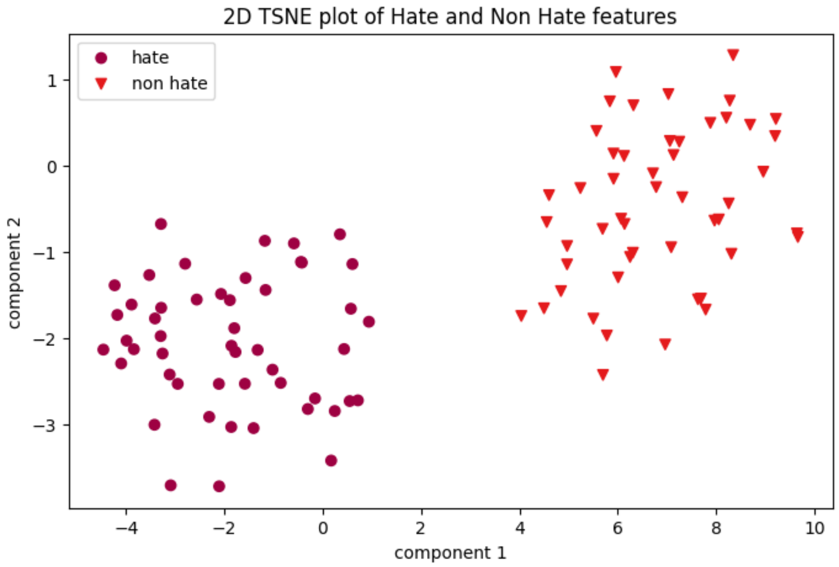
As we can see that accuracy of training data is increasing while validation accuracy fluctuates. Model giving less validation accuracy because model tries to predict harmful memes or not based on only image component of meme which is not sufficient for meme classification as to classify meme we need both text and image feature for classification.

Classification Report



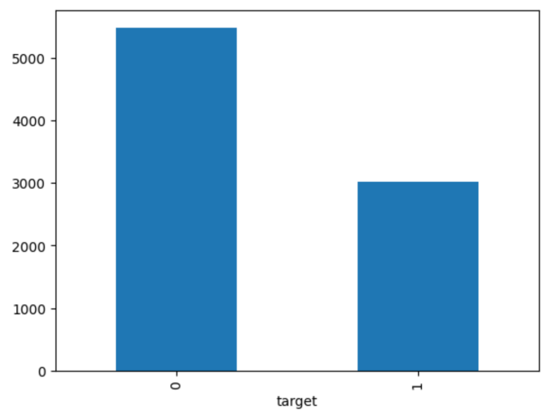
```
              precision    recall  f1-score   support

           0       0.57      0.45      0.50       288
           1       0.48      0.60      0.53       242

    accuracy                           0.52       530
   macro avg       0.52      0.52      0.51       530
weighted avg       0.53      0.52      0.51       530
```

TSNE Plot (Task 1)

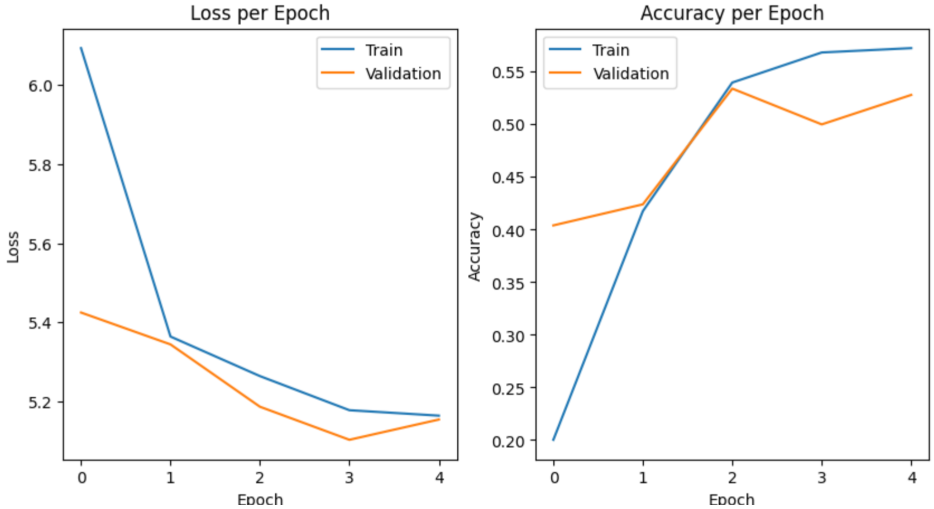


As we can see from the TSNE plot that features extracted by VGG16 in low dimensional space are clustered according to their class labels. Because the features value for Meme to be Hateful is different from non Hateful Meme. So we can say that the model has learned something from input images.

Question 2
Data visualization



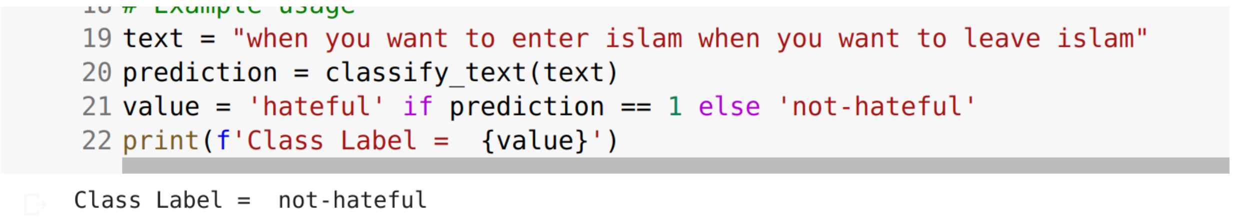
Loss and accuracy plots per Epoch



As the plots suggest, as the models learn, accuracy increases and losses decrease.

```
              precision    recall  f1-score   support

           0       0.73      0.52      0.61       720
           1       0.29      0.51      0.37       280

    accuracy                           0.52      1000
   macro avg       0.51      0.51      0.49      1000
weighted avg       0.61      0.52      0.54      1000
```

Inference on a single sample

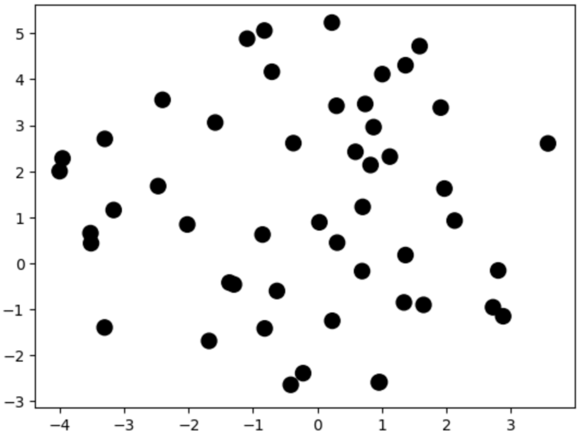


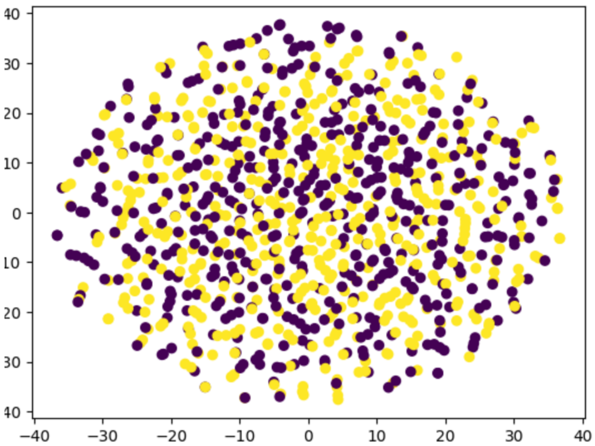
```
# Example usage
text = "when you want to enter islam when you want to leave islam"
prediction = classify_text(text)
value = 'hateful' if prediction == 1 else 'not-hateful'
print(f'Class Label =  {value}')
```

```
Class Label =  not-hateful
```

Due to imbalanced dataset, model is not learning very well and and is slightly biased towards predicting the frequent class ‘non-hateful’. Hence, we have bad F1 scores for non-hateful class over hateful class.

The tsne plots over 50 samples and also over the entire test dataset is shown. This clearly explains the limitation of unimodal models which are unreliable to extract features and map data to different classes. Hence, there is this need of unimodality.

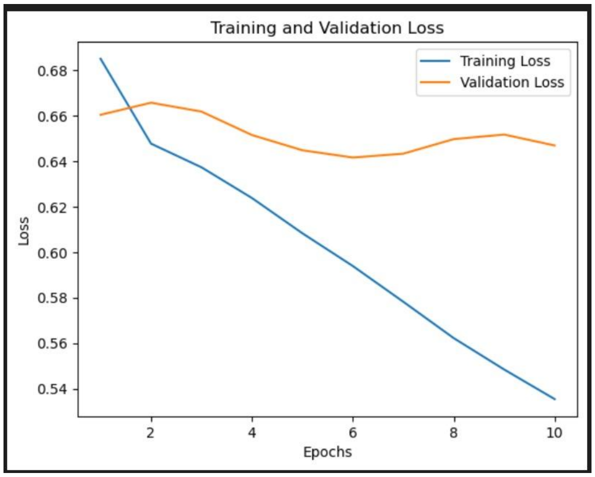


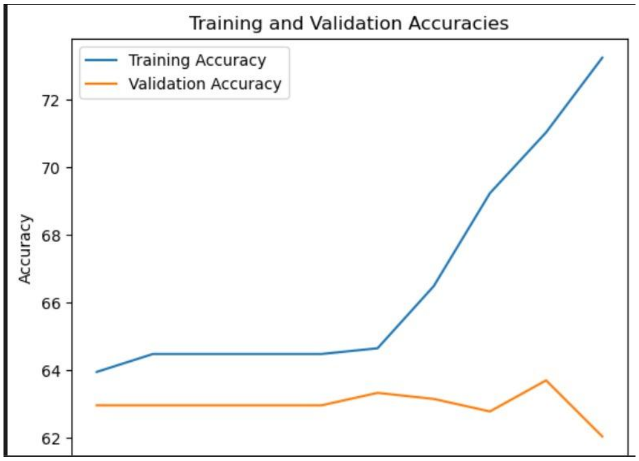


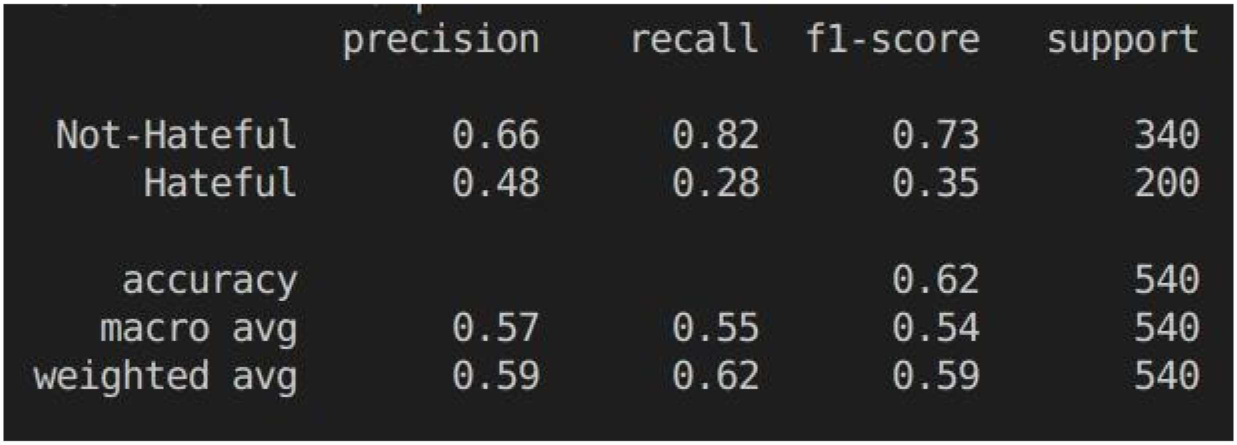
Question 3:

Loss and accuracy plots per epoch

As we can see from the graphs below, as the models learn, the accuracies are improving and losses are decreasing on every epoch.

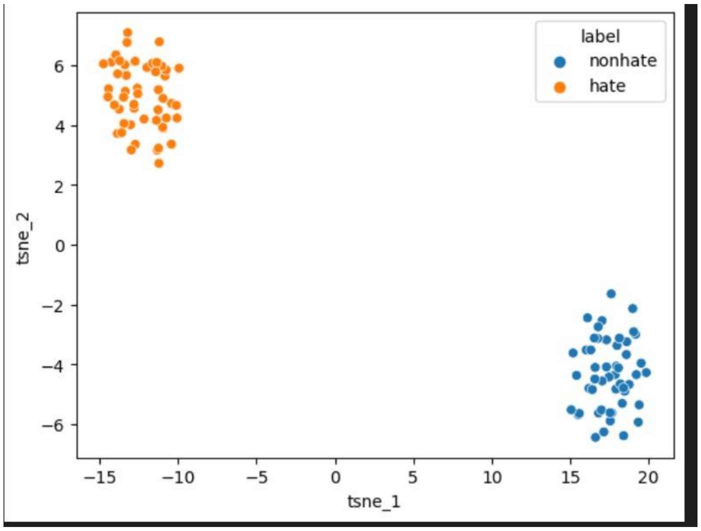






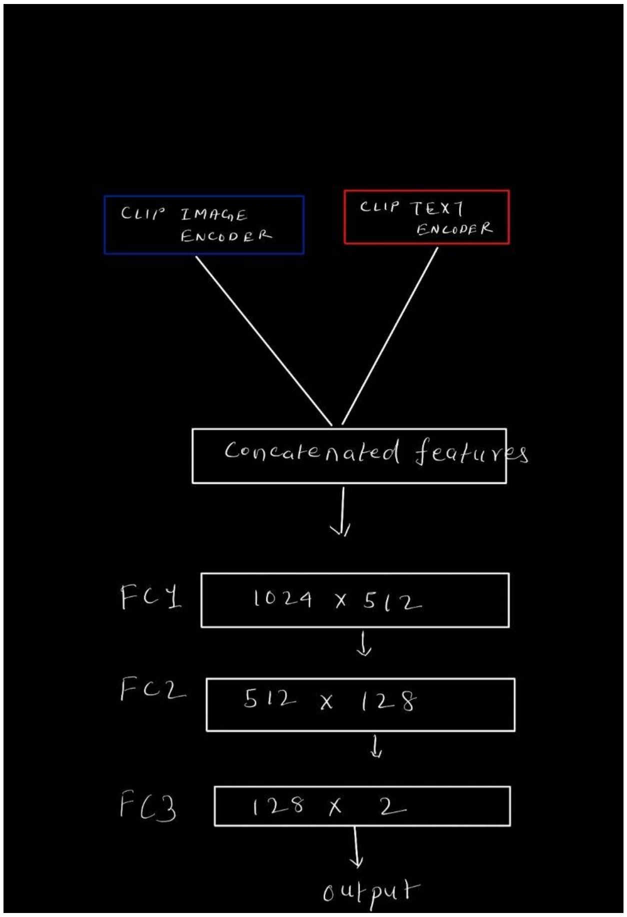
```
              precision    recall  f1-score   support

 Not-Hateful       0.66      0.82      0.73       340
     Hateful       0.48      0.28      0.35       200

    accuracy                           0.62       540
   macro avg       0.57      0.55      0.54       540
weighted avg       0.59      0.62      0.59       540
```

The multimodal values of loss and accuracies, and the results on test data are much better than the unimodal models. This explains the the multimodal nature of the dataset which requires both the caption and corresponding text simultaneously to generate contextual semantics, thereby creating two linearly seperable clusters of the samples in the tsne-feature space.



The above plot shows clear separation between the two classes - hate and non-hate . This shows that our clip model performs efficiently to learn to map embeddings in high dimensional feature space.

Multimodal Model Architechture



The multimodal model consists of pre-trained clip text and image encoders and three fully connected layers. Firstly, encodings from clip were concatenated to form the input to be fed into the neural network which then learns to classify the memes.

Comparison of accuracy on 3 Tasks

Multimodal > Text (unimodal) = Image (unimodal)