# CW1 - Multimodal IMDB Analysis with Keras: Report

**Introduction**

This study delves into a multi-mode IMDB dataset with film posters and their overviews. It aimed at classifying films according to genres using two independent models: a convolutional neural network (CNN) for posters and a long-short term memory (LSTM) network for overviews. This report contains an account of the data processing, model definition, training, and evaluation, along with a critical discussion of results.

**1. Data Processing**

**1.a. Image Processing of Posters**

The film posters used to be a TensorFlow from_tensor_slices dataset. First, an img_process function was denoted for CNN input preprocessing which resized the images to 64x64 and transformed them into 32 float form. Subsequently, the training and validation datasets were built with the tf.data API via performance-efficient constructs: parallel calls, batching, caching, prefetching, and more.

**1.b. Natural Language Processing of Overviews**

Data regarding the film overview have been made like posters, that is, loaded using TensorFlow dataset. The data for training and validation was created by using a batch size of 64. To build vocabulary for the LSTM model, the encoder.adapt() method was directly called on the training overviews data using tf.keras.layers.Text Vectorization. This would create a vocabulary for the most frequently occurring words-to-ensure efficient representation of text for the model.

**2. Definition of the Models**

**2.a. CNN for Posters**

i. To build a CNN to address the given architecture, Keras Functional API was used. Some key features included:

a. Convolutional layers utilized for feature extraction, along with ReLU activation.

b. Dropouts designed to regularize and avoid overfitting.

c. Max Pooling layers for downsampling.

**2.b. LSTM for Overviews**

In constructing the LSTM model, a series of tf.keras.Sequential layers were used by adding embedding layer to represent words as vectors, followed by using bidirectional LSTM layers for sequential information capture. The classification procedure relied on dense layers and dropout, whereas the output layer has a sigmoid activation. Just like CNN, this model was then compiled in combination with a binary crossentropy loss, Adam optimizer, and Precision and Recall metrics.

**3. Training of the Models**

Both models are trained using callbacks for the purpose of checkpointing and learning rate scheduling. Checkpointing saves the best model weights according to the validation loss to be able to use the best model. The learning rate scheduler varies the learning rate during the training possibly to enhance the convergence and improve its performance.

### 3.a. CNN Training

The CNN was trained for 40 epochs with the defined callbacks. The training and validation loss, precision, and recall were recorded for later analysis.

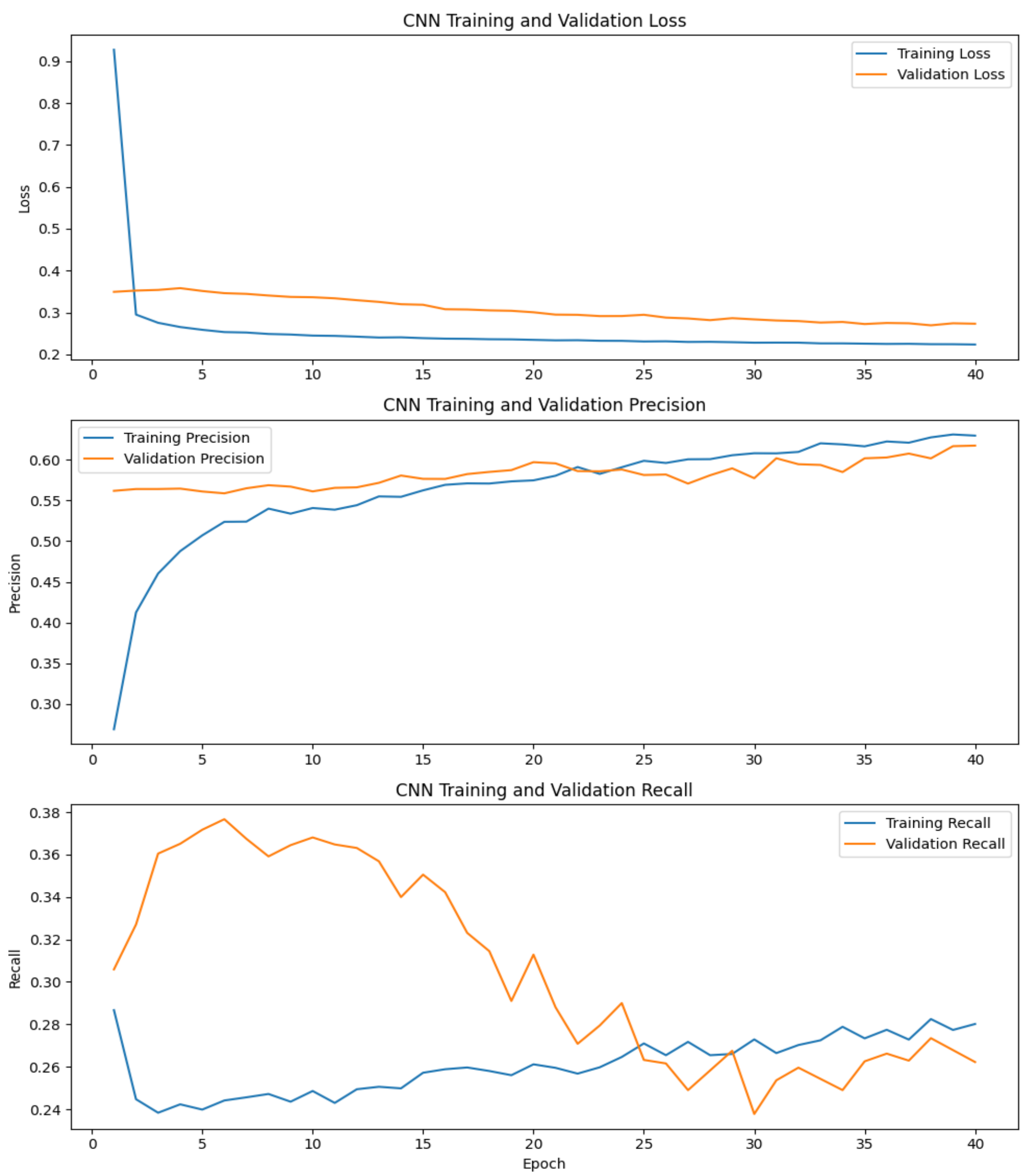
### 3.b. LSTM Training

The LSTM was trained for 20 epochs with similar callbacks and metrics recording.

## 4. Evaluation of the Models

4.a. CNN Evaluation

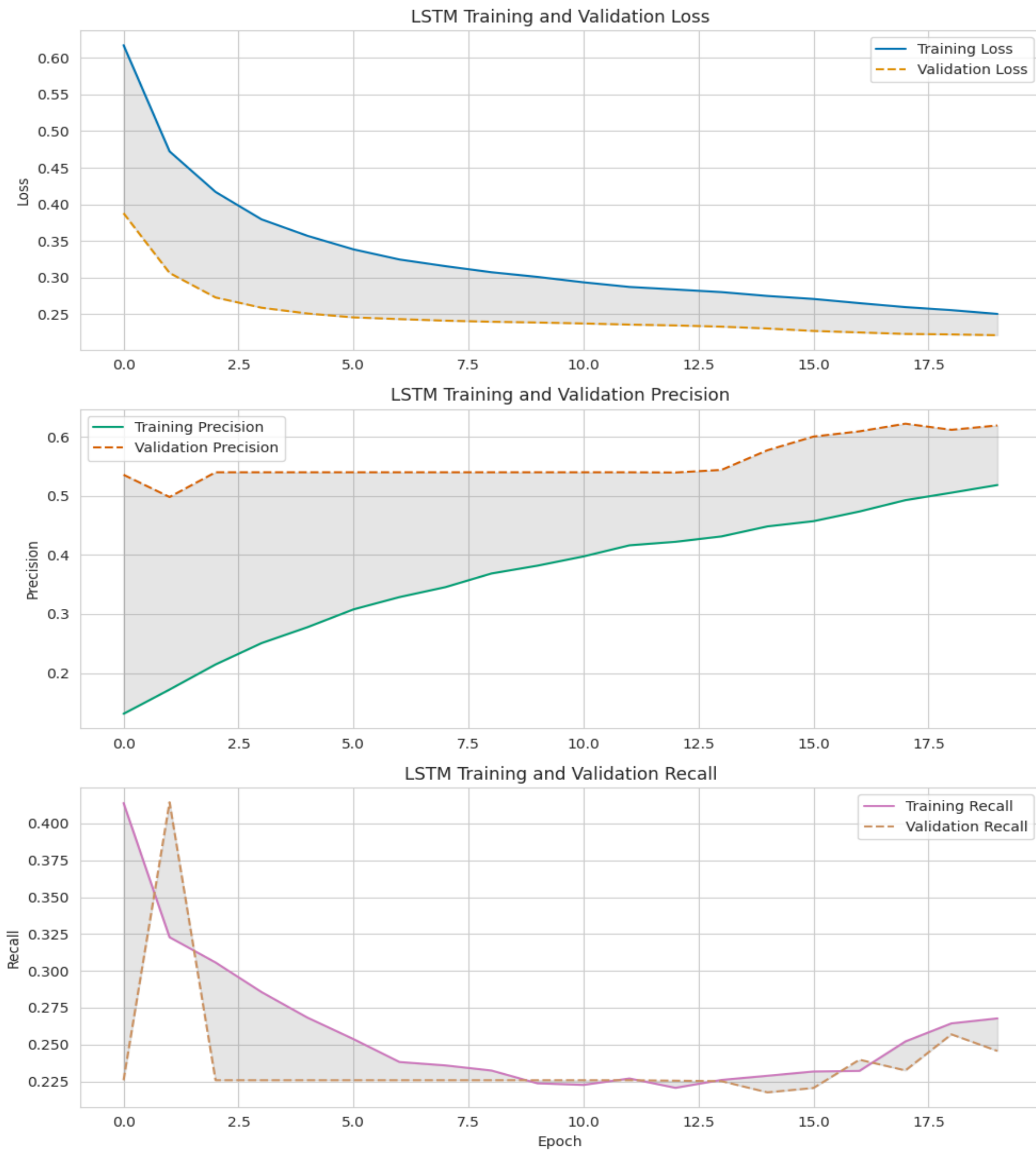
The CNN's performance was evaluated using plots of loss, precision, and recall over epochs. These plots are presented below:



The plots were analyzed to understand the model's learning behavior and identify potential issues like overfitting or underfitting. A critical evaluation of the model's performance across different genres was conducted based on these plots and further analysis.

4.b. LSTM Evaluation

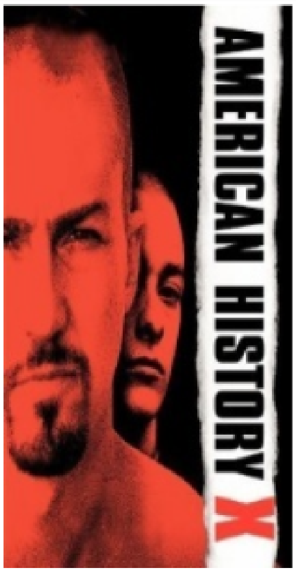
Similarly, the LSTM's performance was assessed using plots of loss, precision, and recall:




The plots were examined, and the model's performance was critically evaluated, considering its ability to classify films based on their overviews.

### 4.c. Examples and Critical Evaluation

The multi-label classification results were drawn from chosen movies selected from the data set. Posters and overviews were shown and compared to their ground truth genres against the top three predicted genres by the CNN and LSTM models. Such examples shed light on the strengths and weaknesses of the models, so to speak:




```
Overview: A former neo-nazi skinhead tries to prevent his younger brother from going down the same wrong p
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
------------------------------
Model      Top Genres               Probabilities
------------------------------
CNN        Drama                    0.6623
CNN        Action                   0.3712
CNN        Crime                    0.3701
LSTM       Drama                    0.6421
LSTM       Comedy                   0.4839
LSTM       Romance                  0.2266
------------------------------
Ground Truth Genres: Crime, Drama
------------------------------
```

An in-depth examination of the performance of the models on the various genres has been undertaken keeping consideration of relevant examples and the overall results. Such discussions included possible reasons for misclassifications, limitations of the models, areas for improvement, and so on.

### Conclusion

The current report presents a multimodal analysis on the comparing movie posters with the synopsis using CNN and LSTM models for genre classification. All the activities from data processing to model definitions, training, and evaluation were recorded thoroughly. Critical analysis of the results was done using plots and examples, with comparative analysis highlighting the performance of the models across the various genres. This project truly demonstrated the possibilities of this combination (visually and in textual form) in classifying against the genre of a film but also signaled some challenging aspects and future areas of possible investigation.