CW1–Multimodal IMDB Analysis with Keras   Student ID: 23031641

For this study, we have been provided a dataset from the internet movie database that has a list of films, their titles in JPEG format, and short descriptions of those films in text format. Two machine learning models, one using Convolutional Neural Networks (CNNs) to categorise movie posters and the other using Long Short-Term Memory (LSTM) networks to categorise movie overviews, are to be developed and trained for this project. This study will present the models' predictions, critically evaluate their performance on specific situations, and investigate genre categorisation triumphs and failures.

1. Data Processing

1.1. Image Processing of Posters

The $img\_process$ method changes the size of pictures to 64×64 pixels and converts them to $float32$ for accuracy. The training and validation datasets are effectively loaded utilising the $parse\_image$ and $img\_process$ functions, incorporating optimisations like as batching, caching, shuffling, and prefetching to guarantee rapid and dependable data processing.

1.2. Natural Language Processing of Overviews

The training and validation datasets for film summaries are generated with $tf.data.$ Dataset using batching and prefetching for efficient data processing, facilitating smooth training with a batch size of 64. The vocabulary is constructed using $encoder.adapt()$ on training overviews and can hold $10,000$ tokens. $TextVectorization$ turns text into integer sequences for model input. $Encoder.adapt()$ retrieved the most common tokens from the training dataset for the vocabulary. The dataset's linguistic structure may be understood by printing the first 200 words, which reveal the most prevalent phrases.

2. Definition of Models

2.1. CNN Model

For the provided architecture, the Keras Functional API builds the CNN model with convolutional, dropout, max-pooling, and fully connected layers. To avoid overfitting, dropout layers follow with rates of 0.2 for convolutional layers and 0.5 for dense layers after ReLU activation with suitable filter sizes, kernel sizes, and paddings. The model uses the Adam optimiser (10^{-4} learning rate), binary cross-entropy loss, and Precision and Recall measures.

2.2. LSTM Model

The LSTM model is constructed with $tf.keras.Sequential$ to handle text data, initiating with an encoder and an embedding layer $(output\_dim = 256,\ mask\_zero = True)$ for text representation. The architecture has two Bidirectional LSTM layers: the initial layer contains 256 units $(dropout = 0.5, recurrent\_dropout = 0.2, return\_sequences = True)$, while the subsequent layer consists of 128 units $(dropout = 0.5,\ recurrent\_dropout = 0.2)$. The design comprises a dense layer with 128 units and ReLU activation, succeeded by a dropout layer $(rate = 0.8)$, and culminating in an output layer with 25 units with sigmoid activation.

3. Training of Models

3.1. General Training Setup

- **Checkpoint Callback for Best Weights**

A **Checkpoint Callback** saves the best-performing epoch weights based on validation loss. This keeps the model's evaluation and prediction weights appropriate, preventing overfitting or deterioration from subsequent epochs.

- **Learning Rate Scheduler**

Training learning rate is dynamically adjusted using the **LearningRateScheduler** Callback. The first 10 epochs are constant, then exponential decay reduces the learning rate as the model converges to optimise training.

3.2. CNN Training

Use of checkpoint and learning rate scheduler callbacks trained the CNN model for 40 epochs. The training approach optimised the binary cross-entropy loss function, with a constant learning rate for 10 epochs and an exponential decline afterward. The best weights were saved at epoch 34 $(val\_loss\ =\ 0.2453)$. Precision and recall measures showed the model's film poster pattern recognition.

3.3. LSTM Training

Using checkpoint and learning rate scheduler callbacks, the LSTM model was trained for 20 epochs. The learning rate dropped after 10 epochs as the model optimised binary cross-entropy loss. The best weights were stored at epoch 20 $(val\_loss\ =\ 0.2254)$. The model's accuracy and recall metrics improved as it classified film overviews, capturing text interpretations

4. Evaluation of Models

4.1. CNN Evaluation

First, we will examine the training and validation losses, together with the accuracy and recall metrics, to assess the CNN's performance in identifying film genres from posters.
The below graphs illustrate the performance of the CNN model:

Loss Plot: This graph depicts the variation of training and validation losses across epochs. If the losses are closely aligned and consistently diminish; the model is functioning effectively. A significant disparity between them may indicate overfitting.

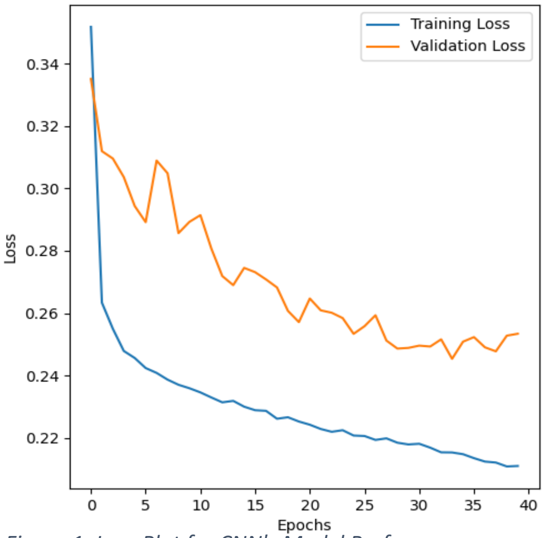


Figure 1: Loss Plot for CNN's Model Performance

Precision Plot: This plot demonstrates the precision. for both the training and validation datasets. A consistently high accuracy curve for both datasets indicate the model's proficiency in accurately predicting genres.

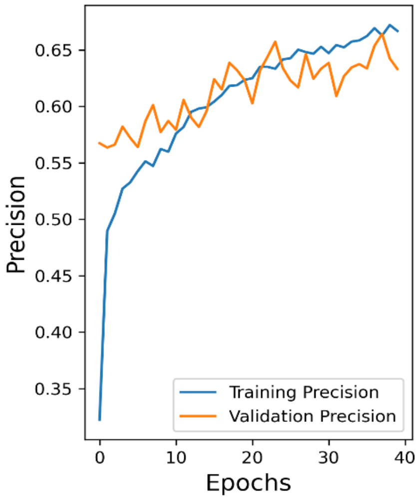


Figure 2: Precision Plot for CNN's Performance

Recall Plot: The recall plot shows the model's proficiency in accurately identifying genres. While high recall is helpful, an excessively elevated recall paired with low accuracy may suggest a model that is overly permissive.

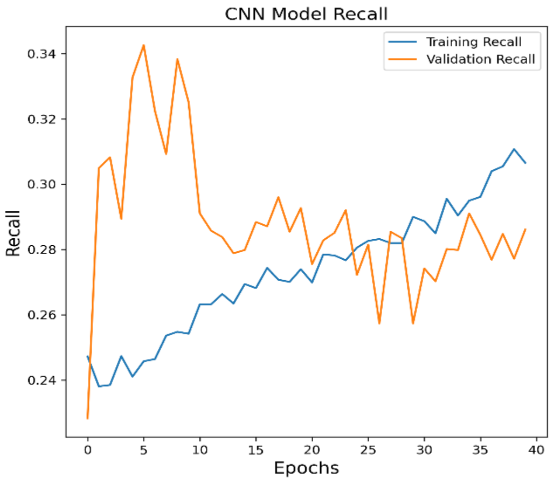


Figure 3: Recall Plot for CNN's Performance

4.2. LSTM Evaluation

LSTM was trained on film textual overviews and quantified using loss, accuracy, and recall like CNN. Training and validation losses reveal the model's ability to fit training data and generalise to new data. Precision and recall measures measure how successfully the LSTM model classifies text genres.

The subsequent diagrams elucidate the performance of the LSTM model:

Loss Plot: This graphical representation illustrates the training and validation losses, which, in an ideal scenario, should converge and stabilise as the model undergoes the learning process.

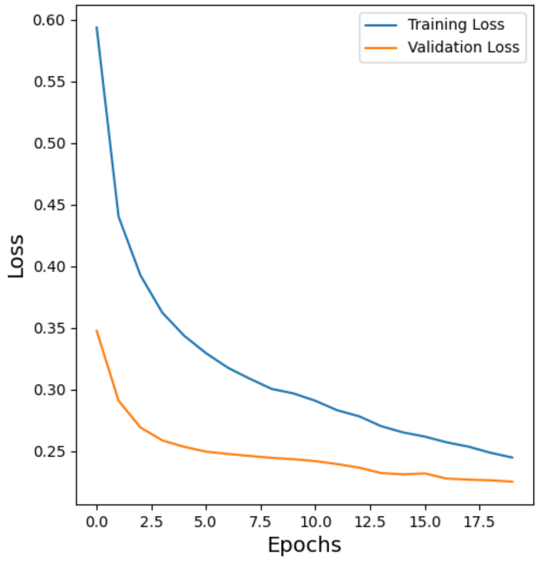


Figure 4: Loss Plot for LSTM's Performance

Precision Plot: Both training and validation sets show a high and steady precision throughout epochs, indicating that the model is accurately predicting the genres.

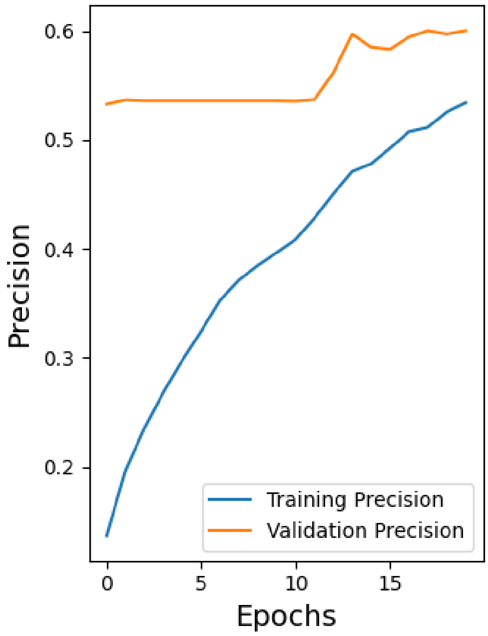


Figure 5: Precision Plot for LSTM's Performance

Recall Plot: A high recall for both the training and validation sets indicates that the model effectively identifies all genres present in the film overviews.

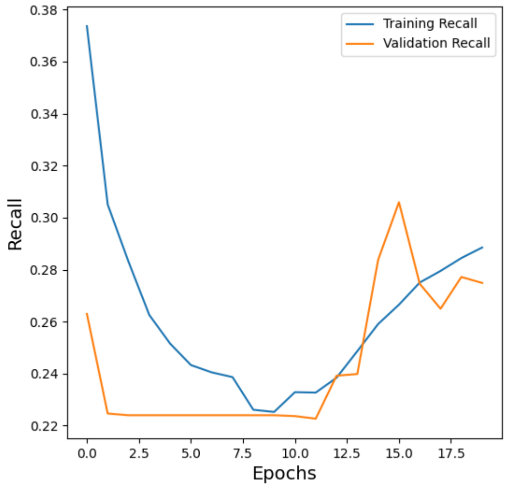
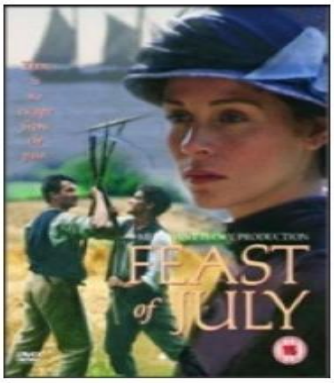


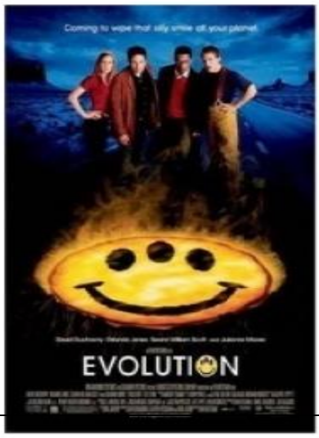
Figure 6: Recall Plot for LSTM's Performance

5.1 Model Predictions for Selected Films

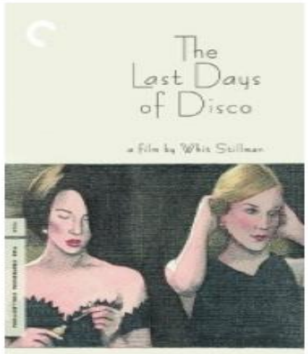
All of examples three have been discussed here:

<table>
<tr><td>Overview:
“After an abandoned young woman in late 19th Century England is taken in by a rural couple with three handsome sons, tragic consequences result.”</td><td rowspan="3">Poster for Film 0

</td></tr>
<tr><td>Top 3 CNN Predictions: [7 4 18]</td></tr>
<tr><td>Top 3 LSTM Predictions: [7 4 5]</td></tr>
<tr><td colspan="2">Ground Truth Genres:
[0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]</td></tr>
</table>

Discussion: The CNN algorithm accurately identified Romance (Column 7) but incorrectly classed the film as Action (Column 4) and Comedy (Column 18) based on visual cues in the poster. The LSTM model accurately recognised Romance (Column 7) but falsely categorised the film as Action (Column 4) and Adventure (Column 5), due to thematic aspects in the synopsis.

<table>
<tr><td>Overview:
“A fire-fighting cadet, two college professors, and a geeky but sexy government scientist work against an alien organism that has been rapidly evolving since its arrival on Earth inside a meteor.”</td><td rowspan="3">Poster for Film 1

</td></tr>
<tr><td>Top 3 CNN Predictions: [7 0 4]</td></tr>
<tr><td>Top 3 LSTM Predictions: [0 7 4]</td></tr>
<tr><td colspan="2">Ground Truth Genres:
[0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]</td></tr>
</table>

Discussion: The CNN model predicts Romance (7), Drama (0), and Action (4), but misclassifies the film by incorrectly predicting Drama and Romance, while correctly predicting Action. Similarly, the LSTM model predicts Drama (0), Romance (7), and Action (4), showing a bias toward Drama and Romance due to generic keywords in the overview.

<table>
<tr><td>Overview:
"Story of two female Manhattan book editors fresh out of college, both finding love and themselves while frequenting the local disco.”</td><td rowspan="3">Poster for Film 3

</td></tr>
<tr><td>Top 3 CNN Predictions: [7 18 4]</td></tr>
<tr><td>Top 3 LSTM Predictions: [7 4 18]</td></tr>
<tr><td colspan="2">Ground Truth Genres:
[0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]</td></tr>
</table>

Discussion: The film is classified as Romance (7), Comedy (18), and Drama (4). The **CNN model** correctly predicts Romance and Comedy but misclassifies it as Action instead of Drama. The LSTM model accurately predicts Romance, Comedy, and Drama. This shows the LSTM's strength in capturing textual themes, while the CNN model relies more on visual cues.

5.2 Discussion of Strengths and Weaknesses

Performance of CNN:

Strengths: The CNN model demonstrated proficiency in categorising films with various visual genres. The poster was efficient in conveying the film's genre through recognisable visual clues. **Weaknesses:** The model had difficulties with unclear posters or those exhibiting overlapping genres, where visual aspects were ambiguous or more generalised. Films 0-3 intricate poster design may make it harder for the algorithm to forecast genres like Action or Comedy. **Performance of LSTM:**

Strengths: The LSTM model exhibited robust performance when the summaries were concise and unambiguously delineated the film's genre. **Weaknesses:** The LSTM model struggled with broader theme overviews, as essential genre indications were less discernible or confusing.

The **CNN model** excelled in films with clear visual cues like **Film 0**, while the **LSTM model** outperformed in films with abstract themes like **Film 2** by accurately identifying **Drama**.

5.3 Conclusion

Both models demonstrated robust performance; nonetheless, there are opportunities for enhancement, especially in managing uncertainty. Improvements may involve integrating more comprehensive data or investigating sophisticated model architectures to enhance generalisation and precision.