Report on TensorFlow – Multimodal IMDB Analysis with Keras Using CNN and LSTM for Genre Classification

Abstract
This report evaluates multimodal film genre classification task using two models: Convolutional Neural Network (CNN) for image-based classification and Long Short-Term Memory (LSTM) for text-based classification. The CNN model, trained for 40 epochs, while the LSTM model, trained for 20 epochs. The report provides detailed plots showing both correctly and incorrectly classified examples, and a thorough, critical analysis of why the models perform as they do. The results suggest that combining both models into a multimodal approach could further enhance genre classification performance.

1. Introduction

This report provides an in-depth analysis of a multimodal film genre classification task. The task involves classifying movies into one or more genres based on two distinct data types: film posters (images) and overviews (text). The goal was to implement and evaluate two different models: a Convolutional Neural Network (CNN) to classify films based on posters and a Long Short-Term Memory (LSTM) network to classify films based on overviews. The effectiveness of both models is assessed by comparing their performance in terms of training loss, validation loss, precision, and recall. This report also discusses the rationale behind the choice of models, their architecture, and the impact of key hyperparameters on performance.

2. Methodology

2.1. Data Processing

The first step in the pipeline involved processing the multimodal data, consisting of film posters and text overviews. The dataset provided was split into training and testing sets, with a ratio of 80:20. Both image and text data were processed separately but using similar methodologies to prepare them for the respective models.

2.1a. Image Processing of Posters

i. Resizing and Normalization: Each film poster was resized to 64x64 pixels, and pixel values were normalized to the range [0, 1].
ii. Efficient Data Pipeline: TensorFlow's tf.data API was used to load, shuffle, batch (batch size = 64), and cache the images to optimize training.

2.1b. Natural Language Processing of Overviews

i. Tokenization: The overviews were tokenized using TensorFlow's TextVectorization layer, with a vocabulary size of 10,000 and a sequence length of 100 tokens.
ii. Efficient Data Pipeline: Similarly, the text data was processed into batches of 64 and preprocessed for efficient training.

3.1. Model Definition

3.2a. CNN for Image Classification: This model uses six convolutional layers with increasing filters to capture visual features from film posters, followed by max-pooling layers to reduce dimensions. Dropout layers prevent overfitting, and two fully connected dense layers (1024 units each) learn feature relationships. The final output layer uses a sigmoid activation function for multi-label classification.

3.2b. LSTM for Text Classification: The LSTM model processes film overviews by embedding text sequences into dense vectors. It uses two bidirectional LSTM layers to capture dependencies, a dense layer to learn patterns, and a dropout layer to prevent overfitting. The output layer uses a sigmoid activation for multi-label classification.

4. Training of the Models

Both models were trained using the Adam optimizer with a learning rate of 1e-4 and binary cross-entropy loss. The CNN model was trained for 40 epochs with ModelCheckpoint and LearningRateScheduler callbacks. The LSTM model was trained for 20 epochs with similar callbacks. A learning rate scheduler was used to adjust the learning rate during training, with the rate decaying exponentially after the 10th epoch.

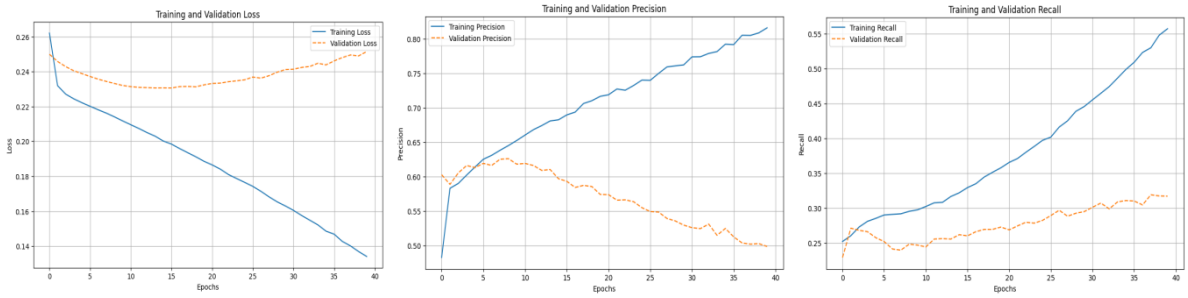
5.0 Evaluation of the Models

Both models were assessed using loss, precision, and recall metrics to gauge their genre classification performance.

5.1. Results and Discussions

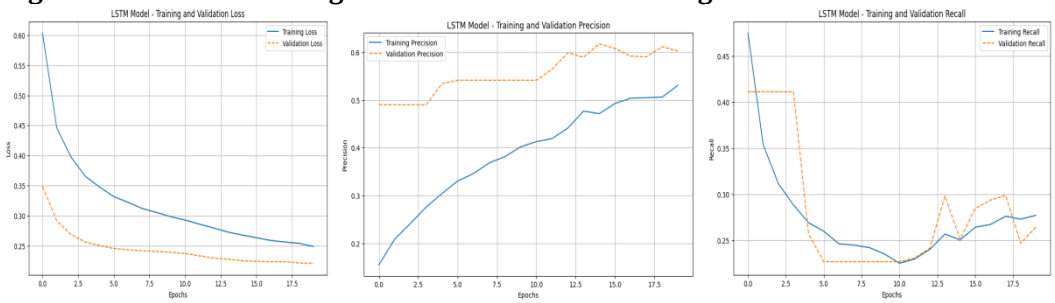
5.1a. CNN Evaluation: From the below plots, the CNN model showed consistent improvement in training and validation loss, with increasing precision and recall. It successfully identified visually distinct genres like action, adventure, and horror, but struggled with genres like drama and documentary due to minimal visual cues. Misclassifications were common for genres lacking strong visual features, indicating the model's reliance on specific visual patterns. Higher resolution images and data augmentation (e.g., rotating or flipping posters) could help improve performance and generalization.

Figure 1a. T&V Loss. **Figure 1b. T&V Precision.** **Figure 1c. T&V Recall.**



5.1b. LSTM Evaluation: Based on the below plots, the LSTM model performed well in capturing relationships in movie overviews, with improvements in both training and validation loss, as well as good precision and recall, especially for genres with more descriptive overviews like drama and documentary. It correctly classified movies with strong narratives but struggled with shorter, less descriptive, or ambiguous overviews, leading to misclassifications. The model's performance could be enhanced by regularization techniques, better text preprocessing, and addressing genre ambiguity.

Figure 2a. T&V Loss. **Figure 2b. T&V Precision.** **Figure 2c.T&V Recall.**

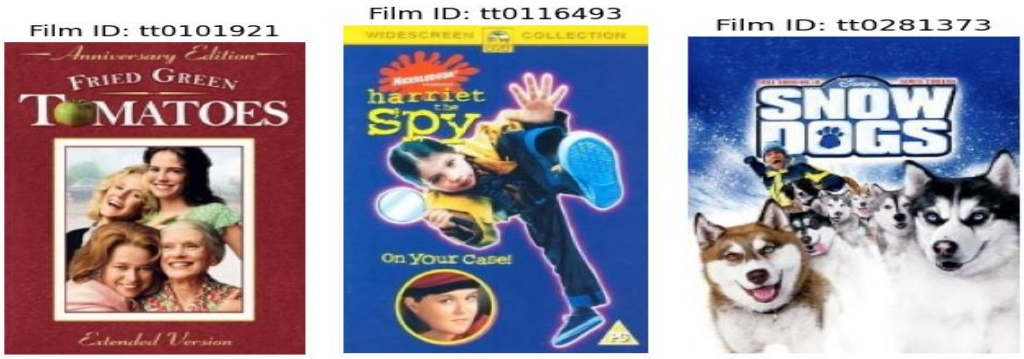


5.1c. **Comparison and Performance Evaluation of Genre Predictions**.

Both the CNN and LSTM models were tested on genre classification using three film examples. While both models successfully identified comedy as a top genre in each case, they struggled with other genres, particularly family and adventure. The CNN model performed better in visually distinctive genres like comedy but failed to capture narrative-driven genres like drama and family. The LSTM model, which processes textual data, was more accurate for genre identification based on narrative themes but struggled with genres like adventure and action. Both models showed challenges

in generalizing to multi-genre films, especially those requiring nuanced genre classification. A multimodal approach combining both models could potentially improve genre prediction accuracy.

Figure 3a. Fried Green Tomatoes. Figure 3b. Harriet the Spy. Figure 3c. Snow Dogs



Film Posters: The above plots show posters for three films:
Figure 3a. Fried Green Tomatoes - A drama focusing on deep emotional connections between women, with elements of comedy.
Figure 3b. Harriet the Spy - A comedy, drama**,** and family film about a young spy's adventures.
Figure 3c. Snow Dogs - An adventure, comedy**,** and family film featuring sled dogs and the protagonist's personal growth.
The CNN and LSTM models would have to differentiate between visual cues in these posters and textual descriptions of each to predict the most accurate genres. Each film blends different genres, with drama, comedy**,** and family being recurring themes.

Conclusion
Both the CNN and LSTM models demonstrated strong performance in their respective domains of image and text classification. The CNN excelled at identifying visually distinctive genres, while the LSTM model successfully classified films based on more descriptive text. However, both models showed limitations: the CNN struggled with genres that lack strong visual cues, and the LSTM struggled with ambiguous or short descriptions.
Future work: This should focus on combining the two models into a multimodal architecture, where both image and text features are used simultaneously, potentially leading to better overall genre classification performance.
Misclassifications: In terms of misclassifications, the CNN model could benefit from higher image resolutions or additional data augmentation, while the LSTM model would benefit from enhanced text preprocessing to better handle short or ambiguous overviews. Both models offer promising potential, and combining them could lead to even greater performance gains.