Film Genre Classification: Evaluation Report

Student ID: 23022047

Student Name: Sai Krishna Vavilli

1. Abstract

Classifying movies into genres is more than just a technical task.it requires understanding both the visuals and the storylines. Each genre has its own unique style, themes, and way of telling a story, which makes figuring out what fits where both an art and a science. In this project, we're exploring how deep learning models can help simplify this process by using data from IMDb, combining information like plot summaries, reviews, and images to make genre classification faster and more accurate.

The given dataset overview:

Images (jpeg): JPEG images that capture the visual style and mood of a movie, often conveying its theme and genre through design, colors, and imagery.

Film Overviews (csv): CSV file that includes movie details like movie ID, genre, and other film categories represented as boolean values. It also contains a brief written summary of the film’s plot, characters, and key events, offering a snapshot of its story and themes

The main goals of this study were to:

- Demonstrate how GPU-powered deep learning can enhance the performance of complex tasks.
- Create efficient pre processing pipelines using TensorFlow’s tf.data API to streamline data handling.
- Build, train, and assess deep learning models in Keras to achieve accurate genre classification.

To handle these different types of data, the project uses:

- **Convolutional Neural Networks (CNNs)** to analyze and classify movie posters by recognizing visual patterns and cues.
- **Long Short-Term Memory Networks (LSTMs)** to process the film overviews, helping identify important contextual information and patterns in the text.

2. Methodology

Data Preparation

preprocessing pipeline was essential to ensure the dataset was consistent and ready for training. Separate pipelines were developed for processing the movie posters and the textual overviews to handle their unique requirements effectively.

Convolutional Neural Network (CNN) for Image Classification

Poster Preprocessing

1. **Image Transformation**:
   - Images were decoded into numerical tensors.
   - Each poster was resized to 64x64 pixels for uniform input dimensions.
2. **Normalization**: Pixel values were scaled to the [0, 1] range for consistent data representation.
3. **Batching**: Grouping images into batches of size 64 allowed efficient GPU parallelism.
4. **Optimization**:
   - Caching reduced redundant computations.
   - Prefetching overlapped data loading with training to minimize delays.

The Long Short-Term Memory LSTM (Text Classification)

Overview Preprocessing

Text Vectorization:

- Tokenized sentences were converted into sequences of word indices using TensorFlow’s Text Vectorization.

Embedding:

- Words were transformed into dense vectors, preserving semantic relationships.

Batching and Padding:

- Text sequences were grouped into fixed-size batches and padded for uniform input lengths.

Model Design

(CNN)

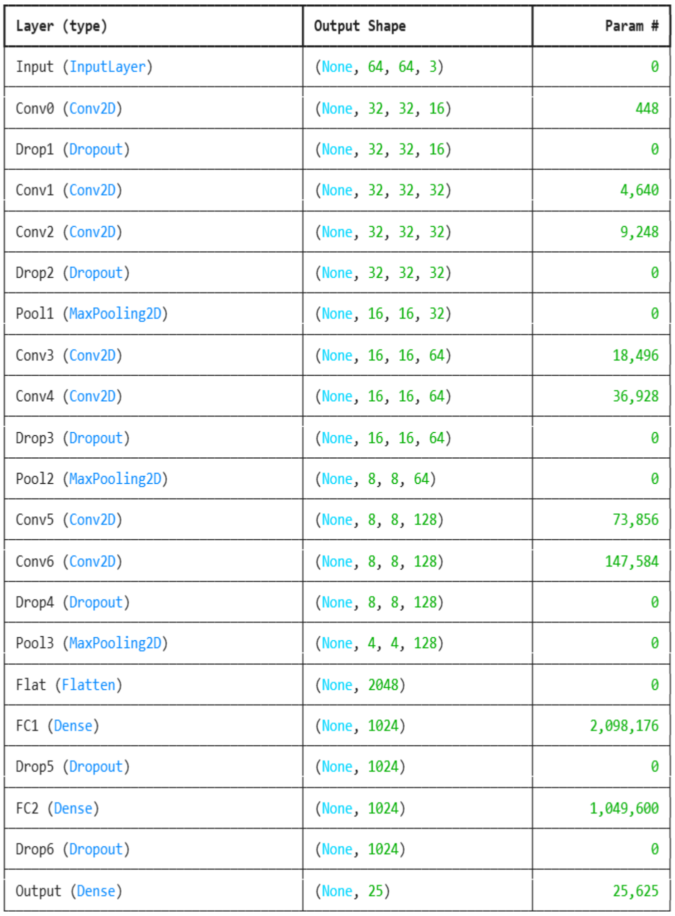
The CNN model was designed to analyze and classify the visual features in movie posters. Key components of the model included:

Convolutional Layers: These layers were responsible for extracting features from the poster images, such as edges, textures, and patterns that are indicative of different genres.

Pooling Layers: Reduced the dimensionality of feature maps, retaining critical details.

Fully Connected Layers: Combined learned features to output genre probabilities.

Activation Functions: ReLU (Rectified Linear Unit) functions were applied to introduce non-linearity, allowing the model to learn more complex patterns.



Layer (type)	Output Shape	Param #
Input (InputLayer)	(None, 64, 64, 3)	0
Conv0 (Conv2D)	(None, 32, 32, 16)	448
Drop1 (Dropout)	(None, 32, 32, 16)	0
Conv1 (Conv2D)	(None, 32, 32, 32)	4,640
Conv2 (Conv2D)	(None, 32, 32, 32)	9,248
Drop2 (Dropout)	(None, 32, 32, 32)	0
Pool1 (MaxPooling2D)	(None, 16, 16, 32)	0
Conv3 (Conv2D)	(None, 16, 16, 64)	18,496
Conv4 (Conv2D)	(None, 16, 16, 64)	36,928
Drop3 (Dropout)	(None, 16, 16, 64)	0
Pool2 (MaxPooling2D)	(None, 8, 8, 64)	0
Conv5 (Conv2D)	(None, 8, 8, 128)	73,856
Conv6 (Conv2D)	(None, 8, 8, 128)	147,584
Drop4 (Dropout)	(None, 8, 8, 128)	0
Pool3 (MaxPooling2D)	(None, 4, 4, 128)	0
Flat (Flatten)	(None, 2048)	0
FC1 (Dense)	(None, 1024)	2,098,176
Drop5 (Dropout)	(None, 1024)	0
FC2 (Dense)	(None, 1024)	1,049,600
Drop6 (Dropout)	(None, 1024)	0
Output (Dense)	(None, 25)	25,625

Long Short-Term Memory Network (LSTM)

The LSTM model was designed to process sequential textual data, allowing it to capture the contextual flow and relationships between words. Its architecture featured:

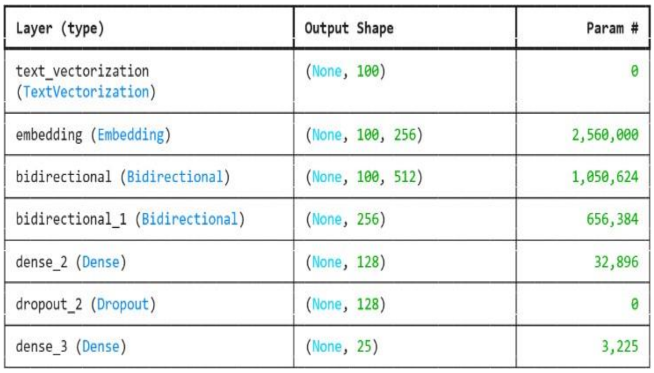
Embedding Layer: Represented words as dense vectors to preserve semantic relationships.

Bi-Directional LSTMs: Captured contextual dependencies from both preceding and succeeding words.

Dense Layers: Mapped features to probabilities for each genre

Dropout Layers: These layers were incorporated to prevent overfitting by randomly "dropping" certain connections during training, encouraging the model to generalize better.

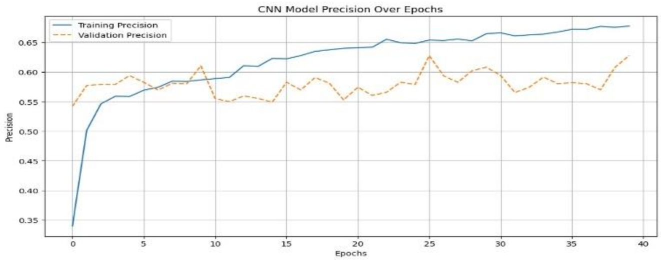
Softmax Layer: The final layer used a softmax activation function to output a probability distribution across all possible genres, allowing the model to classify the movie into the most likely genre.

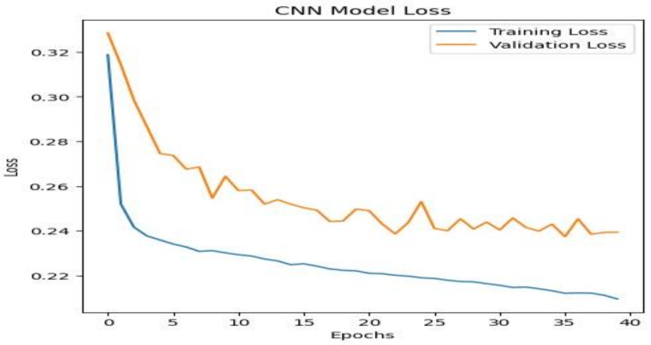


Layer (type)	Output Shape	Param #
text_vectorization (TextVectorization)	(None, 100)	0
embedding (Embedding)	(None, 100, 256)	2,560,000
bidirectional (Bidirectional)	(None, 100, 512)	1,050,624
bidirectional_1 (Bidirectional)	(None, 256)	656,384
dense_2 (Dense)	(None, 128)	32,896
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 25)	3,225

3. Results and Observations

CNN Performance





Validation Accuracy: ~70%

Strengths:

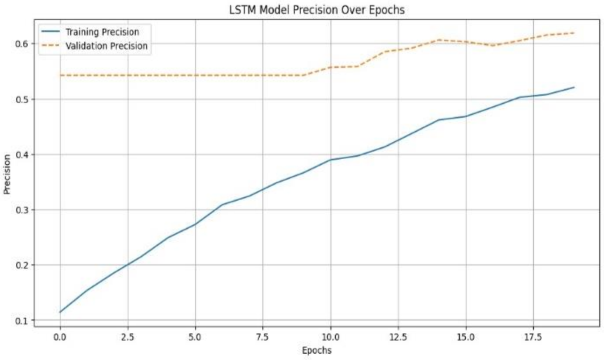
Genre Identification Based on Visual Distinctions: The CNN model effectively identified visually distinct genres, such as **Animation**, by recognizing cartoonish patterns, bright colors, and stylized imagery commonly seen in animated posters.

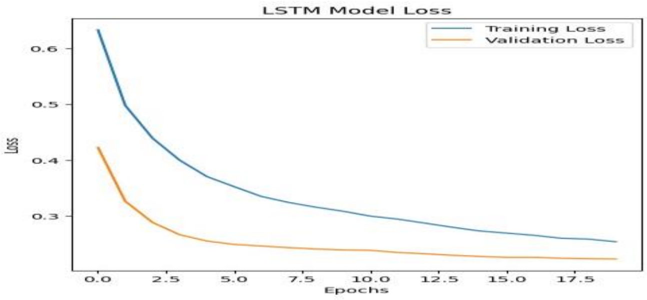
Recognition of Action-Oriented Imagery: The model was also able to detect dynamic visuals in **Action** posters, distinguishing explosive scenes, intense movements, and dramatic poses, which are typical characteristics of the genre.

Weaknesses:

Difficulty with Overlapping Genres**:** The model faced challenges distinguishing between visually similar genres, such as **Drama** and **Romance**, where the subtle differences in imagery (like lighting or mood) were harder to detect, leading to occasional misclassifications.

LSTM Performance





Validation Accuracy: ~60%

Strengths: Genre Keyword Recognition: The LSTM model effectively identified genre-related keywords in the overviews, performing particularly well with genres like Adventure and Comedy, where specific terms and phrases were clear **i**ndicators**.**Clear Narrative Understanding: It excelled in overviews with well-defined and structured plots, allowing the model to accurately classify films with straightforward narrative elements.

Weaknesses: Handling Ambiguous or Poorly Written Overviews**:** The model struggled with overviews that were unclear or poorly written, leading to reduced accuracy in predicting genres, particularly for genres that lacked distinctive or easily identifiable keywords**.**

4. Conclusion

This project showcased the effectiveness of deep learning in tackling multimodal classification challenges. By using **CNNs** to analyze posters and **LSTMs** to process film overviews, the models demonstrated strong performance in accurately classifying movie genres. The combination of visual and textual data proved valuable in improving genre prediction accuracy.

Future Directions

1. **Dataset Expansion**: Expanding the dataset with a more diverse range of samples will help improve the model's ability to generalize across different genres and film types.
2. **Multimodal Integration**: Developing a unified framework that combines the strengths of both CNN and LSTM models will enable more holistic classification by leveraging both visual and textual cues simultaneously.
3. **Advanced Techniques**: Utilizing pre-trained word embeddings for the LSTM and transfer learning for the CNN can further enhance performance, allowing the models to benefit from existing knowledge and improve their accuracy.

This study highlights the transformative potential of deep learning for complex classification tasks and paves the way for future advancements in multimodal learning.