# Bangla Book Genre Classification Using Multimodal Network

by

Abid Al Mamun
19301066
Md. Nahidul Islam
19301237
Md. Hasibul Hasan
19301163
Loknath Saha
20201017

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

# Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

| | |
|---|---|
| Abid Al Mamun<br>19301066 | Md. Nahidul Islam<br>19301237 |
| Md. Hasibul Hasan<br>19301163 | Loknath Saha<br>20201017 |

# Approval

The thesis/project titled "Bangla Book Genre Classification Using Multimodal Network" submitted by

1. Abid Al Mamun(19301066)
2. Md. Nahidul Islam(19301237)
3. Md. Hasibul Hasan(19301163)
4. Loknath Saha(20201017)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 17, 2024.

**Examining Committee:**

Supervisor:
(Member)

Tanzim Reza
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

# Abstract

Bangla literature spans a wide range of genres and has demanded an innovative approach that considers both textual and visual elements for accurate classification. This research focuses on classifying Bangla book genres by using a multimodal system that includes Convolutional Neural Network(CNN) and long short-term memory networks(LSTM). To enhance our classification efficiency with minimal effort, we have focused solely on the book cover image and the Bengali book title for enabling more accurate book genre classification. The research employed CNN for analysing book covers visually and LSTM for textual classification in Bangla language. The combination of these two techniques has resulted in a multimodal genre classification system which will be a significant technical achievement in the field of genre classification. In this research, we evaluated seven models which are VGG16, BILSTM, LSTM and the multimodal models combinations of VGG16 + LSTM, VGG16 + BILSTM, ResNet50 + LSTM, ResNet50 + BiLSTM. Among them the VGG16 + LSTM multimodal model performed the best with an accuracy of 87%. This research's methodology, experiments and insights are exclusively centred on the Bangla books.This research emphasizes the potential of combining visual and textual data for genre classification tasks in Bangla literature and it has set the stage for more effective genre classification systems that benefit libraries, publishers and readers in the diverse and interconnected digital era.

**Keywords:** Bengali Book Genre classification; Multimodal genre classification System; Convolutional Neural Network(CNN); Bangla book cover; Bangla book name; Long short-term memory networks (LSTM)

# Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our advisor Tanzim Reza and co-advisor Rafeed Rahman for their kind support and advice in our work. They were there for us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

# Table of Contents

# List of Figures

# List of Tables

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$BILSTM$ Bidirectional LSTM

$CNN$ Convolutional Neural Network

$F1$ F1-Score, the harmonic mean of Precision and Recall

$LSTM$ Long Short-Term Memory

$Precision$ The ratio of true positive predictions to the total predicted positives

$Recall$ The ratio of true positive predictions to the total actual positives

$ResNet50$ 50-layer Residual Network

$VGG16$ 16-layer Visual Geometry Group Network

# Chapter 1

# Introduction

## 1.1 Introduction

Classifying books into genres is crucial for literature analysis, affecting libraries, publishers and readers. Automated genre classification is a newly emerging field that is gaining positive attention for its potential to transform book management and recommendation in the digital age. As the world has become more connected in the present age, it has expanded the classification system in various languages and literatures [37] [4]. One less explored language in this field is Bengali language, which is brimming with culture and literature. This research tackles the challenging task of "Bangla book genre classification using multimodal System". Bengali language has a rich literary history and expanding modern works that make it a fascinating yet challenging area for genre classification. From various kinds of classical poetry to contemporary novels, each genre in Bengali literature has it's own unique pattern and themes. Therefore, it requires an innovative approach that combines data from both visuals and texts.

This research leverages a multimodal system that harness the power of Convolutional Neural Network(CNN) and long short-term memory(LSTM) techniques to classify Bangla books into specific genres. Multimodal system combines different types of information that shows huge potential for tackling complex tasks that need a complete understanding of contents as the contents combines various features for a diverse dataset.

Convolutional neural network(CNN) is a well-established image classification model, can analyse the visuals of book covers, which is the key to identifying genres. Meanwhile, machine learning algorithm, long short-term memory networks(LSTM) handle the text, extracting language features from book titles. LSTM technique is one of the technique which is specialised for text that can help to improve the accuracy of classifying genres by Bangla text. By combining these methods it will create a more detailed and accurate genre classification system.

This research is not only technical but also a cultural exploration. It explores Bengali literature genres and highlights the relationships between language, visuals and genre categories. By diving into Bangla book genre classification, The work supports preserving, sharing and also appreciating Bengali literature, While also going forward in the field of automated genre classification. In the pages that follow, we have explored our methods, experiments and results offering insights on both the challenges and the opportunities in classifying Bangla books.

## 1.2 Research Objectives

Till today a lot of research has been conducted on classifying the book genres. Even in Bengali Language, research has been done on Bangla articles, newspaper titles etc. There are some research papers on Bengali book genre classification comparing different machine learning and deep learning models. Our goal is to classify the book genres using a deep learning multimodal system , utilising minimal effort by taking just two types of input data : text(book titles) and images(book cover). This approach will simplify the process for future users while ensuring more accurate genre classification. The images are extracted from the book covers and texts are extracted from the book titles. This research can be a blessing to many people in various ways. In an educational context, this research can support teachers to recommend Bengali book genres effectively. Students will also be able to choose their desired books based on their reading preferences and educational goals. Furthermore, there are a lot of Bengali authors who write new books on different exciting topics. Our classification system can help them refer to other books of the same genre. This will help them reach their target audience more efficiently. Additionally, as we know, a book cover and book name are the first thing that attracts a reader's mind. So by using our system they can get a clear idea how the book cover of the specific genre should look like. Moreover, our system can be a great tool to the book publishing industry, the libraries of our country, e-book sites and especially the readers. By categorising books into specific genre, the libraries can easily manage and organise their books and readers can easily find their desired books. This is also true for online book sales platforms, improving the user experience significantly. Besides, developing such a multimodal deep learning framework can promote technological advancement among the Bangladeshi researchers and AI communities.

## 1.3 Problem statement

Automated genre classification systems have become a popular research topic nowadays. Much research has been conducted to classify music, games, movies, books, cancer, heart disease, and many more [36] [35] [7] [37]. We are aiming to develop a model for Bangla book genre classification based on their cover page and book title. We could not find a proper dataset to conclude this research so we had to collect our own dataset. The most concerning problem for this model is accuracy due to minimal input data. As we are classifying the genre of a book based on its cover page and Bengali book title, sometime for limited features it can give us inaccurate results. Classifying a book simply based on colour can also lead to inaccuracy. Also, not all books are classified into a specific genre, a single book can fall under one or more book genre which can also lead to misclassification of a book [5]. Developing a smart system that can automatically figure out what genre a book belongs to, the system needs to be good at classifying even when books have similar or mixed-up characteristics. It is crucial to get this right so that when people search for books online or in a library they get accurate recommendation and find what they're looking for. Sorting books in libraries or online bookstores will be much easier and less time-consuming. Our goal is to develop a model and maximise its accuracy as much as possible to classify books into their respective categories.

# Chapter 2

# Literature Review

## 2.1 Literature Review

Recent academic research showed a group effort to explore diverse methodologies for classifying book genres. This exploration involved both textual and visual elements, adding layers of complexity to an already difficult field. One notable contribution regarding this work came from Lucieri et al., who addressed the formidable challenge of categorising book covers into specific genre [23]. Researchers have emphasized the complexity of this task, particularly due to the complex nature of the datasets at hand. Notably, they observed that convolutional neural networks (CNNs), often used for image classification, tend to heavily rely on recognizable objects within book covers for accurate genre classification. To address this dependency on clear visual hints, they introduced clear attention mechanisms, including the utilisation of simple and guided attention. These method allowed them to point out the most relevant regions within the book covers for genre identification. Their research involved a comprehensive valuation of various image classification models, among them are ResNet, Inception, and DenseNet. They conducted evaluations on a dataset containing 57,000 book covers, consists of 30 different genres which they referred to as the "30cat dataset". Their work focused on deeply understanding the dataset and carefully evaluating model performance which led to identifying critical input regions by effective use of attention mechanism. Lucieri et al.'s research highlighted the difficult nature of classifying book genres and point out the importance of dataset quality and design choices [23]. Also Jolly [11] used Layer-wise Relevance Propagation (LRP) on the book cover image classification results. They used LRP to explain the pixel-wise contributions of book cover design and highlighted the design elements contributing towards particular genres. Furthermore, Krizhevsky [3] trained a large, deep convolutional neural network to classify the 1.3 million high-resolution images in the LSVRC-2010 ImageNet training set into 1000 different classes. To make training faster, they used non-saturating neurons and a very efficient GPU implementation of convolutional nets. To reduce overfitting in the globally connected layers they employed a new regularization method. which proved to be very effective. Zujovic [2] used variable resolution painting data gathered across Internet sources rather than solely using professional high-resolution data to classify various kinds of paintings. Following research by Worsham, J., & Kalita, J. explored genre classification in long literary work emphasizing how each chapter contributes to the overall genre classification [12]. Previous research primarily dealt with shorter texts, using meth-

ods like CNNs and LSTMs. However, traditional approaches such as bag-of-words often failed for long documents. This study presented a hierarchical approach that classifies individual chapters and applies a voting mechanism to determine the final genre of the book. The results showed that XGBoost and CNNs with chapter-based text inputs outperform other deep learning models, highlighting the importance of analyzing compositional subtexts in literature.

Similarly, Biradar et al. [13] broadened the boundaries of genre classification by considering both the visual element of book covers and the text in book titles. Their research explored five different genres and found many underlying challenges. To address this difficult issue, they used colour-based distribution methods and incorporated transfer learning, Convolutional Neural Networks (CNNs). Additionally, they utilised Natural Language Processing (NLP) techniques for the classification of books based on their titles. In their research, CNNs emerged as the frontrunners, showcasing the significant influence of the ImageNet model—a testament to the power of pre-trained models in the field of genre classification. Their work showed that combining visual and textual data and making efficient design choices is crucial for accurate genre prediction. Meanwhile, Buczkowski et al. [9] used deep learning for book cover classification, they used Convolutional Neural Networks as their primary model where they predicted book genres only based on cover images. To assess the effectiveness of CNNs, they have conducted comparative evaluations against other methods, such as multinomial naive Bayes and Doc2Vec-based techniques. Impressively, CNNs emerged as the victors in this contest, achieving a commendabl accuracy rate of 87.5%. However, the significance of their research extends beyond these achievements. They also delved into critical factors affecting model performance, such as dataset size and the number of convolutional layers, shedding light on the complex relationship between these variables and the accuracy of genre classification models. Thus, Buczkowski's research deepens our knowledge of genre classification and shows us how dataset size and design choice interact with each other.

Book genre classification has gained significant attention in recent researchs by using various machine learning and deep learning methods. The traditional approaches often rely on text classification but neural networks like Recurrent neural networks (RNN), has shown significant improvement in maintaining the context and enriching the recommendation accuracy [32]. Recurrent neural network(RNN) is essential for sequential data for example texts and speech but have long short term dependencies. Long short term memory(LSTM) networks have the solution to this problem because LSTM have gate mechanism that selectively retain and discard information which makes them effective for longer sequence. LSTM became the preferred RNN model for tasks where language processing and speech recognition are involved. Enhancement to the basic LSTM model have boosted its performance in time series forecasting and machine translastion [19] [27]. Again, supervised learning techniques like Naive Bayes, Gradient Boosting, and Random Forest have also been applied in book genre classification. Among them, the Naive Bayes showed higher accuracy in classifying the book genres based on the book descriptions [33]. Over the time researchers have used different methods to improve the models and their accuracy. Such an example can be using Stop words. In order to increase the accuracy of the model, stop words can be a significant uplift. We have already seen such examples in the Turkish language. Researchers have highlighted the importance of

stemming and stop-word removal in improving the text classification for agglutinative or inflectional languages [4] [8]. Furthermore, Ul Haque [18] used corpus-based methodology for detection and extraction of Bengali stop word. They have used the machine learning library NLTK from python environment. As there is no introduction and classification available for Bengali stop word, this paper discusses those issues of Bengali Stop word, though they do not show significant improvements for non-agglutinative languages like English, they can be a great tool for languages like Bengali.

Additionally, Goyal & Prem Prakash [30] have done their research on deep learning techniques on larger literary datasets. They used GPU computing to optimise the performance and found that these approaches are more effective for genre classification. Those were all individual models but Multimodal approaches combine multiple forms of data for more accurate genre classification where the dataset are more complex. As seen in Kundu [21] used a multi-modal deep learning framework to solve this problem. The contribution of this paper is four-fold. Firstly, their method adds an extra modality by extracting texts automatically from the book covers. Secondly, image-based and text-based, state-of-the-art models are evaluated thoroughly for the task of book cover classification. Thirdly, they developed an efficient and saleable multi-modal framework based on the images and texts shown on the covers only. Lastly, a thorough analysis of the experimental results was given and future works to improve the performance is suggested. The results showed that the multi-modal framework significantly outperforms the current state-of-the-art image-based models. Also, in the paper by Jayaram, R., [20] explored genre prediction by using machine learning, combining both image and text features.

Turning our attention to the work of Gupta, Agarwal and Jain [14] also Narendra, M [25], they have gain significant progress to use machine learning and natural language processing (NLP) techniques in automated book genre classification. Their research started with the careful collection, preprocessing and feature extraction from a large dataset of Bengali news articles, which were carefully categorised into ten distinct genres. One of the main challenges they faced was the problem of dimensionality, a common problem arising from the large number of words found in books. To manage this complex challenge, they skillfully applied principal component analysis (PCA) for dimensionality reduction. Focusing on machine learning, their approach featured Decision Tree and AdaBoost classifiers. Also seen in Busagala [1] they extracted absolute word frequency from textual documents to be used as feature vectors in machine learning techniques. One of the limitations of this technique is the dependency on text length leading to lower classification rates. They also presented a performance evaluation of feature transformation techniques and regularized linear discriminant function (RLD) in automatic text classification. What distinguishes their research is the innovative use of unlabeled data, which significantly improved the accuracy of genre classification. Also, we can see in Panchal [29] they used K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Logistic Regression (LR) to predict book genre. By constructing a data set with proper structure and data, they predicted the genre by title and abstract of the book. Previous studies, such as Iwana et al. [5], utilized CNNs like AlexNet and LeNet for image-based classification, while others like Buczkowski et al. [9], focused on text based method such as book titles and descriptions. This paper compares image-only, text-only, and multimodal (image + text) methods. Results show that text-based classification

outperforms image only. Multimodal late fusion approach has the highest accuracy (60.1%) among these three approach. The study highlights the effectiveness of combining modalities for genre classification.

ResNet-50 is a well-known deep learning architecture particularly in the field of image classification. Its balance learning framework is allowed for the creation of very deep neworks without encountering the vanishing gradient problem. Due to its proven effectiveness this architecture is often used as a reference point. Researchers commonly utilized pretrained ResNet-50 models from ImageNet and applied transfer learning to adapt it for different research [28]. Which made it a convenient starting point for many studies. In a parallel aim, Islam et al. [34] delved into the detailed field of Bengali handwritten digit recognition. Their research highlighted the potential of pre-trained Convolutional Neural Networks (CNNs) in pattern recognition tasks. They subjected three pre-trained models—Inception V3, EfficientN tB0, and RestNet50—to rigorous evaluation using the Numpad dataset, and their results were remarkable. EfficientNetB0 emerged as the best performer, achieving an accuracy rate of 99.16%. This work highlights how using pre-trained models can tackle pattern recognition, especially for digit recognition. As we explored the complex field of genre classification research, we encountered Chiang et al. [42], who delved into the textual data domain. By using deep neural networks and FastText to predict book genres. They achieved accuracy rate ranging from 30.5% to 62.4% , revealing the challenges of classifying genres based on text alone. Similarly, Iwana et al. [5] did research on image-based models for book genre classification. They employed well-known models such as AlexNet and LeNet which are enhanced by transfer learning techniques on an impressive dataset of 57,000 book covers. Although, their achievement was only 24.7% and 13.5%, respectively, they have highlighted the unique challenges with ambiguous elements in book covers. Lucieri et al. [23] continued to research on multimodal methodologies by merging textual and visual data while expliciting attention mechanisms. Their main contributions are preparing the dataset carefully, through model evaluation and identifying the key input areas. In parallel to that, a research by Lucieri et al. [23] and Kundu, [22] explores the true potential of deep learning by employing ResNet-50 and Inception ResNet v2 in book genre classification. Despite encountered complexities in such advanced domains these collective studies have shown how difficult genre classification can be. They not only showcase the value of multimodal approaches in finding higher accuracy but also show the challenges associated with different genre classification dimensions. Furthermore, Hossain [31] used Resnet50 for classifying Covid-19 x-ray images to detect infection. Their proposed method leverage transfer learning and fine-tuning of deep Convolutional Neural Network based ResNet50 model to classify COVID-19 patients. They achieved an accuracy of 99.31%. This research signifies ResNet50 in the field of image classification. On the other hand Tammina, S [17] explored the limitation of data mining and machine learning algorithms. Which requires identical feature space and distribution for training and test data. To address this they proposed transfer learning, which uses pre-trained models for new tasks. Specifically they used VGG-16 model with a deep convolutional neural network to classify images, which showed the effectiveness of reusing pre-trained knowledge for classification and regression.

In addition, in Bengali language the classification of bengali book titles was explored by Rahman [26] and Ozsarfati [15]. In their research, they explored Bangla news

articles and headlines and classified them into genres. The research has a massive amount of dataset of about 376,226 Bengali news articles that was classified into distinct genres. They used different methods to minimise the complexities in this language. Traditional machine learning models such as Logistic Regression, Naive Bayes, Random Forest and Adaboost was used alongside neural network-driven approaches which features TF-IDF and Word2Vec-based feature extraction. However, what truly distinguished their work was the Long Short-Term Memory (LSTM)-based models which have accuracy value of 65.58% [15]. These models displayed a remarkable accuracy in the classification of both news article and news titles. In addition to that, their research has shown exceptional performance of neural network based approaches using low-dimensional Word2Vec features. Such research is an example of showcasing the power of advanced neural networks in linguistic analysis and is a motivation for us to use neural networks in Bengali language more.
Moreover, nowadays genre classification is not only limited to books. With the help of deep learning even music [16] and movies [10] [24] [6] can be classified in their respective genre. A recent research [35] has been conducted on video games using a multimodal deep learning framework to classify video game genre. In their research, they analysed both the cover images and the textual descriptions. They used a tremendous amount of dataset about 50,000 videos and classified into 15 unique genres. It has not only shown newer opportunities but also increased challenges. Motivated by such valuable research, another research [7] was conducted on movie genre classification focusing especially on the poster of the movies as the primary source of data. Their neural network was fine tuned to extract the visual attributes and identify the objects of the movie posters. As a result the research outperformed all the prior researches in the domain. This research is now used for different applications like movie recommendation systems.

# Chapter 3

# Methodology & Work Plan

## 3.1 Work Plan

The following work plan is the one that we have been using to complete our research efficiently. Following these procedures is what we have planned to follow, with the possibility of adding more if the needs of the study require it.

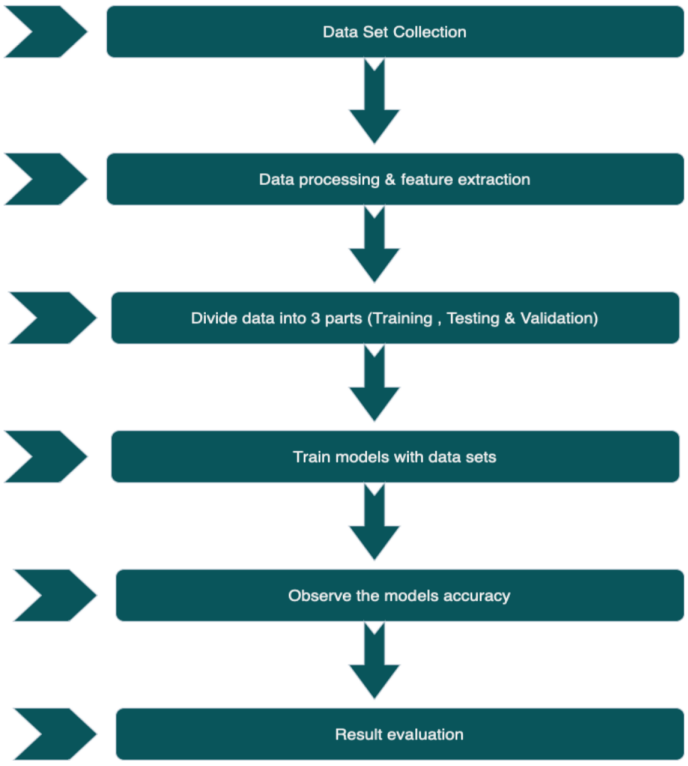


Figure 3.1: Work Plan

Fig 3.1 shows the work process that we have been following. Firstly, We planned to collect data from different resources like libraries and online websites. But, during our research, we have ended up collecting the Bangla book cover images with its corresponding Bengali book title and author names from various Bangla online e-book seller websites and images from google. We intended to divide the distinguished dataset into genres in order to make a specific dataset. After that, we wanted to

modify the data by resizing the image, increasing resolution and augmenting the data by flipping or rotating them. We also wanted to pre-process the Bengali texts by cleaning them, such as removing punctuation, non-Bengali words, English numbers and any extra non-word characters. Additionally, we wanted to try to find the root or base words by removing suffixes or prefixes or by applying other methods like stemming, lemmatizing etc to group words with similar meanings. On top of that, we intended to divide the data into 3 parts of which we planned to use 70% of it for training, 20% of it for testing and 10% of it for validation. Next, we wanted to train the models containing 3 types or modes (Image, text and image-text multimodal fusion) and experiment with different combinations of Image and text models. After that, For each model, we intended to validate and evaluate the result. Thus, based on the comparison, we intended to count the best outputs.

## 3.2 Methodology

### 3.2.1 Data collection

Putting together a dataset is an essential part of building and enhancing machine learning models. This includes gathering data that has unique features which our model aims to address. To be specific, we aimed to develop a model capable of accurately categorising various genres of Bengali books based solely on their cover designs, book titles and the author names. To acquire appropriate data, we have choosen wafilife [41], Baatighar [38], BD books [39], Rokomari [40], Kolkata Comics [43] etc as our primary sources. From the sources, we have collected 4531 book covers along with the book names and the author names. Since, the books are from multiple sub-genres, We merged them in 4 genres. For instance, We have collected different sub genres like Math, Science, Technology, Physics, Chemistry, Biology, Space science and astronomy, etc aiming to make a common genre called Math, science and technology. Again, For making the genre of Story and novels, we have acquired different sub genres like Contemporary story, Romantic story, Horror story, Story Compilation, etc. Furthermore, For the genre of Comics and cartoons, we have gathered sub genres like Comics, Children book, Picture's story, etc. For the genre History and politics, Politics: Governance, Political theory and philosophy, Global Politics, Revolution and rebellion, etc sub genres have been collected.

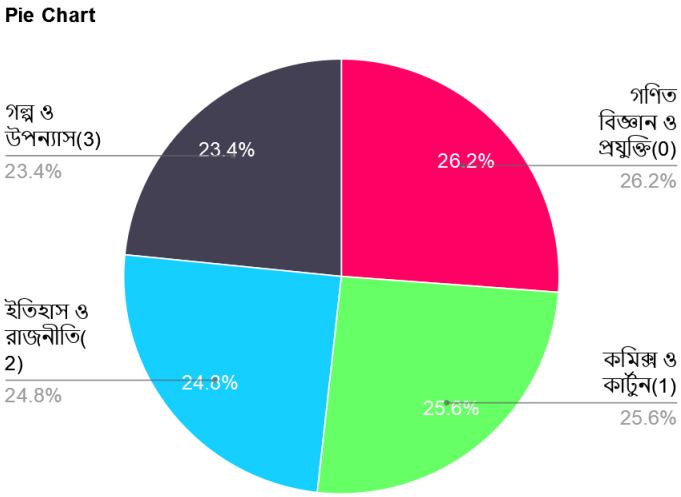


Figure 3.2: Genre Percentage

After collecting our dataset, We have observed different features and have seen some common correlations in our collected dataset. For instance, in the genre Maths, science and technology genre we have found that titles that feature numbers, scientific and mathematical words like science, maths, universe etc have appeared the most. In the genre Story and novels, titles with words like novel, poem, composition, etc have been spotted frequently. In the genre of History and politics, we have noticed titles which include words like movement, history, politics, etc have appeared higher in number. Furthermore, in the Comics and Cartoons genre, we have frequently detected titles containing words like comics, picture, and magic etc. This kind of approach using genre-specific titles along with visual features and statistical features has created a robust dataset that has helped create a strong correlation among the model and the dataset.

### 3.2.2 Data Pre-Processing

In the world of machine learning, making the data ready by cleaning, translating and structuring it into a format that is suitable for both model training and evaluation has been an inhabitable task. Data preparation has played a crucial role in enhancing the performance of machine learning algorithms. In our case, we have had mainly two types of data which are textual data and image based data. For images, we have first loaded the data and have resized all the images into the same shape (224,224,3). After that, we have added some data augmentation on the images randomly so that based on some specific feature the model does not get overfitted, is generalised better and is learning more effectively. For image augmentation, we have used techniques like horizontal_flip: True , shear_range: 0.2, zoom_range: 0.2, width_shift_range: 0.1, height_shift_range: 0.02 , rescaling: 1/255.0, rotation_range: 2 etc using the TensorFlow's Keras ImageDataGenerator. These amplification techniques ensures

that the model has learned from a diverse set of inputs resulting in improvement in its generalization. Here,

- **Horizontal flips** introduce the mirrored versions of the images, allowing the model to learn from various perspectives.
- **Rescale** normalizes pixel values by scaling them to the range [0, 1].
- **Shear** applies distortions that skew the image which adds variance in the data.
- **Zoom:** Random zooming provides both close up and distant versions of the image.
- **Width and height shift ranges** move the image both horizontally and vertically providing different position.
- **Rotation** Random rotation makes the model robust by changing the orientation of the image slightly.

Next, in order to pre-process the Bengali book title, we have cleaned the text by removing the punctuations, spaces, non Bengali characters etc. We have also gathered around 730 Bengali stopwords to remove those from the book titles. After that, we have used a pip package of Python called BnLemma, a lemmatizer to find the root word or base word for each bengali word. In addition to the standard lemmatizer, we have developed a custom list of Bengali words with specific root forms that the standard tool did not cover. This custom approach has been necessary because of the unique structure of some Bengali book titles, which might not be accurately processed by general-purpose lemmatizers. By tailoring the lemmatization to our specific dataset, we have ensured that domain-specific or irregularly formed words are correctly converted to their root forms, improving the quality of text pre-processing. So, we made a custom dictionary for Bengali lemmatization, which includes mappings from commonly encountered words or phrases to their corresponding root forms. This approach has allowed for more accurate text normalisation and significantly improved the model's ability to generalise from the training data. For instance, terms like:

- "হিস্ট্রি" are mapped to "ইতিহাস",
- "অঙ্ক" and its variations to "গণিত", and
- "কমিক্স" to "কমিকস".

The dictionary includes approximately 100 word mappings based on the Bengali word's similar meanings and the common usage patterns in Bengali book titles or book names. The full list of word mappings is available in Appendix [A]. After cleaning and lemmatizing these Bengali book titles, we have tokenized them using the TensorFlow's Keras Text Preprocessing Tokenizer. We have then applied pad_sequences to ensure that all book titles have had uniform length before feeding the text into the model for training.

### 3.2.3 Data Labelling

Using supervised machine learning, we have labeled data instances by assigning them meaningful annotations. This labelling process is very important as it provides necessary information of the machine learning model to recognize patterns and make predictions. Data labelling is a key step in creating datasets used for training and evaluating machine learning models, where the labels signify the specific outcomes or classes that the model aims to predict. So, here, we have collected some sub genres like Math, Science and technology, Physics, Chemistry, Biology, astronomy, etc and have merged them in one genre called Math, science and technology(class 0). Similarly, for the genre Comics and cartoons(class 1), we have merged sub genres like Comics, Children book, Picture's story, etc. Again, For the genre Story and novels(class 3), we have merged different sub genres like Contemporary story, Romantic story, Horror story, Story Compilation, etc. And for the genre History and politics(class 2), we have merged Politics: Governance, Political theory and philosophy, Global Politics, Revolution and rebellion, etc sub genres. Thus, our 4 genres(Math, science and technology, Story and novels, Comics and cartoons, History and politics) have been made. Thus, we have incorporated sub genres book covers and Bengali book titles in four genre categories and labeled our dataset into these four categories using categorical class_mode. We have applied one-hot encoding to represent the genre labels as categorical values and have labeled the genres as the following figure:


Figure 3.3: Data Labelling Example

### 3.2.4 Custom MultiModalDataGenerator:

In this case, we needed to load the Bengali book titles along with their corresponding book cover images for which, we have implemented a custom MultiModalDataGenerator class using the TensorFlow's Keras Sequence. This custom data generator handles the task of loading and labelling both text and image data simultaneously for the multimodal system. Within the class, we have defined key functions such as ___len___, ___getitem___, on_epoch_end and ___init___ etc. Below is a high-level algorithm summarising the key steps involved in the multimodal data generator:

**Input Parameters:**

- `df:` DataFrame containing text and image file paths.
- `labels:` DataFrame containing labels(one-hot encoded).
- `tokenizer:` Tokenizer for text preprocessing.

- `img_size`: Desired image size(default: 224,224,3).
- `batch_size`: Number of samples per batch(default: 16).
- `max_length`: Maximum length of padded text sequences(default: 8).
- `augment`: Boolean to apply image augmentation(default: True).
- `shuffle`: Boolean to shuffle data at the end of each epoch(default: True).

**Key Steps:**

1. Initialize the Data Generator:
   (a) Set up the class attributes like `df`, `labels`, `tokenizer`, etc.
   (b) Initialize an ImageDataGenerator from TensorFlow's Keras Preprocessing Image for augmenting images if `augment`=True.
   (c) Shuffle the data after each epoch if `shuffle`=True.
2. Calculate Total Batches:
   (a) Compute the total number of batches by dividing the number of samples in len of `df` by `batch_size` and rounding it up to ensure all data is included.
3. Generate Batches: For each batch using loop:
   (a) Select Batch Indices: Select the indices of samples for the current batch based on `batch_size`.
   (b) Load Text and Images:
      i. Text: Extract and store the text corresponding to the current batch.
      ii. Images: Load and resize each image to img_size, normalize pixel values to a range of [0, 1], and apply random image augmentation if `augment`=True.
   (c) Prepare Labels: Fetch the corresponding labels for the batch.
   (d) Preprocess Text: Tokenize and pad the text sequences to a fixed length self.max_length.
4. Return Batch
   (a) Returns a batch of preprocessed text sequences, images and corresponding labels
5. End of Epoch:
   (a) At the end of each epoch, shuffle the data sequence if `shuffle`=True to ensure randomness in the next iteration.

**Function Descriptions:**

- ___init___: Defines and initializes the necessary parameters for the class.
- ___len___: Defines the number of batches per epoch.
- ___getitem___: Retrieves a batch of data, returning both the image and text for a given index.
- on_epoch_end: Updates the index list after each epoch, ensuring that the data is shuffled.

By using the custom MultiModalDataGenerator we have loaded our dataset into four genres and generated our train, test and validation data generator.

### 3.2.5 Data Splitting

In machine learning, an important step includes dividing a dataset into separate subsets for training, validation and testing which is a process commonly known as data splitting. This practice contributes significantly to a more generalised and authentic evaluation of a machine learning model's performance. When a model undergoes training on the majority of our dataset, it allows to discover or learn about patterns and correlations within these different portions of data. During this training, to monitor how the model has been learning and performing on unseen data side by side, a smaller sample has been employed which is known as validation split.

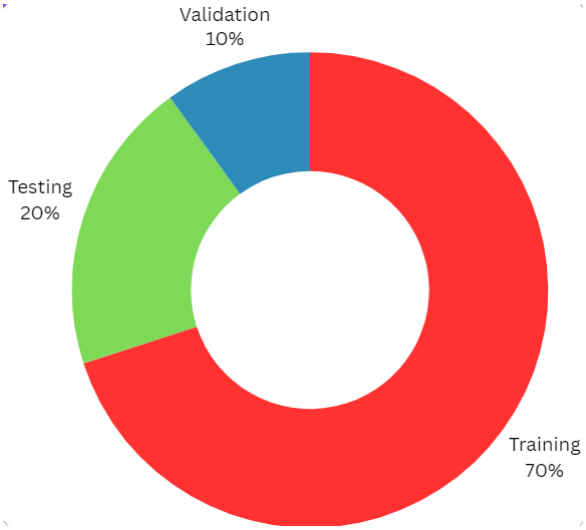


Figure 3.4: Data Splitting

This validation data set has also been very helpful in terms of preventing the overfitting of the model by continuously monitoring the val_loss or val_accuracy. Another fully independent subset is reserved exclusively for the final evaluation of the model's performance which is called the test split. This test set remains unseen for the model during both the training and validation, ensuring an unbiased performance assessment of the model's effectiveness and its ability to generalise to new data. Here is a table of our total data splinting:

| Data Segments | Percentage | Total Images + Book Titles | Parameters | Value |
|---|---|---|---|---|
| Training | 70% of the total data | 3172 | Target Size<br><br>Class Mode<br>Subset<br>Batch Size | (224,224,3)<br><br>Categorical<br>Training<br>16 |
| Validation | 10% of Training data | 454 | Target Size<br><br>Class Mode<br>Subset<br>Batch Size | (224,224,3)<br><br>Categorical<br>Validation<br>16 |
| Testing | 20% of the total data | 906 | Target Size<br><br>Class Mode<br>Subset<br>Batch Size | (224,224,3)<br><br>Categorical<br>Testing<br>16 |

Table 3.1: Bangla Book Genre Classification using Multimodal Network

### 3.2.6 Proposed Methodology

In this research, we have been trying to achieve highly accurate Bengali book genre classification by using a multimodal system. To achieve this, we have focused on developing an efficient model that requires less data or minimal effort for which we have only used Bengali book titles and Bangla book cover images. So, our model had to deal with both Bengali text and book cover images. After doing the pre-processing on both image and Bengali text data, we have split the data into 3 segments(train,validation and test).Then we loaded these 3 segments(train, validation and test) one by one using our custom MultiModalDataGenerator. In our proposed methodology we have taken one Bengali tokenized and padded text sentence array with its corresponding book cover image as input. To handle these two different types of data, we have used One convolutional neural network(CNN) and a Long short-term memory (LSTM) model in our multimodal system to get the prediction.

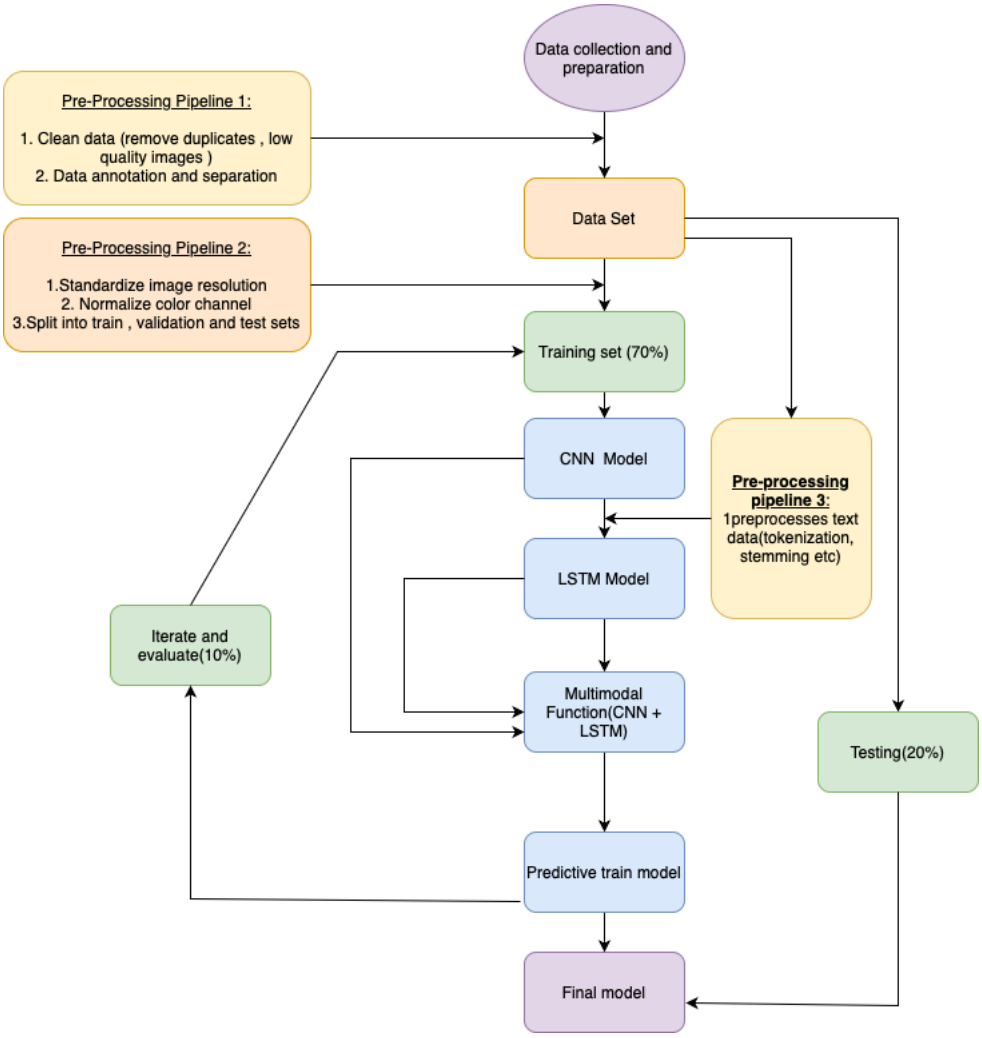


Figure 3.5: Proposed Methodology

### Convolutional Neural Network (CNN)

The CNN component focuses on extracting relevant features from the book cover images, allowing the model to understand visual patterns associated with different genres. In our proposed model we have mainly used VGG16 and Resnet50 for the CNN part. We have set the input image size to 224x224x3, and have used a batch size of 16. We have unfreezed the entire VGG16 and Resnet50 architecture and trained it from scratch without using pre-trained weights from the ImageNet dataset. Rather, we have used the VGG16 and Resnet50 architecture to extract the complex details from different Bangal genres book covers and train this untutored CNN architecture on our collected dataset. This has allowed the model to learn genre-specific features from Bangla book covers. The CNN has had the following components:

- **Image_Input:** This layer takes the input images of size 224,224,3 and we named it image_input.
- **Unfrozen_VGG16:** Applies the VGG16 architecture without including the pretrained weights from imagenet dataset and without including the top layers, with model.trainable = True.
- **Unfrozen_Resnet50:** Uses the Resnet50 architecture without including the top layers and also model.trainable = True.
- **Flatten Layer:** Flattens the convolutional layers output into a one-dimensional array.
- **GlobalAveragePooling2D:** Computes the average of each feature map, reducing the spatial dimensions and retaining essential information for classification. In VGG16 and REsnet50 architectures, it might benefit from using GlobalAveragePooling2D instead of Flatten, especially if we are working with larger image sizes.

### Long Short-Term Memory (LSTM)

This LSTM component of our proposed multimodal model has focused on processing Bengali textual information from the book titles/names. This LSTM has aimed to capture sequential patterns and has tried to discover contextual relations associated with different genres based on the Bengali book's name. We have also used the BiLSTM for handling these Bengali texts as this is a more advanced technique to detect patterns or correlations among the texts. So, we have mainly used a Sequential model to build the LSTM part. Here we also have one text_input and the input consists of tokenized and padded text which has been passed through an Embedding Layer, followed by two LSTM or BiLSTM layers. Each LSTM or BiLSTM layer has had the same units, followed by a Dropout layer with a dropout rate to prevent overfitting. The final Dense layer uses L2 regularisation for better model generalization. So, the structure of the LSTM has been as follows:

- **Text_Input:** This layer receives the input as an array of padded text sentences with a shape of max_length, and we have named it text_input.

- **Embedding Layer:** Converts text data into dense vectors to capture semantic relationships with vocab_size of and input_length of max_length.
- **LSTM Laye:** Two LSTM layers, each with same units of neurons , with the first LSTM layer returning sequences for the second.
- **BiLSTM Layer:** Two BiLSTM layers, each the same units of neurons where the first layer returns sequences for the second one .This RNN can capture sequential patterns in both forward and backward directions, allowing it to better understand the context and relationships within the text.
- **Dense Layer:** A fully connected layer where each sample's output from this layer is a one-dimensional array with L2 regularization for final classification based on Bengali book titles.
- **Dropout Layer:** Dropout layers randomly deactivate or drop a fraction of neurons during training to prevent the overfitting and boosting the generalization of the model.

**Multimodal Fusion Layer**

From both CNN and LSTM components, the model has gotten two 1D output features which is combined in a fusion layer to create a joint representation. For this, we have concatenated both CNN and LSTM last layer's output. Then we have used a dense layer having 4 units and activation of softmax in this layer to get the final prediction. Finally, we have made the multimodal layer where we have taken the CNN and LSTM model's input with corresponding input types and shapes to produce the final prediction. On this multimodal system, we have compiled the model with Adam optimizer with a very low learning rate of 0.0001. Using a low learning rate like 0.0001 has been a good strategy for fine-tuning models such as VGG16 or Resnet50, especially when the dataset is medium-sized. However, it is not necessary but if the model has been learning very fast, a lower learning rate helps prevent overshooting the optimal point and generalize the learning process during training. So, It might be better to say that the low learning rate has helped fine-tune the model and avoid overfitting. To address or handle this overfitting issue we have also used the EarlyStopping method with patience of 4 and a learning scheduler with patience of 4 which have been monitoring the val_loss. For calculating the overall model loss, we have used the categorical_crossentropy as loss function as our data has been labeled in one hot encoding. Finally, we have used the accuracy metrics to measure and monitor the model's performance. With this, we have also used this additional precision, recall and F1-score to get a more comprehensive understanding of your model's performance.

# Chapter 4

# Implementation & Results

## 4.1 Implementation and Performance Evaluation of the Proposed Model

The aim of performance analysis is to evaluate how well a system or process functions. It involves examining the metrics and outcomes to understand efficiency, identify areas for improvement and make informed decisions. This practice is essential across various domains as it helps optimize performance and achieve desired results. In our proposed model, we have used VGG16 by unfreezing all layers and have trained it from scratch with our collected training dataset. The sequential LSTM model has an embedding layer with vocab_size of 400 and max-length of 8, followed by two LSTM layers. Each LSTM layer has 32 units, followed by a dropout layer with a dropout rate of 0.4 to prevent over-fitting. Also, the final dense layer has used L2 regularisation with a strength of 0.01 for better model generalization. We have then concatenated the flattened 1D array from the CNN with the dense 1D array from the sequential LSTM model. Here is our proposed model summary:

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| image_input (InputLayer) | [(None, 224, 224, 3 )] | 0 | [] |
| text_input (InputLayer) | [(None, 8)] | 0 | [] |
| vgg16 (Functional) | (None, 7, 7, 512) | 14714688 | ['image_input[0][0]'] |
| sequential_3 (Sequential) | (None, 32) | 426400 | ['text_input[0][0]'] |
| flatten_1 (Flatten) | (None, 25088) | 0 | ['vgg16[0][0]'] |
| concatenate_3 (Concatenate) | (None, 25120) | 0 | ['sequential_3[0][0]', 'flatten_1[0][0]'] |
| dense_10 (Dense) | (None, 128) | 3215488 | ['concatenate_3[0][0]'] |
| dropout_11 (Dropout) | (None, 128) | 0 | ['dense_10[0][0]'] |
| dense_11 (Dense) | (None, 4) | 516 | ['dropout_11[0][0]'] |

Total params: 18,357,092
Trainable params: 18,357,092
Non-trainable params: 0

Figure 4.1: Proposed model summary

As we have a multimodal system with VGG16 and LSTM for classifying the Bangla book genres, we have first loaded both text and image data with our custom data-generator. We started our model training by splitting the data into training and validation before feeding it into the model. The implementation begins with training and evaluating the data epoch by epoch while using our multimodal(VGG16 + LSTM) model. We have compiled this model with Adam optimizer. Which have a learning rate of 0.001 and categorical_crossentropy as its loss function. During our model training we used 16 epochs and batch size of 16 images, with tokenized arrays of Bengali book titles. The images had a shape of (224,224,3) and text_input consisted of padded tokenized arrays size of 8. It took some time to train and validate the model. Afterward, we evaluated our model with the testing data and got 94% accuracy on training data, 85% maximum on validation data and 87% on testing data. We have shared our results on the proposed model on some other metrics below:
The following table summarises the performance of the implemented multimodal(VGG16 + LSTM) model on test data:

| **Parameter** | **Math, Science and Technology** | **Comics and Cartoons** | **History and Politics** | **Story and Novels** | **Macro Avg** | **Weighted Avg** |
|---|---|---|---|---|---|---|
| Recall | 87% | 92% | 86% | 82% | 87% | 87% |
| f1-score | 88% | 91% | 86% | 81% | 87% | 87% |
| Precision | 89% | 90% | 87% | 81% | 87% | 87% |
| Support | 24.88% | 26.78% | 24.72% | 23.62% | | |

Table 4.1: Performance metrics of proposed model

These metrics used for evaluation are recall, precision, f1-score, and support, which offer a comprehensive understanding of the model's effectiveness. The recall was used to measure how well the model identified all relevant positive instances of a genre. In this case, recall values range from 82% to 92%, indicating that the model has been fairly accurate in predicting each genre. The "Comics and Cartoon" genre showed highest recall at 92%. The precision indicated how many of the instances predicted as a specific genre were correct. The precision values range from 81% to 90%, showing that the model has been consistently good at minimizing false positives. The harmonic mean of precision and recall, provides a single metric that balances both aspects which is called f1-score. This has a range from 81% to 91%, with the "Comics and Cartoon" genre showing the best overall performance at 91%. This score has been crucial for evaluating imbalanced datasets or cases where precision and recall have been of equal importance. Lastly, support refers to the proportion of instances for each genre in the test set. The values range from 23.62% to 26.78%. This distribution suggests a fairly balanced representation of genres, which has ensured that the evaluation has not been biased towards any specific genre. Overall, these metrics have provided a comprehensive view of how well the model has generalized to different genres, with a high performance across the board, though slightly lower scores have been observed for the "Stories and Fiction" genre. Here, we have also plotted the performance in terms of training and validation to evaluate through a visual representation.

## Accuracy

This graph represents the history of training and validation process accuracy while the model is learning from the training data. We know that the validation accuracy indicates the actual accuracy of the model. We have seen here that our proposed model has been learning very well on the training dataset as well as performing well on the validation dataset. However, after the 8 or 9 epochs, the validation accuracy stopped increasing while the training accuracy continued to rise. Which was the indication of slight overfitting. Although, we have run the model for 16 epochs, we have also used EarlyStopping to prevent overfitting. Which successfully stopped the training after 12 epochs.

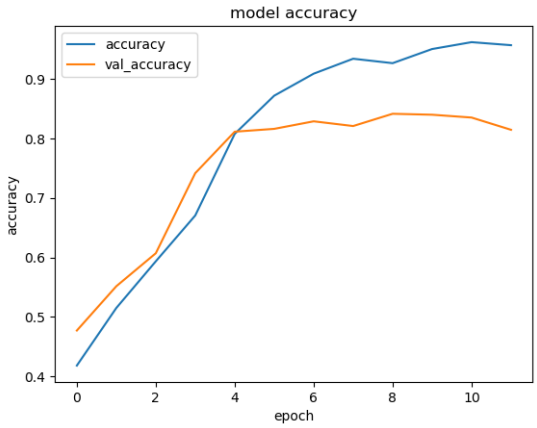


Figure 4.2: Accuracy of the proposed model

## Loss

In this loss function graph we have observed the same pattern as the accuracy where, after 8 or 9 epochs, the validation loss has almost stopped decreasing while the training loss has continued to fall, which indicates the same slight overfitting. Here EarlyStopping has also successfully stopped the training after 12 epochs and has prevented overfitting. So here we have total error of our proposed model throughout the 12 epochs plotted in this graph. The loss graph has revealed a significant flaw that has explained why the data could not be validated across epochs, affecting the accuracy of the model. As shown in this graph, the text and image data of the training dataset has a loss of 24.26%, while the validation dataset has a loss of 62.86% maximum. Despite this, the relatively small difference in loss has indicated decent model generalization.

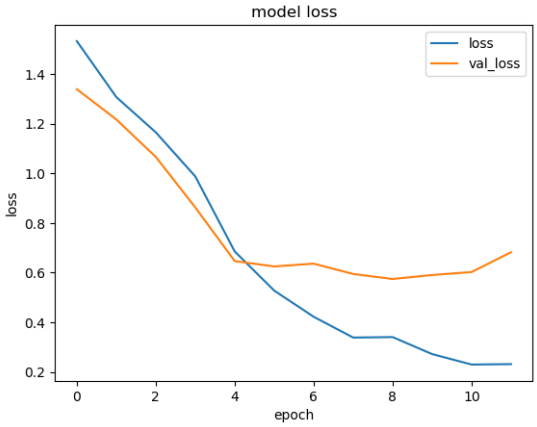


Figure 4.3: Loss of the proposed model

In both accuracy and loss graphs we have seen that when this model started learning, it did better on validation compared to the training curves. This might have happened due to having data augmentation on the training dataset but we have not applied data augmentation on our validation data set which might have made it a bit more easier to identify. Moreover, to prevent overfitting we used regularizer and dropout which may also be a reason for this.

## Confusion matrix

We have also used the confusion matrix to extend the performance evaluation of our multimodal system. We have had multi-class classification problem, where there are total four classes. In the case of multi-class problem, the confusion matrix has become a square matrix with dimensions equal to the number of four classes. From this, we have seen that our model has performed well across all four genre classification, though it has slightly underperformed in the stories and fiction genres, possibly due to more complex and abstract features in the cover images, as in our collected dataset many of them have been based on abstract designs.

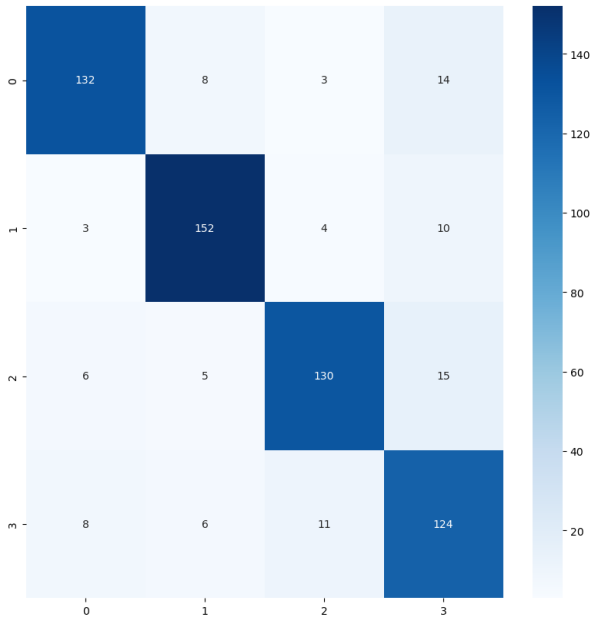


Figure 4.4: Confusion Matrix of the proposed model

## 4.2 Implementation of CNN and LSTM for Book Cover Image and Bengali Title Text Analysis

### BiLSTM

In our total collected 4531 book names with their corresponding book author names, we have identified a total of 4 genres. To predict the corresponding genres with less data, we have only taken the Bengali book titles and put them through a Bi-directional LSTM whose summary follows:

| Layer (type) | Output Shape | Param # |
|---|---|---|
| embedding_2 (Embedding) | (None, 8, 300) | 960000 |
| bidirectional (Bidirectional) | (None, 8, 64) | 85248 |
| dropout_4 (Dropout) | (None, 8, 64) | 0 |
| bidirectional_1 (Bidirectional) | (None, 64) | 24832 |
| dense_4 (Dense) | (None, 32) | 2080 |
| dropout_5 (Dropout) | (None, 32) | 0 |
| dense_5 (Dense) | (None, 4) | 132 |

Total params: 1072292 (4.09 MB)
Trainable params: 1072292 (4.09 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 4.5: BiLSTM model summary

We have trained this sequential BiLSTM model for 60 epochs with batch size of 32 and have found out this accuracy and loss graph:

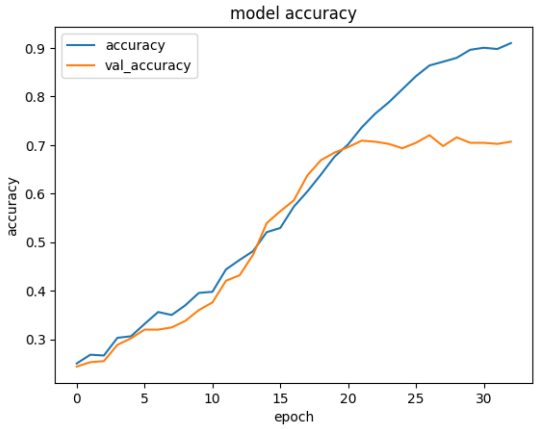


Figure 4.6: Accuracy of BiLSTM model

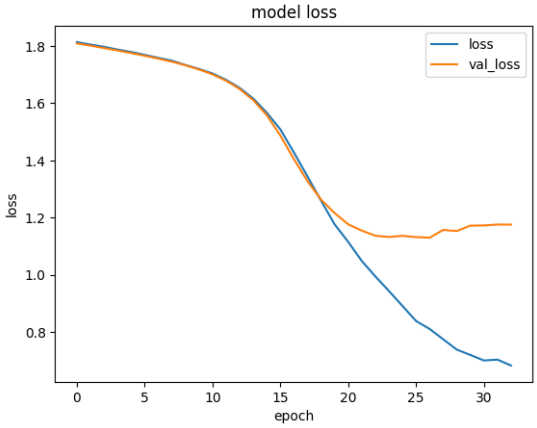


Figure 4.7: Loss of BiLSTM model

From this graph we have observed that after 30 epochs the model has been getting overfitted on the training data as a result due to the early stopping condition on val_loss with a patience of 6 it has stopped the training to prevent overfitting. After that we evaluated the model on test data with different other metrics, here are the results below:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| class 0 | 0.88 | 0.77 | 0.82 | 230 |
| class 1 | 0.59 | 0.61 | 0.60 | 223 |
| class 2 | 0.82 | 0.81 | 0.81 | 244 |
| class 3 | 0.54 | 0.60 | 0.57 | 196 |
| accuracy | | | 0.70 | 893 |
| macro avg | 0.71 | 0.70 | 0.70 | 893 |
| weighted avg | 0.72 | 0.70 | 0.71 | 893 |

Figure 4.8: Performance metrics of BiLSTM model

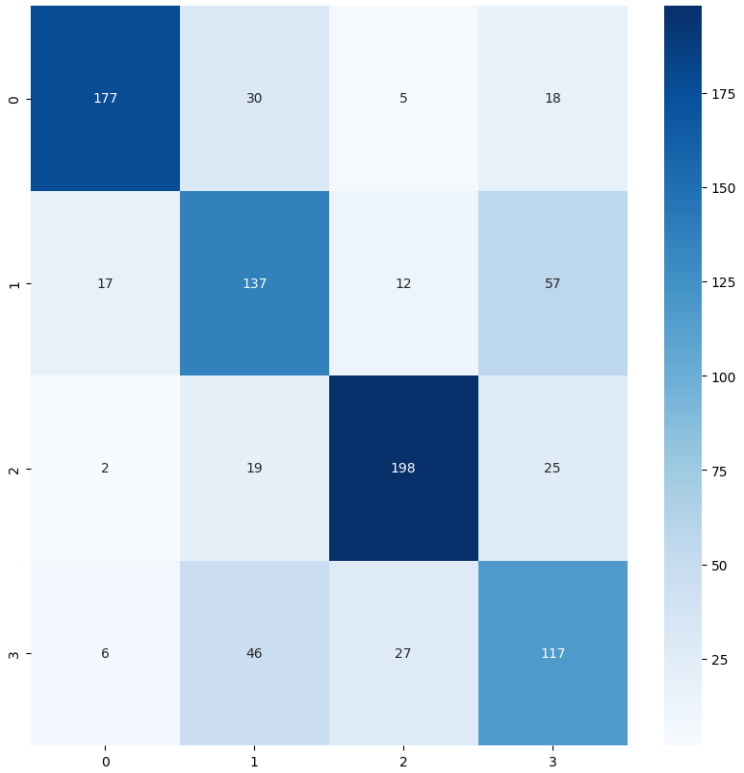


Figure 4.9: Confusion Matrix of the BiLSTM model

From the evaluation results, we can see that the BiLSTM model is biased towards certain classes and performs poorly on others, such as class 3, which represents Story and Fiction. This could be because while BiLSTM captures more complex features by moving both forward and backward, Bengali book titles are often much shorter(typically 2 to 4 words).

### LSTM

We have collected 4531 book names and corresponding book author names which were split into four classes, which are history and politics, Comics and Cartoons, Science, Math and technology and lastly Story and Fiction. In this scenario, we have solely taken Bengali book titles and the model we used is a simple Sequential model with one Embedding layer with a vocabulary size of 300 and an input of max_length of 8, followed by two LSTM layers with 32 units each, followed by a dropout layer with a dropout rate of 0.3, a dense layer with a l2 regularizer with a

strength of 0.01, and finally a Dense layer with 4 units because we have four classes and softmax activation.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| embedding (Embedding) | (None, 8, 300) | 960000 |
| lstm (LSTM) | (None, 8, 32) | 42624 |
| dropout (Dropout) | (None, 8, 32) | 0 |
| lstm_1 (LSTM) | (None, 32) | 8320 |
| dense (Dense) | (None, 32) | 1056 |
| dropout_1 (Dropout) | (None, 32) | 0 |
| dense_1 (Dense) | (None, 4) | 132 |

Total params: 1012132 (3.86 MB)
Trainable params: 1012132 (3.86 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 4.10: LSTM model summary

To achieve good genre prediction with minimal effort, we only use the Bengali book names as input for this model after necessary preprocessing, tokenization and sentence padding. We train this model for 66 epochs with a batch size of 32. The model we used for this is the RNN model which was found to be 87%, 74% and 74.8% at train, validation and test dataset at the end. The data we got from this are shown in the below:

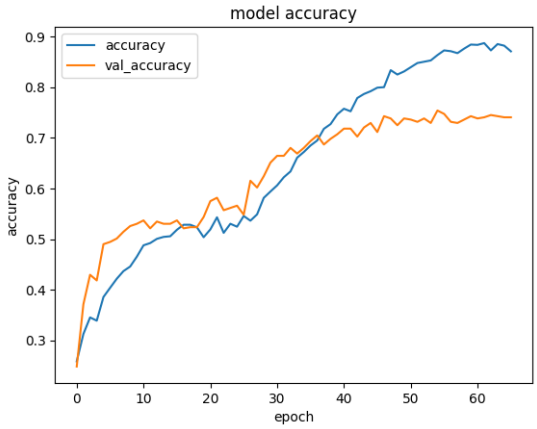


Figure 4.11: Accuracy of LSTM model

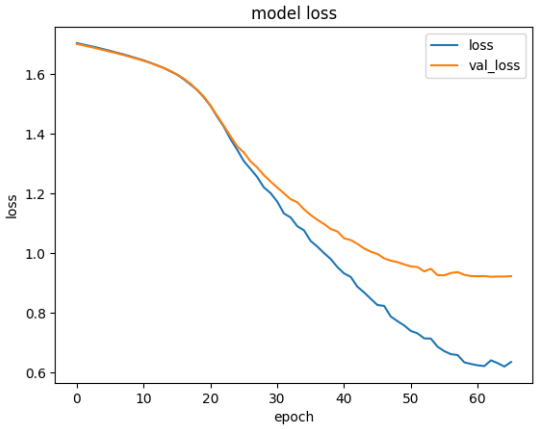


Figure 4.12: Loss of LSTM model

From this accuracy and loss graph we can see that the model is learning well compared to the Previous BiLSTM.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| class 0 | 0.81 | 0.76 | 0.78 | 236 |
| class 1 | 0.81 | 0.64 | 0.71 | 226 |
| class 2 | 0.86 | 0.81 | 0.83 | 235 |
| class 3 | 0.56 | 0.78 | 0.65 | 196 |
| accuracy | | | 0.75 | 893 |
| macro avg | 0.76 | 0.75 | 0.75 | 893 |
| weighted avg | 0.77 | 0.75 | 0.75 | 893 |

Figure 4.13: Performance metrics of LSTM model

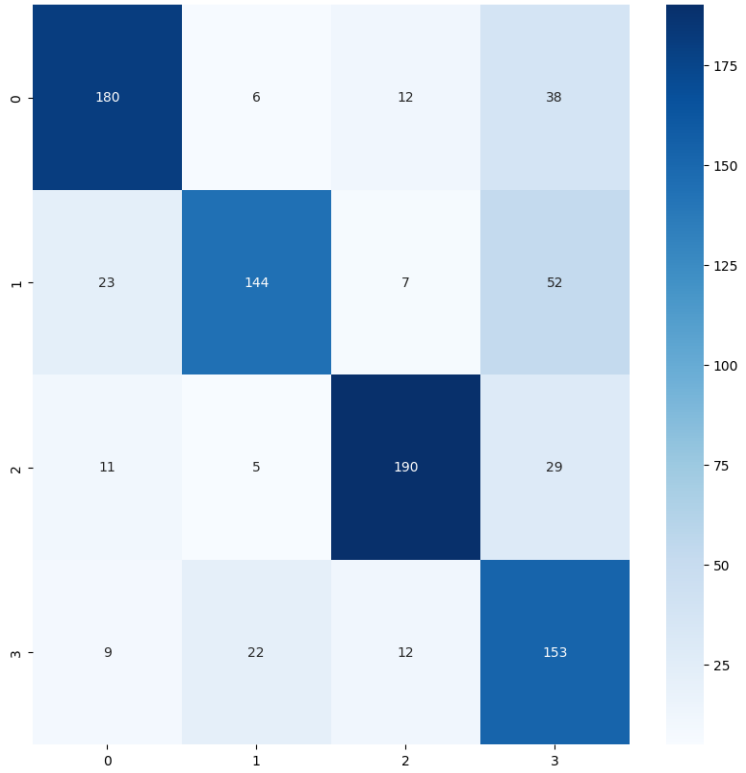


Figure 4.14: Confusion Matrix of the LSTM model

After evaluating this LSTM model we can observe this shows a very promising performance over the BiLSTM model which helps us in developing our proposed model.

### VGG16

For this CNN model, we are using VGG16 architecture, for which we unfreeze the whole VGG16 architecture and then from scratch, we train it. For this method, we use a sequential model where we first take the VGG16 with input_shape = (224, 224, 3), include_top = False and make the model trainable = True. Then following that we use a flatten layer followed by 2 Dense layer and Dropout layer with dropout rate of 0.4. The models summary looks like this:

| Layer (type) | Output Shape | Param # |
|---|---|---|
| vgg16 (Functional) | (None, 7, 7, 512) | 14714688 |
| flatten (Flatten) | (None, 25088) | 0 |
| dense (Dense) | (None, 512) | 12845568 |
| dropout (Dropout) | (None, 512) | 0 |
| dense_1 (Dense) | (None, 4) | 2052 |

Total params: 27,562,308
Trainable params: 27,562,308
Non-trainable params: 0

Figure 4.15: VGG16 model summary

After that, we trained the model for 16 epochs but as the model was getting overfitted and to prevent it, the model stopped learning after 11 epochs with 87% percent of train accuracy and 70% percent of validation accuracy.

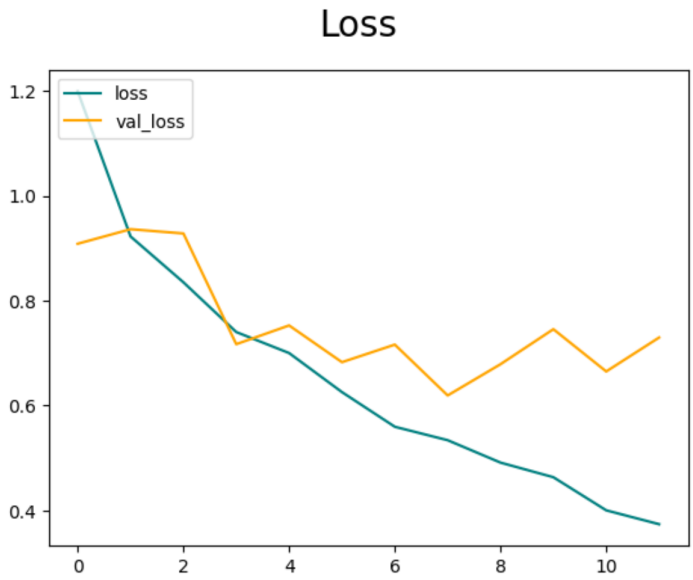


Figure 4.16: Loss of VGG16 model

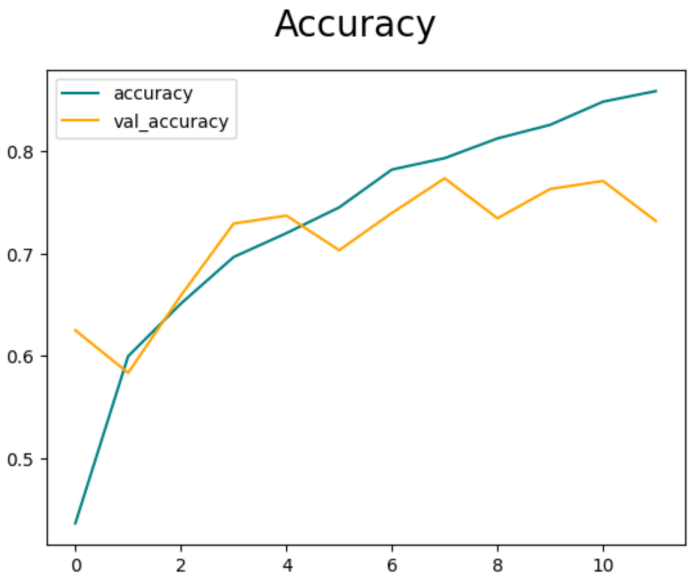


Figure 4.17: Accuracy of VGG16 model

Moreover, we evaluate the model on unseen test dataset and these results :

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.74 | 0.65 | 0.69 | 187 |
| 1 | 0.65 | 0.77 | 0.70 | 195 |
| 2 | 0.87 | 0.88 | 0.87 | 209 |
| 3 | 0.71 | 0.65 | 0.68 | 198 |
| accuracy | | | 0.74 | 789 |
| macro avg | 0.74 | 0.74 | 0.74 | 789 |
| weighted avg | 0.74 | 0.74 | 0.74 | 789 |

Figure 4.18: Performance metrics of VGG16 model

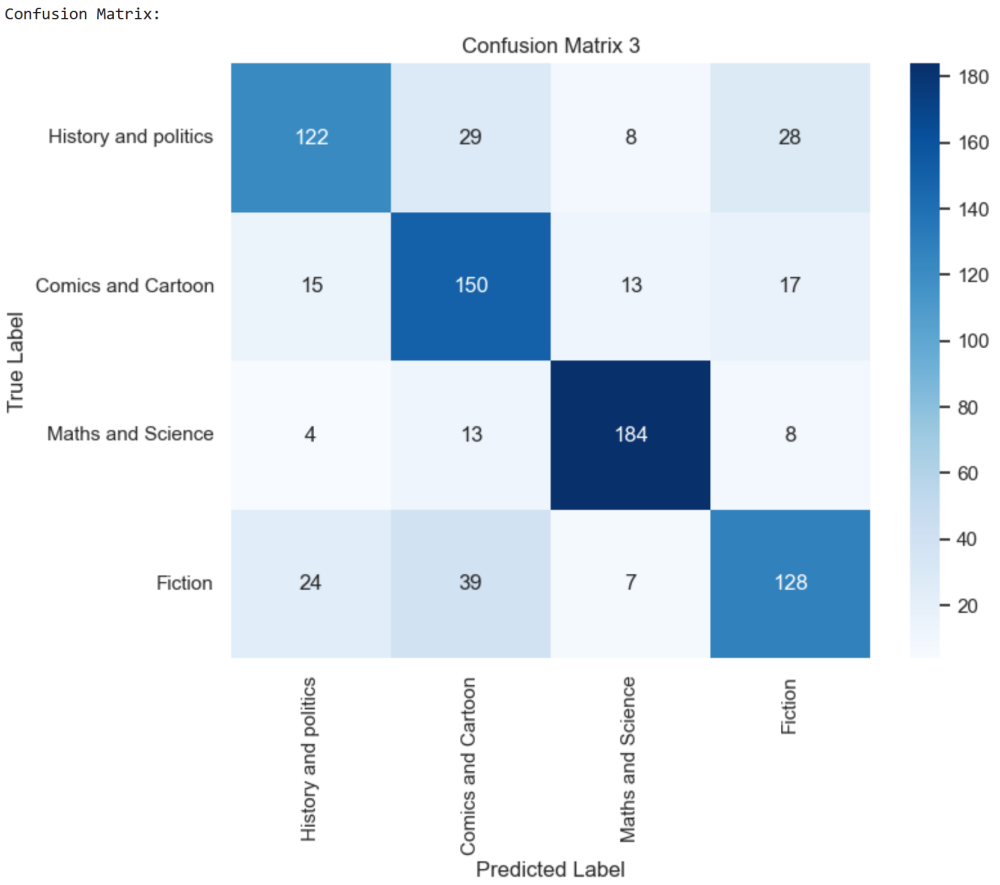


Figure 4.19: Confusion Matrix of the VGG16 model

From this, we can see that the model is learning, but it is slightly overfitting. As it learns we gain some ideas and this indicates how we can merge both CNN and LSTM to develop a better model with more accurate predictions for Bengali book genres.

## 4.3 Implementation of Alternative CNN and LSTM Combinations in a Multimodal System

The efficiency of a model can only be evaluated through its performance evaluation in comparison to some other models' performance. Also, the research works can be extended further according to the limitations and shortcomings of a proposed model through further investigation. So, for this, we have implemented some other combinations of CNN and LSTM models to compare them with our proposed one. Here are some other:

### VGG16+BiLSTM

In this multimodal system, we have implemented the VGG16 by unfreezing the all layers and training it from scratch with our collected train dataset, as well for the BiLSTM we make another sequential model just like used in our proposed model with one Embedding Layer, followed by two BiLSTM layers and each has 32 units, followed by a Dropout layer with a dropout rate of 0.4 to prevent overfitting. The final Dense layer uses L2 regularisation with a strength of 0.1. We trained this model for 16 epochs with the same image (224,224,3) and text (8 length array) shapes. The summary of this model is shown below:

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| image_input (InputLayer) | [(None, 224, 224, 3)] | 0 | [] |
| text_input (InputLayer) | [(None, 8)] | 0 | [] |
| vgg16 (Functional) | (None, 7, 7, 512) | 14714688 | ['image_input[0][0]'] |
| sequential (Sequential) | (None, 32) | 460960 | ['text_input[0][0]'] |
| flatten (Flatten) | (None, 25088) | 0 | ['vgg16[0][0]'] |
| concatenate (Concatenate) | (None, 25120) | 0 | ['sequential[0][0]', 'flatten[0][0]'] |
| dense_1 (Dense) | (None, 128) | 3215488 | ['concatenate[0][0]'] |
| dropout_2 (Dropout) | (None, 128) | 0 | ['dense_1[0][0]'] |
| dense_2 (Dense) | (None, 4) | 516 | ['dropout_2[0][0]'] |

Total params: 18,391,652
Trainable params: 18,391,652
Non-trainable params: 0

Figure 4.20: VGG16+BiLSTM model summary

For this model, we have 92%, 76%, and 77% accuracy on the training, validation, and test datasets, respectively. Here is the accuracy and loss graph from where we can see that the model is getting overfitted over the last epochs:

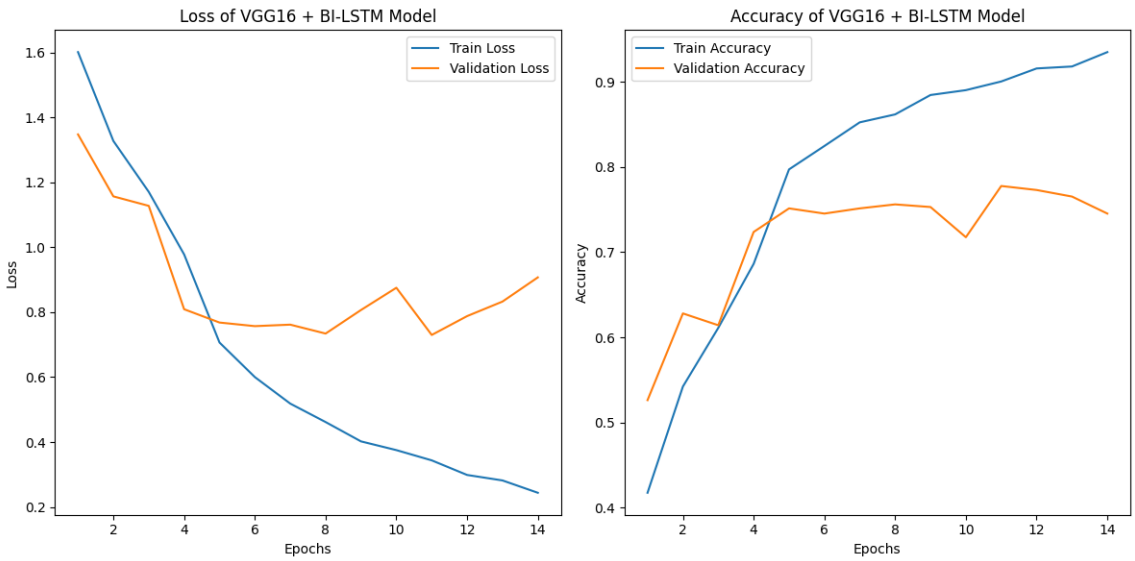


Figure 4.21: Accuracy and Loss of VGG16+BiLSTM model

The confusion matrix and the value for precision, recall and F1 score for all four classifications are also given below.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| class 0 | 0.70 | 0.77 | 0.73 | 181 |
| class 1 | 0.85 | 0.88 | 0.86 | 152 |
| class 2 | 0.81 | 0.74 | 0.77 | 164 |
| class 3 | 0.72 | 0.68 | 0.70 | 150 |
| accuracy | | | 0.77 | 647 |
| macro avg | 0.77 | 0.77 | 0.77 | 647 |
| weighted avg | 0.77 | 0.77 | 0.77 | 647 |

Figure 4.22: Performance metrics of VGG16+BiLSTM model

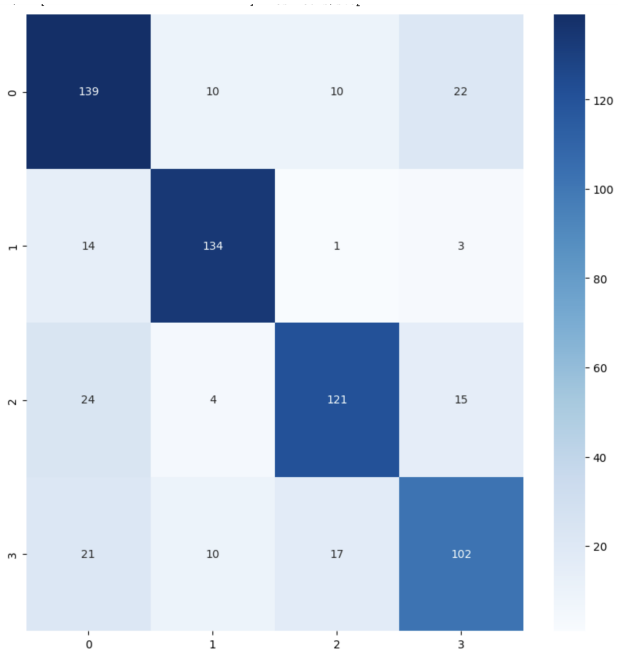


Figure 4.23: Confusion Matrix of the VGG16+BiLSTM model

From this, we have observed that this model has not done that much bad. But it has overfitted over some last epochs. However, this has provided us with valuable understanding for fine-tuning the model Which has given us instincts that we can develop an improved proposed model with higher accuracy.

## Resnet50+BiLSTM

In this combination of multimodal, we have used Resnet50 for the CNN part and a bidirectional LSTM (BiLSTM) for the text part. In this case, we have also unfreezed the Resnet50 architecture rather than using the pretrained weights of imagenet dataset. We have used our collected Train dataset to train the architecture. Then, we used GlobalAveragePooling2D in this CNN layer and have merged or concatenated it with a BiLSTM sequential model. The summary of the model is as follows:

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| image_input (InputLayer) | [(None, 224, 224, 3)] | 0 | [] |
| text_input (InputLayer) | [(None, 8)] | 0 | [] |
| resnet50 (Functional) | (None, 7, 7, 2048) | 23587712 | ['image_input[0][0]'] |
| sequential (Sequential) | (None, 32) | 460960 | ['text_input[0][0]'] |
| global_average_pooling2d (GlobalAveragePooling2D) | (None, 2048) | 0 | ['resnet50[0][0]'] |
| concatenate (Concatenate) | (None, 2080) | 0 | ['sequential[0][0]', 'global_average_pooling2d[0][0]'] |
| dense_1 (Dense) | (None, 128) | 266368 | ['concatenate[0][0]'] |
| dropout_2 (Dropout) | (None, 128) | 0 | ['dense_1[0][0]'] |
| dense_2 (Dense) | (None, 4) | 516 | ['dropout_2[0][0]'] |

...
Total params: 24315556 (92.76 MB)
Trainable params: 24262436 (92.55 MB)
Non-trainable params: 53120 (207.50 KB)

Figure 4.24: Resnet50+BiLSTM model summary

We have also trained this model for 16 epochs with the same image(224,224,3) and text(8 length array) shapes. But, due to the bad performance in validation, it stops earlier. Here is the Accuracy and Loss graph of this model:

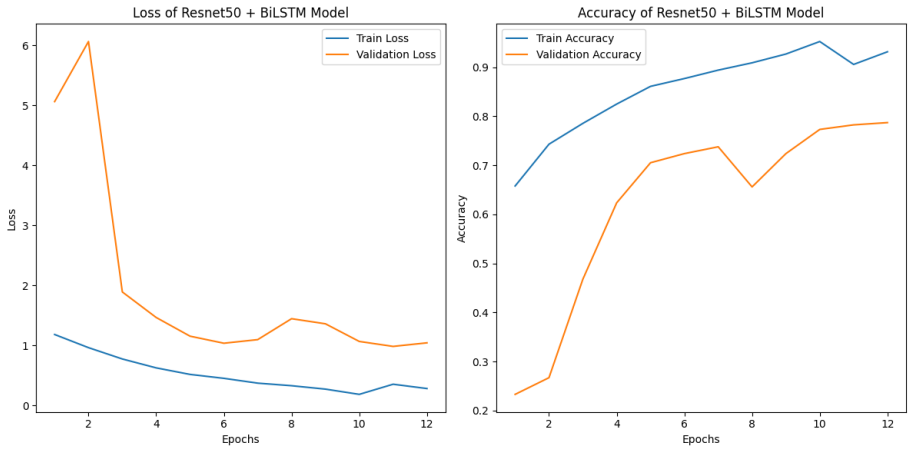


Figure 4.25: Accuracy and Loss of Resnet50+BiLSTM model

The confusion matrix and the values of precision, recall and F1 score for all four classifications are also given below:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| class 0 | 0.68 | 0.79 | 0.73 | 155 |
| class 1 | 0.87 | 0.87 | 0.87 | 159 |
| class 2 | 0.71 | 0.77 | 0.74 | 179 |
| class 3 | 0.78 | 0.58 | 0.67 | 154 |
| accuracy | | | 0.75 | 647 |
| macro avg | 0.76 | 0.75 | 0.75 | 647 |
| weighted avg | 0.76 | 0.75 | 0.75 | 647 |

Figure 4.26: Performance metrics of Resnet50+BiLSTM model

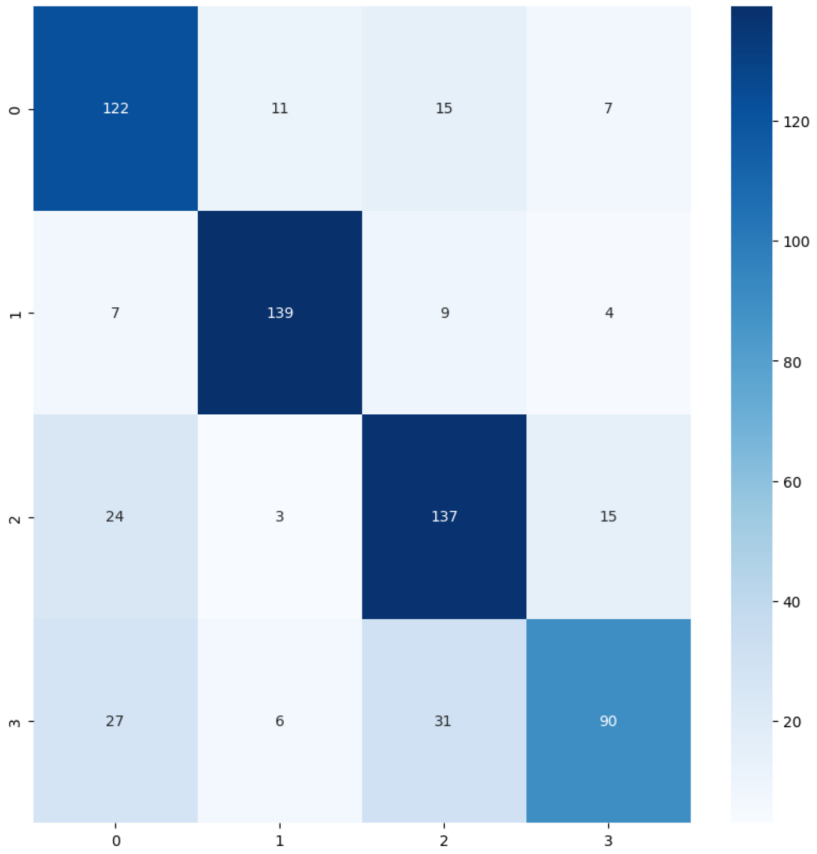


Figure 4.27: Confusion Matrix of the Resnet50+BiLSTM model

This multimodal combination has scored a maximum of 94% in training accuracy and 72% in the validation. In the text dataset, it has scored with 75.68% of accuracy. From the confusion matrix, we can observe how poorly the model has performed in generalization. Moreover, in some specific genres, this model has overfitted and got biased on some specific classes.

## Resnet50+LSTM

For this multimodal system, we have experimented with Resnet50 and LSTM models. Here, we have also used the Resnet50 architecture with a sequential LSTM

model and have trained this multimodal system with our collected data(images + text) for 16 epochs. The model summary looks like below:

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| image_input (InputLayer) | [(None, 224, 224, 3 )] | 0 | [] |
| text_input (InputLayer) | [(None, 8)] | 0 | [] |
| resnet50 (Functional) | (None, 7, 7, 2048) | 23587712 | ['image_input[0][0]'] |
| sequential_1 (Sequential) | (None, 32) | 426400 | ['text_input[0][0]'] |
| flatten_1 (Flatten) | (None, 100352) | 0 | ['resnet50[0][0]'] |
| concatenate_1 (Concatenate) | (None, 100384) | 0 | ['sequential_1[0][0]', 'flatten_1[0][0]'] |
| dense_4 (Dense) | (None, 128) | 12849280 | ['concatenate_1[0][0]'] |
| dropout_5 (Dropout) | (None, 128) | 0 | ['dense_4[0][0]'] |
| dense_5 (Dense) | (None, 4) | 516 | ['dropout_5[0][0]'] |

Total params: 36,863,908
Trainable params: 36,810,788
Non-trainable params: 53,120

Figure 4.28: Resnet50+LSTM model summary

We have used the same image and text shapes in this model and have compiled the model with Adam optimizer with a learning rate of 0.001, categorical_crossentropy as loss function, early stopping and learning scheduler with patience of 4. After only 9 epochs, due to overfitting on the training dataset the model has stopped learning and scored 88% in train accuracy and a maximum of 74% in validation accuracy. On the test dataset, this model has achieved an overall accuracy of 70%. Here are some grapes and other metrics that clearly describe the performance of this Resnet50+LSTM combination:

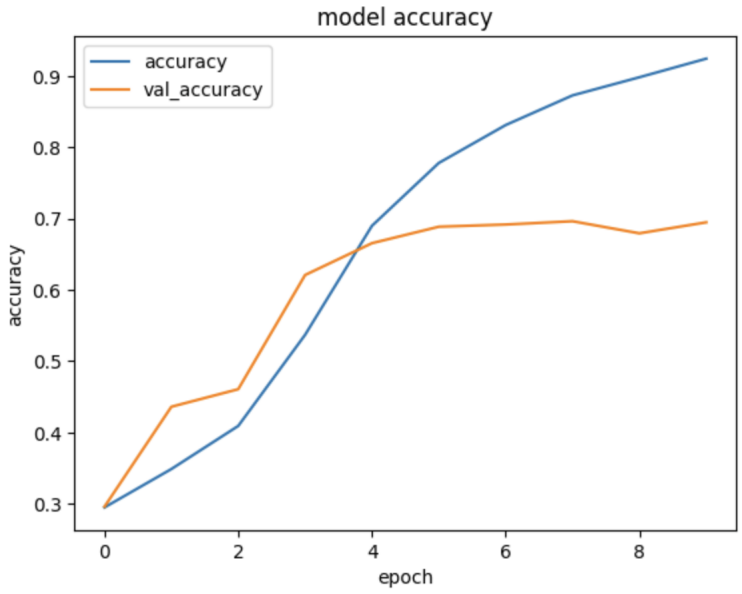


Figure 4.29: Accuracy of Resnet50+LSTM model

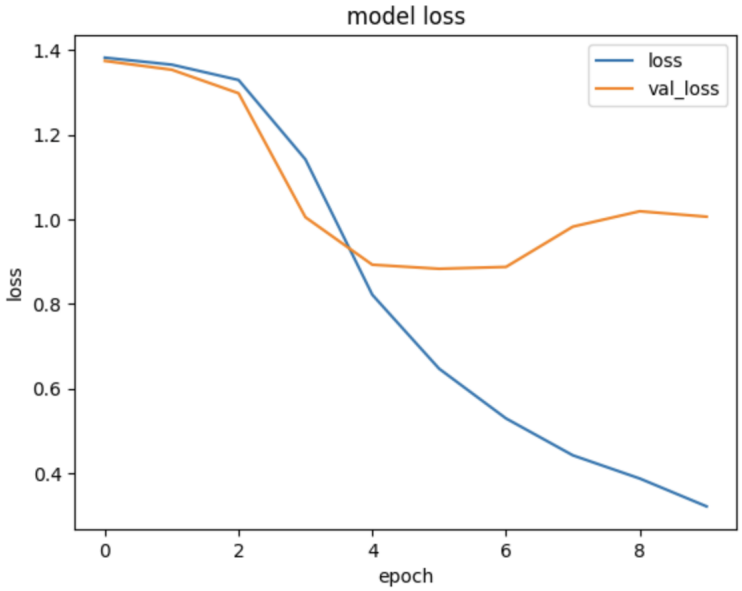


Figure 4.30: Loss of Resnet50+LSTM model

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| class 0 | 0.82 | 0.82 | 0.82 | 169 |
| class 1 | 0.57 | 0.72 | 0.63 | 139 |
| class 2 | 0.86 | 0.65 | 0.74 | 175 |
| class 3 | 0.58 | 0.61 | 0.60 | 165 |
| accuracy | | | 0.70 | 648 |
| macro avg | 0.71 | 0.70 | 0.70 | 648 |
| weighted avg | 0.72 | 0.70 | 0.70 | 648 |

Figure 4.31: Performance metrics of Resnet50+LSTM model

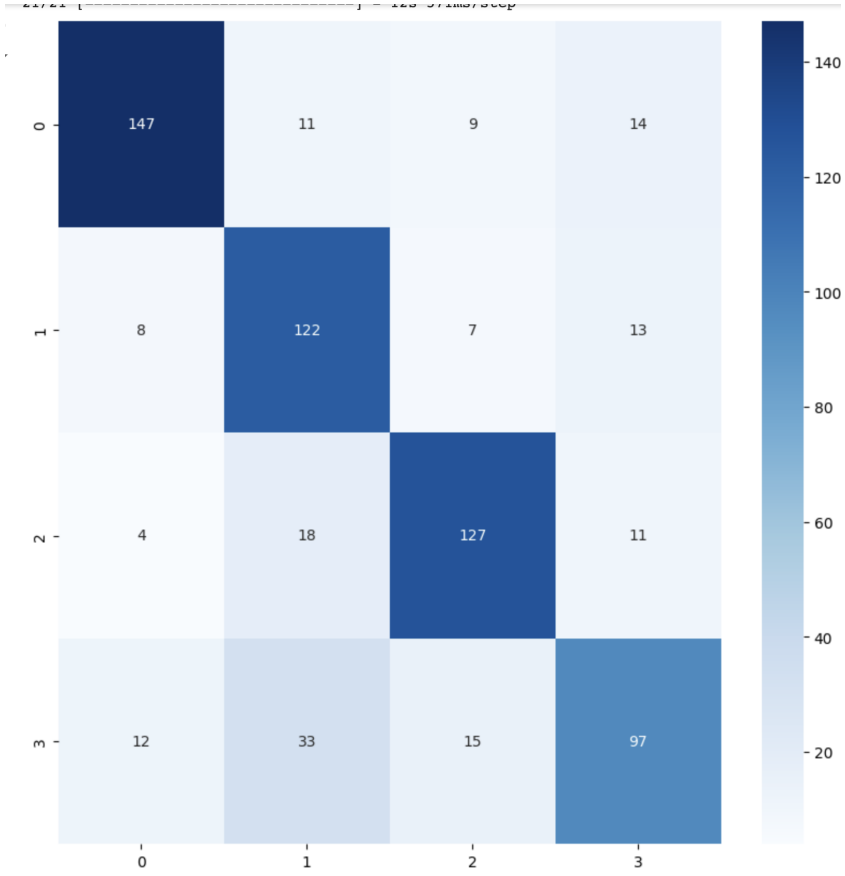


Figure 4.32: Confusion Matrix of the Resnet50+LSTM model

## 4.4 Comparison of the Proposed Multimodal Model(VGG16 + LSTM) with Other CNN and LSTM Model Combinations

The following table shows the performance of our proposed model(VGG16 + LSTM) and other implemented CNN and LSTM multimodal model combinations on the unseen Test dataset.

| **Model** | **VGG16 + LSTM** | **VGG16** | **LSTM** | **BiLSTM** | **VGG16 + BiLSTM** | **Resnet50 + BiLSTM** | **Resnet50 + LSTM** |
|---|---|---|---|---|---|---|---|
| Test_Accuracy | 87% | 74% | 75% | 70% | 77% | 76% | 70% |
| F1-Score Class 0 | 88% | 69% | 78% | 82% | 73% | 73% | 82% |
| F1-Score Class 1 | 91% | 70% | 71% | 60% | 86% | 87% | 63% |
| F1-Score Class 2 | 86% | 87% | 83% | 81% | 77% | 74% | 74% |
| F1-Score Class 3 | 81% | 68% | 65% | 57% | 70% | 67% | 60% |
| Recall Class 0 | 87% | 65% | 76% | 77% | 77% | 79% | 82% |
| Recall Class 1 | 92% | 77% | 64% | 61% | 88% | 87% | 72% |
| Recall Class 2 | 86% | 88% | 81% | 81% | 74% | 77% | 65% |
| Recall Class 3 | 82% | 65% | 78% | 60% | 68% | 58% | 61% |
| Precision Class 0 | 89% | 74% | 81% | 88% | 70% | 68% | 82% |
| Precision Class 1 | 90% | 65% | 81% | 59% | 85% | 87% | 57% |
| Precision Class 2 | 87% | 87% | 86% | 82% | 81% | 71% | 86% |
| Precision Class 3 | 81% | 71% | 56% | 54% | 72% | 78% | 58% |

Table 4.2: Performance comparison of different models for classification

In this table, Classes 0, 1, 2 and 3 correspond to Math, science and technology, Story and novel, Comics and cartoons and History and politics respectively.
From this table and from previous confusion metrics and graphs, we can see that our proposed multimodal system with VGG16 + LSTM has performed the best among all of these 7 models. By evaluating only VGG16, we have got 74% of accuracy and from LSTM, we have got 75% accuracy. These have scored the highest when the model predicts the book genre solely on Bangla Book covers or solely on Bengali book names. We expected that merging the CNN and LSTM models would show better results but we have noticed that this improvement has been gained only when using VGG16 in our CNN part and LSTM for RNN part. As our dataset is a medium-sized dataset and does not have that many complex structures, Resnet50 is a bit more overkill for this. It has learned better on train sets but has performed poorly in validation. Similarly, BiLSTM captures more complex word patterns. But, often the Bengali book titles consist of only a few words. Therefore, BiLSTM is also a bit excessive for this task. As a result, our proposed multimodal system (VGG16 + LSTM) has achieved the best performance by correctly classifying 87% of the book genres.

# Chapter 5

# Conclusion

## 5.1 Conclusion

This research paper has tackled the difficult and challenging task of classification of Bangla book genres by integrating a multimodal system that includes convolutional neural network (CNN), machine learning and natural language processing(NLP) techniques. Our exploration has yielded valuable insights by using Convolutional neural network (CNN) for analysing book covers and natural language processing(NLP) for text based classification for book titles on covers. It has shown how merging different methods improves genre prediction. Although, single models have not perform too poorly, our aim in this paper has been to use both covers and images for better performance. Which proves that multi featured data performs better than single features. This also highlights the importance of data quality and design choices by Thoroughly analysing dataset details and using pre-trained models.Despite various challenges like limited visual attributes and the fine details of Bengali language, this approach enhanced accuracy in genre classification.The models we proposed are very much good enough for classifying bangla book genre with decent accuracy.However, There can still be some error which can be manually resolved by humans. This research advances our understanding of Bangla book genre classification and provides a foundation of future work in multimodal systems for different language and cultural contexts.

# Bibliography


[1] L. S. Busagala, W. Ohyama, T. Wakabayashi, and F. Kimura, "Machine learning with transformed features in automatic text classification," in *Proceedings of ECML/PKDD-05 Workshop on Subsymbolic Paradigms for Learning in Structured Domains (Relational Machine Learning)*, Oct. 2005, pp. 11–20.

[2] J. Zujovic, L. Gandy, S. Friedman, B. Pardo, and T. N. Pappas, "Classifying paintings by artistic genre: An analysis of features & classifiers," in *2009 IEEE International Workshop on Multimedia Signal Processing*, IEEE, Oct. 2009, pp. 1–5.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[4] M. Çağataylı and E. Çelebi, "The effect of stemming and stop-word-removal on automatic text classification in turkish language," in *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I*, Springer International Publishing, 2015, pp. 168–176.

[5] B. K. Iwana, S. T. R. Rizvi, S. Ahmed, A. Dengel, and S. Uchida, "Judging a book by its cover," *arXiv preprint arXiv:1610.09204*, 2016.

[6] G. S. Simoes, J. Wehrmann, R. C. Barros, and D. D. Ruiz, "Movie genre classification with convolutional neural networks," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 259–266. DOI: 10.1109/ijcnn.2016.7727207. arXiv: 1511.7571.

[7] W. T. Chu and H. J. Guo, "Movie genre classification based on poster images with deep neural networks," in *Proceedings of the workshop on multimodal understanding of social, affective, and subjective attributes*, ACM, 2017, pp. 39–45.

[8] S. K. Metïn and B. Karaoğlan, "Stop word detection as a binary classification problem," *Anadolu University Journal of Science and Technology A-Applied Sciences and Engineering*, vol. 18, no. 2, pp. 346–359, 2017.

[9] P. Buczkowski, A. Sobkowicz, and M. Kozlowski, "Deep learning approaches towards book cover classification," in *ICPRAM*, 2018, pp. 309–316.

[10] A. M. Ertugrul and P. Karagoz, "Movie genre classification from plot summaries using bidirectional lstm," in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, 2018, pp. 248–251. DOI: 10.1109/ICSC.2018.00043.

[11] S. Jolly, B. K. Iwana, R. Kuroki, and S. Uchida, "How do convolutional neural networks learn design?" In *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, Aug. 2018, pp. 1085–1090.

[12] J. Worsham and J. Kalita, "Genre identification and the compositional effect of genre in literature," in *Proceedings of the 27th international conference on computational linguistics*, Aug. 2018, pp. 1963–1973.

[13] G. R. Biradar, J. M. Raagini, A. Varier, and M. Sudhir, "Classification of book genres using book cover and title," in *2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT)*, IEEE, Jun. 2019, pp. 72–723.

[14] S. Gupta, M. Agarwal, and S. Jain, "Automated genre classification of books using machine learning and natural language processing," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, Jan. 2019, pp. 269–272.

[15] E. Ozsarfati, E. Sahin, C. J. Saul, and A. Yilmaz, "Book genre classification based on titles with comparative machine learning algorithms," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, Feb. 2019, pp. 14–20.

[16] L. Shi, C. Li, and L. Tian, "Music genre classification based on chroma features and deep learning," in *2019 Tenth International Conference on Intelligent Control and Information Processing (ICICIP)*, IEEE, 2019, pp. 81–86. DOI: 10.1109/ICICIP47338.2019.9012215.

[17] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.

[18] R. Ul Haque, P. Mehera, M. F. Mridha, and M. A. Hamid, "A complete bengali stop word detection mechanism," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, IEEE, 2019.

[19] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[20] R. Jayaram, H. Mallappa, S. Pavithra, M. B. Noor, and K. J. Bhanushree, "Classifying books by genre based on cover," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 5, pp. 530–535, 2020.

[21] C. Kundu and L. Zheng, "Deep multi-modal networks for book genre classification based on its cover," *arXiv preprint arXiv:2011.07658*, 2020.

[22] C. S. Kundu, "Book genre classification by its cover using a multi-view learning approach," 2020.

[23] A. Lucieri, H. Sabir, S. A. Siddiqui, *et al.*, "Benchmarking deep learning models for classification of book covers," *SN Computer Science*, vol. 1, pp. 1–16, 2020.

[24] R. B. Mangolin, R. M. Pereira, A. S. Britto, *et al.*, "A multimodal approach for multi-label movie genre classification," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19 071–19 096, 2020. DOI: 10.1007/s11042-020-10086-2.

[25] M. Narendra, K. M. Rayudu, T. S. Sai, K. Rajshekar, and V. Lingala, "A survey on book genre classification system using machine learning," *Mathematical Statistician and Engineering Applications*, vol. 69, no. 1, pp. 147–160, 2020.

[26] R. Rahman, "A benchmark study on machine learning methods using several feature extraction techniques for news genre detection from bangla news articles & titles," in *Proceedings of the 7th International Conference on Networking, Systems and Security*, Dec. 2020, pp. 25–35.

[27] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132 306, 2020.

[28] B. Koonce, "Resnet 50," in *Apress eBooks*, Apress, 2021, pp. 63–72. DOI: 10.1007/978-1-4842-6168-2_6.

[29] B. Y. Panchal, *Book genre categorization using machine learning algorithms (k-nearest neighbor, support vector machine, and logistic regression) using customized dataset*, 2021.

[30] A. Goyal and V. Prem Prakash, "Statistical and deep learning approaches for literary genre classification," in *Advances in Data and Information Sciences: Proceedings of ICDIS 2021*, Singapore: Springer Singapore, 2022, pp. 297–305.

[31] M. B. Hossain, S. H. S. Iqbal, M. M. Islam, M. N. Akhtar, and I. H. Sarker, "Transfer learning with fine-tuned deep cnn resnet50 model for classifying covid-19 from chest x-ray images," *Informatics in Medicine Unlocked*, vol. 30, p. 100 916, 2022.

[32] M. Saraswat and Srishti, "Leveraging genre classification with rnn for book recommendation," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3751–3756, 2022.

[33] A. Sethy, A. K. Rout, C. Gunda, S. K. Routhu, S. Kallepalli, and G. Garbhapu, "Book genre classification system using machine learning approach: A survey," in *International Conference on Soft Computing and Signal Processing*, Singapore: Springer Nature Singapore, Jun. 2022, pp. 231–241.

[34] M. Islam, S. A. Shuvo, M. S. Nipun, *et al.*, "Efficient approach to using cnn-based pre-trained models in bangla handwritten digit recognition," in *Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2022*, Singapore: Springer Nature Singapore, 2023, pp. 697–716.

[35] Y. Jiang and L. Zheng, "Deep learning for video game genre classification," *Multimedia Tools and Applications*, pp. 1–15, 2023.

[36] A. Rasheed, A. I. Umar, S. H. Shirazi, Z. Khan, and M. Shahzad, "Cover-based multiple-book genre recognition using an improved multimodal network," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 26, no. 1, pp. 65–88, 2023.

[37] N. Vanetik, M. Tiamanova, G. Kogan, and M. Litvak, "Genre classification of books in russian with stylometric features: A case study," *Information*, vol. 15, no. 6, p. 340, 2024. DOI: 10.3390/info15060340.

[38] *Batighar: The empire of books in bangladesh — baatighar.com*, https://baatighar.com/, [Accessed 17-10-2024].

[39] *Bdbooks.net*, https://bdbooks.net/, [Accessed 17-10-2024].

[40] *Buy Book Online - Best Online Book Shop in Bangladesh | Rokomari.com — rokomari.com*, https://www.rokomari.com/book, [Accessed 17-10-2024].

[41] *Buy Islamic Books - Online Book Shop in Bangladesh — wafilife.com*, https://www.wafilife.com/, [Accessed 17-10-2024].

[42] H. C. Y. Ge and C. Wu, *Classification of book genres by cover and title*.

[43] *KolkataKomics/1st one stop bengali comics/graphic novel store — kolkatakomics.com*, https://www.kolkatakomics.com/, [Accessed 17-10-2024].

# Bangla book title domain based custom word lemmatizer dictionary

| Original Word | Root Word |
|---|---|
| "হিস্ট্রি" | "ইতিহাস" |
| "ইতিহাসের" | "ইতিহাস" |
| "অঙ্ক" | "গণিত" |
| "অঙ্কের" | "গণিত" |
| "গাণিতিক" | "গণিত" |
| "অংকের" | "গণিত" |
| "রাজ্যে" | "গণিত" |
| "কৈশোরীর" | "কৈশোর" |
| "কৈশোরের" | "কৈশোর" |
| "কৈশোরী" | "কৈশোর" |
| "কৈশোরদের" | "কৈশোর" |
| "শিশু" | "ছোট" |
| "ছোট্ট" | "ছোট" |
| "ছোটদের" | "ছোট" |
| "শিশুর" | "ছোট" |
| "শিশুদের" | "ছোট" |
| "ছোটদের" | "ছোট" |
| "বালক" | "কৈশোর" |
| "বালকের" | "কৈশোর" |
| "বালিকা" | "কৈশোর" |
| "বালিকার" | "কৈশোর" |
| "কমিক্স" | "কমিকস" |
| "গ্রাফিক" | "কমিকস" |
| "কমিক্সের" | "কমিকস" |
| "কার্টুন" | "কমিকস" |

Figure 5.1: Original word and Root word

| | |
|---|---|
| "কার্টুন" | "কমিকস" |
| "কমিক্স" | "কমিকস" |
| "সিনডরেলা" | "কমিকস" |
| "সিনডেরেলার" | "কমিকস" |
| "স্পাইডারম্যান" | "কমিকস" |
| "স্পাইডারম্যানের" | "কমিকস" |
| "ডোরেমন" | "কমিকস" |
| "ডোরেমনের" | "কমিকস" |
| "টম" | "কমিকস" |
| "জেরি" | "কমিকস" |
| "সিসিমপুর" | "কমিকস" |
| "বাবু" | "কমিকস" |
| "বেসিক আলী" | "কমিকস" |
| "গুড্ডুবুড়া" | "কমিকস" |
| "নোমান" | "কমিকস" |
| "প্যারাহীন" | "কমিকস" |
| "টম এবং জেরি" | "কমিকস" |
| "টিনটিনের" | "কমিকস" |
| "টিনটিনে" | "কমিকস" |
| "গুড্ডুবুড়া" | "কমিকস" |
| "বাঘরা" | "প্রাণি" |
| "পশুপাখির" | "প্রাণি" |
| "ব্যাঙ" | "প্রাণি" |
| "বাঘ" | "প্রাণি" |
| "মাছের" | "প্রাণি" |
| "শালিকছানা" | "পাখি" |
| "হাঁস" | "পাখি" |
| "পাখির" | "পাখি" |
| "পাখিদের" | "পাখি" |

Figure 5.2: Original word and Root word

| | |
|---|---|
| "ময়নাপাখি" | "পাখি" |
| "ময়না" | "পাখি" |
| "শালিক" | "পাখি" |
| "বুবলিদের" | "পাখি" |
| "বুবলি" | "পাখি" |
| "পায়রা" | "পাখি" |
| "ফড়িং" | "পাখি" |
| "প্রজাপতি" | "পাখি" |
| "টিয়া" | "পাখি" |
| "পিঁপড়ে" | "প্রাণি" |
| "ইঁদুর" | "প্রাণি" |
| "বিড়াল" | "প্রাণি" |
| "বিড়ালের" | "প্রাণি" |
| "বিড়ালদের" | "প্রাণি" |
| "বিড়ালেরা" | "প্রাণি" |
| "কাঠবিড়ালির" | "প্রাণি" |
| "নেকড়ে" | "প্রাণি" |
| "নেকড়ের" | "প্রাণি" |
| "বাদুড়ের" | "প্রাণি" |
| "বাদুড়" | "প্রাণি" |
| "কাঠবিড়াল" | "প্রাণি" |
| "জলপরি" | "প্রাণি" |
| "ব্যাঙের" | "প্রাণি" |
| "হাতি" | "প্রাণি" |
| "হাতির" | "প্রাণি" |
| "তেলাপোকা" | "প্রাণি" |
| "মাছ" | "প্রাণি" |
| "সিংহ" | "প্রাণি" |
| "হনুমান" | "প্রাণি" |

Figure 5.3: Original word and Root word

| | |
|---|---|
| "বানর" | "প্রাণি" |
| "বানরের" | "প্রাণি" |
| "কুমিরের" | "প্রাণি" |
| "কুমির" | "প্রাণি" |
| "যাদুকরের" | "জাদু" |
| "যাদুকর" | "জাদু" |
| "জাদুকর" | "জাদু" |
| "প্রেম" | "ভালোবাসা" |
| "প্রেমের" | "ভালোবাসা" |
| "প্রিয়তমা" | "ভালোবাসা" |
| "ভালোবাসার" | "ভালোবাসা" |
| "ভালোবাসায়" | "ভালোবাসা" |
| "লাভিং" | "ভালোবাসা" |
| "প্রিয়তমার" | "ভালোবাসা" |
| "রচনাবলি" | "রচনা" |
| "রচনাসমগ্র" | "রচনা" |
| "রচনাবলী" | "রচনা" |
| "হরর" | "ভয়" |
| "ভূত" | "ভয়" |
| "যকের" | "ভয়" |
| "কাব্য" | "কবিতা" |
| "জননী" | "মা" |

Figure 5.4: Original word and Root word