Deep Neural Networks for Book Cover Classification

Ikram Ait Taleb Naser

Contents

List of Figures

List of Tables

Declaration of Authorship

I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work, and including any code produced using generative AI systems. I/We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.

Abstract

The project aim is the implementation of a scalable deep learning approach for classifying book covers from the Amazon Books Review dataset. This report describes the implementation of a convolutional neural network (CNN) based solution using transfer learning with MobileNetV2 as the base model. Training the model on the full dataset requires substantial computational resources (approximately 5 hours). For this reason, the experiments presented here are conducted on a stratified subsample of the dataset.

1 Introduction

The aim of this project is to develop a deep learning solution for classifying book covers into different categories using convolutional neural networks (CNNs). Our approach leverages transfer learning with pre-trained models and, to ensure the scalability of our solution, including a caching system for images and metadata.

The central aspects of this work are:

- A CNN-based approach for book cover classification
- A caching system for images and metadata to optimize resource usage
- A two-stage training approach with transfer learning
- An overfitting analysis

2 Dataset Description

2.1 Amazon Books Review Dataset

The Amazon Books Review dataset contains 2 files: The first file **reviews** file contain the feedback about 3M user on 212404 unique books. The second file **Books Details** file contains details information about 212404 unique books, built by using google books API to get details information about books rated in the first file. For this project task, I only used the Book Details one, containing:

#	Column	Non-Null Count	Dtype
0	Title	212403	object
1	description	143962	object
2	authors	180991	object
3	image	160329	object
4	previewLink	188568	object
5	publisher	136518	object
6	publishedDate	187099	object
7	infoLink	188568	object
8	categories	171205	object
9	ratingsCount	49752	float64

Table 1: DataFrame summary: 212404 entries, 10 columns

2.2 Selected Data Subset

To focus on the most representative categories and since we only care about the image and category columns we selected a subset of the dataset with the following characteristics, Table 3 shows the statistics of the selected dataset subset:

- Only books with valid cover image URLs and category information
- Only select the 2 relevant column (image and category)

Column	Missing	Percent (%)
ratingsCount	162652	76.576712
publisher	75886	35.727199
description	68442	32.222557
image	52075	24.516958
categories	41199	19.396527
authors	31413	14.789270
publishedDate	25305	11.913617
previewLink	23836	11.222011
infoLink	23836	11.222011
Title	1	0.000471

Table 2: Percentage of missing values by column

- Focus on the top 10 most frequent book categories, to simplify the classification problem I only took the primary category for multi label books. Figure 2 illustrates the distribution of books across the top 10 categories in the dataset.

Table 3: Dataset Statistics

Metric	Value
Total number of books (after filtering)	151884
Number of categories	10
Image dimensions	128 × 128 pixels
Training set size	60%
Validation set size	20%
Test set size	20%

Figure 1 illustrates a sample of book covers:

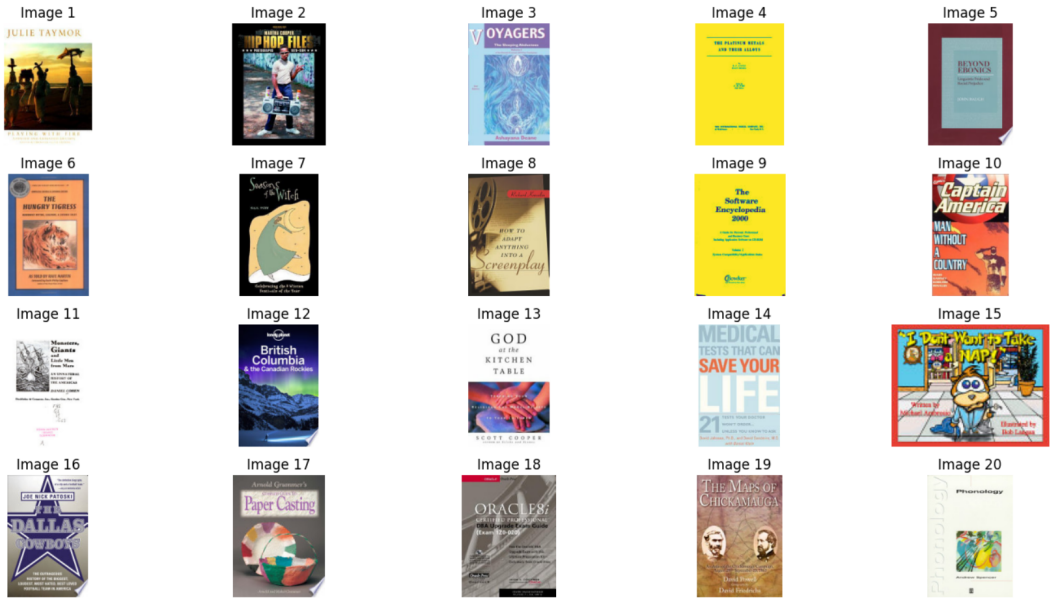


Figure 1: Cover samples from the dataset

The model solutions will be evaluated only on a subset of the original data, given the computational cost constraints. I uniformly subsampled 500 examples per category, resulting in a dataset of 5000

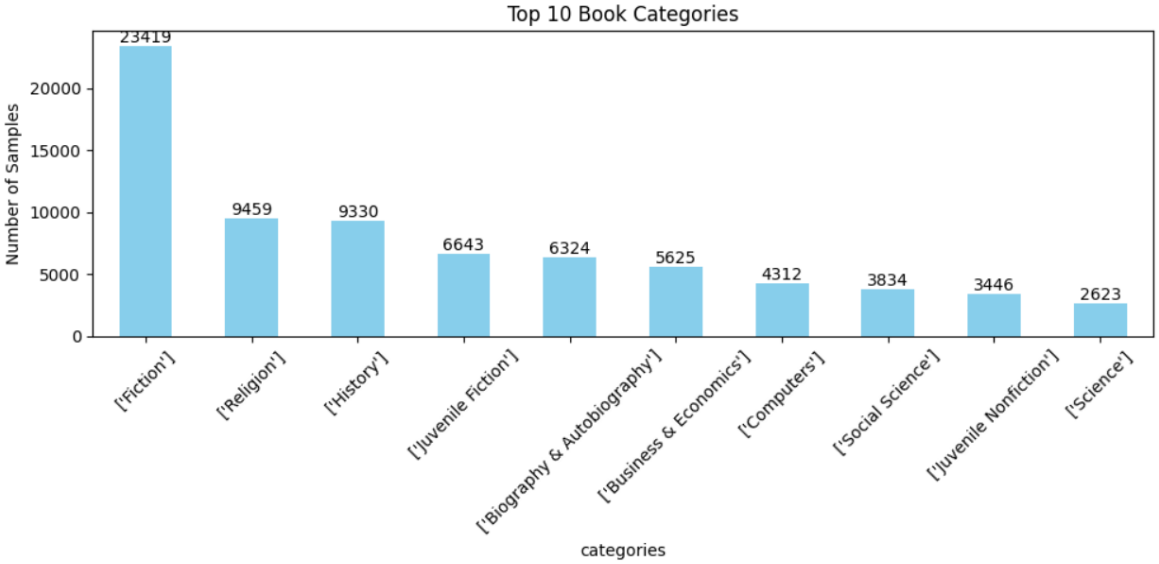


Figure 2: Distribution of books across the top 10 categories

total samples, split into 3000 (train), 1000 (val), 1000 (test), Each subset maintained a stratified class distribution as summarized in the table 10.

3 Data Organization and Preprocessing

3.1 Data Organization

A structured approach to ensure efficient processing and scalability is essential when dealing with massive datasets. Therefore, all book cover images were downloaded once and cached locally using MD5 hashes of URLs as filenames. In fact, image caching eliminates redundant downloads and ensures consistent image access across multiple runs. Additionally, dataset splits, category mappings, and other metadata were cached as JSON files for quick access and to maintain consistency between runs.

3.2 Data Preprocessing

We applied the following preprocessing techniques to prepare the data for training:

- **Image Resizing**: All images were resized to 128×128 pixels to ensure uniform input dimensions for the neural network. Then, images were converted to RGB format to ensure consistent color channels (3 channels).
- **Normalization**: Pixel values were normalized to the range [0,1] by dividing by 255.
- **Category Extraction**: For books with multiple categories, we extracted the primary category (first listed category) as the target label.
- **Label Encoding**: Category names were encoded as integer indices using scikit-learn's LabelEncoder.
- **Train-Validation-Test Split**: The dataset was split into training (60%), validation (20%), and test (20%) sets using stratified sampling to maintain class distribution.
- **Subsampling**: From an original dataset of 72970 samples, I am subsampling 500 book covers for each of the 10 categories, leading to a total of 5000 samples. (10)

Table 4: Class Weights to Handle Imbalance (when using the whole dataset)

Class ID	Class Name	Weight
0	Biography & Autobiography	1.1785
1	Business & Economics	1.3271
2	Computers	1.7196
3	Education	2.8634
4	Fiction	0.3224
5	History	0.8048
6	Juvenile Fiction	1.1186
7	Juvenile Nonfiction	2.2123
8	Religion	0.7899
9	Social Science	1.9589

4 Experimental Methods

4.1 Experimental Setup

Our experimental protocol was designed to ensure replicability:

1. Dataset split into training (60%), validation (20%), and test (20%) sets using stratified sampling
2. Fixed random seed (42) for reproducibility
3. Two-stage training approach with early stopping based on validation accuracy
4. Evaluation on the held-out test set
5. Multiple runs to ensure consistency of results

4.2 Architecture

Our model architecture consists of:

- **Base Model**: Pre-trained MobileNetV2 (trained on ImageNet), a lightweight CNN architecture designed for mobile and embedded vision applications.
- **Global Average Pooling**: To reduce spatial dimensions
- **Batch Normalization**: For training stability
- **Dense Layer**: The first dense layer has 512 units. The second dense layer has 256 units.
- **Dropout**: a dropout rate of 0.7 in the transfer learning model function, and a slightly lower rate of 0.7 * 0.8 (which is 0.56) for the second dense layer. The model uses the relu activation function for both dense layers.
- **Output Layer**: Dense layer with softmax activation (10 neurons for 10 categories)

4.3 Training Strategy

A two-stage training approach was implemented:

- **Stage 1:** Training with Frozen Base Model

  The MobileNetV2 base model, pre-trained on ImageNet, is loaded. All the layers in the base model are initially frozen (base model.trainable = False). Meaning that their weights will not be updated during this stage of training. The newly added dense layers on top of the base model are trained. These layers are responsible for learning to classify your specific book cover categories based on the high-level features extracted by the frozen MobileNetV2 base. The model is compiled with a learning rate of 0.0005. Training proceeds for up to 20 epochs, with

early stopping monitoring validation accuracy to prevent overfitting to the training data. After training, the weights of the model that resulted in the best validation accuracy during this stage are restored.

- **Stage 2:** Conservative Fine-tuning

  A few of the top layers of the MobileNetV2 base model are unfrozen (base model.trainable = True for the last 10 layers). This allows these layers to be fine-tuned. The rest of the base model layers remain frozen. The model is recompiled with a much lower learning rate (0.00005). A lower learning rate is crucial during fine-tuning to avoid drastically changing the pre-trained weights and potentially disrupting the learned features. Training continues for a shorter period (up to 10 epochs) with early stopping and a reduced learning rate on plateau, still monitoring validation accuracy. Again, the weights corresponding to the best validation accuracy achieved in this stage are restored. The idea behind this two-stage approach is to first train the new classification layers on top of a stable feature extractor (the frozen base model) and then, in the second stage, adjust the top layers of the base model to better suit the specific characteristics of the book covers without forgetting the features learned from ImageNet.

4.4 Data Augmentation

To prevent overfitting, the training pipeline included data augmentation to simulate data diversity. The transformations applied were:

- **Geometric transformations**: Rotation (±15°), width/height shifts (±10%), shear (±10%), zoom (±10%)
- **Photometric augmentation**: Brightness variation (0.8-1.2 range)
- **Spatial augmentation**: Horizontal flipping
- **Fill strategy**: Nearest neighbor interpolation for boundary pixels

4.5 Regularization Strategies

To prevent overfitting, we implement multiple regularization techniques:

1. **Dropout**: High dropout rates (0.7 and 0.56) in dense layers
2. **L2 Regularization**: Weight decay ($\lambda = 0.01$) on dense layers
3. **Batch Normalization**: Stabilizes training and provides implicit regularization
4. **Early Stopping**: Monitors validation accuracy with patience=5
5. **Data Augmentation**: Increases training data diversity

5 Results and Discussion

5.1 Evaluation Metrics

We used the following metrics to evaluate our model:

- **Accuracy**: The proportion of correctly classified book covers
- **Precision**: The ability of the model to avoid false positives
- **Recall**: The ability of the model to find all positive samples
- **F1-Score**: The harmonic mean of precision and recall
- **Confusion Matrix**: To visualize classification performance across categories

Table 5: Class Weights to Handle Imbalance (when using the whole dataset)

Class ID	Class Name	Weight
0	Biography & Autobiography	1.1785
1	Business & Economics	1.3271
2	Computers	1.7196
3	Education	2.8634
4	Fiction	0.3224
5	History	0.8048
6	Juvenile Fiction	1.1186
7	Juvenile Nonfiction	2.2123
8	Religion	0.7899
9	Social Science	1.9589

5.2 Classification Performance on full dataset

Although this report focuses on the subsampled dataset, it is important to note that overfitting was observed when training on the full dataset, as evidenced by the training and validation accuracy and loss curves presented in Figure 5. The transfer learning model achieved 44.72% accuracy on the test set. Table 7 shows the classification performance of our model on the test set.

Table 6: Classification Results on Test Set

Class	Precision	Recall	F1-Score	Support
Biography & Autobiography	0.34	0.35	0.35	1238
Business & Economics	0.35	0.36	0.35	1099
Computers	0.56	0.61	0.59	849
Education	0.28	0.28	0.28	510
Fiction	0.66	0.54	0.60	4527
History	0.39	0.41	0.40	1813
Juvenile Fiction	0.47	0.57	0.51	1305
Juvenile Nonfiction	0.26	0.33	0.29	660
Religion	0.37	0.41	0.39	1848
Social Science	0.18	0.18	0.18	745
Accuracy			**0.45**	14594
Macro Avg	0.39	0.40	0.39	14594
Weighted Avg	0.46	0.45	0.45	14594

Table 7: Classification report

Figure 4 shows the confusion matrix for our model on the test set. The model demonstrates strong performance in distinguishing genres with highly distinctive visual features, such as Computers, which shows minimal confusion with other categories. Conversely, genres with more semantically or visually overlapping characteristics such as Fiction, Religion, Juvenile Fiction, and Juvenile Nonfiction exhibit significant misclassifications between one another. For instance, while the model correctly classifies a large number of Fiction books (2448), it frequently confuses them with History (365) and Religion (445). Similarly, the confusion between Juvenile Fiction and Juvenile Nonfiction suggests a need for additional context or metadata, as their visual cues may be insufficient for accurate separation. The Education category also suffers from poor classification accuracy (only 141), likely due to weak visual distinctiveness.

5.3 Classification Performance on subsampled dataset

This report focuses on the results on the subsampled dataset. To ensure balance and computational efficiency, 500 examples were subsampled per category (10 primary categories), resulting in a dataset of 5000 total samples, split as 3000 (train), 1000 (val), 1000 (test), Each subset maintained a stratified

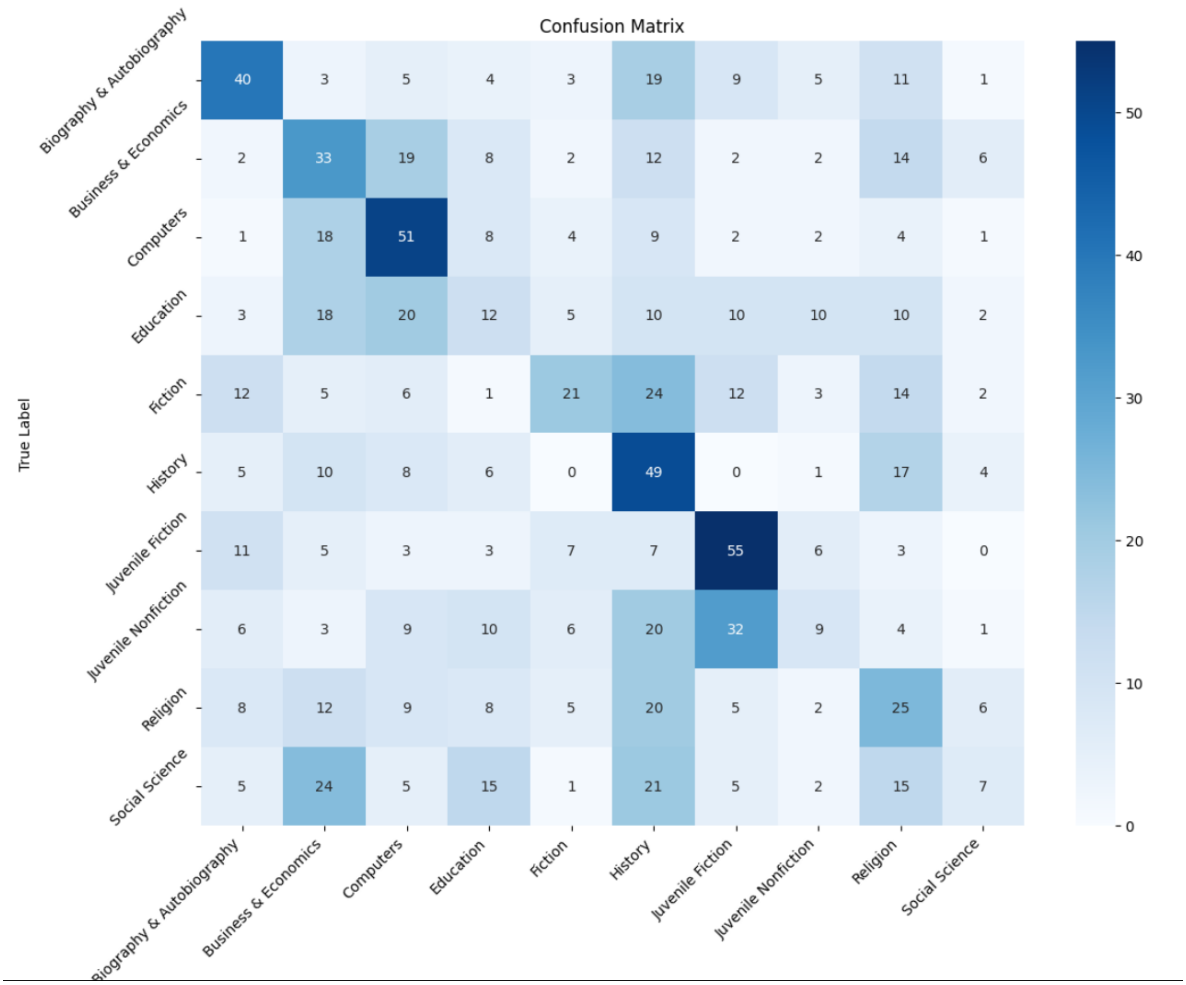


Figure 3: Confusion matrix on test set (subsampled data)

class distribution as summarized in the table 10. For subsampled data the classification performance is summarized in 8. The results of the CNN model, as depicted in the classification report and confusion matrix, reveal limited classification performance across the 10 book genre categories. The overall accuracy stands at 30% (Table 9), with a macro-averaged F1-score of 0.28, indicating that the model struggles to generalize across all classes equally. Performance is highly imbalanced: while categories like Juvenile Fiction (F1 = 0.44), Biography & Autobiography (F1 = 0.42), and Computers (F1 = 0.36) show comparatively stronger results, other classes such as Education (F1 = 0.10), Juvenile Nonfiction (F1 = 0.15), and Social Science (F1 = 0.07) perform poorly. The confusion matrix confirms substantial misclassification, particularly among conceptually similar genres. For instance, Education, Juvenile Nonfiction, and Social Science frequently overlap with predictions for Business & Economics and History, suggesting semantic and visual similarities the model fails to disentangle. Notably, the model misclassified 34% of Social Science samples as Business & Economics, indicating potential overfitting to their visual or textual features.

5.4 Discussion

As we can see from table 9, the Train-Validation Gap: 0.0053 and Validation-Test Gap: 0.0030, both gaps are negligible, indicating strong generalization The two-stage training strategy culminated in a final test accuracy of **30.2%**. In Stage 1, with the MobileNetV2 base frozen, the model trained for 13 epochs, improving training accuracy from **12.5%** to **29.9%** and validation accuracy from **23.7%** to **31.0%**, with a maximum train-validation gap of just **−1.4%**, indicating no overfitting. Loss curves for both training and validation decreased smoothly, confirming stable convergence (Fig 6). In Stage

Figure 4: Confusion matrix on test set (full dataset)

2, conservative fine-tuning of the top 10 layers over 10 epochs further refined performance, reaching **31.0%** train accuracy and **30.9%** validation accuracy, with a gap of **+0.1%**. The model's generalization ability remained consistent across all phases, as reflected in the final test accuracy of **30.2%**, with balanced train-validation-test gaps. (Fig 7) The model may benefit from adding more diverse and distinctive samples, especially for Education (misclassified as Computers and Business), Juvenile Nonfiction (misclassified as Juvenile Fiction and History) and Social Science, the category that performed the worst (misclassified as Business & Economics (24), History (21), Fiction (21), Education (15),and only correctly classified 7.

6 Conclusion

In this project, I developed a deep learning solution for book cover classification using convolutional neural networks. The focus of the study is the implementation of a solution on a subsampled dataset by leveraging transfer learning with MobileNetV2 and, to optimize resource usage and improve scalability, I implemented a comprehensive caching system.

In summary, the two-stage transfer learning strategy with MobileNetV2 achieved stable convergence and strong generalization, as indicated by negligible train-validation and validation-test gaps. However, the overall classification performance remained modest, with a final test accuracy of 30.2% and a macro-averaged F1-score of 0.28. The results show that while the model effectively avoids overfitting, it fails to discriminate among visually similar categories. In particular, Education, Juvenile Nonfiction, and Social Science exhibited severe misclassification patterns, suggesting that these genres require more diverse and representative training data.

Class	Precision	Recall	F1-score	Support
Biography & Autobiography	0.43	0.40	0.41	100
Business & Economics	0.25	0.33	0.29	100
Computers	0.38	0.51	0.43	100
Education	0.16	0.12	0.14	100
Fiction	0.39	0.21	0.27	100
History	0.26	0.49	0.34	100
Juvenile Fiction	0.42	0.55	0.47	100
Juvenile Nonfiction	0.21	0.09	0.13	100
Religion	0.21	0.25	0.23	100
Social Science	0.23	0.07	0.11	100
Accuracy			0.30	1000
Macro avg	0.29	0.30	0.28	1000
Weighted avg	0.29	0.30	0.28	1000

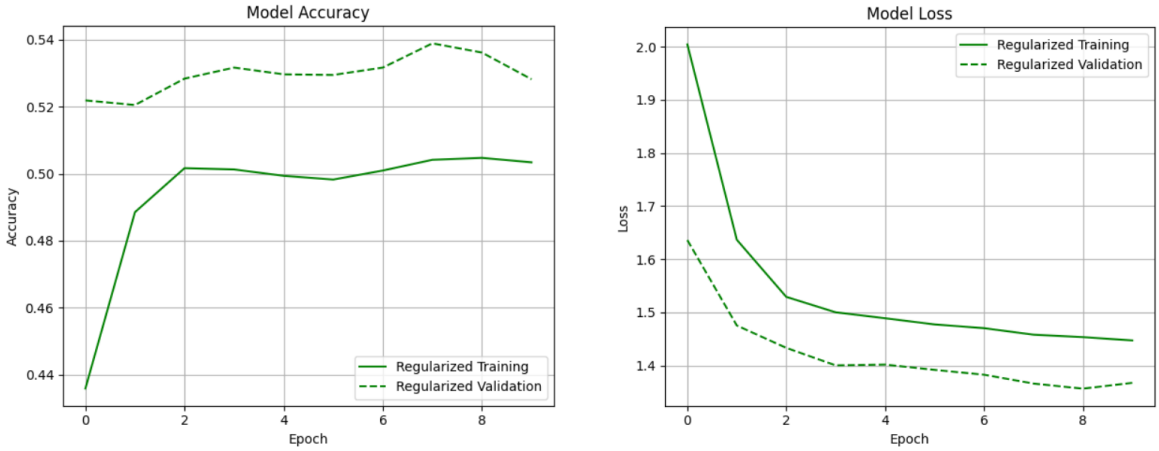
Table 8: Classification Report Summary

Metric	Accuracy
Training Accuracy	0.3103
Validation Accuracy	0.3050
Test Accuracy	0.3020

Table 9: Final Model Accuracy Scores

Category	Original Samples	Subsampled
Fiction	22,636	500
Religion	9,239	500
History	9,067	500
Juvenile Fiction	6,523	500
Biography & Autobiography	6,192	500
Business & Economics	5,497	500
Computers	4,243	500
Social Science	3,725	500
Juvenile Nonfiction	3,299	500
Education	2,549	500
Total	**72,970**	**5,000**

Table 10: Subsampling summary by category



(a) Accuracy curves

(b) Loss curves

Figure 5: Training and validation curves (full dataset)

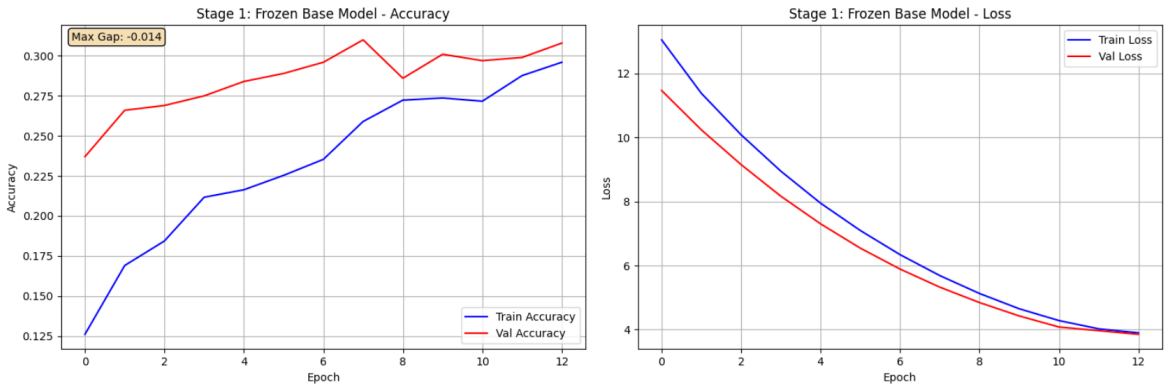


Figure 6: stage 1

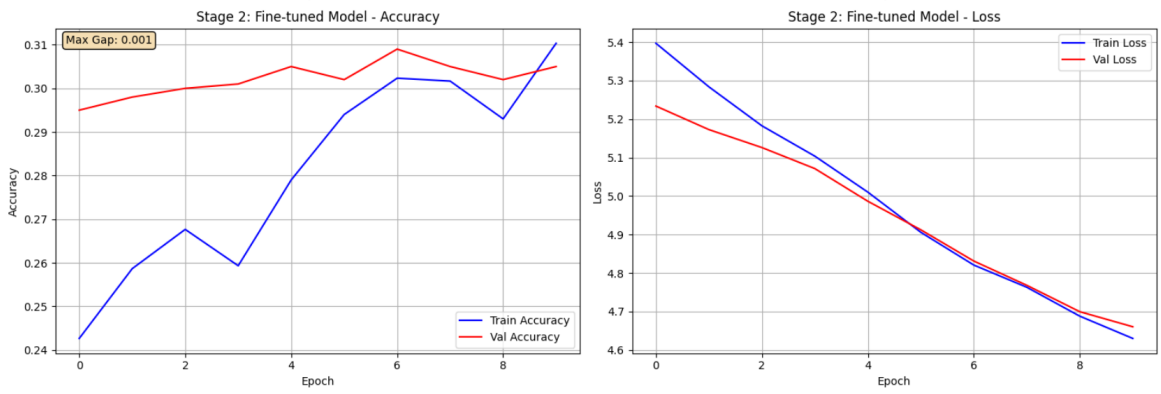


Figure 7: stage 2