New Deep Multi-Modal Network for Book Genre Classification Based on Cover Images using Transformers

Justin Goh
jgoh@scu.edu

Aaron Pluemer
apluemer@scu.edu

Abstract

A book's cover can be considered the first point of interaction between a book and its prospective reader. Because of this, publishers and designers attempt to create titles and covers that convey the content of the book in a single image to lure the reader and are heavily influenced by the genre of the book. Previous work done so far in genre classification based on book covers have shown promising results. However, they have not considered newer techniques in image classification for extracting feature information. Our paper explores using an updated multi-modal approach to extract image-and-text-based features using newer state-of-the-art architectural models, particularly using a shifted window (SWIN) image-based transformer model for the cover and a RoBERTa text-based transformer model for the title of the book. We also intend to explore the effects of image pre-processing techniques on our classification performance.

1 Problem Statement

Most real-world interactions are usually multi-modal or multi-sensory, but historically, machine learning models are usually single-modal. Previous work in multi-modal book genre classification by book cover have shown promising results. However, with a 56.1% top-1 accuracy for 30 classes, even the state-of-the-art models of yesteryear have plenty of room for improvement. Previous work in this area have employed limited methods and have not explored solutions using newer deep learning models. Limited work has also been done in determining the effects of applying pre-processing techniques for genre classification. For these reasons, this paper can serve as an additional data point to the community with respect to the larger question of deep learning architectures beyond CNNs for classification and the role of transformers in machine learning in general.

2 Introduction

The common saying goes "Don't judge a book by its cover". However, when it comes to genre classification, convolutional neural networks (CNN) have made significant progress in the last several years doing just that [4]. This is not particularly surprising; CNNs have been deployed for various image-related tasks, including classification, tracking, detection, facial recognition, and many others.

Most real-world interactions are usually multi-modal or multi-sensory, but historically, machine learning models are usually single-modal. Previous work by Kundu et al. [6] takes a multi-modal approach to book genre classification, which is where we drew inspiration for our project. The approach taken by Kundu et al. was to use ResNet-50 and a Universal Sentence Encoder for image and text feature extraction respectively. Results from that paper demonstrated that the multi-modal models outperformed image based and text based models alone, and the approach achieved a top-1 accuracy of 56.1% across 30 classes. In recent years, large language models (LLM) have become better at tasks such as prediction, text classification, translation, and named entity recognition to name a few. In this paper, we present a similar multi-modal approach to Kundu et al., except we use newer state-of-the-art image and language models for our image and text modalities.

For our image-based modality, we implement a Vision Transformer (ViT) proposed by Dosvitsky et al. [10] and Shifted Window (SWIN) Transformer proposed by Liu et al. [8] For our text-based modality, we implement a Bidirectional Encoder Representation from Transformers (BERT) and a

Robustly Optimized BERT Pretraining Approach (RoBERTa) language model. Lastly, we experiment with some image pre-processing techniques to determine if there is any impact on our models performance, particularly mean normalization, standardization, and Principal-Component-Analysis (PCA). This is based on work by Pal et al. [9]

Since our investigation revolves around assessing the potential superiority of ViT, SWIN, BERT, RoBERTa, and image-preprocessing over ResNet-50 and Universal Sentence Encoder, we'll be using the baseline of Kundu et al. of 56.1% top-1 accuracy on the BookCover30 dataset for training and validation. This dataset provided by Iwana et al. [4] consists of of 57,000 labeled book cover images split equally across 30 classes.

3 Related Work

3.1 Deep Multi-Modal Network for book genre classification based on its cover

The work of Kundu et al. forms the baseline of this paper. In it, they trained a variety of CNN-based networks (LeNet, AlexNet, VGGNet-16, MobileNet-V1, MobileNet-V2, Inception-V2, and ResNet-50) on the images of book covers and determined the best-performing model for their dataset. They then trained an RNN-LSTM and Universal Sentence Encoder on the titles of the books and determined the better-performing model.

Using the best image-based model (ResNet-50) and the best text-based model (Universal Sentence Encoder), they froze the models, extracted the features with a ReLU activation function, concatenated them, and fed them into a final, fully connected layer with a softmax activation function, producing a top-1 accuracy of 56.1%. This step is the backbone of our implementation for multi-modal classification. Although it's not surprising that the multi-modal approach is superior to any single-mode approach, what's especially interesting is the low top-1 accuracy of the image-only ResNet-50 model at 29.6%; most of the classification accuracy is contributed by the text-based Universal Sentence Encoder, which independently achieves a top-1 accuracy of 52.6 percent. This really demonstrates the subjective and artistic nature of the book cover as opposed to the title, which is echoed by Biradar et al. [2]

3.2 Attention is all you need

In this paper, Vaswani et al. [10] introduce the transformer, a network architecture that is based on attention mechanisms. It replaces traditional recurrent and convolutional neural networks. The self-attention layers in the model enables the transformer to take in all input positions from an embedding simultaneously to perform encoding and decoding tasks. It uses a multi-head self-attention mechanism that calculates the attention of each word in a sequence with respect to every other word in the sequence. The self-attention mechanism is made up of 3 vectors – Key, Query, and Value – which are derived from the input sequence. These vectors are used to calculate the scaled dot-product attention defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

The final output of the self-attention mechanism is a weighted sum of the Value vectors for all words in the sequence, where the weights are given by the attention scores. This weighted sum is then passed through a feed-forward network and residual connections to produce the final representation for the input sequence. The multi-head attention mechanism computes attention scores using multiple sets of Key, Query and Value vectors, with each of which producing a different weight matrix. The transformer has since been used for many applications within the field of natural language processing (NLP) and increasingly for computer-vision-related tasks.

3.3 Vision Transformer

The Vision Transformer (ViT) was introduced by Dosvitsky et al. due to the wide spread popularity of Transformers [10] in the field of NLP. To cater as images as inputs, Dosvitsky et al modified the original Transformer architecture by changing the preprocessing steps to tokenize an image instead

of a sentence. This is done by dividing an image into a sequence of 2D patches. These patches are then linearly projected to form patch embeddings. Positional embeddings are added to the patch embeddings to give contextual information. A class token is also added to caputre global features of the image to allow for classification tasks. This final sequence of embedding vectors are used as the input to the Transformer encoder. This encoder is the same as the one used by Vaswani et al. [10] From the experiments performed Dosvitsky et al had shown that the ViT was able to achieved performances on par or better than other SOTA models for image classification if trained on a large enough dataset.

3.4 SWIN Transformer

The transformer is a recent natural language processing (NLP) model notable for its use of sequence modeling of long-range dependencies in data. However, as Liu et al. demonstrate, applying the transformer to images is problematic for two reasons. First, images have far more data than text and transformers are quadratic in computation intensity; therefore, the technique doesn't scale. Second, tokens in language transformers are fixed size, but objects in images are of varying sizes. [3]

Liu et al. [8] address this by using a shifted window on the image patches as opposed to a sliding window and also use hierarchical feature maps instead of a single, low-resolution feature map.

One of the key elements of the SWIN Transformer is the Patch Merging (or Patch Partition) operation. This acts as a downsampling operation. The image is initially divided into 4 patches. Each of the 4 patches are done stacked depth-wise before finally being combined. The SWIN transformer block is similar to the encoder block of a regular transformer [10], with the difference being that the self-attention layer is replaced with a Window-based multiheaded self attention layer (W-MSA) Fig1. The W-MSA calculates the attention score between every pixel with every other pixel within the same patch. This has advantage over regular attention due to the reduce big-O complexity (O(M*N) time vs $O(N^2)$). The second encoder block has a shifted window multiheaded self attention layer (SW-MSA). This SW-MSA works the same way as the W-MSA, except the location of the window is shifted to other parts of the image. This however may result in some pixels not being overlayed by the window and the window not being fully populated by pixels. In order to address this, the authors implement a method known as cyclic shift. What this does is move the "orphaned pixels" into the empty spaces of the window. Since these relocated orphaned pixels are not necessarily adjacent to pixels they were placed next to in the original image, the orphaned pixels are masked.

With these modification, the SWIN Transformer, outperforms previous state-of-the-art CNNs in image classification, object detection, and semantic segmentation. Among its high performance results, it achieves a top-1 classification accuracy of 87.3% on ImageNet-1K database.

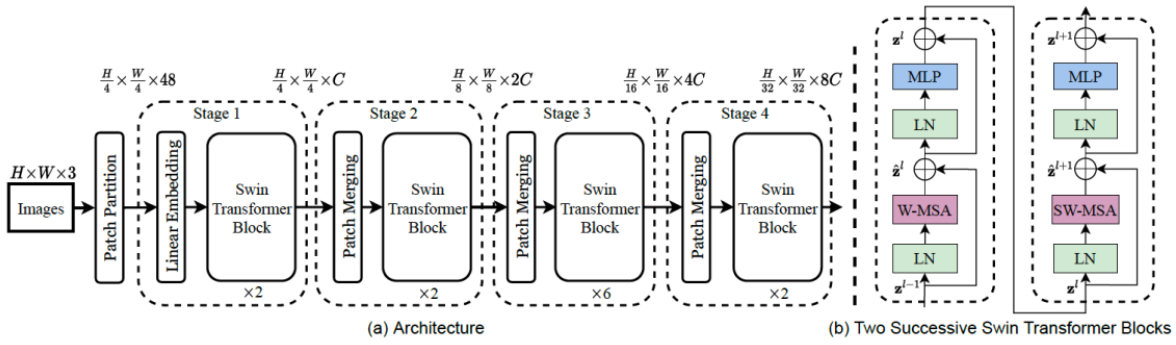

Figure 1: Block diagram of SWIN Transformer

3.5 Bidirectional Encoder Representation from Transformers

There are four primary features that make BERT stand out as a transformer. They are 1) Bidirectionality, 2) Encoder-only, 3) Sub-word tokenization, and 4) Masking. By processing text bidirectionally, BERT learns relationships between words in sentences in various orders and structures, which heavily affect the meaning. By being an encoder, it's optimized for classification (but also unoptimzed for generation). By using sub-word tokenization, BERT is able to break words with common roots down

into parts, resulting in similar tokenization for similar words. And finally, masking allows it to infer the meaning of words from context by selectively trying to predict words during training.

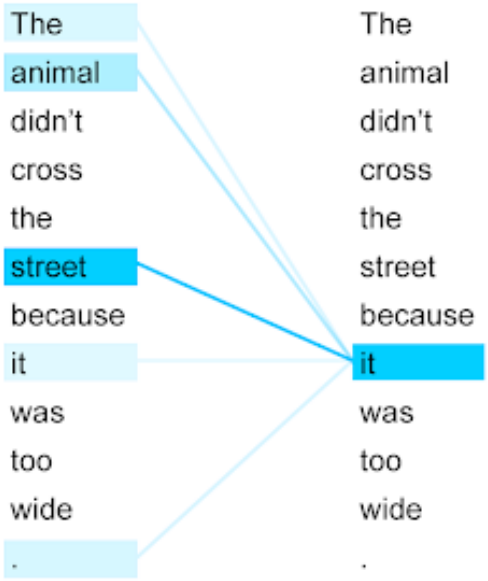

Figure 2: BERT's bidirectionality checks before and after for the object of "it" (source: neptune.ai).

3.6 A Robustly Optimized BERT Pretraining Approach (RoBERTa)

RoBERTa is an extension of the Bidirectional Encoder Representation from Transformers (BERT). The authors posit that BERT was under-trained. This is why they present RoBERTa, a replication study of BERT which includes a careful evaluation of the effects of hyperparameter tuning and training set size. The modifications made are as follows: (1) training the model for longer; (2) removing next sentence prediction objective; (3) training on a larger training corpus, including books, web pages, and text from Wikipedia, among other things; and (4) dynamically changing the masking pattern applied to the training data. These changes resulted in RoBERTa achieving state-of-the-art performances for a variety of NLP tasks such as sentiment analysis, question answering and natural language inference.

4 Solution

Our approach is as follows:

1. Implement the ViT and Swin transformer on the dataset with mean normalization and standardization applied and selecting the highest performing one.
2. On the top performer in (1), apply PCA to the dataset and evaluate the new performance selecting the higher performing one.
3. Implement the BERT and RoBERTa transformer on the dataset selecting the highest performing one.
4. Concatenate the models in (2) and (3) for the multi-modal approach and evaluate the result.

4.1 Overview

For our investigation into multi-modal approaches, we first establish baseline results by recreating the multi-modal approach used by Kundu et al. [6] We planned on evaluating our image-based and text-based models individually and comparing their performance. After that, we combine the features of our image and text based models and evaluate our classification accuracy. This will be done with and without applying image pre-processing. By making these modifications incrementally, we can assess

the contribution of each technique toward the overall performance. This allows us to conclude which combination(s) of techniques yields the best overally performance.

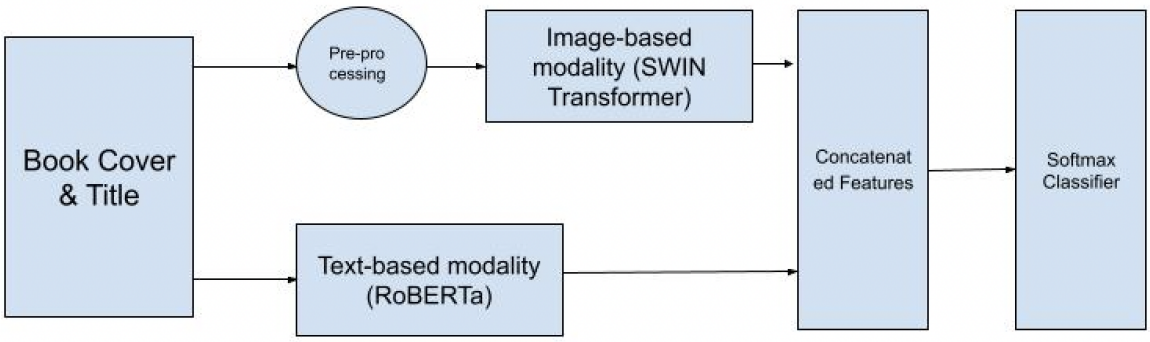

Figure 3: Block diagram of our multi-modal approach

4.2 Methodology

The multi-modal approach used by Kundu et al.[6] uses a ResNet-50 and Universal Sentence Encoder as the image-and-text-based modalities respectively, selected based off the individual performances of these models. We begin by recreating this model and evaluating it against the Book Cover dataset first introduced by Iwana et al.[4] This dataset contains 57,000 images of book covers divided into 30 classes. Each data point contains information on the cover, title, author and category/genre that the book belongs to. After establishing this baseline, we incrementally swap in our model pieces into the architecture to evaluate an improvement in performance independently.

When we perform the machine learning, we so with a training, validation, and test set. The final accuracy is based on the performance of the test set only. This is to prevent overfitting. Furthermore, to keep inline with our baseline, we did not perform data augmentation.

4.2.1 Data Pre-processing

The approach used by Kundu et al.[6] found that data pre-processing did not improve performance; in fact it reduced it. However, findings by Pal et al.[9] showcased three pre-processing techniques that improved performance of CNNs. They were mean normalization, standardization, and Zero Component Analysis (ZCA). Mean normalization is the process of shifting the image data so that it is centered around the mean value, and thereby reducing the clipping of the CNN's computation at the extremes of the data. Standardization is the process of normalization the standard deviation of each feature dimension of the data, which ensures that certain parameters are arbitrarily more sensitive, leading to most robust convergence of the CNN. Finally, ZCA applies a whitening filter to the image, which forces the model to learn higher-order correlations in the images by suppressing the lower-order correlations. We implement Principle Component Analysis (PCA), which resembles ZCA with a rotation.[1]

4.2.2 Deep Learning Models

The models that we primarily incorporate is the SWIN Transformer and RoBERTa introduced by Liu et al.[8] and Liu et al. [7] respectively. The SWIN transformer was introduced as a general purpose transformer capable of computer vision tasks. It is a more efficient vision transformer that can limit self-attention computation to non-overlapping local windows. These properties allowed it to achieve state-of-the-art performances on COCO object detection and ADE20K semantic segmentation, beating out previous best methods. RoBERTa is a commonly used large language model which also achieved state-of-the-art performances for multiple NLP tasks. Because it was pretrained on a large dataset, RoBERTa can learn from a variety of text and is capable of generalizing to new tasks. It is also relatively flexible and can be fine-tuned depending on the application. Both of these models are available on Hugging Face, including the feature extractor and tokenizer respectively.

For both our models, we train them on the BookCover30 dataset[4] individually before we apply our multi-modal approach using both models simultaneously. To combine the features of both models, we apply simple concatenation on the feature vectors and then input the concatenated vector into a linear layer with a softmax function. Since the training of each mode independently requires different hyperparameters and tuning, we feed the multi-modal approach a frozen copy of the trained modalities. Since the problem we are trying to tackle is a classification task, we evaluate our model using metrics such as precision, recall, accuracy and F-score.

4.2.3 Libraries

Our solution was implemented in Python primarily relying on the PyTorch library. Since our solution is dependent on transfer learning of transformers, we acquired pre-trained models from Hugging Face. A full listing of the libraries used can be found in the code. The computation was done with Google Colab Pro in a "Premium"-tier GPUs and "High Ram" environment. The image pre-processing steps were performed in MATLAB with the general data pre-processing done in Python.

5 Experiments and Results

5.1 Image Based

For the imaged based experiments, we tested with a standard ViT and SWIN tiny, imported from the Hugging Face library. We also ran a baseline using ResNet-50 as seen in the code repo of Kundu [5]. The Top 1% Test Accuracy can be seen in Table 1. For our initial image preprocessing, we normalized, standardized, and resized the images to fit into their respective models. The second set of data we tested on had PCA applied to the images on top of the other preprocessing.

Given our limited computing resources, when training the data only half the dataset was used (around 25000 images). The training was done with the following parameters for both models: learning rate = 0.00002, batch size = 20, epochs = 1. It should be noted that the training of the ViT was stopped before training for 1 epoch was complete. The training time so far was about 5 hours and 30 minutes. This was due to the validation loss and accuracy not improving as well as an attempt to preserve computing resources. Despite this, the ViT achieved a test accuracy of about 31.1%, whereas ResNet-50 had an accuracy of 29.6%.

When analyzing the training accuracy and loss curves of the ViT, we note that the accuracy and loss appear to begin levelling off after 2k samples. For the SWIN, we noted the sporadic nature of the training loss curve. This could possibly be due to overfitting. The validation loss curve on the other hand is smoother. We notice convergence for the accuracy and loss curves at around 4.5k samples. The time it took to train 1 epoch of the SWIN Transformer was about 3 hours and 20 minutes, over 2 hours shorter than the ViT. This could be due to the implementation of the W-MSA and SW-MSA layers which are not present in the ViT model.

The SWIN model had promising results achieving a test accuracy of 32.3% on the standard preprocessing dataset. However when applied to the dataset with PCA applied, this accuracy dropped to about 28.6%, which is less than the accuracy achieved by ResNet-50.

Model	Top-1 Accuracy (%)
ResNet-50	29.6
ViT (with normalization)	31.1
SWIN (with normalization)	**32.2**
SWIN with PCA dataset	29.1

Table 1: Imaged-Based Transformer: Top 1% Test Accuracy

	precision	recall	f1-score	support
0	0.17	0.18	0.17	190
1	0.23	0.35	0.28	190
2	0.17	0.08	0.11	190
3	0.42	0.37	0.39	190
4	0.31	0.44	0.36	190
5	0.18	0.14	0.16	190
6	0.48	0.66	0.56	190
7	0.34	0.46	0.39	190
8	0.53	0.62	0.57	190
9	0.27	0.28	0.28	190
10	0.36	0.37	0.37	190
11	0.24	0.11	0.15	190
12	0.24	0.27	0.25	190
13	0.21	0.07	0.10	190
14	0.21	0.26	0.24	190
15	0.10	0.07	0.08	190
16	0.25	0.24	0.25	190
17	0.31	0.42	0.35	190
18	0.30	0.35	0.32	190
19	0.10	0.04	0.06	190
20	0.23	0.06	0.10	190
21	0.15	0.19	0.16	190
22	0.44	0.62	0.52	190
23	0.16	0.12	0.14	190
24	0.30	0.45	0.36	190
25	0.19	0.23	0.21	190
26	0.30	0.32	0.31	190
27	0.16	0.07	0.10	190
28	0.54	0.58	0.56	190
29	0.30	0.38	0.34	190
accuracy			0.29	5700
macro avg	0.27	0.29	0.27	5700
weighted avg	0.27	0.29	0.27	5700

(a) ViT precision, recall, and f1-score by class

	precision	recall	f1-score	support
0	0.19	0.18	0.19	190
1	0.26	0.29	0.28	190
2	0.17	0.21	0.19	190
3	0.52	0.57	0.54	190
4	0.35	0.45	0.39	190
5	0.13	0.07	0.09	190
6	0.53	0.69	0.60	190
7	0.40	0.49	0.44	190
8	0.65	0.61	0.63	190
9	0.35	0.35	0.35	190
10	0.41	0.33	0.37	190
11	0.24	0.16	0.19	190
12	0.24	0.29	0.26	190
13	0.16	0.12	0.14	190
14	0.26	0.30	0.28	190
15	0.14	0.08	0.10	190
16	0.23	0.33	0.27	190
17	0.36	0.39	0.37	190
18	0.32	0.32	0.32	190
19	0.19	0.09	0.13	190
20	0.28	0.14	0.19	190
21	0.20	0.27	0.23	190
22	0.50	0.57	0.53	190
23	0.15	0.12	0.13	190
24	0.39	0.48	0.43	190
25	0.19	0.29	0.23	190
26	0.40	0.34	0.37	190
27	0.22	0.15	0.18	190
28	0.64	0.63	0.63	190
29	0.36	0.35	0.35	190
accuracy			0.32	5700
macro avg	0.31	0.32	0.31	5700
weighted avg	0.31	0.32	0.31	5700

(b) SWIN precision, recall, and f1-score by class

Figure 4: Comparison of ViT and SWIN classification reports

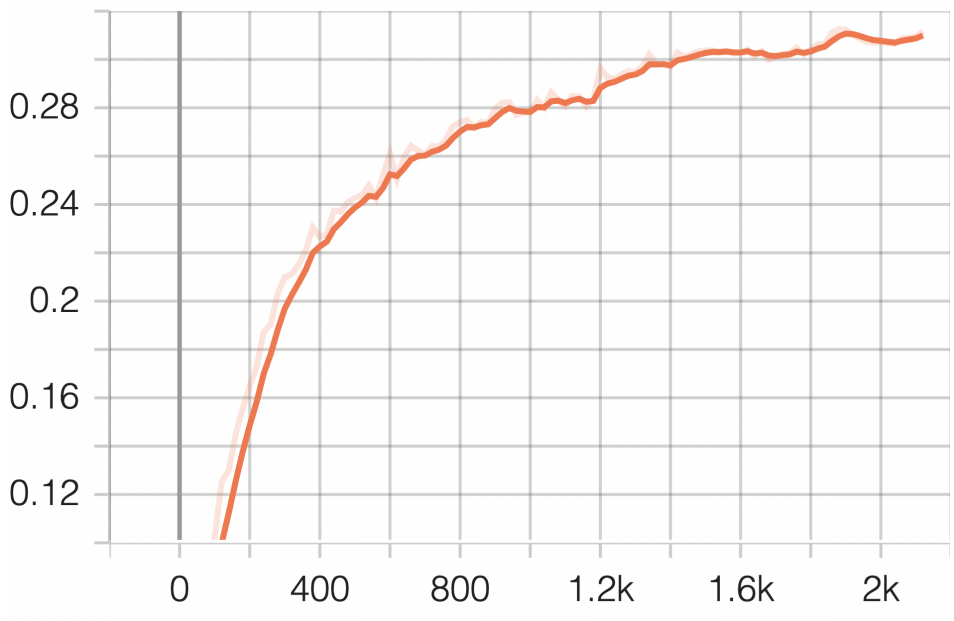

Figure 5: Validation Accuracy: ViT, 0.311

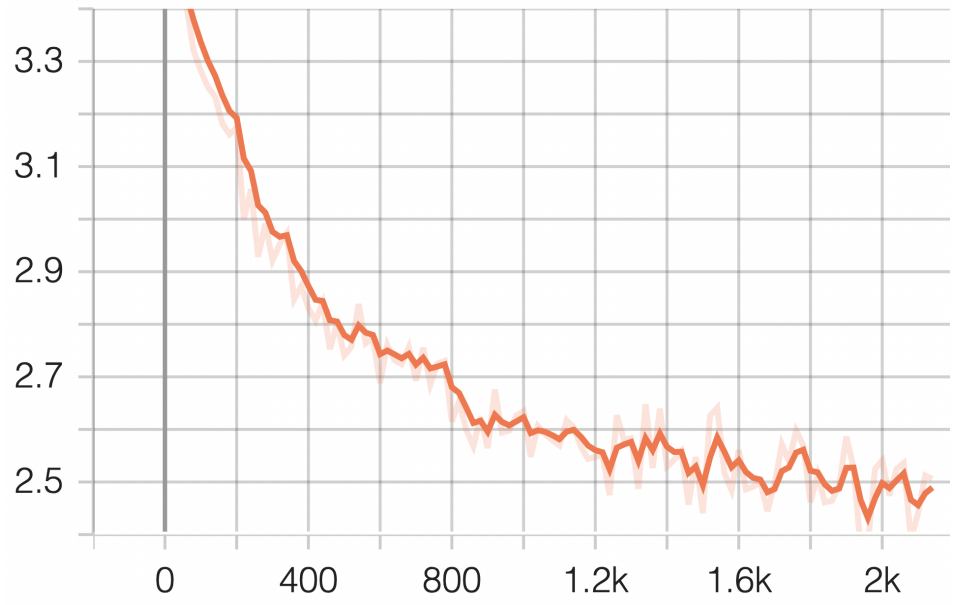

Figure 6: Training Loss: ViT, 2.46

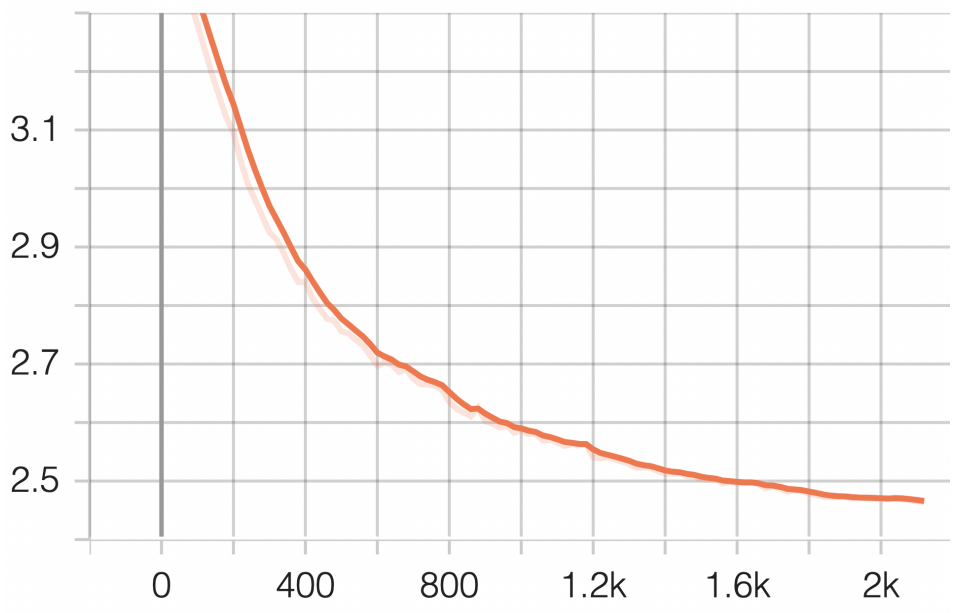

Figure 7: Validation Loss: ViT, 2.49

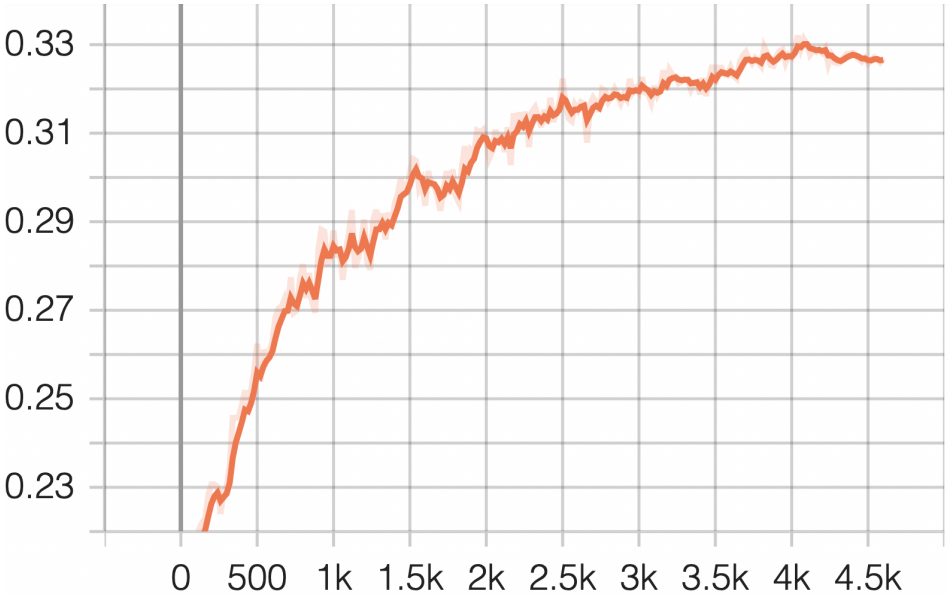

Figure 8: Validation Accuracy: SWIN, 0.326

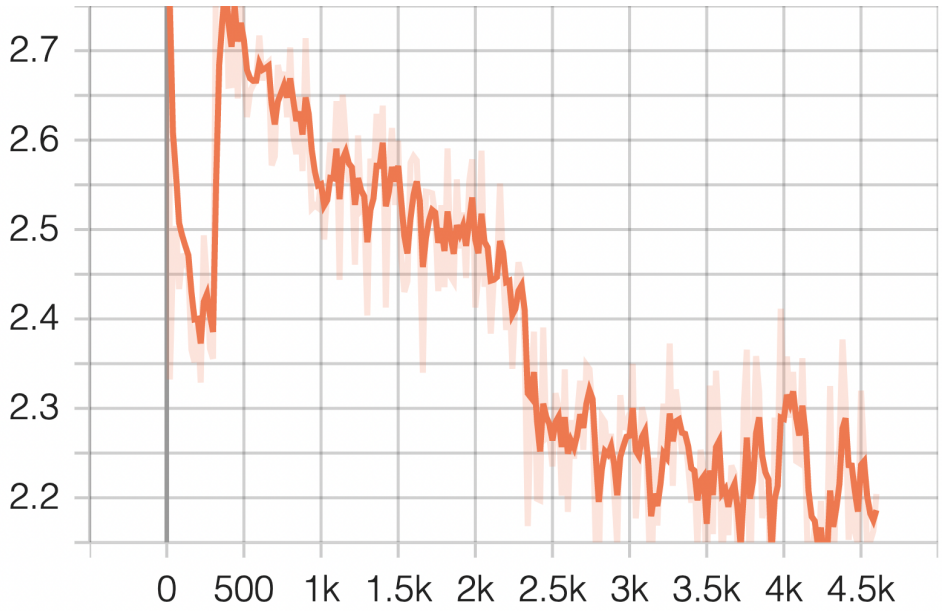

Figure 9: Training Loss: SWIN, 2.20

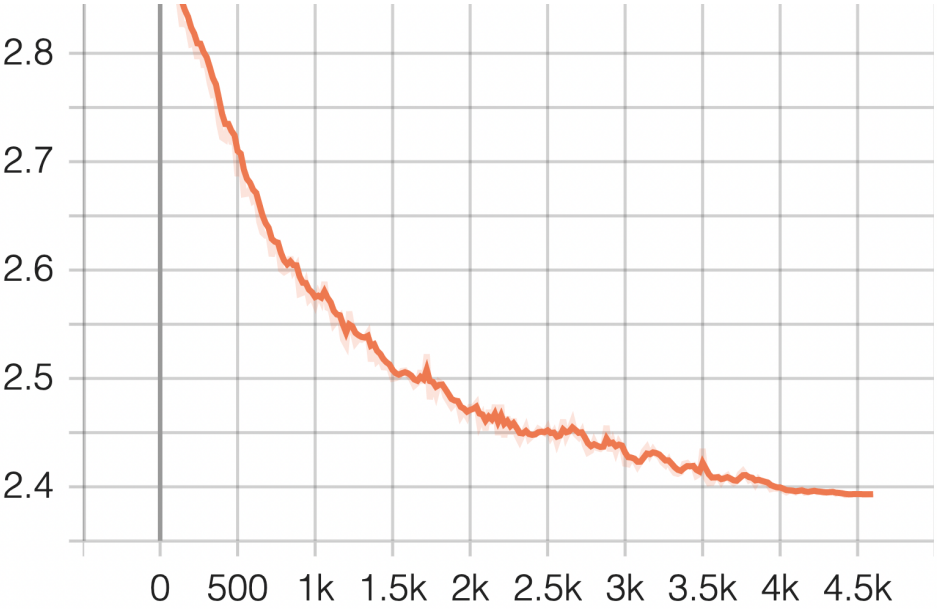

Figure 10: Validation Loss: SWIN, 2.39

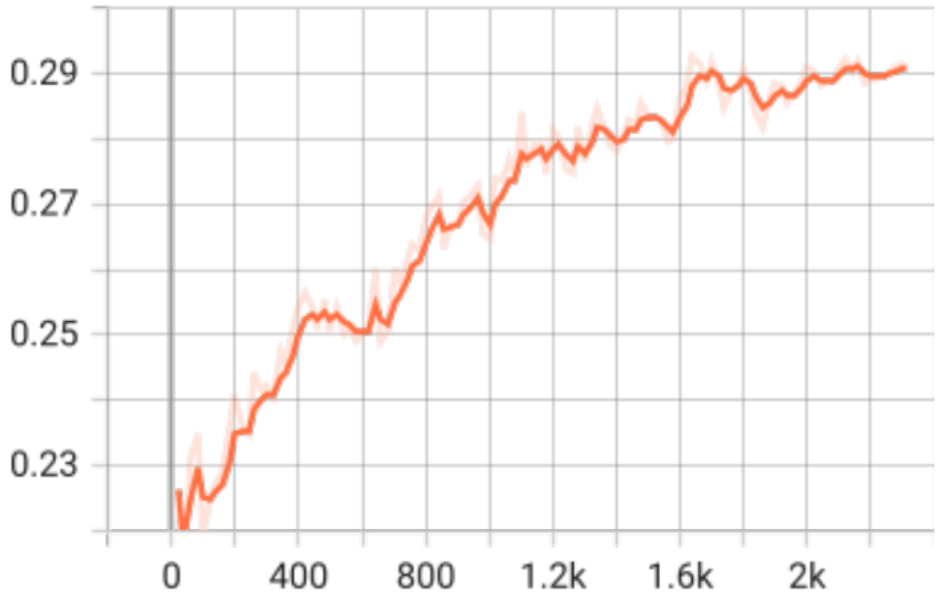

Figure 11: Validation Accuracy: SWIN w/ PCA dataset, 29.1

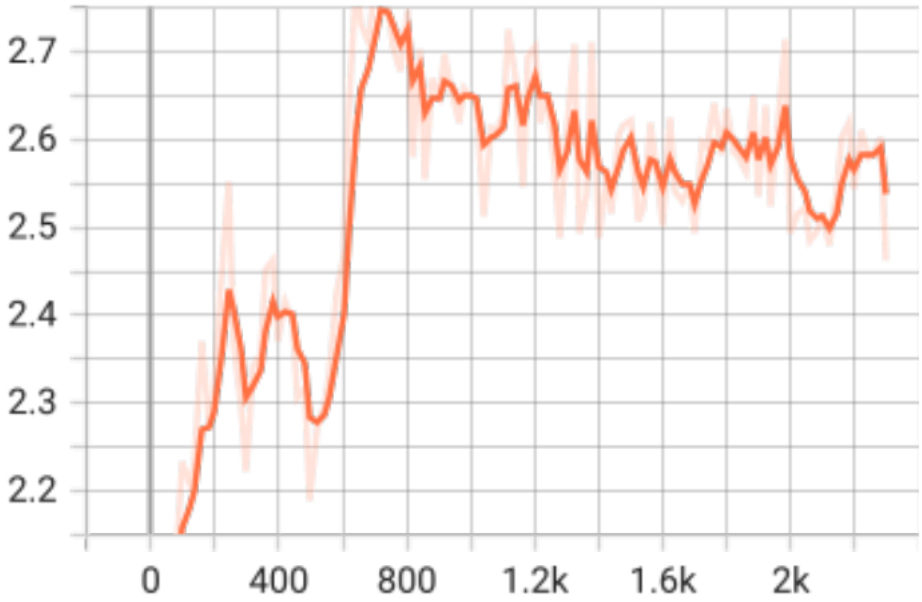

Figure 12: Training Loss: SWIN w/ PCA dataset, 2.55

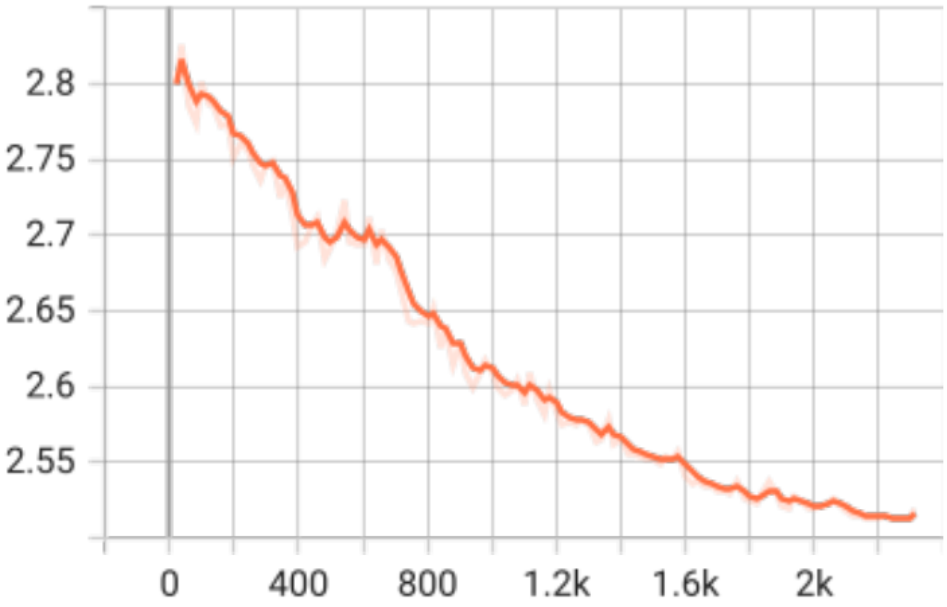

Figure 13: Validation Loss: SWIN w/ PCA dataset, 2.52

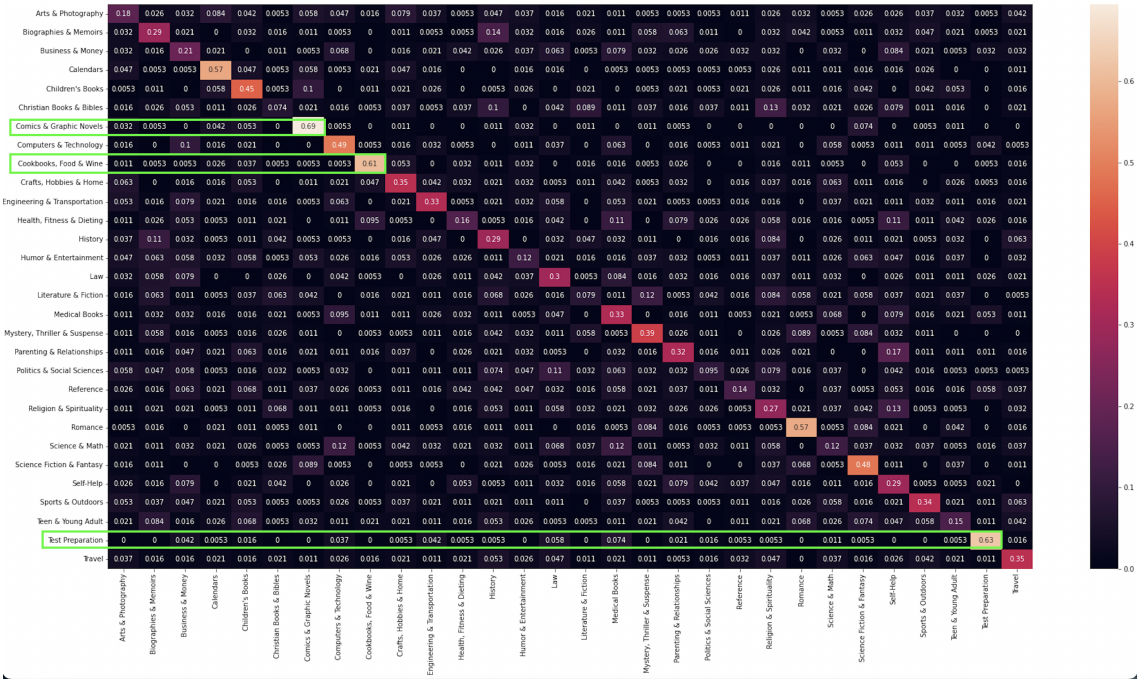

Figure 14: Confusion Matrix: ViT

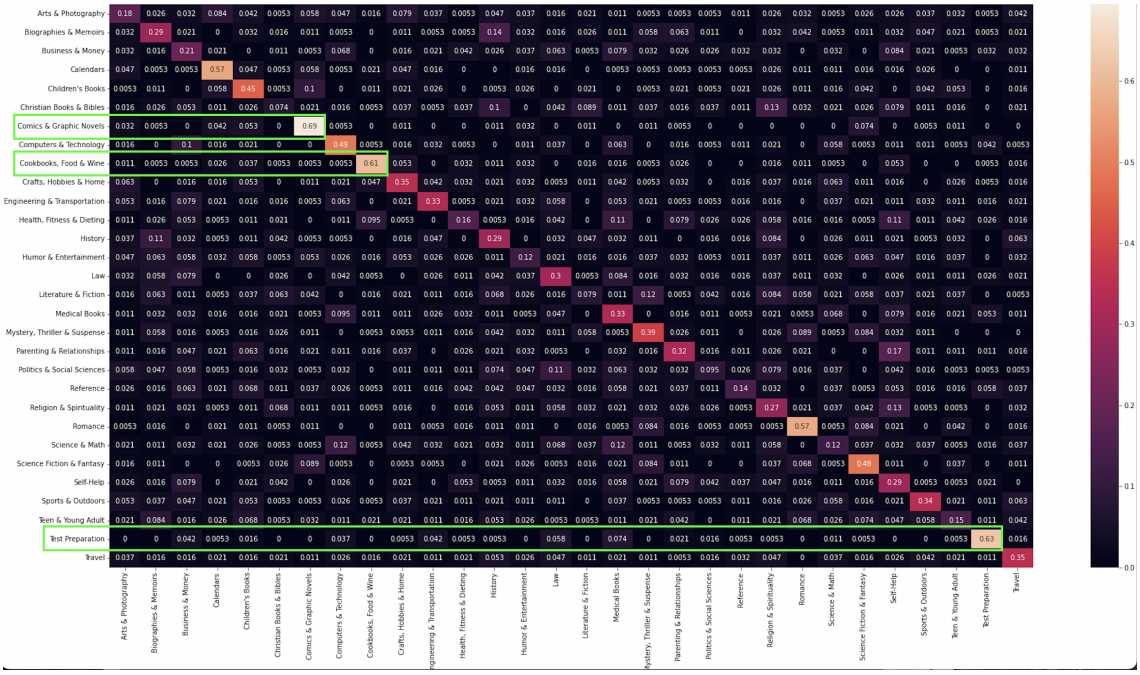

Figure 15: Confusion Matrix: SWIN

Looking at the confusion matrix for the SWIN and VIT models, we notice that 3 classes performed particularly better than others. Those classes being "Comics and Graphic Novels", "Cookbooks Food and Wine" and "Test Preparation". These genre of books tend to have more notable image features, for example a Test Preparation book having words like "SAT" or "GRE" imprinted on the cover.

5.2 Text Based

Model	Top-1 Accuracy (%)
Universal Sentence Encoder (USE)	52.6
BERT	56.5
RoBERTa	**57.7**

Table 2: Text Based Transformer Top 1% Accuracy

The text-base classification for BERT and RoBERTa included a pre-trained transformer and tokenizer. This was followed by the addition of dropout, a linear layer with relu activation, and then softmax. Both model were training on the whole dataset for 10 epochs with a learning rate of 10^{-6}. The dropout used was 50%, the loss function used was Cross Entropy and the optimizer was Adam. In our high-performance computing environment, a single epoch could be performed in under 16 minutes.

Accuracy improved epoch-over-epoch on both the training and validation sets. The loss improved epoch-over-epoch on both the training and validation sets. The plots for validation accuracy and loss are shown below.

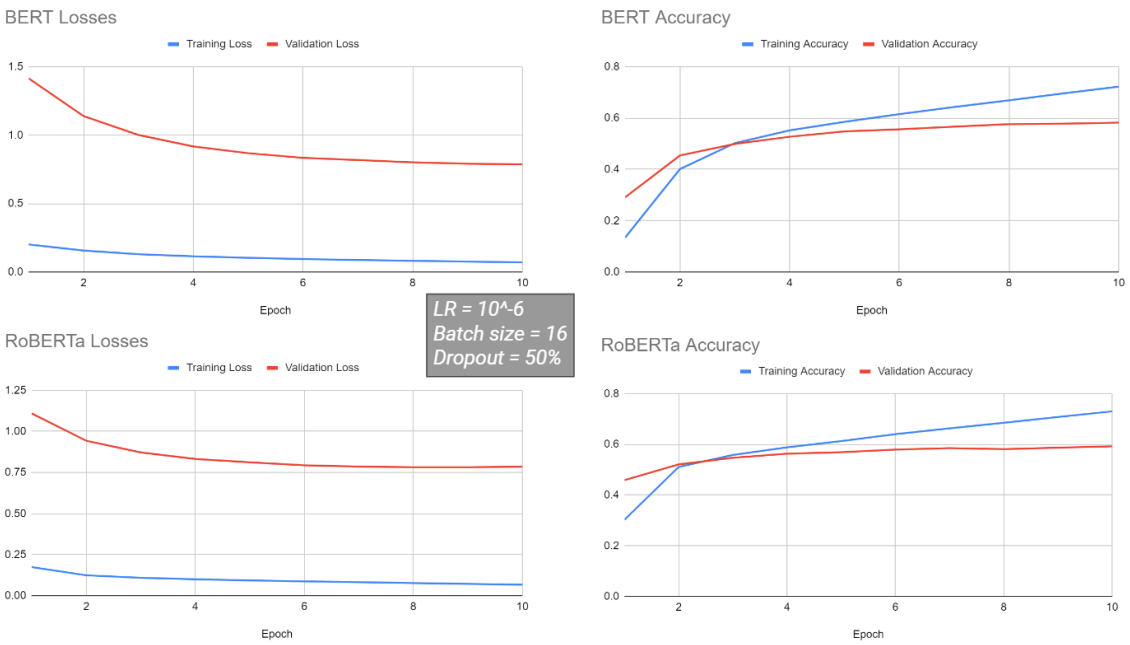

Figure 16: BERT validation loss (top left), BERT validation accuracy (top right), RoBERTa validation loss (bottom left), RoBERTa validation accuracy (bottom right)

Both these models performed very well, definitively beating the baseline by several percentage points. In fact, the models individually outperformed the multi-modal model of Kundu et al.

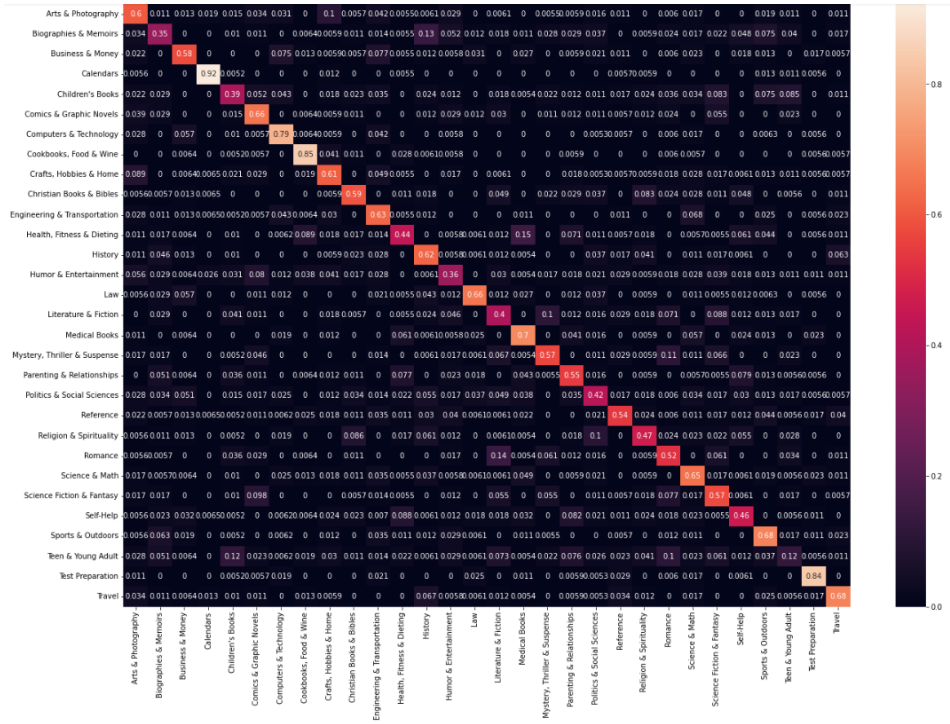

Figure 17: BERT confusion matrix across all classes

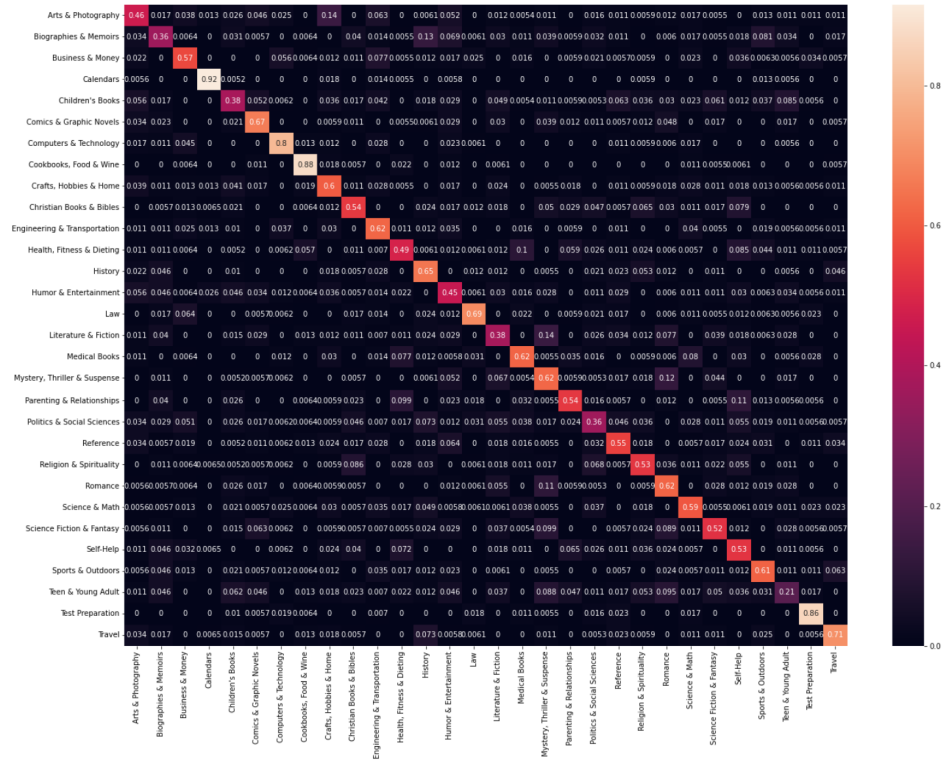

Figure 18: RoBERTa confusion matrix across all classes

An inspection of these confusion matrices highlights which classes are easily identifiable to the classifier by their text and which are more ambiguous. Looking at the popular classifications reveals that the classifier identified a closely-related genre. Some examples include History vs. Biographies & Memoirs, Teen & Young Adult vs. Childrens' Books, among others. This finding highlights the rather subjective nature of genres. There is no quantitative way to determine a book's genre and this dataset's methodology to split it into these genres could be disputed.

	precision	recall	f1-score
0	0.52	0.60	0.56
1	0.40	0.35	0.37
2	0.62	0.58	0.60
3	0.91	0.92	0.92
4	0.47	0.39	0.43
5	0.57	0.66	0.61
6	0.69	0.79	0.74
7	0.76	0.85	0.80
8	0.57	0.61	0.59
9	0.64	0.59	0.62
10	0.54	0.63	0.58
11	0.52	0.44	0.47
12	0.52	0.62	0.56
13	0.44	0.36	0.40
14	0.74	0.66	0.70
15	0.39	0.40	0.39
16	0.60	0.70	0.64
17	0.61	0.57	0.59
18	0.52	0.55	0.53
19	0.44	0.42	0.43
20	0.64	0.54	0.58
21	0.55	0.47	0.51
22	0.47	0.52	0.49
23	0.54	0.65	0.59
24	0.48	0.57	0.53
25	0.50	0.46	0.48
26	0.59	0.68	0.63
27	0.25	0.12	0.17
28	0.82	0.84	0.83
29	0.70	0.68	0.69
accuracy			0.57
macro avg	0.57	0.57	0.57
weighted avg	0.56	0.57	0.56

(a) BERT's precision, recall, and f1-score by class

	precision	recall	f1-score
0	0.39	0.32	0.35
1	0.28	0.25	0.26
2	0.53	0.43	0.48
3	0.84	0.88	0.86
4	0.27	0.04	0.06
5	0.38	0.45	0.41
6	0.54	0.84	0.66
7	0.68	0.93	0.79
8	0.41	0.51	0.46
9	0.44	0.47	0.45
10	0.45	0.52	0.48
11	0.39	0.36	0.37
12	0.39	0.60	0.47
13	0.21	0.03	0.06
14	0.60	0.69	0.64
15	0.23	0.22	0.22
16	0.47	0.69	0.56
17	0.32	0.57	0.41
18	0.44	0.52	0.47
19	0.44	0.09	0.15
20	0.49	0.37	0.42
21	0.40	0.37	0.38
22	0.24	0.18	0.21
23	0.56	0.56	0.56
24	0.38	0.51	0.43
25	0.37	0.37	0.37
26	0.45	0.63	0.53
27	0.00	0.00	0.00
28	0.69	0.78	0.73
29	0.57	0.75	0.64
accuracy			0.46
macro avg	0.43	0.46	0.43
weighted avg	0.43	0.46	0.42

(b) RoBERTa's precision, recall, and f1-score by class

Figure 19: Comparison of BERT and RoBERTa classification reports

5.3 Multi-Modal

The multi-modal approach was implemented by loading in a saved version of the trained SWIN transformer and trained RoBERTa transformer. Although we considered training the whole network together, the issue presented itself that the text modality takes several epochs to reach high performance, which is an amount of computation that is too cumbersome for the image modality. The multi-modal model loaded in our saved pretrained modalities, concatenated their output tensors and fed it into a fully-connected layer with ReLU activation, followed by softmax. The models were frozen using a PyTorch method which disables the gradient calculation for certain layers.

The computation of the multi-modal model was so demanding that our results were regularly hindered by the 83.5 GB RAM limit on Google Colab Pro. This led to several crashes and significant amount of paid compute resources. To our surprise, we still managed to train a single epoch on the dataset, resulting in a modest test accuracy of 31.1%, which is lower than either modality on its own.

Epoch: 1 — Training Loss: 2.8376 — Training Accuracy: 0.2578 — Validation Loss: 2.4483 — Validation Accuracy: 0.3405

Furthermore, we were unable to download the PyTorch model data with the determined weights and biases in the architecture for further testing and reproducibility - the file size was 581 MB and the Colab UI was unresponsive to the Download command.

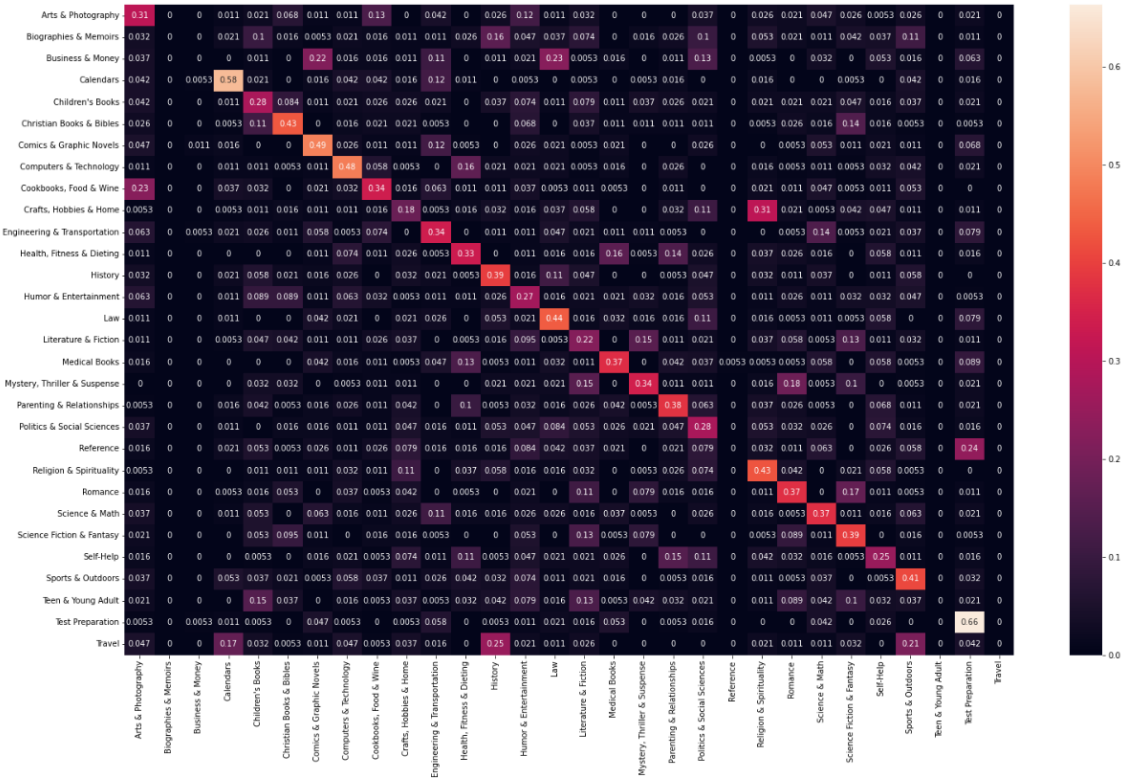

Figure 20: Multi-modal confusion matrix across all classes

	precision	recall	f1-score
0	0.25	0.31	0.28
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.55	0.58	0.56
4	0.21	0.28	0.24
5	0.41	0.43	0.42
6	0.41	0.49	0.44
7	0.41	0.48	0.44
8	0.34	0.34	0.34
9	0.19	0.18	0.18
10	0.28	0.34	0.31
11	0.31	0.33	0.32
12	0.30	0.39	0.34
13	0.19	0.27	0.22
14	0.33	0.44	0.38
15	0.15	0.22	0.18
16	0.41	0.37	0.39
17	0.40	0.34	0.37
18	0.35	0.38	0.36
19	0.20	0.28	0.23
20	0.00	0.00	0.00
21	0.33	0.43	0.37
22	0.32	0.37	0.34
23	0.33	0.37	0.35
24	0.30	0.39	0.34
25	0.24	0.25	0.25
26	0.30	0.41	0.34
27	0.00	0.00	0.00
28	0.41	0.66	0.51
29	0.00	0.00	0.00
accuracy			0.31
macro avg	0.26	0.31	0.28
weighted avg	0.26	0.31	0.28

Figure 21: The multi-modal model's precision, recall, and f1-score by class

It's clear that certain neural paths did not get a chance to learn certain weights and biases, since several classes don't get predicted. The learning rate use was 10^{-4}. More epochs are needed to see what the loss function would truly converge to and what the final test accuracy could be. We wish we had more computational resources to find out what that would be, although we consider ourselves lucky to have a run where the environment didn't crash.

Model	Top-1 Accuracy (%)
ResNet + USE	**56.1**
SWIN + RoBERTa	31.1*

Table 3: Multi-modal Top 1% Test Accuracy

6 Conclusions and Future Work

Overall, our models outperformed the baseline, with three out of four transformers outperforming their respective modality. The text-based modality alone outperformed the concatenation of Kundu et al. This is a significant achievement and an additional piece of evidence that transformers can play a role

in reaching much higher levels of performance in both image-based and text-based classification tasks. That being said, we believe is there significant room for even further improvement in this field, as follows:

1. **Consideration for increased computational resources into training our current models.** The computational demands of training the image transformers was so overwhelming that they couldn't get more than a single epoch of training. This was with reasonably high computing power. For this reason, the current architecture may be capable of performance that is much higher than what we achieved.

2. **Larger and cleaner dataset.** BookCover30 contains 51,300 images and titles. However, there are tens of millions of published books in the world. Furthermore, the base transformers we used were originally trained on a much larger dataset. Finally, BookCover30's methodology for selecting a genre is slightly arbitrary; the associated genre is the first one listed for that book on Amazon, even when multiple genres are listed. This should either suggest that genre classification be a multi-label problem, or that the dataset contain books which purely fall into a particular genre. There is also the not-insignificant issue that several files were poorly formatted or misformatted and led to significant wasted computational time.

3. **Additional research into the role of image preprocessing for image classification.** The use of PCA did not improve the performance of the SWIN transformer. Whether this is preprocessing technique is ideal for CNNs only is not clear. For this reason, our findings add on to the conflicting findings of Kundu et al. that preprocessing didn't improve performance and Pal et al. that preprocessing did improve performance.

6.1 Division of work

1. Topic brainstorming (Aaron & Justin)
2. Proposal (Aaron & Justin)
3. Baseline analysis (Aaron & Justin)
4. Data pre-processing (Justin(image), Aaron(text))
5. Image-based classification (mostly Justin)
6. Text-based classification (mostly Aaron)
7. PCA pre-processing (mostly Aaron)
8. Final presentation (Aaron & Justin)
9. Multi-modal classification (Aaron & Justin)
10. Final report (Aaron & Justin)

References

[1] Anthony Bell and Terrence Sejnowski. *Edges are the 'Independent Components' of Natural Scenes*. 1994. URL: https://www-n.oca.eu/Bijaoui/doc_ab/cardon/edge.pdf.

[2] Ganeshprasad Biradar et al. *Classification of Book Genres using Book Cover and Title*. June 2019. DOI: 10.1109/ICISGT44072.2019.00031.

[3] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2020. DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929.

[4] Brian Kenji Iwana et al. *Judging a Book By its Cover*. 2016. DOI: 10.48550/ARXIV.1610.09204. URL: https://arxiv.org/abs/1610.09204.

[5] Chandra Kundu. *BOOK GENRE CLASSIFICATION BY ITS COVER USING MULTI-VIEW LEARNING APPROAC*. 2020. URL: https://github.com/chandrakundu/multiview-learning-book-genre-classification.

[6] Chandra Kundu and Lukun Zheng. *Deep multi-modal networks for book genre classification based on its cover*. 2020. DOI: 10.48550/ARXIV.2011.07658. URL: https://arxiv.org/abs/2011.07658.

[7] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. DOI: 10.48550/ARXIV.1907.11692. URL: https://arxiv.org/abs/1907.11692.

[8] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. DOI: 10.48550/ARXIV.2103.14030. URL: https://arxiv.org/abs/2103.14030.

[9] Kuntal Kumar Pal and K.S. Sudeep. *Preprocessing for image classification by convolutional neural networks*. 2016. DOI: 10.1109/RTEICT.2016.7808140. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7808140.

[10] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.