Judging a book by its cover
(COL774: Machine Learning)
Prof. Parag Singla

Tushar Verma (2022AIY7514)
ScAI, IIT Delhi

1 Introduction

The project involves utilizing a book cover dataset that includes both images of book covers and their corresponding titles to develop a machine learning model capable of predicting the genre of a book. The primary objective is to create a model that accurately classifies books into predefined genre categories based on visual and textual features.

2 Dataset

The dataset [1] consists of a folder named `Images`, which contains all the required images. The image names are read from the corresponding CSV files, and the corresponding images are retrieved from the `Images` folder. The goal of the model is to generate the `comp_test_y.csv` file in the same format as `train_y.csv` and `non_comp_test_y.csv`.

2.1 Files

The following files are part of the dataset:

- `train_x.csv` - the training set with columns `Image_Name`, `Title`
- `train_y.csv` - the training set with columns `Genre label`
- `non_comp_test_x.csv` - the non-competitive test set with columns `Image_Name`, `Title`
- `non_comp_test_y.csv` - the non-competitive set with columns `Genre label`
- `comp_test_x.csv` - the competitive test set with columns `Image_Name`, `Title`

The ultimate goal is to predict the `comp_test_y.csv` file.

2.2 Genre Labels and Corresponding Names

These genre names provide additional context about the data. Ultimately, the model's objective is to predict the genre labels.

Genre Label	Genre Names
0	Information Technology
1	Crafts and Hobbies
2	Romance
3	Comics
4	Bibles
5	Medicine
6	Engineering
7	Parenting
8	Reference
9	Health & Fitness
10	Self-help
11	Sports
12	Maths and Science
13	History
14	Politics
15	Calendars
16	Law
17	Religion
18	Test Preparation
19	Biographies
20	Humor
21	Young Adult
22	Cookbooks
23	Business
24	Sci-fi
25	Children's books
26	Photography
27	Literature
28	Travel
29	Mystery

Table 1: Table lists the genre labels along with their corresponding names

3 Model

We used a **ResNet-Bert** model is a multi-modal architecture that effectively integrates textual and visual information. By combining the strengths of BERT for text classification and ResNet for image classification, the model can handle tasks that require understanding both images and their corresponding textual descriptions, such as predict-

ing the genre of a book based on its cover image and title.

Layer Type	**Output Size**	**Notes**
Image Input	Variable (C, H, W)	RGB image input.
ResNet18	(256,)	Feature extraction from image.
BERT Input	Variable	Token IDs and attention mask.
BERT	(768,)	Contextual text representation.
Dropout	(768,)	Prevents overfitting.
Linear (BERT)	(256,)	Reduces BERT output.
Concatenate	(512,)	Combines image and text features.
Linear (Classifier)	(30,)	Outputs for 30 genres.

4 Results

Class	Precision	Recall	F1-Score	Support
0	0.70	0.80	0.75	1134
1	0.54	0.58	0.56	1155
2	0.43	0.60	0.50	1148
3	0.52	0.69	0.60	1116
4	0.35	0.60	0.44	1133
5	0.65	0.74	0.69	1153
6	0.69	0.52	0.60	1132
7	0.47	0.50	0.49	1132
8	0.65	0.39	0.49	1149
9	0.49	0.53	0.51	1144
10	0.40	0.42	0.41	1129
11	0.64	0.69	0.66	1135
12	0.62	0.65	0.64	1175
13	0.49	0.64	0.56	1107
14	0.50	0.22	0.31	1098
15	0.87	0.95	0.91	1149
16	0.63	0.82	0.72	1136
17	0.40	0.36	0.37	1121
18	0.78	0.79	0.78	1146
19	0.47	0.11	0.17	1156
20	0.43	0.15	0.22	1149
21	0.35	0.02	0.04	1100
22	0.72	0.91	0.80	1116
23	0.63	0.65	0.64	1158
24	0.44	0.47	0.46	1172
25	0.48	0.19	0.27	1132
26	0.46	0.44	0.45	1135
27	0.30	0.32	0.31	1168
28	0.59	0.78	0.67	1169
29	0.38	0.67	0.48	1153

Table 2: Classification Report Training Data

4.1 Performance on Training Data

Accuracy: The overall accuracy of the model is 54%. This means that the model correctly classifies about half of the instances. This is relatively low, especially for a classification task, suggesting that there may be issues with the model, the data, or both.

4.2 Class-Level Insights

- **Class 0:** Shows the best performance with a precision of 0.70 and recall of 0.80. This indicates that the model does a good job of identifying this class and is relatively reliable.
- **Class 15:** Has a very high precision of 0.87 and a recall of 0.95, indicating excellent performance for this class, making it the best-performing class overall.
- **Class 21:** Exhibits very poor results with precision of 0.35 and a recall of 0.02, suggesting that almost all predictions for this class are incorrect. This indicates a severe issue, possibly due to an imbalanced dataset or insufficient examples of this class.
- **Class 19:** Also shows very low performance with a precision of 0.47 and a recall of 0.11, indicating it is rarely predicted correctly.

4.3 Averages

- **Macro Average:** All classes are treated equally. Here, precision, recall, and F1-score are all around 0.54, suggesting an average performance across classes.
- **Weighted Average:** This takes into account the support of each class. The weighted averages remain around 0.54, indicating that the model performs similarly across different classes, with no significant improvements from the macro average due to the imbalanced support.

4.4 Performance on Test Data

Overall Accuracy: The overall accuracy of the model is **48%**. This indicates that nearly half of the predictions made by the model are correct, but there is considerable room for improvement.

Class-wise Performance:

- **Precision**: The average precision across all classes is **49%**. This metric indicates how many of the predicted positive instances were actually positive. Classes like **15** (0.89) and **22** (0.73) have high precision, suggesting that when the model predicts these classes, it is likely to be correct. In contrast, classes like **21** (0.38) have low precision, indicating many false positives.
- **Recall**: The average recall across all classes is **49%**. This metric measures how many of the actual positive instances were correctly identified by the model. The model performs particularly well in class **15** (0.95), which means it successfully identifies almost all instances of this class. However, classes such as **19** (0.10) and

Class	Precision	Recall	F1-Score	Support
0	0.61	0.77	0.68	162
1	0.44	0.56	0.49	189
2	0.33	0.49	0.39	184
3	0.56	0.61	0.58	216
4	0.29	0.52	0.37	184
5	0.55	0.71	0.62	150
6	0.69	0.49	0.58	209
7	0.39	0.51	0.44	168
8	0.61	0.28	0.38	198
9	0.44	0.42	0.43	199
10	0.37	0.38	0.38	217
11	0.55	0.62	0.58	185
12	0.52	0.55	0.53	186
13	0.46	0.51	0.48	203
14	0.39	0.16	0.23	177
15	0.89	0.95	0.92	194
16	0.64	0.77	0.70	193
17	0.28	0.27	0.27	198
18	0.74	0.79	0.76	175
19	0.44	0.10	0.16	202
20	0.39	0.16	0.23	190
21	0.38	0.02	0.05	204
22	0.73	0.87	0.80	212
23	0.59	0.55	0.57	188
24	0.30	0.39	0.34	158
25	0.49	0.17	0.25	200
26	0.42	0.36	0.38	198
27	0.25	0.28	0.26	197
28	0.48	0.72	0.58	165
29	0.38	0.66	0.48	199

Table 3: Detailed Classification Test Data

21 (0.02) have very low recall, indicating that the model fails to capture most of the actual instances for these classes.

- **F1-Score**: The average F1-score across all classes is **46%**. This score is a harmonic mean of precision and recall, providing a balance between the two. Classes like **15** (0.92) and **22** (0.80) excel in F1-score, while others, such as **21** (0.05), struggle significantly.

Macro vs. Weighted Averages:

- **Macro Average**: This treats all classes equally, regardless of their support (the number of true instances for each class). The macro average precision, recall, and F1-score are all around **49%** to **46%**, indicating a balanced performance across classes.

- **Weighted Average**: This takes into account the number of instances in each class, which may skew results towards classes with larger support. The weighted averages are similar to the macro averages, reflecting a consistent performance across the dataset.

5 Conclusion

- The model demonstrates varying performance across classes, with some classes performing well while others perform poorly.
- The overall accuracy is modest, and the low performance for certain classes, particularly Class 21 and Class 19, indicates potential issues in data representation or model training.
- Consider investigating class imbalances, refining the model, or augmenting the dataset to improve performance, particularly for underrepresented classes.

5.1 Future works for Improvement

Class Imbalance: certain classes have significantly more instances than others, it can skew the model's performance. Consider techniques such as:

- **Oversampling** the minority classes or **undersampling** the majority classes.
- Using **class weights** during training to give more importance to underrepresented classes.

Model Architecture:

- Exploreing different model architectures or hyperparameter tuning to improve performance, especially for classes with low precision and recall.
- Experimenting with ensemble methods or transfer learning from pre-trained models, which can sometimes yield better results.

Data Augmentation: Increasing the diversity of your training data through augmentation techniques may help the model generalize better, especially for underrepresented classes.

References

[1] V. B. Chirag Mohapatra, "Judging a book by its cover," 2022.