Can you Judge a Book by its Cover?

Sigtryggur Kjartansson
Stanford University
sigkj@stanford.edu

Alexander Ashavsky
Stanford University
ashavsky@stanford.edu

Abstract

Judging a book by its cover is an old adage that warns against evaluating the merit of something based strictly on its outward appearances. However, taken literally, we set out to see if we can in fact judge a book by its cover, or more specifically by its cover art and title.

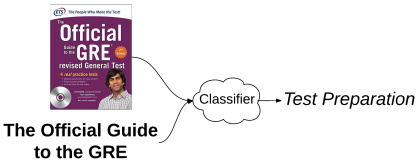
On one hand, we will explore several different neural network approaches for predicting the genre from the cover image alone, such as Fully-Connected networks, Conv Pool networks, VGGNet, SqueezeNet and ResNet. On the other hand, we will explore some simple text classification algorithms such as Multinomial Naive Bayes and SVM classifiers, in addition to Neural Networks using a Bag-Of-Words text representation and CNN-based methods for predicting the genre from the title text alone. Ultimately, we find that ResNet performs the best on the image classification task, while Fully-Connected network using a Bag-of-Words representation performS the best on text classification. Overall, as we expected, text classifiers perform the best on the task though image classifiers come close. The best overall model was one trained with a softmax gating network on top of the best ensembles of each classifier type.

1. Introduction

The cover of a book is usually the first contact the reader has with the literature, and therefore makes an important impression that shapes the readers expectations. Understanding this impression is important for publishers when marketing the book, and is also useful for merchants when categorizing books. This problem could also be extended to other domains, such as album covers and movie trailers.

1.1. Problem Description

Our goal is to determine a book's genre by information from its cover. Specifically, given a book's cover artwork and/or title, we will attempt to predict its genre.



2. Related Work

The set of related studies we investigated can be broken down by classification task; image or text, and by learning type; classical machine learning and neural networks.

2.1. Image Classification

The first type of networks we examined were fully connected neural networks (NN) for image classification. We saw examples where networks with as few as two layers were able to accurately classify movie genres from posters [16], as well as a study that was able to get an accuracy of over 90% on a three-category classification task [18]. The advantage of these networks is that all the neurons between layers are connected, allowing it to learn deeper features as layers are added. However, they are not as well suited for the task of image classification as Convolutional Neural Networks (CNN), which aim to learn features locally and as such handle slight perturbations in pixel value well.

Our investigation of CNNs lead to a few studies very similar to ours, with one also attempting to predict the genre of a book strictly by its book cover [9]. One major difference between our study and Iwana et al., 2016 is the breadth of networks applied to the task, in addition to ensembles, and gating mechanisms between models, and text classification as discussed later in this section. It should be noted that we also used the same dataset as the one used by Iwana et al, which is why it served as such a good early benchmark for ours. Another CNN experiment in a similar domain to ours attempted to identify a movie's genre from its trailer [5] and provided hope for the potential success for this approach, as it was able to beat out previously established benchmarks using document similarity and neural networks.

Finally, we went through numerous studies that did not

use deep learning as well. The first notable study we investigated was a good introduction into how Multinomial Naive Bayes (MNB) can be applied to image classification. [21]. One pattern we saw here was how important image preprocessing is in image classification when not using a deep learning approach. Similarly, in a study that attempted to leverage machine learning for artwork recognition, it found that using Naive Bayes to find important features, and then using an SVM classifier to separate artwork by artist, performed reasonably well. [1] Finally, we we examined a study aiming to classify album genre using only the album cover [19]. Here the researchers used document similarity, identified what the most representative images of each genres looked like, and applied various distance measures to evaluate similarity.

2.2. Text Classification

First, we investigated some non-neural net approaches. The first such study we used to guide our implementation of multinomial naive bayes (MNB) for text classification. Here, the author transformed the input data into vector representations using summary statistics (sentence and document frequency). [12] Another approach we found useful used both a video's captions as well as its actual content to improve genre classification. [25] Specifically, the authors used document similarity techniques on the video captions, as well as on cosine transformations of the video, with the best model using a weighting between the models to make classifications.

For deep learning, we first analyzed recurrent neural networks (RNN). One such paper used a RNN model with attention to classify music. While the authors achieved good results, it was a bit different in that it relied on time series data, and as such the structure of the input was important, which we did not believe would apply to our title data. [24]

We also examined approaches using CNNs for text classification, transforming the titles to vectorized representations that can be convulsed on. In one such study that had strong results, each word was projected to an encoding vectors with static and non-static (learnable) entries. [13] The final paper we examined tried to use different representations of text with both deep and shallow features, finding the more shallow representation actually performed better when inputted into a CNN, something we kept in mind throughout our research. [11]

2.3. Techniques

Two other powerful techniques we applied to speed up and enhance our results were Domain Transfer [4] and Ensembling [7]. The specific paper we reference for domain transfer showed that genre classifiers can be used across multiple topics. We made use of this knowledge in using networks pretrained on ImageNet [3], applying them to classification tasks on a different dataset with different labels. Also, given the number of models we trained, using an ensemble was a natural approach for our problem. We explored the approach discussed in this paper of using local minima along the approach, as well as ensembling various models from popular implementations.

In conclusion we analyzed an experiment that used a voting scheme between the best models for the final classifications. It combined four relatively lower power models, showing that when properly calibrated in a weighted voting scheme, they could produce powerful results. [26][10]. We applied a similar voting scheme between models (text and images) for our final classification predictions.

3. Methods

Our study relies on a variety of methods for both image and text classification. We develop a model using ensembles of the best methods from each classification task, finally combining both these models using a gating mechanism to compute final class probabilities.

3.1. Fully-Connected Network

The first model we developed was a fully connected neural network that has two fully-connected layers with 4,096 units each, batch normalization and a ReLU activation. We use a cross-entropy loss function as our basis for evaluation, computing normalized class probabilities to make our predictions.

3.2. Conv Pool Network [17]

Our next approach was a convolutional neural network, noted for achieving state-of-the-art results for image classification and other image related tasks. Our implementation has two convolutions layers with two convolutions, batch normalization and ReLU activation, followed by max pooling. Each convolution has filter size 3×3, 64 filters and stride 1. Finally, the output is fed into a fully-connected layer with 4,096 units to computer class probabilities.

3.3. SqueezeNet

SqueezNet [8] is a popular CNN architecture notable for achieving results on par with AlexNet with 50x fewer parameters and 510x smaller in size. The key insight in this model is its use of squeeze layers that down-sample the filter space as the image passes through the network. We used an implementation with pretrained weights [23], retraining the top 10 layers on our data.

3.4. VGG16

Our VGG-16 is a slight variant on Simonyan & Zisserman 2014 [22] where we replace the fully-connected layers with global average pooling. We use a model pre-trained on

Imagenet [3] and only re-train the top 10 layers. VGG was a natural next progression as it is twice as deep as AlexNet by leveraging the use of 3x3 filters, and achieving better error rates in the Imagenet challenge.

3.5. ResNet50

ResNet50 [6] is the most state-of-the-art CNN architecture we implemented, having more than 3x the layers of our vgg model. A known issue with deeper models is that the number of parameters tend to increase almost linearly with depth. As such, they become harder optimize, resulting in sometimes poor test results. However, the ResNet model overcomes this by using "residual" mappings from shallower layers, allowing it to take advantage of the deeper structure and additional nonlinearities, while not increasing training difficulty significantly.

3.6. SVM

The first text classifier we implemented was a Support Vector Machine (SVM), using a hinge loss function to evaluate our error. [2] A strength of this model is there are a few parameters you can tune (most notably the margin) in finding the best model. We used results from this model as a baseline for subsequent text classifiers, as we expect a neural network architecture to perform better.

3.7. Multinomial Naive Bayes

Though an older approach, multinomial naive bayes is still one of the best models for genre classification of text. Specifically, in data preprocessing we convert the book titles into vectorized representations using occurrence count, and decreasing the scores for words that do not add much to the classification decision. [12]

3.8. Text CNN

We use CNNs on the titles of the books in addition to the cover images. In a study done by Kim 2014 [13] it was shown that it is possible to achieve high accuracy using pre-trained vectors to represent features of different words in a CNN architecture. Similarly, we use previously learned embedding matrix to encode the titles, then feeding them into our convolutional network.

3.9. Two Layer FC-Net with Bag-of-Words

Similar to the naive bayes approach, using a bag-of-word (BOW) approach loses grammatical information in favor of summary statistics. Each title is converted into a dictionary where each key is a word and each value refers to the frequency of that word. These vectors are then inputted into a fully connected neural network for training similarly for prediction on the test set.

4. Dataset

We use a dataset of 19,000 book covers and titles, and 10 human-curated genres, collected from Amazon.com, Inc.

This dataset is adapted from `BookCover30` used by Iwana and Uchida 2016 [9] (https://github.com/uchidalab/book-dataset). A modification we will be making to this dataset is we will only use 10 categories, which we believe will be representative of the dataset and will significantly reduce training time.

We refer to the image-only dataset as `BookCover10` and the text-only dataset as `BookTitle10`.

The ten genres are *Children's Books*, *Comics & Graphic Novels*, *Computers & Technology*, *Cookbooks, Food & Wine*, *Romance*, *Science & Math*, *Science Fiction & Fantasy*, *Sports & Outdoors*, *Test Preparation*, and *Travel*. Each genre has 1539 training samples, 171 validation samples and 190 test samples.

We analyzed the text data and saw that 90% of the titles were between 1-20 words in both the train and test data. The training data had a total of 168,498 words and 18,555 unique words. Only about the first $\approx$ 8000 words occurred more than once. We also plotted the log-log word frequencies and ranks, and observed that the both the training and test data followed a fairly Zipfian distribution.

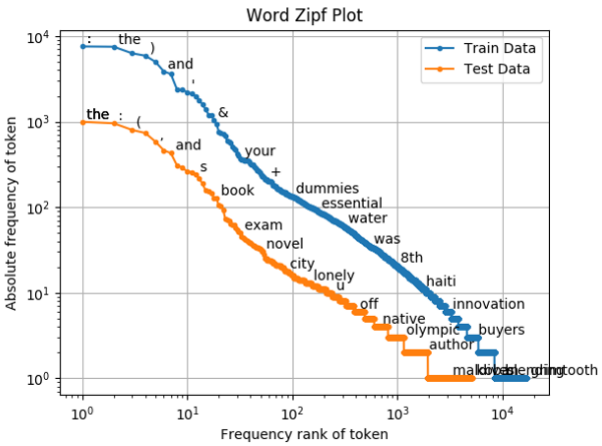


Figure 1. Zipf log-log plot of word frequencies.

One potential issue we found with the dataset is how a single label was chosen per record. The source data had several multi-label samples, however in preprocessing the data one of these labels was selected at random. While this process was done randomly, if a secondary genre is chosen (for instance a book that is mainly a *Children's Book* but also touches upon *Travel*) it will negatively affect our results (albeit in a minor way).

4.1. Preprocessing

In image preprocessing, we rescale all the images to by 227x227. We picked those dimensions for three reasons: (1) it's larger than the max size (50 KB), (2) it's square, (3) there are pre-trained weights for VGG-16, SqueezeNet and ResNet50 for those dimensions. Additionally, we subtract the training set mean from each image.

In text preprocessing, we tokenized the input using the `nltk` toolkit [20].

5. Experiments

We survey successively deeper & more state-of-the-art image and text classifiers. We compare these standalone models using the quantitative metrics listed below and analyze the predictions qualitatively by inspecting the confusion matrices and t-SNE plot of the softmax activations. Using our findings we combine the best image and text classifiers, respectively, using various ensembling methods. Ultimately, we combine the best ensembles using a mixture of experts model [10].

5.1. Quantitative Metrics

Our main evaluation metric is the overall categorical accuracy:

$$\frac{\texttt{\#correct}}{\texttt{\#total}}$$

We keep track of the top-2 and top-3 categorical accuracy, as is common with multi-class classification. We also calculate precision and recall.

We record all of these metrics overall and per-class.

5.2. Training

We use Adam optimization [14] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a vertical dropout of 0.5, and gradient clipping beyond an absolute value of 5.

We train on a single GPU in batches of 64 randomly selected training pairs, annealing an initial learning rate according to the standard learning rate schedule depicted in figure 2.[1] We actively selected the initial learning rate as the largest learning rate that led to good activations and large but possible updates. We were able to saturate multiple models within 20 epochs with this process.

During training, we kept track of the categorical accuracy on the validation set, and checkpointed the model parameters only when there was an improvement on the validation set.

When training all the text classifiers, we had the problem of the model quickly over-fitting the training data (within 10 epochs), with the validation accuracy trailing far behind. In an attempt to mitigate this issue, we added dropout layers, increased the weight decay and simplified the models. Through these methods we managed to bridge the gap a little bit, but not completely.

[1] We augmented the halving points on a per-model basis, and used cyclic cosine annealing only for snapshot ensembles.

5.2.1 Ensembles

Once we had achieved the best possible results for each classification type, we experimented with a few different ensembling techniques; we combined a set of fully-trained constituent models. The trained ensemble generates a single hypothesis, from a larger hypothesis space than any individual constituent, and thus can represent a wider variety of functions.

We have two choices when ensembling models: (1) model selection, and (2) combining method. In our experiments, we always combined the models by taking the average of their scores, selecting the trained models in three different ways; snapshotting, different initializations, and different model architectures.

As demonstrated by Kuncheva et al., 2003 [15], ensembles obtain better predictive results when the models are diverse, so we actively searched for models with different predictive behaviors.

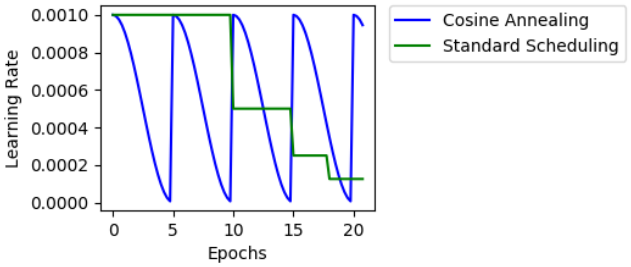
1. **Snapshot Ensemble**:
   We hand-selected the best checkpoints of a model within a single run. Using the standard learning rate annealing in 2, the predictive results tended to simply average out, producing worse results than the single best one. This was expected, since the models produced are not sufficiently diverse. As suggested by Huang et al., 2017 [7], we used cyclic cosine learning rate annealing:

   $$\texttt{lr}(e) = \frac{\texttt{lr}_0}{2}\left(\cos\frac{\pi\texttt{rem}(e,P)}{P} + 1\right)$$

   where P is the period. This schedule helps the model converge to multiple local minima during a single training run.
2. **Cross-Run Ensemble** In an effort to get models to converge to disparate minima, we trained models with different initialization and hyperparameters, and selected the best ones with the most different predictive behavior.
3. **Cross-Architecture Ensemble** Lastly, we combined the best models from different architectures, as they will likely learn vastly different features.

As the results below suggest, the cross-run and cross-architecture ensembles perform the best, with snapshotting not far behind.

Figure 2. Standard and Cyclic Cosine Annealing Learning Rate Schedules.



5.2.2 Softmax Gating Network

Once we had trained the best classifiers of each type, we wanted to combine the two into a single classifier over both domains. We first attempted simply averaging the scores as before, which produced only marginally better results than text classification alone. Analyzing the per-class predictive powers of each, we noticed that there was some complementary behavior. This drove us to train a softmax gating network on top of the best two ensembles [15].

5.3. Results

In table 1 we see top the results of all the image-only classifiers on `BookCover10`. As expected, the quality of predictions increases with deeper and more advanced networks, with ResNet50 achieving the best invidiual accuracy of 0.58 and a cross-run ResNet50 ensemble achieving the best accuracy overall [2]

In table 2 we see the top results of all the text-only classifiers on `BookTitle10`. Unsurprisingly, Multinomial Naive Bayes performs really well. What was most surprising was that the Text CNN performed worse than Naive Bayes, even with an extensive hyperparameter search. We hypothesized that this implied that the structure of the title was mostly irrelevant, and only the presence of keywords was important. We tested that hypothesis by implementing a simple two-layer fully-connected network using a bag-of-words feature representation. This simple network outperformed both Text CNN and Naive Bayes, supporting our hypothesis. As we did for the best image classifiers, we trained cross-run and cross-model ensembles, [3] with both achieving nearly identical results.

Comparing tables 3 and 4 we see that the text classifier performs significantly better in almost every class, with the notable exception of *Children's Books*. Combining these two cross-run ensembles with a softmax gating mechanism,

[2]The cross-model ResNet50 and SqueezeNet Ensemble achieved almost the same accuracy as the cross-run ResNet50 ensemble. The former score 0.588 and the latter 0.592

[3]We didn't train a snapshot ensemble, as it's performance was not as convincing.

Table 1. Results from Image-Only Classification Methods on `BookCover10`

Model	Top 1	Top 2	Top 3	Prec.	Rec.
Fully-Connected	0.27	0.42	0.53	0.27	0.27
Conv Pool	0.39	0.54	0.65	0.39	0.39
VGG16	0.53	0.67	0.80	0.53	0.53
SqueezeNet	0.54	0.71	0.80	0.54	0.54
ResNet50	0.58	0.74	0.83	0.58	0.57
ResNet50 Snapshots	0.58	0.75	**0.84**	0.58	0.58
ResNet50 Cross-Run	**0.59**	**0.76**	0.82	**0.59**	0.58
ResNet50 + SqueezeNet	**0.59**	0.75	**0.84**	**0.59**	**0.59**

Table 2. Results from Text-Only Classification Methods on `BookTitle10`

Model	Top 1	Top 2	Top 3	Prec.	Rec.
SVM	0.70	0.76	0.89	0.71	0.71
Naive Bayes	0.74	0.85	0.90	0.74	0.74
Text CNN	0.71	0.83	0.88	0.74	0.71
BOW FC	0.75	0.85	0.90	0.75	0.75
BOW FC Ensemble	**0.76**	**0.86**	**0.91**	**0.76**	**0.76**
BOW FC + TextCNN	**0.76**	**0.86**	**0.91**	**0.76**	**0.76**

Table 3. Per-Class Accuracy of ResNet50 Ensemble

Genre	Top 1	Top 2	Top 3
Children's Books	0.66	0.81	0.86
Comics & Graphic Novels	0.62	0.77	0.85
Computers & Technology	0.62	0.77	0.83
Cookbooks, Food & Wine	0.61	0.74	0.81
Romance	0.66	0.82	0.91
Science & Math	0.40	0.63	0.76
Science Fiction & Fantasy	0.48	0.73	0.80
Sports & Outdoors	0.49	0.70	0.81
Test Preparation	0.73	0.84	0.89
Travel	0.49	0.68	0.76

we achieve incredible results; **0.80 top 1 accuracy**, **0.91 top 2 accuracy**, and **0.94 top 3 accuracy**. Looking at table 5, we see that in almost every category the mixture achieves better results than either individual ensemble, with the exception of *Science & Math* and Sports & Outdoors. In these cases, it appears that the poor image classification results are weighing the mixture down.

In order to compare our results directly to the Iwana et al., 2016 results [9], we took our best ResNet50 ensemble approach and trained it on `BookCover30`, using the same hyperparameter settings as we did on `BookCover10`, and significantly improved on the benchmark. As shown in table 6, our approach achieved 0.40 categorical accuracy, which is a 1.6x improvement on the 0.25 they achieved using AlexNet.

5.4. Qualitative Analysis

We perform qualitative analysis of the model, including exploring the types and frequencies of its errors and suc-

Table 4. Per-Class Accuracy of BOW Fully-Connected Ensemble

Genre	Top 1	Top 2	Top 3
Children's Books	0.54	0.73	0.78
Comics & Graphic Novels	0.76	0.86	0.90
Computers & Technology	0.88	0.94	0.95
Cookbooks, Food & Wine	0.88	0.94	0.96
Romance	0.75	0.87	0.93
Science & Math	0.65	0.79	0.86
Science Fiction & Fantasy	0.72	0.83	0.90
Sports & Outdoors	0.67	0.82	0.88
Test Preparation	0.96	0.98	0.99
Travel	0.76	0.86	0.92

Table 5. Per-Class Accuracy of BOW Fully-Connected Ensemble and ResNet50 Ensemble Mixture

Genre	Top 1	Top 2	Top 3
Children's Books	0.77	0.89	0.95
Comics & Graphic Novels	0.84	0.94	0.96
Computers & Technology	0.91	0.98	0.98
Cookbooks, Food & Wine	0.96	0.98	0.99
Romance	0.81	0.89	0.93
Science & Math	0.62	0.79	0.85
Science Fiction & Fantasy	0.75	0.86	0.92
Sports & Outdoors	0.66	0.82	0.91
Test Preparation	0.96	0.99	1.00
Travel	0.79	0.89	0.93

Table 6. Image-Only Classification Comparison on `BookCover30`

Model	Top 1	Top 2	Top 3
AlexNet (Iwana et al. 2016)	0.25	0.33	0.40
LeNet (Iwana et al. 2016)	0.14	0.21	0.28
ResNet50 Ensemble	**0.40**	**0.54**	**0.66**

cesses. For every model we trained, we generated a t-SNE plot of Softmax Activations, as well as confusion matrices from its predictions. We found the t-SNE plots particularly useful when determining which models to select for ensembling, as they allow you to visually determine whether the predictive behaviors of each model are different enough for it to yield better ensembling results.

Figures 3 and 4 demonstrate the clustering of the softmax activations from the best ensembles for each domain. We see very strong clustering for both. Consistent with the results, the image classification clustering is a lot noisier than the text classification one. We also observe similar clustering behavior in *Cookbooks, Food & Wine* and *Travel*, on one hand, and very different behavior in *Children's Books*, *Science & Math* and Test Preparation, on the other. The partially complementary behavior indicates that a good combination of the two should yield stronger results, which is precisely what we see. Looking at figure 5 we observe three things: the similar clusters stay the same, dissimilar clusters have mixed into new ones, clusters are further apart.

Examining the confusion matrix in figure 6 we see that our model makes most of it's mistakes when deciding between similar genre's; *Science & Math* is most often misclassified as *Computers & Technology*, *Travel* is most often misclassified as *Sports & Outdoors*[4] (and vice versa), *Romance* is most often misclassified as *Science Fiction & Fantasy*.

Figure 3. t-SNE plot of Softmax Activations from ResNet50 Ensemble.

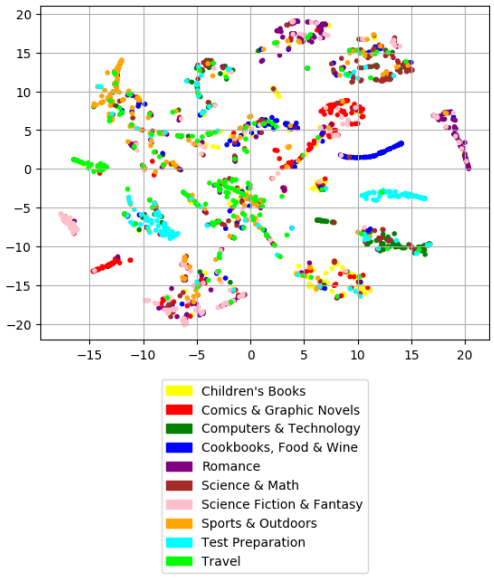
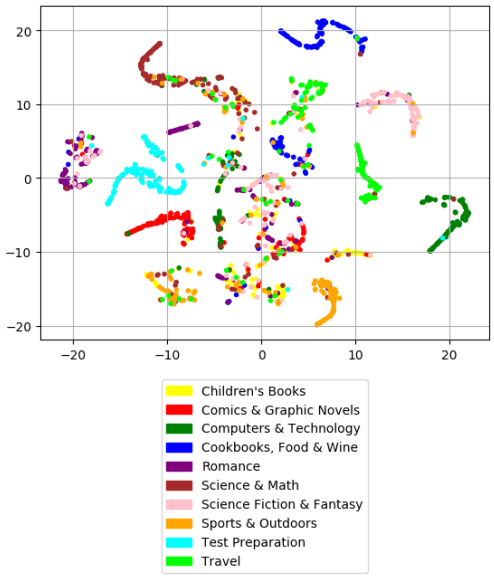


Figure 4. t-SNE plot of Softmax Activations from BOW Fully-Connected Ensemble.



[4]Mostly "Outdoors" books get misclassified as *Travel*

Figure 5. t-SNE plot of Softmax Activations from ResNet50 and BOW Fully-Connected Mixture.

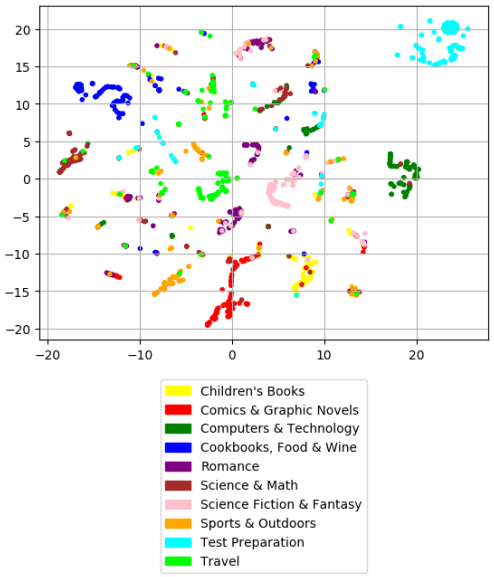
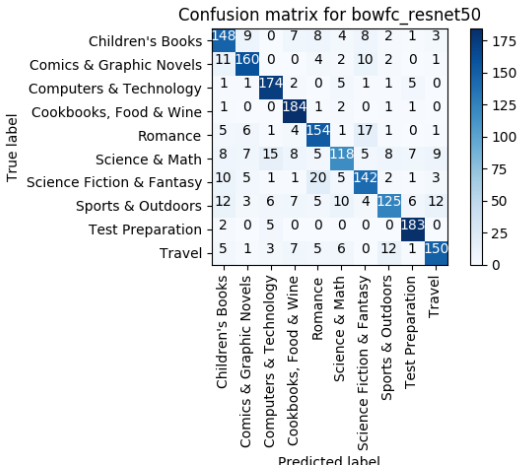


Figure 6. Confusion Matrix for the ResNet50 and BOW Fully-Connected Mixture.



5.4.1 Misclassified Samples

We conduct a blind hand-analysis of 30 misclassified samples selected at random to gain qualitative insights into the performance of the best model. A human analysis indicates for about 8 of the misclassified samples, the ground truth for the book's genre is either invalid or ambiguous enough for the predicted genre to be valid. Four of the samples contain a red herring in the image, where something in the image is very common in the predicted genre, but should not be the main focus. Two of the samples contain a red herring in the text. Five of the samples do not contain a lot of visual or textual information. The rest appear to be nonsensical, perhaps due to lack of representation in the test data.

6. Conclusion & Future Work

This paper sought to prove that a book can be judged (classified) by its cover, and accomplished just that. We have shown that information from the cover of the book can be used to classify its genre, achieving 80% top 1 accuracy, 91% top 2 accuracy and 94% top 3 accuracy. Furthermore, using just the cover image we had accuracy of 59% (5.9x random), using just the title an accuracy of 76% (7.6x random), or a gating mechanism using both accuracy of 80% (8x random). It can be seen as a natural extension of previous experiments showing the power of deeper neural networks [9, 16, 5], and how important data representation and optimization efficiency techniques are in their use.

Our initial benchmark was set to the results provided by Iwana et al., 2016 [9], namely 24.7% top 1 accuracy, 30.1% top 2 accuracy, and 40.3% top 3 accuracy. When training our models on the full dataset and evaluating on the full test set, we obtained results of 40%, 54% and 66% for the top 1, top 2, and top 3 accuracy respectively.

An important conclusion from our analysis is how a cover's text is a more important signal of genre than the artwork. This is in line with what we expected; that books tend to have titles very explicitly similar to their genre, while the cover art might have some weaker latent clues but by-and-large has less of a connection.

We encountered numerous example where are model failed. One such example is when the artwork is lacking visual information or the title is non-descriptive, leaving the classifier little to use to aid in its task. In a similar vein some cover art was a red herring, matching patterns seen for another genre though having no relation (a good example being an image of a man and a woman being a children's book and not a romance novel). Finally an inherent issue in this task is identifying what constitutes the ground truth for a books genre. Many books can be classified as a combination of multiple genres, and when forced to choose only one genre it is entirely possible that the genre that is more represented is discarded, causing the training example to be adversarial to the network.

Some suggestions for future work address some concerns we had throughout the study, as well as extend in other ways. One such improvement would be the implementation of an attention mechanism, allowing the network to disregard possible misinformation. Another such improvement would be a study that took into account multigenre classification of books, perhaps weighting how representative each genre is of a books true genre.

An advancement that could be made to this study would be to use a neural network to extract text from the book cover, giving it more contextual clues of what is going on in the image (for example the text of a billboard in the fore-

ground). Another extension to this study could be to use generative networks to then generate mockups of books by genre. Such a model could allow a user to them input partial designs or other parameters and a target genre, and the network would an appropriate cover or the genre. Taking this a step further a future investigation could then look more into the features/weights that seem to be most useful to the model, attempting to infer latent characteristics in book covers/titles. Such an investigation could find the effect of coloration, font style, illustration style and other factors to be critical, allowing illustrators to know which parts of the book *truly* capture a reader attention.

References

[1] A. Blessing and K. Wen. Using machine learning for identification of art paintings. 2010.

[2] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] A. Finn and N. Kushmerick. Learning to classify documents according to genre: Special topic section on computational analysis of style. *J. Am. Soc. Inf. Sci. Technol.*, 57(11):1506–1518, Sept. 2006.

[5] R. C. B. Gabriel S. Simes, Jnatas Wehrmann and D. D. Ruizl. Movie genre classification with convolutional neural networks. *IJCNN*, 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger. Snapshot ensembles: Train 1, get M for free. *CoRR*, abs/1704.00109, 2017.

[8] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

[9] B. K. Iwana and S. Uchida. Judging a book by its cover. 2016.

[10] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87, Mar. 1991.

[11] R. Johnson and T. Zhang. Convolutional neural networks for text categorization: Shallow word-level vs. deep character-level. *CoRR*, abs/1609.00718, 2016.

[12] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes. *Multinomial Naive Bayes for Text Categorization Revisited*, pages 488–499. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[13] Y. Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[15] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, 51(2):181–207, May 2003.

[16] B. Kuprel. Judging a movie by its poster using deep learning. 2016. Done in CS221 at Stanford.

[17] Y. LeCun, K. Kavukvuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Proc. International Symposium on Circuits and Systems (ISCAS'10)*. IEEE, 2010.

[18] J. H. Lee, S. W. Baik, K. Kim, C. Jung, and W. Kim. *IGC: An Image Genre Classification System*, pages 360–367. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[19] J. Libeks and D. Turnbull. You can judge an artist by an album cover: Using images for music annotation. *IEEE MultiMedia*, 18(4):30–37, April 2011.

[20] E. Loper and S. Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[21] D.-C. Park. Image classification using naive bayes classifier. *IJCSEE*, 2016.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[23] T. Yang. squeezenet demo. `https://github.com/DT42/squeezenet_demo`, 2017.

[24] D. Zhang and D. Wang. Relation classification via recurrent neural network. *CoRR*, abs/1508.01006, 2015.

[25] K. Zhang, A. Ball, F. Gu, and Y. Li. A hybrid model with a weighted voting scheme for feature selection in machinery condition monitoring. In *2007 IEEE International Conference on Automation Science and Engineering*, pages 424–429, Sept 2007.

[26] K. Zhang, A. Ball, F. Gu, and Y. Li. A hybrid model with a weighted voting scheme for feature selection in machinery condition monitoring. In *2007 IEEE International Conference on Automation Science and Engineering*, pages 424–429, Sept 2007.