Book Cover Classification with VGG16 and ResNet50

Emelie Aalto
University of Gothenburg
gusaaltoem@student.gu.se

Nadina Suditu
University of Gothenburg
gussudna@student.gu.se

Abstract

In this paper we chose to experiment with the task of multilabel classification on book genres inferred by their cover images. We compared the CNN models VGG16 and ResNet50 with transfer learning, and discussed our results.

1 Introduction

1.1 Background

In a recent Guardian article (Bramley, 2021), its author highlights the case of the book cover nowadays. It's argued that in recent years, the designs have 'taken on a higher profile' (Bramley, 2021), possibly as a consequence of pandemic (Bramley, 2021). Although there are people in the industry who view this as a positive change, showcased by the fact that cover designers have started to be more recognised, such as in the case of the reinstatement of the Designer of the Year category at the British Book Awards (Bramley, 2021), there are opposing views being voiced as well. In the Observer article "Let's Get Rid of the Blobby Book Cover", the focus is shifted on how, in the vein of 20th century expressionism, book designers have gone too far with their artistry, leading into a regression of the book cover as an ultimate colorful, crowded blob (Klee, 2022).

It is argued that some of the blame lays on Amazon's recommendation engine: "a mechanism that constantly equates products as interchangeable and therefore incentivizes a kind of uniformity" (Klee, 2022). It is therefore not surprising that a rising sentiment in the literary world is to yearn for the past century's arguably more vivid and expressionistic covers (Kreider, 2013). In the midst of this cultural dilemma, we thought that it would be interesting to look at it from the machine learning standpoint. If "the covers of most contemporary books all look disturbingly the same, as if inbred" (Kreider, 2013), then how would AI fare in succeeding to recognise a book's genre by its cover? Although this isn't always the case, a lot of the times the clues on the cover communicate a great deal of information to the potential reader (Thorp, 2020), usually including its genre. This can be considered an overall beneficial aid that AI can bring, from the marketing standpoint, since it allows for a book to signal its presence faster and more efficiently to its targeted audience, as well as from the reader's position, making it possible to identify the preferred genre and not be 'tricked' into an undesirable read.

1.2 Proposed question and structure of paper

For our study, we chose to test two pretrained models, VGG16 and ResNet50, and compare their performance in predicting the genre of a book based on its cover. In the next section we talk about the dataset that was used, as well as examine the models that we chose and why we chose them, followed by a step by step presentation of our methodology. This is followed by an analysis and discussion of our results, a comparison of the two models' performance and a conclusion.

2 Related work

There have been papers focusing on this particular issue, out of which most inspiration was drawn from Iwana et al. 's "Judging a Book by its Cover" (2016). In it, the authors explore the matter of automatically deriving a book's genre by its cover, using deep Convolutional Neural Networks (CNN). Neural networks have become the state of the art in this area of research, specifically in image classification tasks, because of their ability to successfully recognise patterns (Iwana et al., 2016).

CNNs are multilayer neural networks that make use of convolutional layers as a method of feature extraction (Iwana et al., 2016). They are made up of three components: the convolutional layers, which consist of feature maps produced by repeatedly applying filters across the input (Iwana et al., 2016). Next are the filters, which represent shared weights and are trained with backpropagation (Iwana et al., 2016). The last component consists of the final layers which are fully connected and are given a vector representation of the images from a preceding pooling layer and continue like standard feedforward neural networks (Iwana et al., 2016). For this paper, the authors experiment on pretrained AlexNet with transfer learning, as well as the 5-layer model LeNet (Iwana et al., 2016). AlexNet returns satisfactory results, the model performing at an accuracy of 24.7% for Top 1, 33.1% for Top 2, and 40.3% for Top 3 in 30-class classification (idem), while LeNet performs overall more poorly, with accuracies of 13.5.7% for Top 1, 21.4% for Top 2, and 27.8% for Top 3 (Iwana et al., 2016).

In a paper from 2021, Mascarenhas and Agarwal made a comparison of the different architectures for image classification. In their paper they compared VGG16, ResNet50 and VGG19, a version of VGG with 19 layers instead of 16. They used a dataset of images with both textual data and objects within the categories of shoes, beauty, jewelry, watches and bags. For this task within image classification they came to the conclusion that ResNet50 is the best architecture with 97.33% accuracy compared to an accuracy of 96.67% for VGG16.

3 Materials and methods

3.1 Dataset and pre-processing the data

The dataset used for this research was obtained from Kaggle and is called the ”Book Covers Dataset” (n.d.). The dataset comprises book covers and associated metadata from the open book retailer https://www.bookdepository.com. However, it should be noted that the dataset was created in 2020, so it does not include very recent book releases. It consists of one CSV file containing metadata about the books and a directory named ”book-covers” containing images in JPG format. For the purpose of image classification, only the ”book-covers” directory was utilized, which includes 33 folders named after different book categories. Each category includes roughly 1000 images.

To be able evaluate the performance of our models, we first divided the dataset into a training set and a test set, where the training set includes 90% of the book covers and the test set includes the remaining 10%. After that, we conducted an exploratory data analysis by creating a dictionary of book categories in the training set, which confirmed the successful split of the dataset into 33 distinct categories, or classes. We then created a function that would prepare the dataset with images and labels for our task. This was done by using the os library to list all the folders and then we used the cv2 library to read the image file and convert it to RGB color space. We also resized it to 224 x 224 using the cv2 library and appended the images to one list and the labels to another list. Next we normalized the images by dividing them by 255 and converting them to numpy arrays. Finally we shuffled the arrays and then visualized the distribution of book covers in different categories by using the Seaborn library (sn) and we used the matplotlib library to display a grid of images with their corresponding labels. The distribution of categories can be seen in figure 1 and the grid images and their labels can be seen in figure 2.

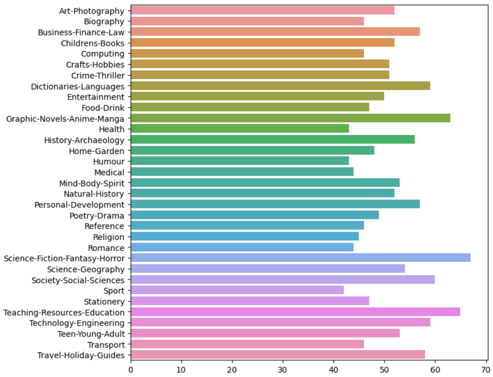

Figure 1: Distribution of book genres

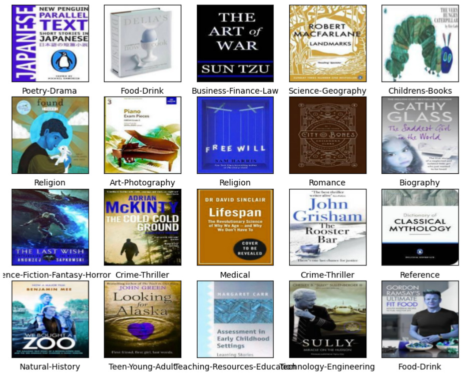

Figure 2: Book covers and their genre

Next we used the ImageDataGenerator class from Keras to load the book cover images. We applied image augmentation techniques such as shearing and zooming to the images in order to prevent overfitting. By using image augmentation, we aimed to create a more diverse training set and a more robust model. A validation set was also created, consisting of 10% of the generated training data. We employed the flow_from_directory method to generate batches of augmented data from the training folder. The image size for all covers was set to 224x224 and the categorical class mode was used, as there are more than two classes in the dataset. This resulted in a training set consisting of 26,393 images across 33 classes and a validation set consisting of 2,920 images across 33 classes. Some of the augmented images are displayed in Figure 3.

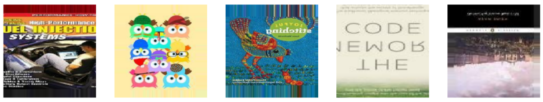

Figure 3: Example of augmented images

3.2 VGG16

VGG16 is a convolutional neural network model trained on the ImageNet dataset introduced by Simonyan and Zisserman (2014). The architecture of VGG16 is characterized by its simplicity, using only 3x3 convolutional layers stacked on top of each other in increasing depth. It also makes use of max pooling layers and is 16 layers deep and can classify images into 1000 object categories. VGG16 has been trained on the imagenet dataset and therefore we set the “trainable_set” parameter to False so that we wouldn’t train the base model. The “include_top” parameter was also set to False because the output layer has to be modified accordingly. To read more on how VGG was created and trained see Simonyan & Zisserman, 2014.

3.3 ResNET50

ResNet, short for Residual Network, is also a CNN architecture that was introduced in 2015 by Microsoft Research (He et al). It utilizes the concept of ”residual connections” to mitigate the vanishing gradient problem that occurs in deep neural networks. ResNet50, one of the variations of ResNet, comprises 50 layers and is widely used in image classification tasks. These residual connections enable the network to stack additional layers and build a deeper network that can bypass less relevant layers during training. The ResNet50 parameters were set in the same way as VGG16, with “trainable_set” and “include_top” being set to false because it had also been pretrained on imagenet.

3.4 Training the models

For compiling and training the above-mentioned models, VGG16 and Resnet50, we used the pre-trained architecture from the Keras library. Both models were loaded with ImageNet weights, with the fully connected layers removed and the input shape set to (224, 224, 3) using the applications module from Keras. We also set res_base to False so that the pre-trained weights will not be updated during training. Then we created an input layer with the same shape as the input images which connects to the pre-trained model. The code then applies a GlobalAveragePooling2D layer to reduce the spatial dimension of the feature map, followed by adding a Dense layer with 1024 neurons, a Dropout layer with a rate of 0.5 to prevent overfitting, and finally, a Dense output layer with 33 neurons and a softmax activation depending on the model, this is used to predict the probability for each class. Once the model architecture was defined, it was compiled with an optimizer (Adam), loss function (CategoricalCrossentropy) and metrics (CategoricalAccuracy). Then the model is trained on the dataset using the fit() method, and the training process is being monitored by an early stopping callback, which stops the training if the validation loss does not decrease for 5 consecutive epochs. We used a batch size of 32 and 25 epochs.

4 Results and Discussion

Training the models took approximately 5 minutes per epoch and accuracy was increasing but not by much for each epoch. The final epoch for ResNet50 ended like this:

```
Epoch          24/25          459/459
[================================]
- 397s 865ms/step - loss: 3.4415 - categorical_accuracy: 0.0539 - val_loss:
3.4408 - val_categorical_accuracy: 9
0.0563
```

and the final epoch for VGG16 ended like this:

```
Epoch          25/25          459/459
[================================]
```

```
- 395s 861ms/step - loss: 2.9577 - categorical_accuracy: 0.1864 - val_loss: 3.0599 - val_categorical_accuracy: 0.1672
```

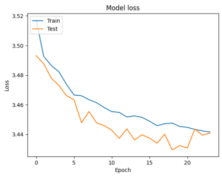

Figure 4: ResNet loss

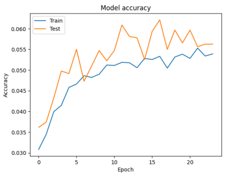

Figure 5: ResNet accuracy

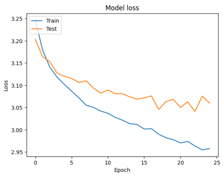

Figure 6: VGG16 loss

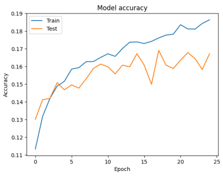

Figure 7: VGG16 accuracy

Looking at the results, VGG16 has the best accuracy out of the two models. It predicts with 16.72% accuracy on the validation set and 18.64% on the training set. ResNet on the other hand reaches only 5.63% accuracy on the validation set and 5.39% accuracy on the training set. To put things into perspective, randomly guessing would result in an accuracy of around 3%Ġiven that previous papers have reported better results while using ResNet, we consider that training for more epochs and tuning the hyper parameters could have resulted in better accuracy.

We are also taking into consideration the possibility that something might have gone wrong in the code leading to training the model, however at the time of writing the report there was no more available time for pinpointing the specific problematic area or retraining. We suspect that the learning rate might have been too low, or the dropout too high.

Looking again at the VGG16 results, we have to take into consideration the fact that the accuracy might be so low because of the choice to look only at the first prediction, instead of the first 3 or 5. We are aware that realistically it would have been better to go with the latter approach, since there are books which pertain to more than one category. As we can see in the example images the book cover relies on the context and knowledge of the target audience. For example a Jamie Oliver (famous chef) book that is normally part of the food and drinks can also be classified as a lifestyle book. This means that a book can be part of multiple classes as the classes themselves are not well defined and separated. However, we chose to approach the problem in this way because we were curious of how a first guess attempt would pan out, since we haven't seen other examples of experiments on this matter focusing only on first guesses.

5 Conclusions and Future Work

Our goal was to test the task of classifying books by their covers using two different models, VGG16 and ResNet50, and see which one performs better, without fine tuning them. The results for VGG16 were somewhat satisfactory, while the results for ResNet turned out very low. Therefore, we don't believe that they are conclusive or representative for the question of which model is better suited for this task. We conclude that the low results might have been caused by a number of factors, such as: underfitting, poor data quality, or a lack of training data. It could also be the case that the model architecture is not well suited for this task. In order to see improvements, we believe that the following could be of help: training on more epochs, using a bigger dataset, tuning the hyper-parameters, the learning rate, the layers or kernel sizes. Given that we know about the overlap between the classes it would have been interesting to look first few predictions as well and check if there are any cover types that are often seen together.

6 Acknowledgements and contributions

We want to note that most of this project was done as a collaborative effort between Emelie and Nadina, to which we employed a strategy of meeting up, planning and discussing each section together. For the code part, we followed a Multiclass Image Classification+Transfer Learning Kaggle tutorial by EXASPERATED DUCK, as well as the TensorFlow Image classification tutorial.

References

E. V. Bramley. In the instagram age, you actually can judge a book by its cover. the guardian. URL `https://www.theguardian.com/books/2021/apr/18/in-the-instagram-age`.

R.E. Hawley. Behold, the book blob. URL `https://www.printmag.com/book-covers/the-book-cover-behold-the-book-blob`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

M. Klee. Let's get rid of the blobby book cover. observer. URL `https://observer.com/2022/10/lets-get-rid-of-the-blobby-book-cover/`.

T. Kreider. The decline and fall of the book cover. the new yorker. URL `https://www.newyorker.com/books/page-turner/the-decline-and-fall-of-the-book-cover`.

Chandra Kundu and Lukun Zheng. Deep multimodal networks for book genre classification based on its cover, 2020. URL `https://arxiv.org/abs/2011.07658`.

lukaanicin. Book covers dataset. URL `https://www.kaggle.com/datasets/lukaanicin/book-covers-dataset`.

Sheldon Mascarenhas and Mukul Agarwal. A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification. In *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, volume 1, pages 96–99, 2021a. doi: 10.1109/CENTCON52345.2021.9687944.

Sheldon Mascarenhas and Mukul Agarwal. A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification. In *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, volume 1, pages 96–99, 2021b. doi: 10.1109/CENTCON52345.2021.9687944.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. URL `https://arxiv.org/abs/1409.1556`.

TensorFlow. Image classification. URL `https://www.tensorflow.org/tutorials/images/classification`.

C. Thorp. What makes an iconic book cover? bbc. URL `https://www.bbc.com/culture/article/20200604-the-best-book-covers-in-history`.