MSc Data Science Proposal: Can a Convolutional Neural Network judge a book by its cover?

Ryan Hill MSci

Supervisor: Dr Hubie Chen

Birkbeck, University Of London

rhill06@mail.bbk.ac.uk

May 2, 2020

Abstract

In this work we make a proposal for training convolutional neural networks to predict the genre of a book, based on its cover. We evaluate the prior work done in this area and the proposed improvements of the existing dataset known as *BookCover30*. We break the work into 3 stages and provide detail on the proposed hardware and software usage within the project.

Contents

1 Introduction **1**

1.1 Background . . . 1

1.2 Problem Statement . . . 1

2 Literature Review **2**

2.1 History of Computer Vision . . . 2

2.2 Prior Work . . . 3

2.3 CNN Architectures and Transfer Learning . . . 4

2.4 Existing Datasets . . . 5

2.5 Debugging CNNs . . . 7

3 Proposal **9**

3.1 Planned work . . . 9

3.1.1 Part 1: Data Collection and Preparation . . . 9

3.1.2 Part 2: Genre Predictor via Transfer Learning . . . 10

3.1.3 Part 3: Feature Visualisation . . . 11

3.2 Hardware and Software . . . 12

4 Conclusion **12**

1 Introduction

1.1 Background

Don't judge a book by its cover is a common English idiom, with its origin somewhere around the mid 19th century, often cited as being from the newspaper *Piqua Democrat* as follows:

> "Don't judge a book by its cover, see a man by his cloth, as there is often a good deal of solid worth and superior skill underneath a jacket and yaller pants."

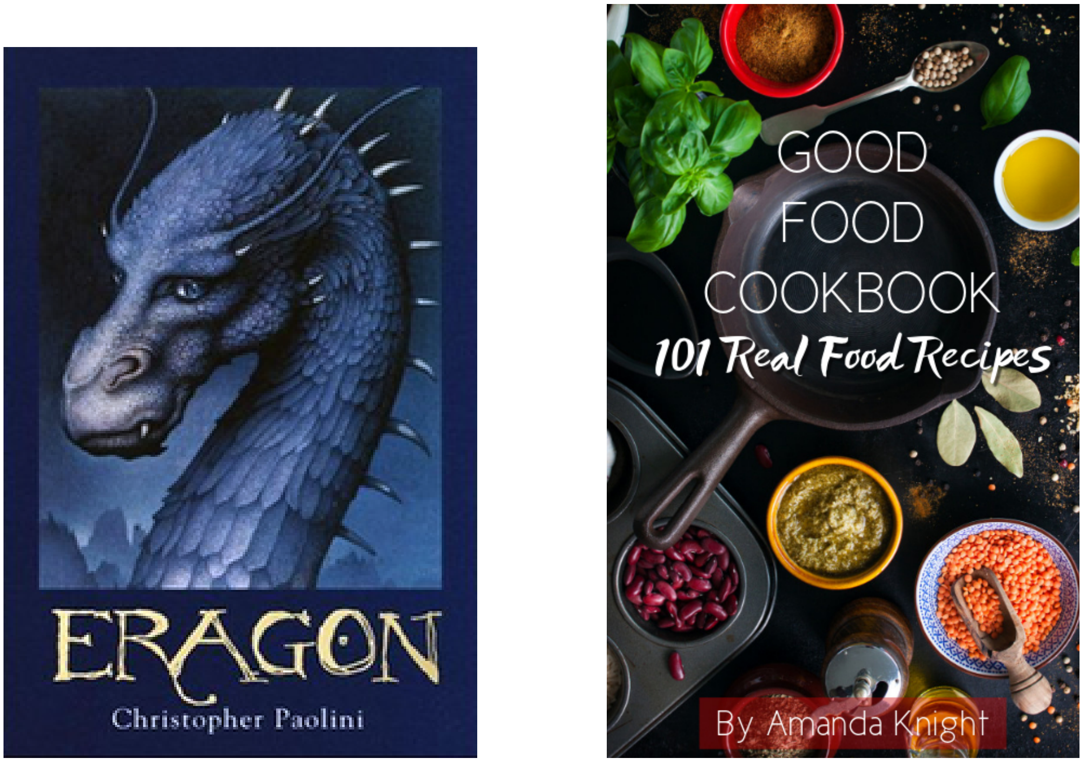
The phrase itself refers to the idea that one should not judge an item, or person, by how it first appears, but instead you should take the time to understand it better. That being said, a huge amount of cost and effort can go into the design and choice of a cover for a book; costs for cover design alone can range from a few dollars to a few thousand[1], and revenue from publishing in 2018 was around $122bn[2]. This value, combined with the fact that humans have been shown to make judgments within 100ms of exposure to another person[3], means it is reasonable to assume that this advice is not often taken literally and that as humans we do, in fact, judge a book by its cover.

Beyond the focus of the judgement placed on book covers, they are designed to help the potential reader understand the type of story, or contents, a book may contain. As an example, consider the covers shown in fig. [1.1] that should, even if they had no words on them, help the reader identify immediately the type of content contained within. The type of this content is what we refer to as the *genre* of the book. Even within this example it is clear that there could be some subjectivity, as the reader you may see a dragon and assume it is a fantasy novel, but the Teen & Young Adult genre often encompasses many genres just due to the age of the intended audience.

1.2 Problem Statement

In this work we will try to answer one seemingly simple question; *can a Convolutional Neural Network predict the genre of a book based solely upon an image of the front of the book?* Or put another way, *can a Convolutional Neural Network judge a book by its cover?* This may seem like a fairly simple question, but in many cases it might already be quite difficult for a human to distinguish which genre a given book belongs to as just discussed. Beyond answering this question we will also attempt to take the easiest and hardest to identify genres (as determined by the accuracy of the classifier) and use feature visualisation to generate ideal covers for these genres, as the Convolutional Neural Network *sees* them.

The rest of this work is structed as follows; first we review the existing literature and history of computer vision, predicting genre from book covers, and feature visualisation in section [2]. We then discuss in more detail the plans for how to tackle this problem statement, including the hardware and software we plan to use, in section [3]. Finally, we summarise this work in section [4].



(a) Cover of Eragon, Teen & Young Adult Genre

(b) Cover of Good Food Cookbook, Cookbooks, Food & Wine Genre

Figure 1.1: Two book covers with distinctly different designs and genres

2 Literature Review

2.1 History of Computer Vision

The work in this section is mostly informed by the comprehensive article by R Demush[4]. Computer vision was originally, like most theoretical work in artificial intelligence (AI) and machine learning, conceived in the 1960s. A 1966 summer project by MIT was intending to solve the computer vision problem within a few months, simply put by attaching a camera to a computer it would then 'describe what it saw'[5]. Clearly, that was a little over-ambitious.

One of the most influential papers in computer vision actually came in 1959 when two neurophysiologists, Hubel and Wiesel, published their work studying the response of visual neurons of a cat's brain, and how these experiences shape the neural architecture[6]. Paraphrasing some of their results, they discovered that there are both simple and complex neurons, with visual processing beginning with simple structures such as edges and then going to more complex structures. This is essentially the basis behind deep learning today, especially in computer vision. However, even with this knowledge, it wasn't until 1982 when neuroscientist David Marr published the next piece of the puzzle in the form of a book entitled *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*[7]. While his representational framework was abstract and high-level, it was still ground-breaking work and helped us understand how we create more complex images from the simple building blocks, and gave us a path to understanding how we could build computer models to do this.

Around the same time *Neocognitron*, a multilayer artificial neural net, was proposed by Kunihiko Fukushima which contained several convolution layers, the types of layers still used today. In 1989 Yann LeCun took this network and developed LeNet-5 from it, a Convolutional Neural Network (CNN) that contains many elements used by modern CNNs. This model had the goal of classifying images, as has become the main use of CNNs in the modern era. This work[8] also included the production of the now famous MNIST dataset of handwritten digits.

For a few years, work continued to try and construct 3D models from 2D images, before instead moving towards object recognition around the turn of the new millennia. Development continued over the coming years, including a new focus on having a set of communal benchmark datasets to enable easy comparison. These included the original MNIST, before the development of Pascal VOC[9], and finally ImageNet[10]in 2010. ImageNet has now become the de-facto benchmark dataset for image classification with over a million manually cleaned images covering 1000 object classes. At this point, development started coming specifically within the architecture of CNNs, which is discussed briefly in section [2.3].

2.2 Prior Work

Design has been explored in many areas before with regards to its history and the various styles, however it is a relatively new area for machine learning in comparison to other fields. CNNs have been used to identify the artists of works[11], they have been suggested for use in forgery detection[12], and with the advances in Generative Adversarial Networks (GANs) we can also produce realistic examples or enhance images of a given classification[13].

CNNs have also been used to help classify artwork[14], clothing types[15], and music[16]; but only Iwana et al.[17] have applied this work to book covers to attempt to classify the genre so far as we can tell. It is their work that we propose to build on. They produced a cleaned dataset, discussed further in section [2.4], of 57,000 images spanning 30 equally represented genres and presented the classification results of an unseen test set of 10% of the data after training AlexNet, a specific CNN architecture. They report a Top 1 accuracy of 24.7% and a Top 3 accuracy of 40.3% across the 30 classes (i.e. a random chance of 3.3% and 9.9% respectively). Class specific Top 1 accuracies reach up to 68.9% for the Test Preparation genre, but are as low as 6.8% for the Politics & Social Sciences genre. We will explore the results of their work further, and discuss the potential reasoning for this range of performance by class if we see similar results, in our main work.

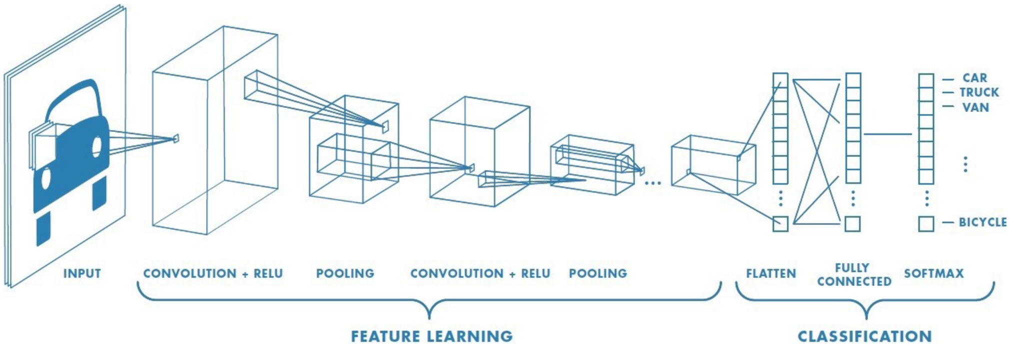


Figure 2.1: An simplified example of a CNN architecture

2.3 CNN Architectures and Transfer Learning

For all of the development of CNNs discussed in this section, they continue to maintain the same overall structure they started with, that of having a clear delineation on the network where they go from feature extraction to classification as shown in fig. [2.1]. The first part of the network consists of various convolution/filter and pooling layers, used to extract image features (e.g. edges) and to reduce the image dimension respectively. The second part of the network is a traditional densely connected artificial neural network used for the actual classification; a few fully connected layers followed by the output/target layer with as many neurons are there classes in the problem. Much of the development has gone into the feature extraction part of the network over the last 8 years or so; starting in 2012 with AlexNet[18], with the paper hailed as one of the most influential papers in applied computer vision. A few years later these were joined by what have been come to be known as the VGG CNNs (named after the Visual Geometry Group at Oxford)[19]. The following year the family of Inception models (also known as GoogLeNet) start to appear[20]; around the exact same time ResNets were proposed by He et al. at Microsoft[21] which later led to the combination of these in 2017. We do not detail here the specifics of CNN layers, including the types of convolution or pooling, nor the differences in all these architectures as we will detail these in the work itself, but those chosen to be used within this work are mentioned in section [3.1.2].

We will be using transfer learning to reduce the effort involved in training our network. Transfer learning was first proposed in 1993 by Lorien Pratt[22] and has been said by Andrew Ng to be the next driver for commercial machine learning success. A comprehensive survey of transfer learning has been completed by Zhuang et al.[23] so we do not detail here all of the developments and differences in various transfer learning algorithms, especially as we plan to take the most basic approach. Whilst many have worked to develop detailed algorithms that take into account the overlap of the source data domain and the target data domain and adjust a network accordingly, we are taking a more simple approach that is discussed in section [3.1.2] and reusing all but the (dense) classification layers of a network pretrained on ImageNet as the starting point for our training.

2.4 Existing Datasets

The dataset collected by Iwana et al.[17] is the largest, and as far as we can tell only, clean dataset to collate together the book covers of over tens of thousands of books. The dataset exists in two distinct forms; the *BookCover30* dataset and the *Book32* dataset that contain 57,000 and 207,572 records respectively, split across 30 and 32 classes again respectively. Both datasets have the same columns; *Amazon Index, Filename, Image URL, Title, Author, Category Id, Category*. For *BookCover30* there is also a reduced dataset containing just the filename and category ID that can be used for training and testing after image download. A comparison of the volumes, as well as the split between train and test sets, and a key between the category ID and category label can be seen in table [1]. In the case that a book belonged to multiple genres, the reported genre was chosen at random.

The reduced *BookCover30* dataset has already been pre-processed to remove 2 underrepresented classes, and to scale and shape the images; this will likely have to be re-sourced before the processing has taken place as the various CNNs we are going to use have different input shapes. There is also some other issues with the dataset; in particular that no manual filtering was done to ensure that the images are representative of the actual book cover, an example where this is not the case can be seen in fig. [2.2]. The data also looks to cover multiple languages so it is possible that there will be duplication of items (although often foreign editions of books have different covers) and this could also lead to additional difficulty if there is a difference in design choices between different countries. Furthermore, this dataset and paper were first published in 2016; while using the same dataset would be useful for comparison it is possible that the same script they used to generate their data could be repurposed to allow for a more up-to-date dataset to be produced, with higher resolution images and a potentially more representative sample of modern books. More discussion on this is repeated in section [3.1.1].

We mention here for completeness that during our research we did come across a potential alternative source of data, that of the Open Library[24] who as part of the Internet Archive project have the goal to have *"One web page for every book ever published"*. The regular data dumps[25] are a possible alternative source to using the Iwana dataset, however this dataset would come with some uncertainty attached. Whilst the volume of records is undoubtedly larger in size, it is not clear without processing all of the records how many of these actually have a genre classification at all, and given that their book covers data dump has not been updated since 2016 it does not provide any more recent data than Iwana's does. We would also be remiss to not mention that as this is an open-source project, the quality of the data itself would be more in question than with the *Book32* records. Finally, it would remove our ability to use Iwana et al's paper as a reference in terms of performance of the networks as not only would the training data itself be vastly different, but the genres are also different in this source.

Overall the *Book32* dataset is the best quality and most label-complete dataset available that we could find, and to produce a new one from scratch even using Open

Label	**Category Name**	**Size Book32**	**Size BookCover30 Training**	**Size BookCover30 Test**
0	Arts & Photography	6,460	1710	190
1	Biographies & Memoirs	4,261	1710	190
2	Business & Money	9,965	1710	190
3	Calendars	2,636	1710	190
4	Children's Books	13,605	1710	190
5	Comics & Graphic Novels	3,026	1710	190
6	Computers & Technology	7,979	1710	190
7	Cookbooks, Food & Wine	8,802	1710	190
8	Crafts, Hobbies & Home	9,934	1710	190
9	Christian Books & Bibles	9,139	1710	190
10	Engineering & Transportation	2,672	1710	190
11	Health, Fitness & Dieting	11,886	1710	190
12	History	6,807	1710	190
13	Humor & Entertainment	6,896	1710	190
14	Law	7,314	1710	190
15	Literature & Fiction	7,580	1710	190
16	Medical Books	12,089	1710	190
17	Mystery, Thriller & Suspense	1,998	1710	190
18	Parenting & Relationships	2,523	1710	190
19	Politics & Social Sciences	3,402	1710	190
20	Reference	3,268	1710	190
21	Religion & Spirituality	7,559	1710	190
22	Romance	4,291	1710	190
23	Science & Math	9,276	1710	190
24	Science Fiction & Fantasy	3,800	1710	190
25	Self-Help	2,703	1710	190
26	Sports & Outdoors	5,968	1710	190
27	Teen & Young Adult	7,489	1710	190
28	Test Preparation	2,906	1710	190
29	Travel	18,338	1710	190
30	Gay & Lesbian	1,339	0	0
31	Education & Teaching	1,664	0	0

Table 1: Comparison of the number of records per category between the Book32 and Book30 training and test sets

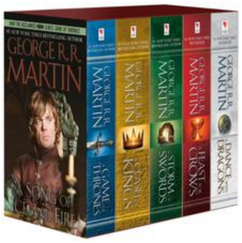


Figure 2.2: An image in the *BookCover30* dataset showing an anthology box collection rather than the cover

Library as a foundation would take a large amount of work for likely very little gain, so it is this dataset we will use in our work.

2.5 Debugging CNNs

CNNs, along with most other deep-learning methods, have been seen for a long time as *black box* methods i.e. it is not possible to understand what a specific neuron or layer within the network is doing. The attitude to black box methods has soured over the years for 2 major reasons. The first being that black box methods are illegal in some fields such as underwriting; a lender must be able to explain to an applicant why they were rejected for a loan, and black box methods do not allow for this. The second is that they are almost impossible to debug and understand where the network may be making an error. Many examples of this were first shown by Nguyen et al[26] in 2015, where images could be evolved to trick CNNs that return high precision results for obviously garbage images. More recently Su et al.[27] managed to show that perturbation of even a single pixel in an image can generate a change in the predicted class from otherwise high performing classifiers. These examples, combined with a drive for more explainable AI, has led to a few key developments in the debugging of CNNs via representative visualisation.

In 2018 Qin et al.[28] produced a survey of the major visualisation methods and whilst there has been some developments, mostly in the implementation of these methods in common software packages, their work remains a good summary of the state of the work. They call out four key visualisation methods:

1. Activation maximisation - A visualised image is generated such that it maximises a specific neuron's activation.

2. DeconNet - Uses an inverted CNN structure to identify the pattern in the input image that triggers specific neuron activation.

3. Network Inversion - Recreates an image based on the input image from a specific layer, helping identify what information is preserved by this layer.

4. Network dissection - Applies semantic concepts to each neuron such as scene, material, object etc.

For the purpose of this work we focus only on activation maximisation as this is what we plan to use to generate representative images of some of the genres.

Activation maximisation was proposed by Erhan et al.[29] in 2009 and the idea itself is quite simple; iteratively perturbate an image to maximise the activation of a given neuron. The resultant image then represents what that neuron has learned. The actual algorithm uses backpropagation instead of perturbation to improve the synthesised image over many iterations. Compared to normal backpropagation where we calculate the change in the network error with respect to each weight and bias and use that to update the values, here we calculate the change in activation with respect to the input and use this as the base for our update rule:

$$input_t = input_{t-1} - \eta \times \frac{\partial ActivationMaximisationLoss}{\partial input_{t-1}} \tag{2.1}$$

where $ActivationMaximisationLoss$ is some loss measure that decreases as the activation increases, and η is a learning rate. We stop when the activation of the neuron cannot be increased further. This can be applied to almost any neuron in the network with no alteration to the structure except for those in the final layer. The final layer activation function, the transformation applied to the inputs of the neuron, must be changed from the pre-existing SoftMax (σ) as defined in eq. (2.2), which takes a vector of activations for the K output classes and normalises them to be between 0 and 1 with a sum of 1, and be replaced with a linear function instead (often just the linear sum of inputs is used). The reason for this is that to maximise the output of a SoftMax function for a single neuron, it is possible to minimise the values of the other neurons. This leads to a case where the generated image would not be representative of our target class, but just be more representative than it is of the other classes. By using the unbounded linear function we can continue to apply updates to the input until we see little to no increase in the activation for our target neuron, irrespective of the other class activations.

$$\sigma(\vec{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{2.2}$$

Often, this approach is taken together with some form of normalisation to improve the interpretability of these images. As mentioned, pictures that are composed entirely of noise can trigger activation in the output neurons in a network with high certainty, so these normalisation methods are applied to reduce the production of these types of images and return more visually understandable results. Some of these normalisations, such as l_2 decay (which usually uses the sum of squared weight, but pixel values in this case) are used to prevent a few pixels dominating the image, or applying a Gaussian Blur can be done to ensure that no area of the image has too high an information frequency.

The final extension to this work is that of *Deep Generative Network Activation Maximization* (DGN-AM)[30] which adds in a generative network, a network with the goal of producing an item of the specified class rather than to classify an object into a

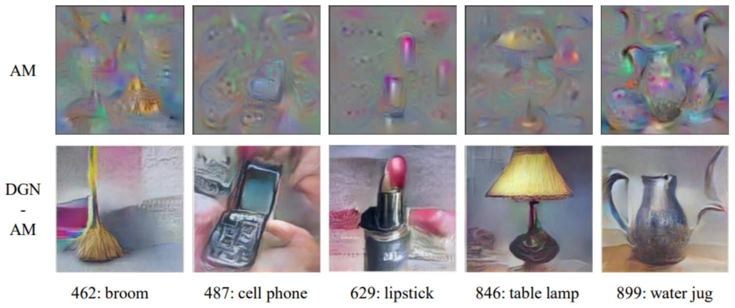


Figure 2.3: Activation Maximization vs Deep Generative Network Activation Maximization from ”How convolutional neural network see the world - A survey of convolutional neural network visualization methods” by Qin et al, 2018

class, to produce more realistic images as opposed to the abstract ones often produced by activation maximisation. They were first used in 2016 and have shown greatly improved results, but also require greatly increased network complexity and training time. DGN-AMs have also been shown to provide worse outputs when the generative network is not the same architecture as the CNN and so multiple generative networks would need to be created for this work which would be far too computationally expensive; as such we chose to not use them within this work. An example taken from Qin's survey is shown in fig. [2.3] highlighting the huge change that using DGN-AMs produce.

3 Proposal

3.1 Planned work

The proposed work is to be completed in 3 distinct parts, starting with the work detailed in section [3.1.1] focusing on the collection and reprocessing of the data first generated by Iwana et al.[17] to ensure completeness and correctness. We then in section [3.1.2] propose to reproduce the work of Iwana et al. and then develop this further using recent advances in CNN architecture. Finally section [3.1.3] discusses the plans to use activation maximisation on the CNN to produce images based on the output class.

3.1.1 Part 1: Data Collection and Preparation

The work first produced by Iwana[17] uses a reduced dataset of 57,000 book covers that span 30 classes with 1,900 records each. We will first step back to the *Book32* dataset to have available to us the full use of 207,572 books; we will download and ensure via some manual testing that the URLs of the images are still the correct ones from the time of writing.

At this stage any images will all be pre-processed including scaling to be a sensible size. The complexity is that of the shape of book covers in general, whilst there is some

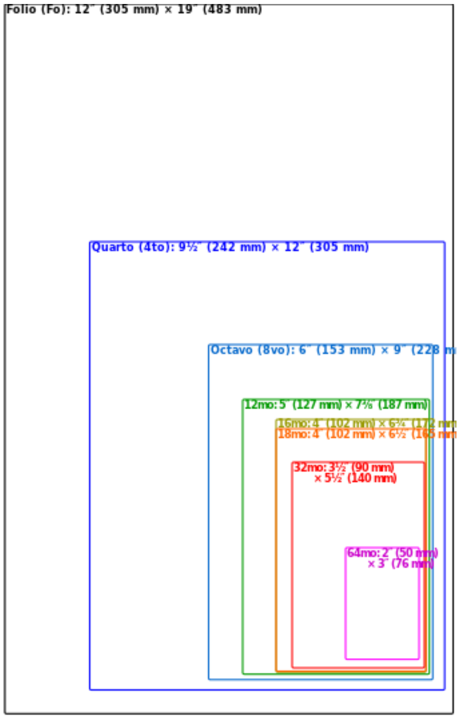

Folio (Fo): 12" (305 mm) × 19" (483 mm)
Quarto (4to): 9½" (242 mm) × 12" (305 mm)
Octavo (8vo): 6" (153 mm) × 9" (228 m
12mo: 5" (127 mm) × 7⅜" (187 mm)
16mo: 4" (102 mm) × 6¾" (172 mm
18mo: 4" (102 mm) × 6½" (165 mm
32mo: 3½" (90 mm)
× 5½" (140 mm)
64mo: 2" (50 mm)
× 3" (76 mm)

Figure 3.1: Comparison of book sizes based on[32]
By Cmglee - Own work, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=15264645

standardisation across the industry, is not fixed as shown in fig. [3.1] and specific genres of books such as cookbooks are likely to take a different shape due to their usage and content. This means that by scaling the images we are likely to impact some classes more than others. Analysis will be conducted to identify the distribution of shapes and if this varies by class, as well as a review of various methods used within the literature to approach such a problem. In particular an approach similar to the NIST dataset as reproduced by Cohen at al.[31] could be used with some adaptations for colour images.

Once this has been completed we will need to re-sample the dataset down to the same size as the original work, however we may not use the exact same subset of books they selected if we believe we can responsibly remove unrepresentative images such as the one displayed in fig. [2.2]. This will lead to an inability for exact comparison between our work and that of Iwana, however it should help improve the accuracy of the model by removing non-representative training and test data. Where possible the same records will be used within the training and test sets, and only those that we identify to be non-standard will be swapped for another record.

3.1.2 Part 2: Genre Predictor via Transfer Learning

In their work, Iwana et al[17] trained 2 CNNs (LeNet and AlexNet) on the training set and report the Top 1 and Top 3 predicted class accuracy by book genre on the test

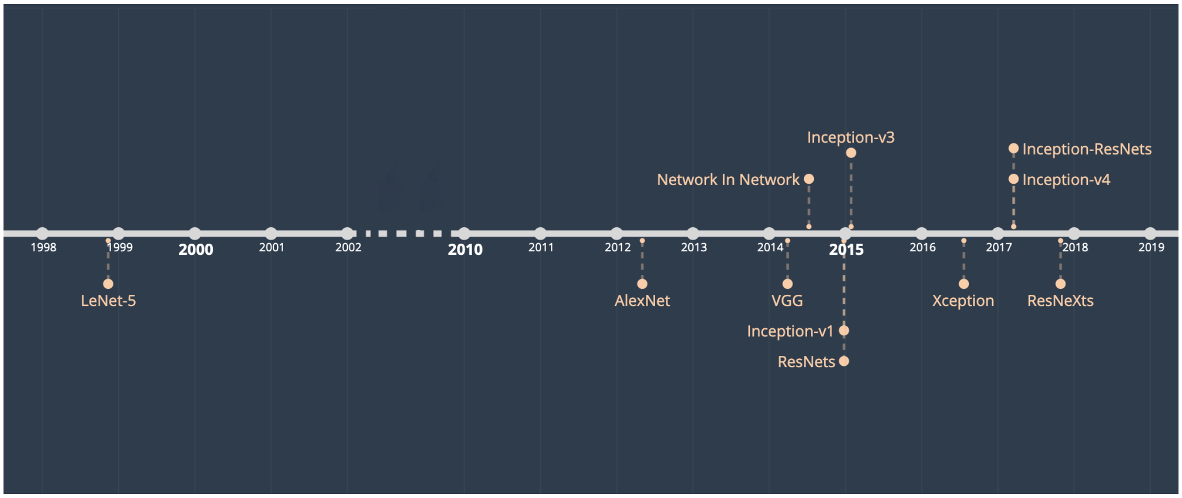


Figure 3.2: Key CNN architectures and their year of publication[33]

set. We will attempt to reproduce the results achieved by them using AlexNet on our adjusted dataset and just use the LeNet results for an approximate comparison. We will use the same method as they did to train AlexNet i.e. apply transfer learning to a pre-trained version of the network on the ImageNet dataset, removing the classification layers of the network, those densely connected non-convolutional layers, including the final 1000 neuron dense layer (representing the 1000 classes) and replace it with a similar network ending with a layer of 30 densely connected neurons instead, 1 for each of our genres, freezing the weights of the feature-extraction part of the network before beginning training.

Once a baseline has been established we will intended to improve the per-class accuracy by using some of the newer architectures as shown in fig. [3.2], namely Inception-ResnetV2 and a ResNeXt, and see if these yield better results. We will also look at using any of the techniques in the literature for differing shaped input images at this point to see if it is possible to provide not just the cover image to the network, but also information about the shape and size of it as well in the chance that this may improve accuracy. We again will use transfer learning here to greatly reduce the computational cost and time to train these models. The specific details of these CNN architectures will be discussed more within the work itself.

3.1.3 Part 3: Feature Visualisation

Once we have compared the performance of the different CNN architectures on the dataset we will attempt to use activation maximisation as discussed in section [2.5] with the network and instead of predicting the genre based on an image, it will produce an image based on a requested genre. This is sometimes also referred to as feature visualisation, although rather than individual features we are trying to reproduce a whole input. We will only take the best and worst 2-3 performing genres based on the previous stage to attempt this for.

To do this we propose the use of the relatively new and still under active development python package *tf-keras-vis*[34]. This package offers 2 key feature visualisation techniques; activation maximisation for filter layers and final output layers, and attention (i.e. what area of the image is driving the output class the network has predicted). Using this package we should be able to produce images that maximise the activation of our target class neuron. As mentioned in section [2.5] it is probable that these images will be abstract and may be entirely uninterpretable, but they may shed some insight on how specific genres are easier to predict than others and so we believe it adds value to the project.

3.2 Hardware and Software

Even using transfer learning, the amount of epochs the networks will have to train for to get a reasonable accuracy is still likely to be quite high, and as these are complex deep networks this will take a lot of time. Because of this we propose the use of Google Colab[35] to enable the use of GPUs, and potentially Tensor Processing Units (TPUs), for an order of magnitude speed up in the training of these networks. There is a limit on the amount of time a single session can run for, and there is no guarantee on the specific G/TPU brand/architecture that will be used, but without this or an approach similar it is unlikely that we posses the computing power to complete the work. As a point of note, TPUs are likely to offer a faster training however require a non-trivial amount of additional setup and so the choice to use them will be evaluated on a case by case basis. Because of this it is not possible to specify the exact hardware that we will use for this work.

For software, the majority of the work in terms of downloading, pre-processing, modelling, evaluations, and visualising features will be done in python using Tensorflow 2.x with a Keras backend, with heavy support from both the sklearn package and the aforementioned tf-keras-vis package. Visualisation outside of features may be done using R after some pre-analysis in python due to the extensive variety of libraries that support the production of graphics in R compared to python, as well as our own familiarity with the language.

4 Conclusion

In this work we have laid out a problem which can be summarised as *Can a CNN judge a book by its cover* and highlighted some of the previous work already completed in this space, namely that of Iwana et al. We discussed the existing available datasets, known as *BookCover30* and *Book32*, and explored some of the limitations this dataset currently has and we may potentially wish to address. Following this we laid out a 3 stage plan for trying to answer the question, starting with data collection and pre-processing, then using transfer learning to train high-performing CNN architectures for this problem and evaluating their performance. Finally we spoke of plans to use feature visualisation techniques to produce idealised covers for the most and least accurate genres from our models. Overall the proposed work should produce, as far as we can tell at time of writing, the only follow-up work to the Iwana paper and update their work using the more recent developments in CNN architecture and computing power.

References

[1] A. G, "Book Cover Design Prices in 2019 - Rocking Book Covers." [Online]. Available: https://www.rockingbookcovers.com/book-cover-design/book-cover-design-prices-2017/

[2] A. Watson, "Global book publishing revenue 2018-2023," 2019. [Online]. Available: https://www.statista.com/statistics/307299/global-book-publishing-revenue/

[3] J. Willis and A. Todorov, "First Impressions," *Psychological Science*, vol. 17, no. 7, pp. 592–598, jul 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16866745http://journals.sagepub.com/doi/10.1111/j.1467-9280.2006.01750.x

[4] R. Demush, "A Brief History of Computer Vision (and Convolutional Neural Networks)," 2019. [Online]. Available: https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3

[5] S. A. Papert, "The Summer Vision Project," *MIT Home*, jul 1966.

[6] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, oct 1959. [Online]. Available: http://doi.wiley.com/10.1113/jphysiol.1959.sp006308

[7] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Co., Inc., 1982.

[8] Y. Le Cun, L. Jackel, B. Boser, J. Denker, H. Graf, I. Guyon, D. Henderson, R. Howard, and W. Hubbard, "Handwritten digit recognition: applications of neural network chips and automatic learning," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, nov 1989. [Online]. Available: http://ieeexplore.ieee.org/document/41400/

[9] M. Everingham, L. Van~Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, jun 2010.

[10] 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.

[11] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller, “Recognizing Image Style,” *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, nov 2013. [Online]. Available: http://arxiv.org/abs/1311.3715

[12] S. J. Gideon, A. Kandulna, A. A. Kujur, A. Diana, and K. Raimond, “Handwritten Signature Forgery Detection using Convolutional Neural Networks,” *Procedia Computer Science*, vol. 143, pp. 978–987, jan 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050918320301

[13] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 105–114, sep 2016. [Online]. Available: http://arxiv.org/abs/1609.04802

[14] J. Zujovic, L. Gandy, S. Friedman, B. Pardo, and T. N. Pappas, “Classifying paintings by artistic genre: An analysis of features & classifiers,” in *2009 IEEE International Workshop on Multimedia Signal Processing*. IEEE, oct 2009, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/5293271/

[15] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms,” aug 2017. [Online]. Available: https://trends.google.com/trends/explore?date=all{&}q=mnist,CIFAR,ImageNethttp://arxiv.org/abs/1708.07747

[16] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, jul 2002. [Online]. Available: http://ieeexplore.ieee.org/document/1021072/

[17] B. K. Iwana, S. T. R. Rizvi, S. Ahmed, A. Dengel, and S. Uchida, “Judging a Book By its Cover,” *Journal of Architectural Education*, vol. 72, no. 1, pp. 180–181, oct 2016. [Online]. Available: http://arxiv.org/abs/1610.09204

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, may 2017. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3098997.3065386

[19] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” sep 2014. [Online]. Available: http://www.robots.ox.ac.uk/http://arxiv.org/abs/1409.1556

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” sep 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” dec 2015. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/http://arxiv.org/abs/1512.03385

[22] L. Y. Pratt, “Discriminability-Based Transfer between Neural Networks,” Tech. Rep., 1993.

[23] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A Comprehensive Survey on Transfer Learning,” nov 2019. [Online]. Available: http://arxiv.org/abs/1911.02685

[24] “About Us | Open Library.” [Online]. Available: https://openlibrary.org/about

[25] “Open Library Data Dumps | Open Library.” [Online]. Available: https://openlibrary.org/developers/dumps

[26] A. Nguyen, J. Yosinski, and J. Clune, “Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images,” Tech. Rep. [Online]. Available: http://evolvingai.org/fooling.

[27] J. Su, D. V. Vargas, and S. Kouichi, “One pixel attack for fooling deep neural networks,” oct 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8601309http://arxiv.org/abs/1710.08864http://dx.doi.org/10.1109/TEVC.2019.2890858

[28] Z. Qin, F. Yu, C. Liu, and X. Chen, “How convolutional neural network see the world - A survey of convolutional neural network visualization methods,” *Mathematical Foundations of Computing*, vol. 1, no. 2, pp. 149–180, apr 2018. [Online]. Available: http://arxiv.org/abs/1804.11191

[29] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, “Visualizing Higher-Layer Features of a Deep Network,” *Technical Report, Univeristé de Montréal*, 2009.

[30] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, “Synthesizing the preferred inputs for neurons in neural networks via deep generator networks,” Tech. Rep.

[31] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “EMNIST: an extension of MNIST to handwritten letters,” feb 2017. [Online]. Available: http://arxiv.org/abs/1702.05373

[32] American Library Association. Committee on Library Terminology. and E. H. E. H. Thompson, *A.L.A. glossary of library terms : with a selection of terms in related fields*. American Library Association, 1971.

[33] R. Karim, “Illustrated: 10 CNN Architectures - Towards Data Science,” 2019. [Online]. Available: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

[34] “keisen/tf-keras-vis: Neural network visualization toolkit for tf.keras.” [Online]. Available: https://github.com/keisen/tf-keras-vis

[35] Google, “Colaboratory – Google.” [Online]. Available: https://research.google.com/colaboratory/faq.html