What is a book cover worth?

Najim Yaqubie
najim.yaqubie@columbia.edu

Charles Gordon Summers
charles.summers@columbia.edu

Daniel Carpenter Silver
dcs2180@columbia.edu

Abstract

Is a picture worth a thousand words? Book covers are the first glimpse into the book and need to immediately convey a convincing message to induce readers. But how do you design a book cover and what should you prioritize? We use Amazon product and review data to explore book covers. Using nearest neighbor and convolutional neural network models, we aim to learn what is a good book cover and help design novel covers by generating suggestions.

1. Introduction

We shouldn't judge a book by its cover, but we are often influenced by its immediate outward appearance. Covers are designed to catch the eye of the would be reader otherwise it may never be opened. [7] Over three million books are published each year, but only a tiny fraction are read widely. [18] What features does a cover contain? Can a cover be indicative of a book success?

Often, conventional cover design requires hiring expensive art designers over a prolonged undefined and non-standardized process. Designers often have their own style derived by a personal artistic system, all of which is designed to entice potential readers to open a new book. [7] However, if the goal is to have more readers, or be a more successful book, why not explore whether there is a relationship between cover design and success directly?

We use image feature extraction and machine learning techniques to learn successful covers for books. We discover that certain elements of book covers are related to the eventual success of a book as defined by Amazon reviews and overall rating. We then generate simple suggestions for improvements. Can we actually judge a book by its cover?

2. Related Work

Our approach incorporates active research areas such as book success prediction and convolutional neural networks.

2.1. Book Success Prediction

Extracting features from books, whether through textual content or visual cover features, is a lightly explored research area. This may be due to the artistic element as well as shifting trends in book cover design. Book covers have shifted from representing the text between the covers to representing literary genres and brands. [10] This commodification of book covers as marketing devices may actually lead to stronger correlation between cover design and overall expected success.

As the content still remains text based, it's no surprise that a lot of attention and research goes into the typeface or font used on the cover as well as the book. [7] Previous work has explored the relationship between font and book covers to discover latent patterns based on genre. [16]

Finally, there has been direct work in using metadata and content to predict book sales. Studies have show that different genres have different driving factors for success. [18] Universally, it does seem that the publishing house behind a book is the biggest indication of success while metadata about the author comes second. Given such research, we shouldn't expect a very strong correlation between book cover and success, however, these studies did not take into account the effect of a cover on sales.

We will restrict ourselves to analyzing book covers and measuring success by average review on Amazon, thereby discovering how much of a relationship exists between cover design and book success.

2.2. Convolution Neural Networks

Convolutional Neural Networks (CNNs) are multi-layer neural networks that utilize learned convolutional filters to extract features at different semantic levels. Instead of using predefined image features, CNNs learn the most important feature representation for image recognition for a particular task, whether it be generation or classification. [9][17]

CNNs have been used to predict the genre of a book simply by visual clues on the cover successfully demonstrating a shared contextual relationship between content and form. [5] Furthering this study, it was found that Residual Neural Networks (ResNet) outperformed CNNs with respect to classification within the same task, strengthening the existence of this relationship that may be exploited for future cover design. [8] Interestingly, they found a cover's text is a more important signal of a genre than the artwork.

There has been work to use CNN's to generate cover art based on content. In a recent study, they were able to use a Deep Convolutional Generative Adversarial Network (DCGAN) to generate cover art trained on music lyrics from an album. [4] The adversarial network allows for unsupervised learning on a massive scale to discover whether album art can be fully automated and designed for success. [13] However, their results were not promising and points towards similar difficulty in attempting to generate novel book cover art based on the textual content of a book.

We will explore using CNN's for book cover feature extraction and estimating book success based on those features if there is a latent relationship.

3. Data

We use a number of sources for our analysis. Mainly, we will rely on the Amazon Review Data with respect to books. This contains 51,311,621 reviews for 2,935,525 books containing product information, links to images, and metadata. [11] Related to this data is a curated set of 207,572 cleaned book cover images of size 224x224 that may be easier to rely upon as some product images are more than just the cover. [6] This normalized data set linked to amazon review data forms the backbone of our analysis.

We explored augmenting our data set by adding sales numbers from NPD BookScan and body text from Project Gutenberg. We use average review rating as a proxy for book success by revenue but could not gain access to the NPD BookScan dataset. [12] Lastly, we explored using text available online to enrich our feature vectors, however, most of the books on our dataset were not available. [2]

4. Methodology

We aim to discover and extend current studies on analyzing book covers. First, we examine the review data set and consolidate metadata with book cover images. Second, we use traditional image feature extraction to cluster similar covers together and examine whether these features correlate with book success. Then we explore using neural networks to use learned features to predict book success on new proposed covers. Lastly, we combine these models to build a suggestion engine web app.

4.1. Data Extraction

Using the Amazon review data set, we extract a list of books and URLs for book cover images. However, these URL's are of book cover images that are not normalized in terms of image content, image size, nor even angle. We then used the book cover dataset which had 207,572 cleaned book cover images that are normalized with respect to the aforementioned concerns. We use this set of books to filter out reviews from the original data set.

After filtering out reviews, we then analyzed each review and consolidated ratings across reviews for books. We also extracted books that were also bought with that book through the metadata associated with the book cover set. Together, we were able to attribute an average rating per book weighted by the total number of reviews as seen in Figure 1. Essentially, a book with 1 5 star rating is ranked lower than a book with 10 5 star ratings. These ranks are also persisted as seen in Figure 2.

Row	total_reviews	rating	ASIN
1	115455.0	3.984641932700604	038568231X
2	91711.0	4.105053489100756	B000X1MX7E
3	88491.0	4.76244550885313	0312577222
4	87513.0	4.683847141939627	B003156C4E
5	87279.0	4.683857464849201	B001C4VLZQ

Figure 1. Reviews are consolidated by book ASIN and provided an average rating weighted by total number of reviews.

Row	ASIN	ranking	also_bought
1	B01HI5V3HI	3921775	
2	B01HIP8KLU	2655897	
3	B01HJOAYR2	583884	B078V7G3XY,B01M0C8BVU,B0747HDXCX,B00L43QLD2,B00VEE42DW,B012JPGK3E,B01MR8C5PW,B01EH10DAE,B009B2YKTU,B00L1BW4AA,B00RHCX25O,B005T54LCY,B00MMNTO14,B0114OL8SG8,B00SQ4DLBM,B00IFSU3OM,B009FR9CCG,B01L0Y8P20,B01B5BMVXQ,B00CENJBSG,B01L0Y8P1A,B008MM88BG,B00DPZZVBU,B07DMZB97R,B00FD9WTZA,B07CQ7P24Y,B009CQ9T4,B00KP8DA72,B07JWFDM2F,B004RKXHEU,B07HDMC4GH,B07K6DDW7J,B07B26JHMD,B07FCQV31V,B077WXP3KG,B076X2TNRM,B002G54Y2M,B06XCM8VPB,B079KX1XFD,B078CY7TLY,B078X12Y75,B074N1XJR2,B07BR6RNS3,B07BK3M7V7,B01ER0MI1S,B074PD2QR3,B07D4M7N3L,B071X2K4S6,B075C9TMR1,B002361LHU,B002DW92V6,B01H03IBHS,B01MRF794N,B00OBT9B60,B0783P29QP,B01NCUSJYC,B00UI4ERXY,B075JGPRR5,B07634GR5R,B000OCXHT0,B01LW8CYI,B00KWFZ7CE
4	B00T3PMFHO	1103611	B00408PIOA,B007WKFMGS
5	B00T884AME	126974	B01D6RFQPC,B01NC30DNW,B07652YGR9,B01CNZV0Q2,B01BIFXV88,B07CK1MRPJ,B00PMVTNQO,B01N3QYKNX,B073W9DBPS,B075KS2V9N,B07BBX3FMN,B01A86GUTK,B07CZBV5LQ,B07K4TS4SR,B00KECZLG2,B01N6SFVMS,B00I5Q71YE,B07D67FG2H,B010XQ4WDG,B01FLJKZNY,B00LKGBK2A,B01MYIZPZW,B07K1FJ75F,B077X4LQ1S,B01MYC4RV7,B07FZTSFXY,B00ZJG39BU,B01NABG44N,B013CFG1HA,B072L1NZ16,B07GHYQ4T1,B008Y0XH1C,B00WHXYZ08,B01BECUPJS,B0793X6RJQ,B01A7RJ94S,B00495Z8E0,B07HLC6R6L,B077T898H9,B01LVXR8M8,B07L5R5KWZ,B076HKP2PM,B07D8JM9W9,B0153IIWDK,B00NCYZ2M2,B0166SHTVM,B000O5INIA,B07BX14NRW,B01JX6EFKW,B00EBC0ZFI,B074WXMKVJ,B00BZED006,B017RMH400,B07C2NC58K,B078RQNCDT,B00MX0EC8G,B01MT3UWAM,B005GOCD6A,B00CFQJNKA,B00LB6DKD6,B000ZU9AFA,B008H7CEQ0,B0193K5IRW,B0000VLIQU,B01DLBGDSC,B005FVPFEK,B01N27XYYO,B07FH84SVH,B075LYJWN5,B01FUV0WJW,B00JUUZOD4,B00IHGB70WM,B00YQKID04,B01M662J9X,B0130L

Figure 2. Rankings for all books also take in consideration other books that were bought with that book.

4.2. Image Feature Extraction

In order to investigate what specific qualities of a book cover correlate to the average customer rating, we extract interoperable features using traditional image processing techniques with Python's image library to extract images into red, green, and blue (RGB) vectors. Although there are endless features one could extract from pixel vectors, we focused on 3 distinctive features that proved to be highly influential in the overall quality of a book cover. These features are colorfulness, brightness and dominant colors.

There are many ways to define colorfulness of an image, however, we follow the perceptual metric developed by in [3]. This study asked 20 non-expert participants to rate images on a 1-7 scale of colorfulness. This survey was conducted on a set of 84 images. They found through these experiments that a simple opponent color space representation along with the mean and standard deviations of these values correlates to 95.3% of the survey data.

$$rg = R - G, yb = \frac{1}{2}(R + G) - B$$
$$C = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} + 0.3 \times \sqrt{\mu_{rg}^2 + \mu_{yb}^2}$$

In these equations, R, G and B refer to the respective vectors in the red, green, and blue vector representation of pixels. This turned out to be a simple and effective method of computing colorfulness as can be seen in Figure 3.

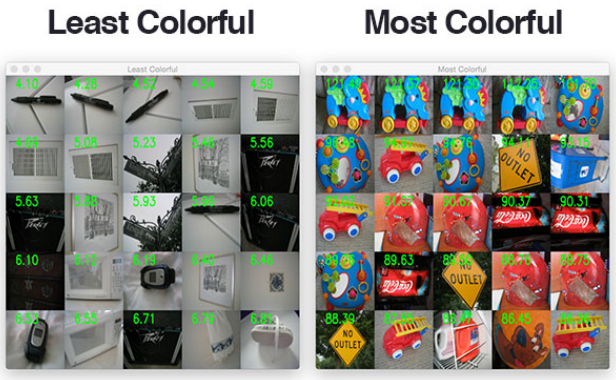

Figure 3. Notice how the image colorfulness metric has done a good job separating non-colorful images (left) that are black and white from "colorful" images that are vibrant (right). [14]

In order to extract the brightness of an image, we first explore different models for color. The traditional HSL (hue, saturation, lightness) and HSV (hue, saturation, value) are alternate representations of the RGB color model designed to more closely align with how human vision perceives color. In the HSV model, the V represents the brightness of a color by taking the maximum value of any of the three RGB components of color, while the HSL model is an average of the minimum and maximum RGB value.

Instead, it was found that these weighting mechanisms have severe shortcomings as seen in Figure 4 where converting an image to gray-scale fails. Using a different weighting mechanism that can be described as HSP, where P stands for perceived brightness, yields better results. [1]

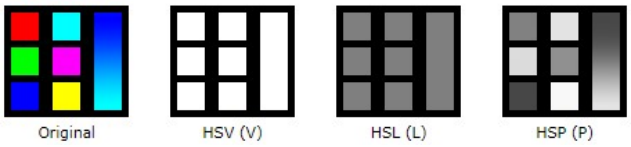

Figure 4. This simple color palette of red, green, blue, cyan, magenta, and yellow, plus a blue-to-cyan gradient, painfully exposes the limitations of HSV and HSL in their attempts to gauge the brightness of a color. [1]

In this model, we define brightness as a root mean square equation:

$$brightness = \sqrt{.299R^2 + .587G^2 + .114B^2}$$

Lastly, to determine the dominant colors in an image, a K-means clustering algorithm is used on pixel RGB values. K-means is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. By clustering pixels based on colors, we can determine the overall color palette of a book cover as well as the most dominant color. As color palettes are typically made of 5 colors, we set the number of clusters to 5. The most dominant color is defined as the mean of the largest cluster. We can see an example of a color palette extracted from a sample book cover in Figure 5.

Figure 5. Book cover (left). Palette extracted by K-means clustering with 5 clusters demonstrating the 5 colors in the palette (center). Other metrics quantified (right).

4.3. Nearest Neighbors Model

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a lazy non-parametric method used for classification and regression. In our case, we used a k-NN model to give us the k most similar book covers to an input book cover based on the 3 defined image features in Section 4.2, that is colorfulness, brightness, and dominant color.

This is a relatively straightforward model that allows us to find similar covers and map their associated metadata such as average rating to give us an idea of how successful the input cover will be. Given that book covers are typically correlated by our image features as shown in Figure 6, this model can give us information of how this book cover will be associated with other books.

Figure 6. Book covers were successfully classified by common moods or color palettes. Genres have a particular color association with respect to dominant color. [5]

4.4. Convolutional Neural Network Model

CNN's are a powerful technique for extracting information from images that is highly correlated to the target values. However, one drawback of this technique is that the extracted features used to make the predictions are not easily interoperable by humans.

We trained a CNN model on over 50,000 cover images for books with at least 20 reviews. The the average review of each book was used as the target value for the regression model. The model consists of 3 convolutional layers each with progressively larger filters in order to learn more discriminate features. Then the output from the convolutional layers is flattened and passed to two fully connected layers with 16 and 4 nodes respectively and rectified linear activation functions. Lastly, a single regression node is added to preform the linear regression book rating. An illustration of our network can be seen in Figure 7.

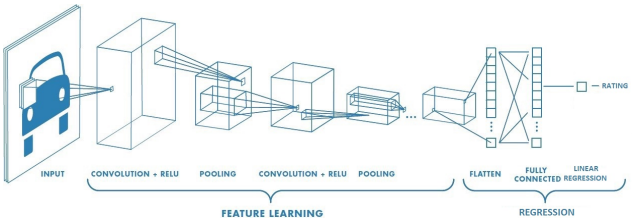

Figure 7. A deep CNN with 3 layers and a regression output. [15]

The book cover ratings were then normalized to range from 0-1 and the input data was split into a training and test set (75% and 25%). The model was trained over 10 epochs with mean absolute percent error used as the loss function. The final model achieved a testing loss of 6.79% error with a standard deviation of 7.02%. This means the model was, on average, able to predict the ratings of the books within the test set to +/- 0.34 stars with a standard deviation 0.35 stars. This strongly indicates that there is latent context within a book cover image that correlates to a book's overall success. However, discovering which specific features describe that latent context is very difficult.

4.5. Suggestion Engine

Given both the k-NN and CNN models, we are able to create suggestions for improving a candidate book cover in terms of overall expected success of the book. We use the k-NN model to get the 5 most closely related book covers and their metadata to gain an understanding of what this candidate book cover looks like as well as the success of those other books. We then run the CNN model to get an expected overall rating of the candidate book cover. If the candidate book cover has a statistically significantly lower rating than its neighbors, we then make suggestions such as raising the brightness of the candidate cover to increase its expected success as seen in Figure 8.

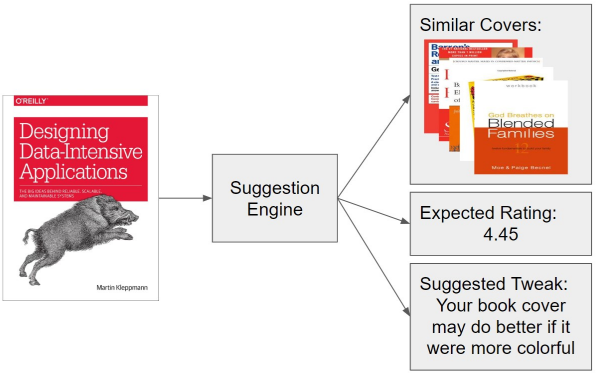

Figure 8. A candidate book cover (left) is passed into our suggestion engine (middle). We get a series of similar covers (right top) that share the same genre. We get an expected rating and a suggestion to improve the candidate cover.

5. System Overview

Our system centers around building a straightforward web application. Our backend environemt is built on top of Flask deployed onto Google Cloud. We deploy the application via Docker images built on Google Cloud Build and deployed via Kubernetes using Google Kubernetes Engine.

We used Google Dataproc as our main engine to generate our backend data and models. We ran our analysis via a series of Python scripts that stored their results in Google BigQuery. We trained our k-NN and CNN models using SKLearn and Keras frameworks respectively. All our images and trained models are on Google Cloud Storage.

The actual Flask app is deployed with both pre-trained models. When a user uploads a candidate cover, our app first normalizes the cover image. We then run our image feature extraction methods to pre-process the book cover and feed it into our suggestion engine. Our suggestion engine fetches similar book cover images from Google Cloud Storage and augments these covers with book rating, reviews, and metadata from Google BigQuery. Then, we run the image through our CNN model to get a predicted book rating. Lastly, we serve these results to the user via HTML as can be seen in Figure 9.

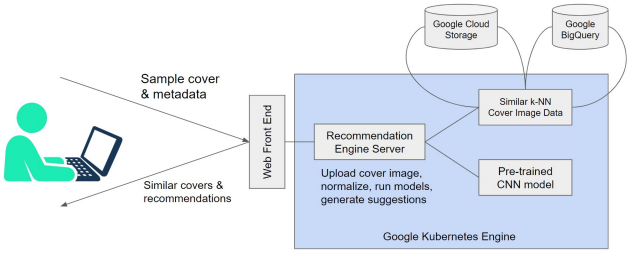

Figure 9. Our web application uses quite a few distinct services, but is one seamless experience from the perspective the user.

All relevant code, gathered data, and results are accessible via https://github.com/najerama/bookcovers.

There are 2 main bottlenecks in our system. First, training our models on images takes a very long time, so to incorporate further images, training time increases. Secondly, our Flask application does the same image processing on the server. While Kubernetes should load balance effectively, each image does take a bit of time to process sequentially. This may be sped up by concurrent calculations.

6. Experimental Results

We ran our suggestion engine on all book covers in out dataset and have seen very interesting results. Our engine provided suggestions for improvements on 29484 book covers, or 51.7% of our dataset, with an average expected rating increase of 0.322. This is greator the standard deviation of our CNN model, however, when we look only at rating increases that are greater than the standard deviation, we provided suggestions for improvements on 12294 book covers, or 21.6% of our dataset, with an average expected rating increase of 0.534. Figure 10 illustrates expected improvements or how much better a candidate book cover did when compared to similar book covers.

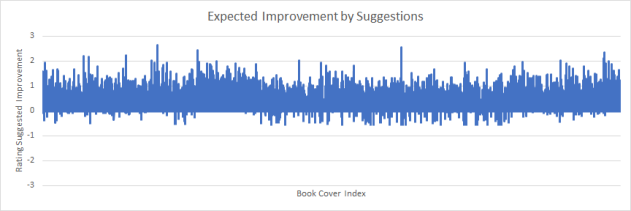

Figure 10. In general, our suggestion generation framework improvements to most covers.

When looking at the suggestions that provide the highest expected improvements, we do see a common theme such that they are poor visual covers. As we can see in Figure 11, the cover in question uses a color palette that is uncommon and not seen in similar colors. Additionally, the suggested covers that are similar look significantly better designed at first glance. This may be an example of a book that did not use a book cover designer and we can see the overall rating of the book and number of reviews suffered.

Figure 11. In this scenario, the candidate cover in question had a pre-existing average rating of 2.75 from 33 Amazon reviews. Our suggestion of lowering brightness has a higher expected rating.

7. Conclusions

Our CNN and k-NN models have shown promising results in suggesting improvements and predicting future success of book covers. There exists a strong relationship between a book cover and a book's metadata, such as genre and context. Given this information, generating sensible suggestions for book cover improvement in terms of success is not too difficult and proved via our suggestion engine. Lastly, our CNN model demonstrates a very strong relationship between a book cover and its expected success using automatically learned image features with less than 10% error. Given these successes, we published a web application to take advantage of these models as seen in Figure 12. Therefore, you can judge a book by its cover.

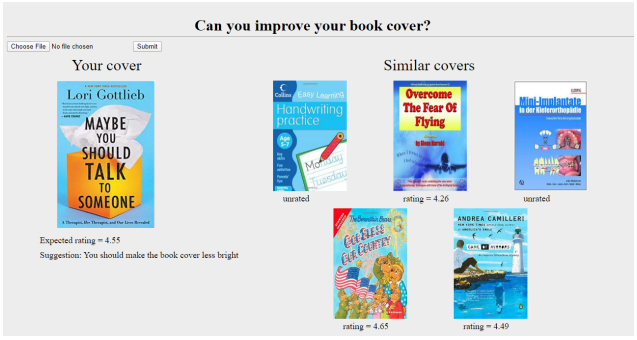

Figure 12. This is a screenshot of our actual application that can be run from our git repository. It's a very simple UI that takes an uploaded cover.

8. Future Work

There are many directions future work can take with respect to analyzing book covers and predicting book success. First, we can extend our models to employ a neural network to learn the relationship between a book's metadata such as publisher and author history with by extending upon [18].

We can also employ the lessons learned by the effect of typeface on a book cover extending upon [7]. There are also many other factors such as the content and topical popularity of a book that can affect it's success.

Importantly, our study makes the underlying assumption that Amazon Book Reviews are a good proxy for overall book success. However, the biggest improvement to our work would come in augmented out success parameters with actual sales numbers and determine whether this proxy relationship is fair.

Lastly, as touched upon slightly in [4], generating book covers based on book context, metadata and desired success is the hope in the not-so-distant future.

9. Individual Contributions

Charles Gordon Summers (33%) - Image gathering, System, WebApp Suggestion Engine
Daniel Carpent Silver (33%) - Image Feature Extraction, Convolutional Neural Network
Najim Yaqubie (33%) - Project Management, Data Extraction, Nearest Neighbors Model, Deliverables

References

[1] Darel Rex Finley. Hsp color model — alternative to hsv (hsb) and hsl, 2006.

[2] Project Gutenberg. Free ebooks, 2019.

[3] David Hasler and Sabine E. Suesstrunk. Measuring colorfulness in natural images. In Bernice E. Rogowitz and Thrasyvoulos N. Pappas, editors, *Human Vision and Electronic Imaging VIII*, volume 5007, pages 87 – 95. International Society for Optics and Photonics, SPIE, 2003.

[4] Alexander Hepburn, Ryan McConville, and Raul Santos-Rodriguez. Album cover generation from genre tags. 10 2017.

[5] Brian Kenji Iwana and Seiichi Uchida. Judging a book by its cover. *CoRR*, abs/1610.09204, 2016.

[6] Brian Kenji Iwana and Seiichi Uchida. Bookcover30 dataset, 2017.

[7] Anne Jordan and Mitch Goldstein. Uncovering the cover design, 2015.

[8] Sigtryggur Kjartansson and Alexander Ashavsky. Can you judge a book by its cover? 2017.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[10] Lisa Lau. *Contextualising Book Covers and Their Changing Roles*, pages 1–26. 05 2015.

[11] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and finedgrained aspects. *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[12] Nielson. Bookscan data, 2019.

[13] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 11 2015.

[14] Adrian Rosebrock. Computing image "colorfulness" with opencv and python, 2017.

[15] Sumit Saha. A comprehensive guide to convolutional neural networks — the eli5 way, 2018.

[16] Yuto Shinahara, Takuro Karamatsu, Daisuke Harada, Kota Yamaguchi, and Seiichi Uchida. Serif or sans: Visual font analytics on book covers and online advertisements. *CoRR*, abs/1906.10269, 2019.

[17] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *CoRR*, abs/1606.05328, 2016.

[18] Xindi Wang, Burcu Yucesoy, Onur Varol, Tina Eliassi-Rad, and Albert-László Barabási. Success in books: predicting book sales before publication. *EPJ Data Science*, 8(1):31, Oct 2019.