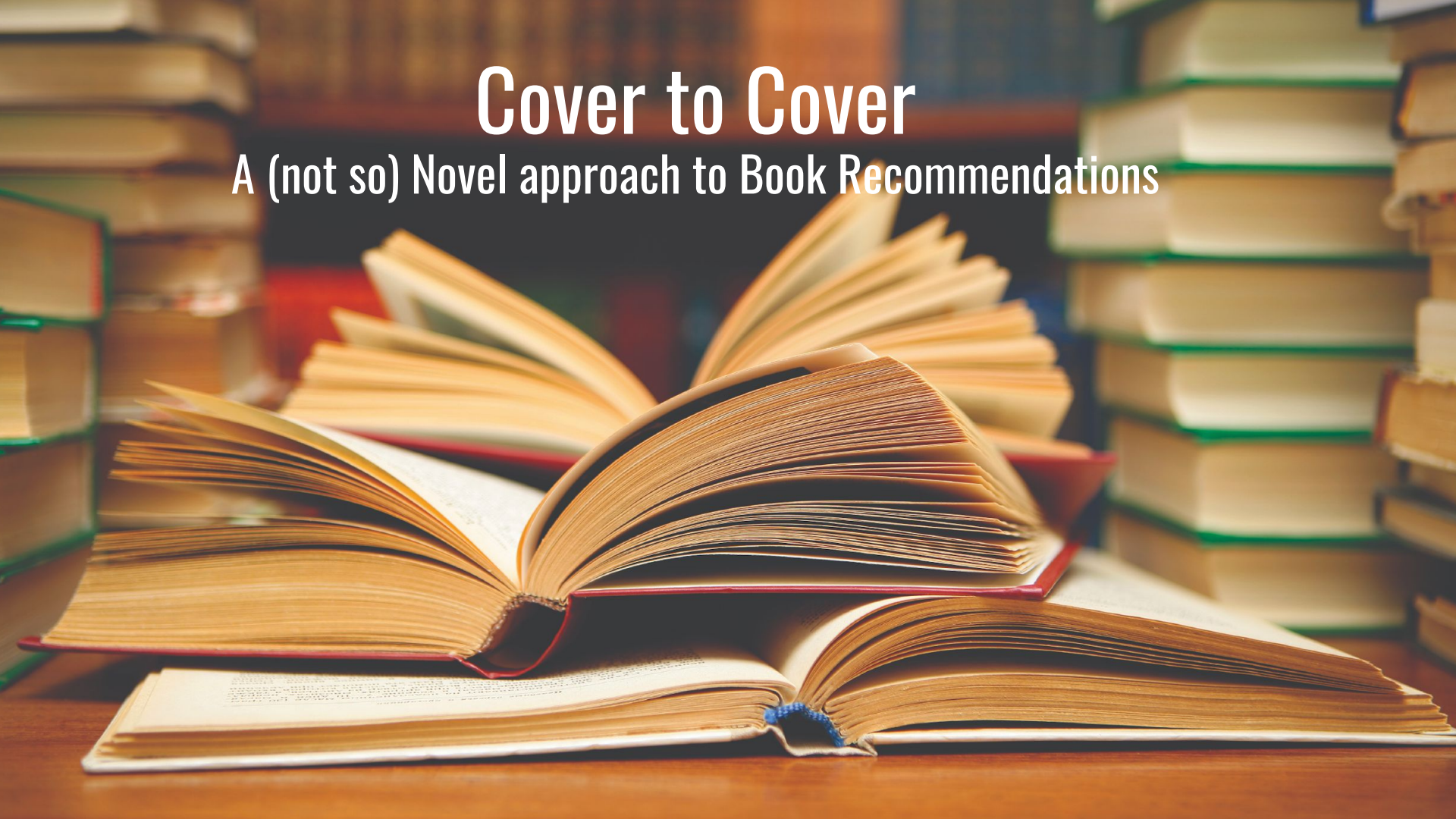

Cover to Cover
A (not so) Novel approach to Book Recommendations

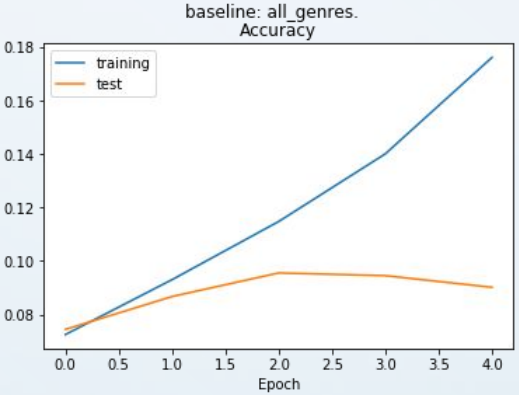
The Data

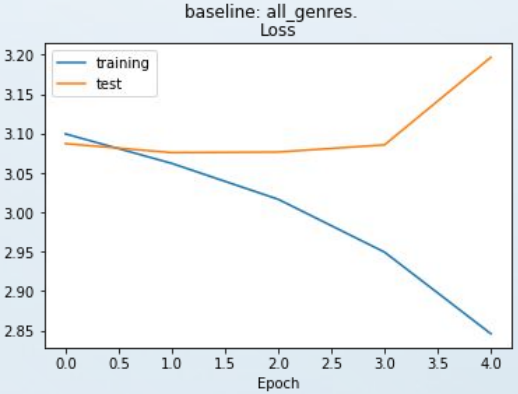
Sampled from Amazon book-covers dataset (40,000 Images scraped from Amazon.com across 23 genre classes, 1700 per class)

Original Dataset contains 200,000+ images from 32 classes

Exploring Genre classification with Neural Networks

Exploring the data

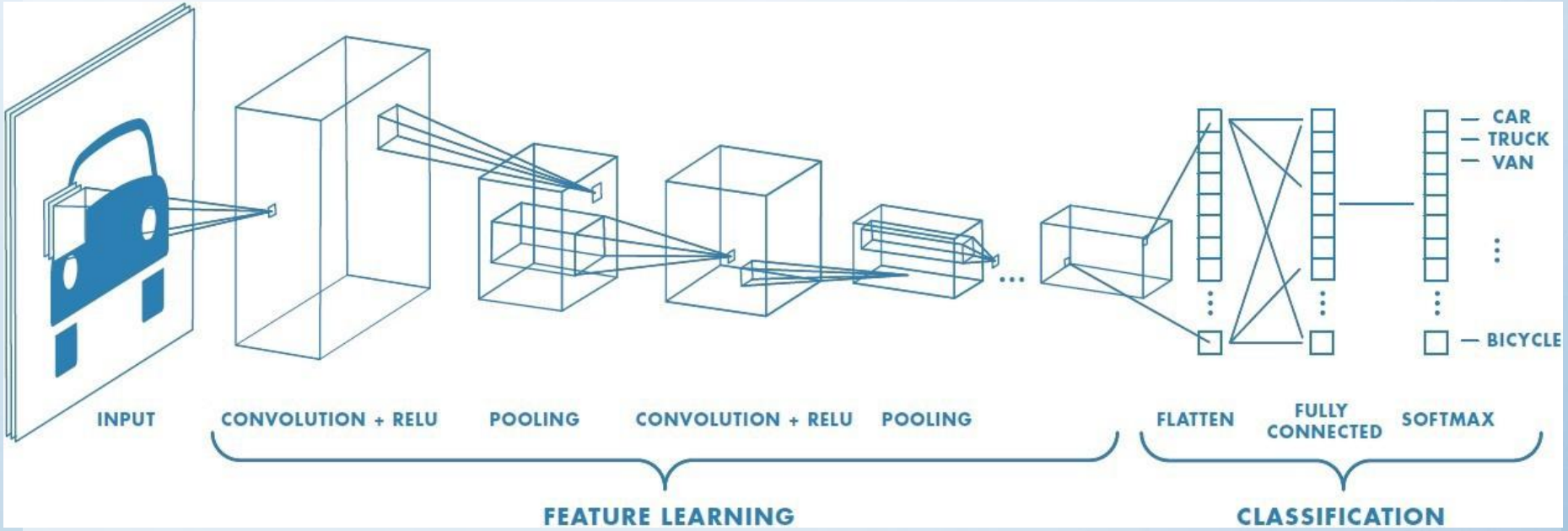




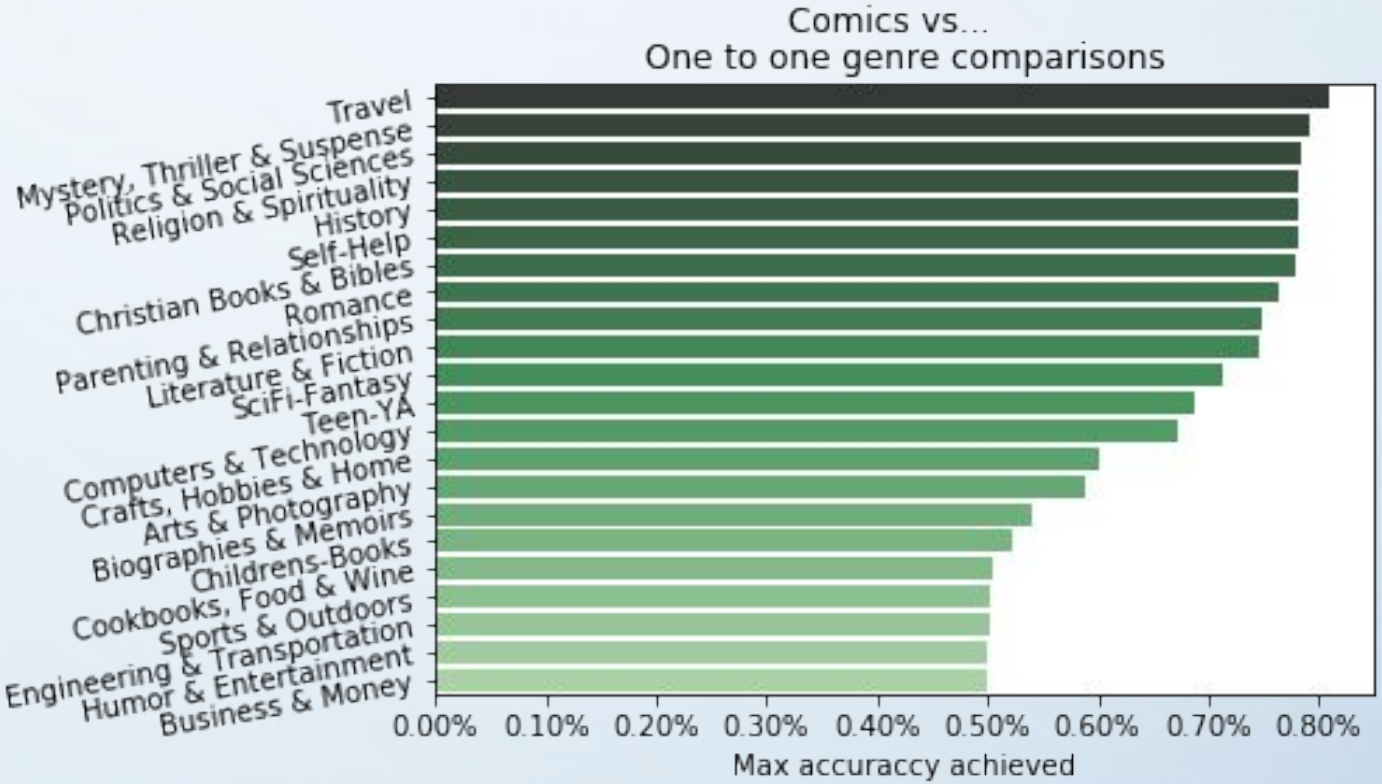
Initial attempts to Classify every genre results in very low test accuracy <10%

Transfer Learning Results from Original Research paper achieve max accuracy of 24%

Convolutional Neural Networks



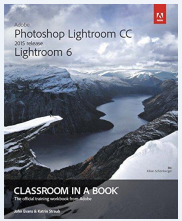
One to one genre classification with CNNs

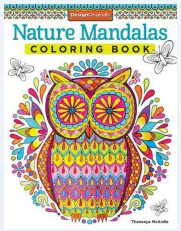


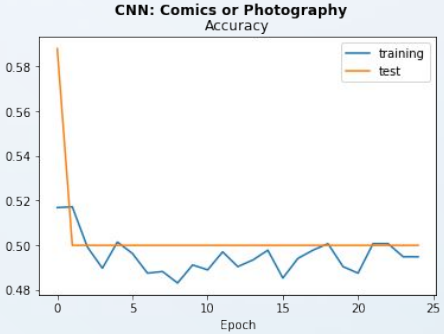
*Results varied from run to run

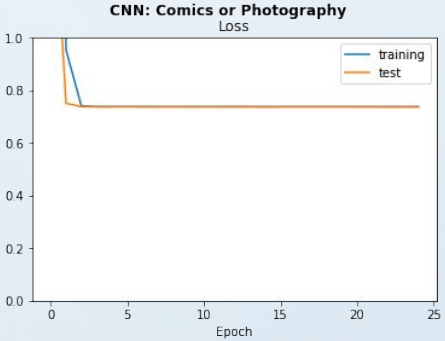
Poor performing categories had too much variance and/or poor sampling resulting of books belonging to multiple genres

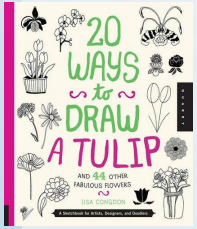
Photography

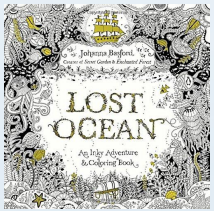




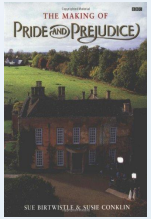


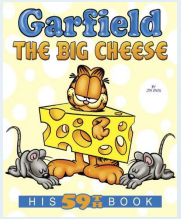


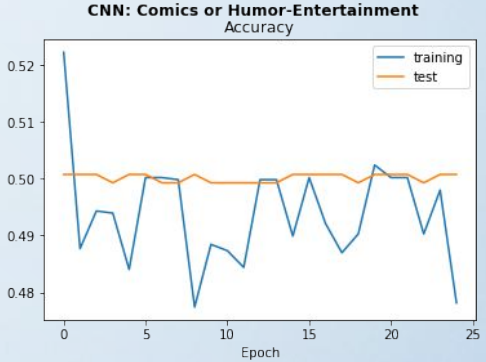


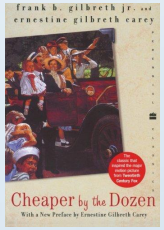


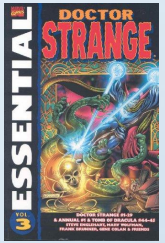
Humor & Entertainment

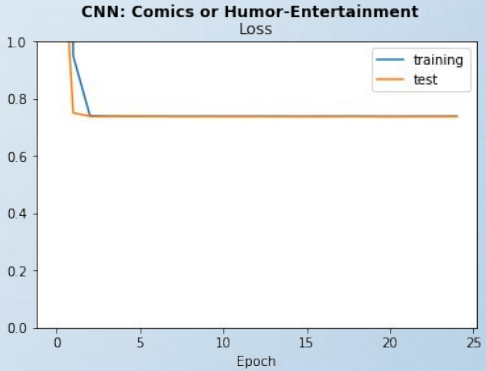




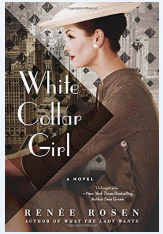


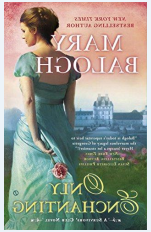


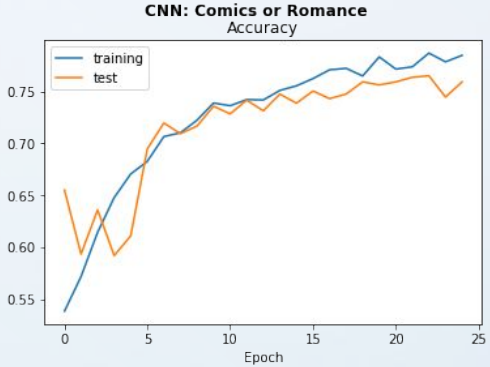


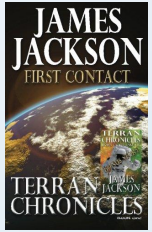


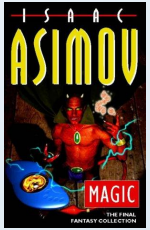
Categories with learnable features

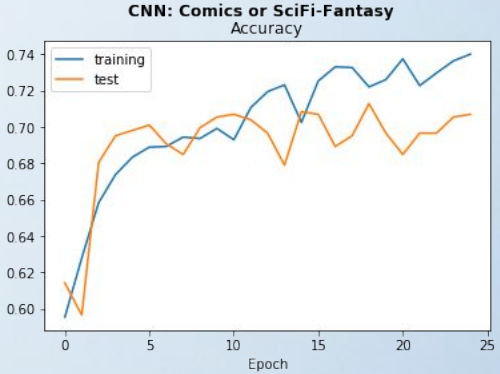


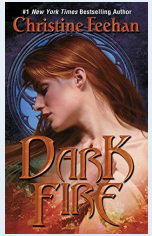


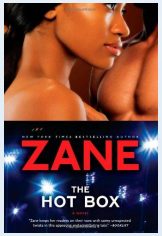


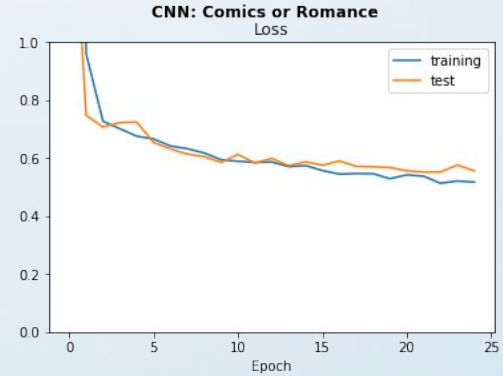


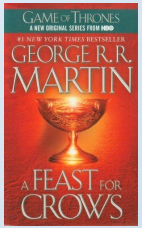


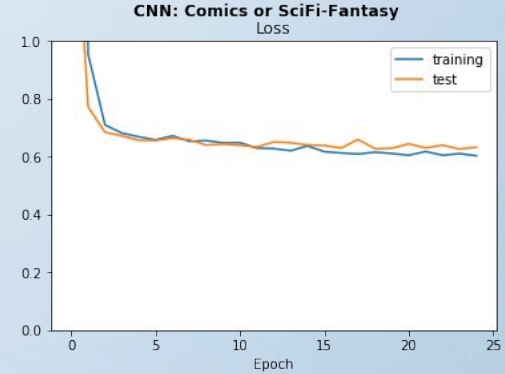












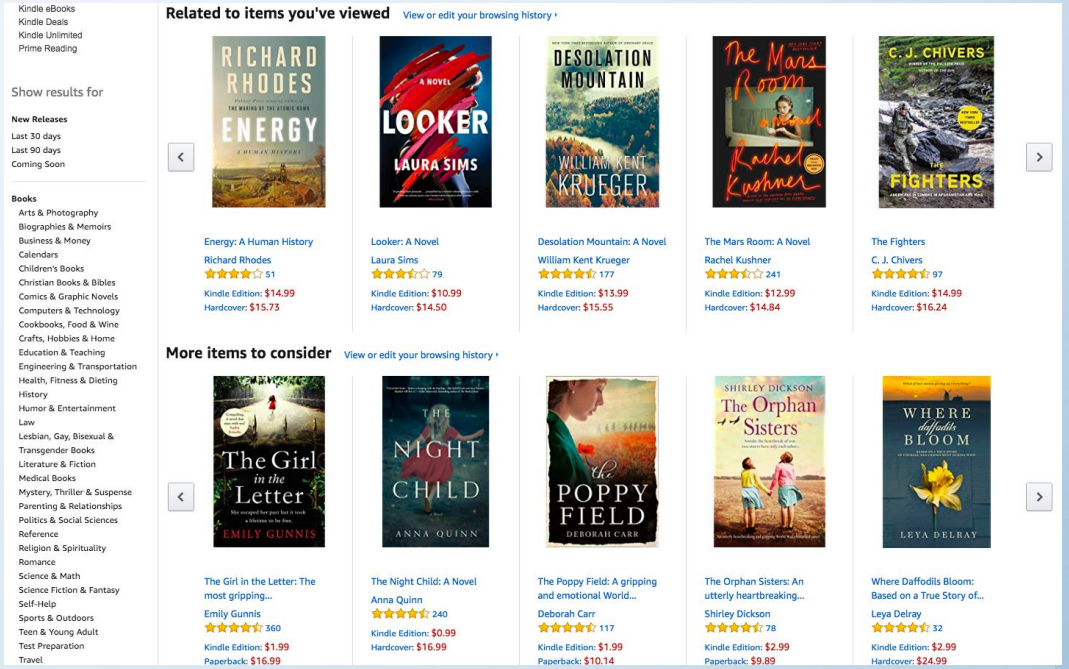
Transfer Learning Results of the Original research paper

Using pre-trained Alexnet and Lee-net architectures, more complex design patterns (including **color**) of a particular genre are proven to be learnable. [1]

Examples correctly classified by Alex-net [2]

“I just want a good book…let’s check amazon”

[3]



"You cannot invent an algorithm that is as good at recommending books as a good bookseller, that's the secret weapon of the bookstore — is that no algorithm will ever understand readers the way that other readers can understand readers."

John Green, author of The Fault in Our Stars [4]

A (not so) novel approach:
Feature Extraction with Similarity Models

Filter level 1: Comparing color histograms with Chi-squared

Filter Level 2: Measuring image similarity with Structural Similarity Index (SSIM)

Comparing Histograms

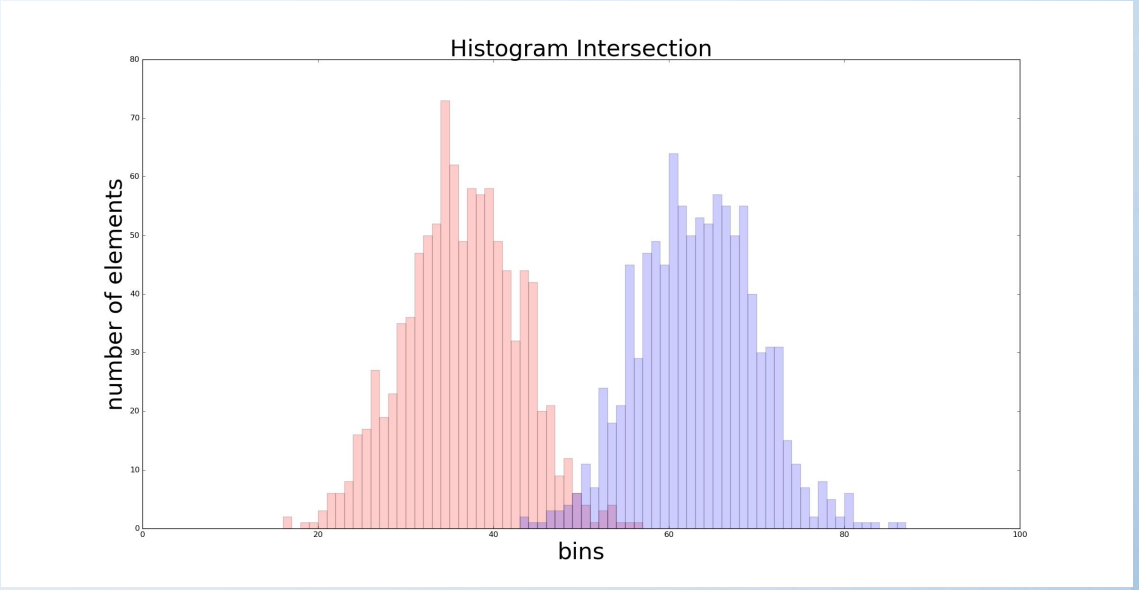
OpenCv contains seval built-in methods for comparing histogram.

Correlation

Intersection

Chi-Square

Bhattacharyya Coef



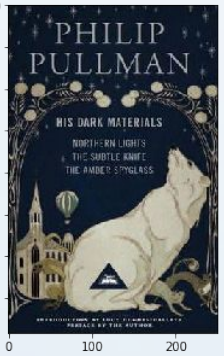
Structural Similarity Index

The Structural Similarity (SSIM) Index is based on the computation of three terms, namely the luminance term, l the contrast term, c and the structural term, s. The overall index is a multiplicative combination of the three terms. SSIM is between 1 and -1. A SSIM of 1 is a perfect match , 0 indicated no similarity and -1 indicate an inverted image. [5]

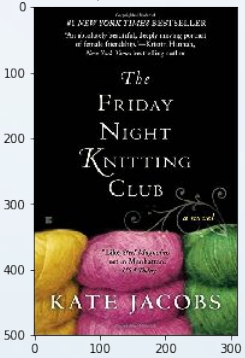
$$SSIM(x,y)=[l(x,y)]\alpha \cdot [c(x,y)]\beta \cdot [s(x,y)]\gamma$$

Let's get some Recommendations!

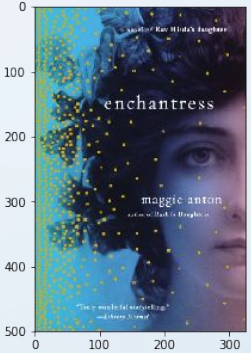
Result 1:



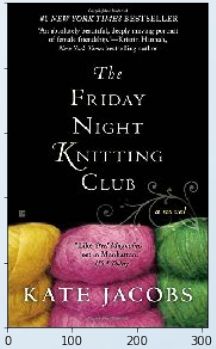
SSIM 0.42, intersect coef 8.86



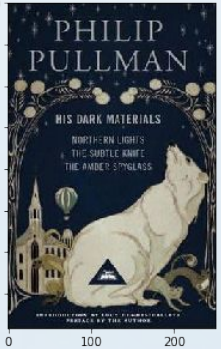
Result 4:



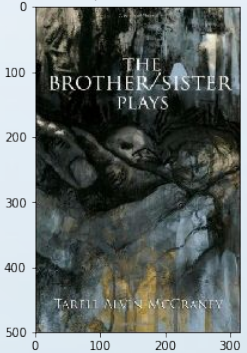
SSIM 0.42, intersect coef 18.44



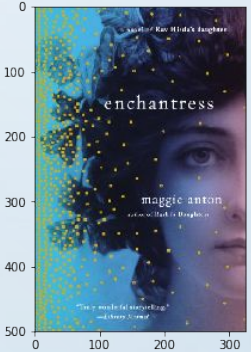
Result 8:



SSIM 0.30, intersect coef 16.45



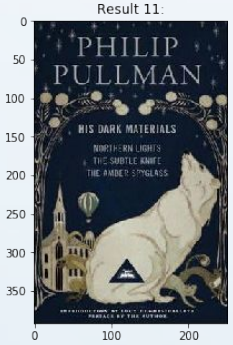
Result 19:

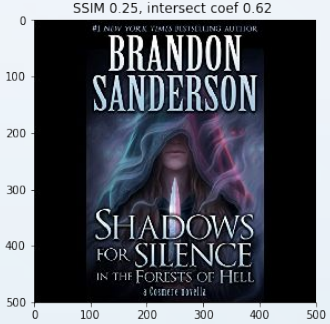


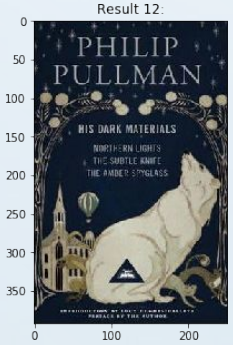
SSIM 0.30, intersect coef 34.19

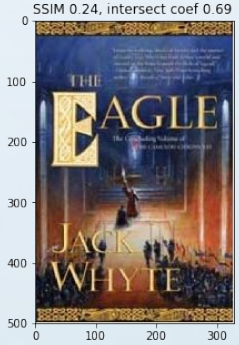
“Modeling Discovery”

*Results using alternate histogram comparison method

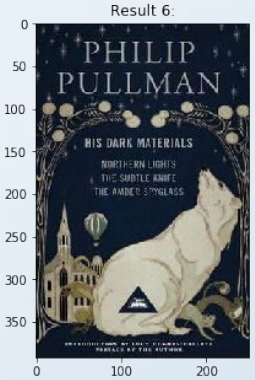


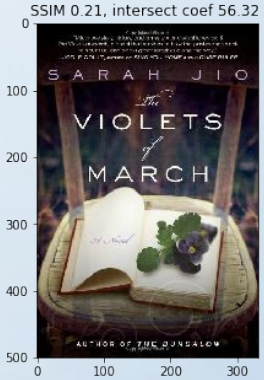


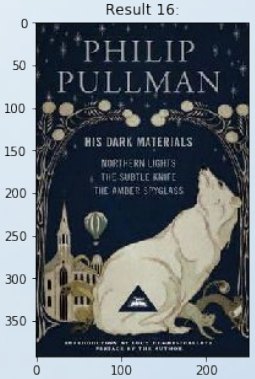


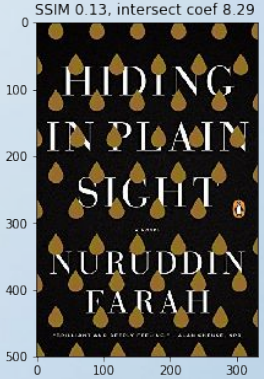


*Results querying alternate genre









Next Steps

Add Feedback weighting functionality to model

Scrape more data from Amazon to improve results

Create Front-end for website/IOS app

References

[1] “Judging a Book by its Cover” Iwana et al (2017) https://arxiv.org/pdf/1610.09204.pdf

[2] “Judging a Book by its Cover”, Fig. 4, Iwana et al (2017)

[3] Xkcd “Decision Paralysis” https://xkcd.com/1801/

[4] “The Rise Of Independent Booksellers In The Time Of Amazon” https://www.buzzfeed.com/michelefilgate/the-rise-of-the-independent-bookseller-in-the-time-of-amazon

[5] “Structural Similarity” Wikipedia https://en.wikipedia.org/wiki/Structural_similarity

The End

Questions?