Feature Film Features
CSCI 5622 Group L

Grant Baker **John Dinkel** **Derek Gorthy** **Jeffrey Maierhofer** **Luke Meszar**

University of Colorado Boulder

Abstract

Feature film advertisers often produce striking posters to help promote their films. These posters vary heavily in stylistic choice, looking extremely different from each other. Others fall into common tropes, ending up with many features in common. We posited that similar posters were likely to belong to the same genre. We sought to design several algorithms to aid in classifying movie genre based on poster design. We found that posters were difficult to fully classify using only visual data, as even humans struggle with this task. We then created a secondary network that brought in additional meta-data about the movie to help improve accuracy.

1 Introduction

In the film industry, posters created for movies are an important form of advertising that can make or break a film's performance in the box office. As a result, advertisers spend a significant amount of time and money constructing posters that are able to convey themes of the movie in order to appeal to their target audience. We believe that creating an algorithm that is able to take in data from a movie poster and use it to predict the movie's genre could be very helpful to advertisers and producers. For example, consider an inexperienced designer that needs to be able to design an iconic poster. They would likely bring in a lot of their own bias when designing it. However, if they were able to use an algorithm that could double check their work, they would be more likely to eliminate their bias and develop an image more appealing to the target audience.

In this paper, we describe several candidate algorithms for attempting to solve this problem. For a simple baseline, we developed an algorithm that would create 3 histograms of the posters' color distributions, and would then use these as features to perform a logistic regression on the genres. Next, we developed and trained a Convolutional Neural Network to take in the poster to predict the genre. Next, we decided to employ transfer learning from the Inception Neural Net [3] to improve results. This network has trained on millions of images, and has developed a robust feature extractor in its early layers. We used these feature extractors, and trained on the features in a CNN. Finally, we determined that most posters include information about actors and directors on the poster. We included this data in our network and sought to train on this.

CSCI 5622 Project Group L. Fall 2017, University of Colorado Boulder, CO, USA.

2 The Data

To populate our dataset, we found a spreadsheet listing approximately $40,000$ movies with links to their IMDB pages. Using Python's IMDbPY package, we developed a program that would take these links and download the corresponding poster thumbnail from IMDb as well as meta-data such as the feature film's director, top five actors, and the film's associated genres. We were able to generate 38,497 movie posters of size 182x268 pixels. We also generated a spreadsheet of the meta-data. There were a total of 29 different genres, and we encoded these in a one-hot vector. The data can be found in the project GitHub directory.

2.1 F-Score Metric

Our goal was to classify each film into its genres from 29 possibilities. Since our dataset contained movies classified into up to 4 different genres, we needed to perform sparse multi-label classification. Because the labelling was sparse, any algorithm would be able to generate a relatively high accuracy by simply guessing that the movie didn't have a single genre, a vector of all zeros. This created a local minimum that would be difficult for any classifier to train out of. Since a traditional accuracy metric would not be useful, we turned toward other methods. After a little research, we decided that the F-Score[1] metric would be most insightful. The F-Score metric is calculated as follows:

$$F = \frac{2TP}{2TP + FN + FP} \tag{1}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. This is equivalent to the weighted average of precision and recall. This metric is optimal for a sparse positive count, as it ignores true negatives. This allowed us to train our network without predicting all zeroes, and is a much better indicator of success than accuracy for a sparse problem of this nature.

3 Human Case Study

To get a basic indication of the difficulty of the problem, we first decided to do some research on how well humans can solve the problem. We asked a few family members and friends to classify ten posters by selecting three genres for each poster from our list of 29. The 12 participants averaged an f-score of 0.32. Although this is a very small sample size, the goal was only to gain a general intuition of how well humans do at the problem, rather than obtain statistically accurate results. From the F-Score, we concluded that humans can perform somewhat well while guessing movie genres, and can usually guess one genre accurately, but will only occasionally do much better. However, we also recognize that humans might have somewhat of an advantage with the problem, as they can recognize actors' faces and read the poster if it has words, in addition to simply looking at the picture, and this can influence their decisions. For example, one of our test posters had a picture of Arnold Schwarzenegger on the front, and since he is a prominent action actor, every one of our testers was able to successfully guess that the move was indeed classified as an 'action' movie.

4 Histogram of Colors

Our initial naive solution to the problem was to develop a logistic regression algorithm for prediction. We noticed that many horror movie posters feature black and red heavily, while comedies and animated movies tend to use a brighter color scheme. Because of this, we hypothesized that many movie genres tended to follow specific color schemes. We guessed that if we created a histogram of the color distributions, it would serve as a good feature to train on. We wrote a program to take in a poster image and separate pixels into their RGB values and group the densities of each color into 8 integer ranges. From this, we created histograms to train our model on. See an example of one of these histograms visualized in Figure 1.

We implemented the regression algorithm using a Keras sequential operator. It acted as a single layered neural network with 96 nodes and predicted the probability of each genre. Because we were performing multi-label classification, most loss functions were not appropriate. We did, however, find that the binary cross entropy loss function performed the task moderately well. Through the training,

we found that we were more than able to fully fit sample training set, but no amount of regularization would allow us to learn a relationship between genre and color. This seems to imply that there is not a strong enough relationship between color and genre to accurately determine the genre beyond simply guessing.

It wasn't entirely shocking that the histogram of colors wasn't an effective model given the large number of posters that do not conform to a color scheme, but it still was a bit disappointing. Clearly, a better method was needed.

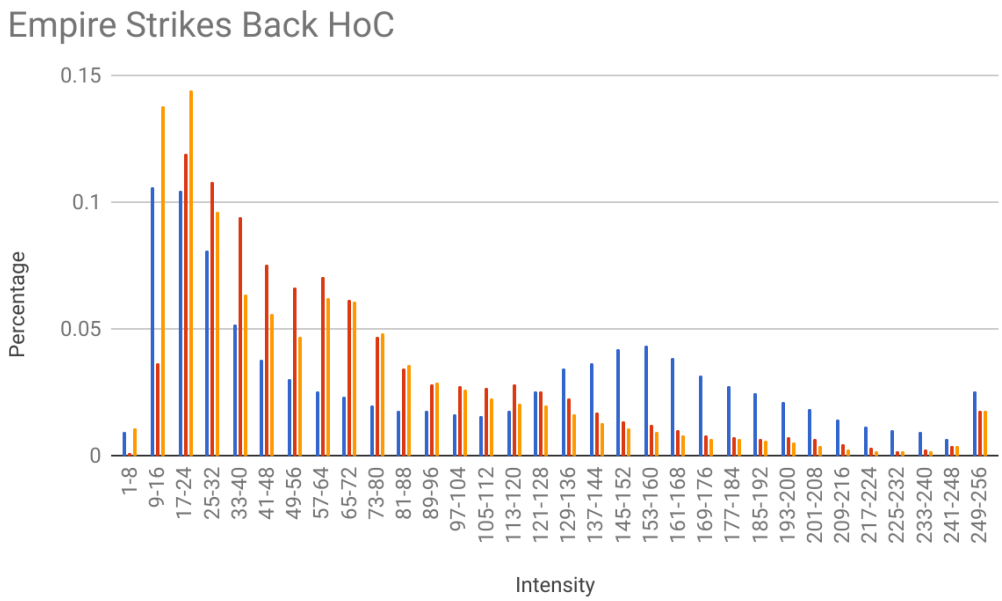


Figure 1: A histogram of colors for the poster of *Star Wars: The Empire Strikes Back*

5 Sparsemax Classification

The classification in the CNN model presents a difficult problem. Movies belong to up to 4 genres, chosen from a possible 29, so the output is a sparse multi-label classification. In order to help with this, we researched a classifier called Sparsemax, which is modelled after softmax classification, but is better able to output sparse probability distributions [2].

6 Convolutional Neural Network

Our next approach toward solving the feature film poster problem was to develop a Convolutional Neural Network. A CNN is able to take in an entire image as an input, and learn to find key features in it, so there was no need to generate a histogram or otherwise transform the input data. We thought a CNN might be useful, because the architecture of a CNN emphasizes local features, like finding a face in an image, and is invariant to translation, and so might be able to find more useful features. This model was trained entirely on our data set, and was able to learn to identify genres with some reliability.

The final architecture for our CNN is shown in Figure 2.

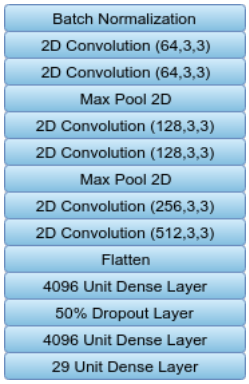


Figure 2: The architecture for the Convolutional Neural Network

After being trained for 20 epochs, this model achieve an F-score of 0.27.

7 Inception

After creating our initial CNN, we did some research about transfer learning using a pretrained InceptionV3 model. A common technique in Computer Vision is to use the first layers from state of the art CNNs to extract features. This is useful, as most CNNs look for very similar features in an image. The later layers then learn how to interpret these images. As a result, the state of the art networks have the leverage of millions of training images, as opposed to our small training set. When we implemented this feature extractor into our network, and trained a net following this, we were able to dramatically increase the ability of our network, pushing it past even the ability of humans to identify poster genre. This network achieved a F-score of 0.4 after 20 epochs.

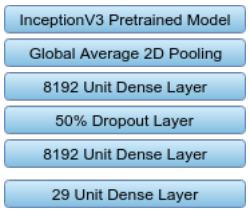
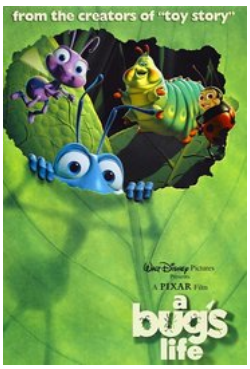


Figure 3: The architecture for the InceptionV3 Pretrained Convolutional Neural Network



(a) **Predicted**: Drama, Animation, Comedy
Actual: Adventure, Animation, Comedy

(b) **Predicted**: Drama, Comedy, Horror
Actual: Action, Fantasy, Adventure

Figure 4: Example Predictions from Inception model. One that performed well (A Bug's Life) and one that did poorly (Star Wars)

8 Meta-data

One of the things we learned from our human case study is that humans gain an advantage because they can read additional information such as actors and director off of a poster. To reproduce this advantage, we decided to supplement the images that we initially used in our analysis with the director and actor meta-data we downloaded from IMDb. The model we developed is an extension of the above neural net models, but with the added component of the relevant meta-data merged with the poster images.

8.1 Metadata Preprocessing

To reduce the dimensionality of the director data, we determined that a director with only one movie in the dataset would not help to predict the genre of future movies, as the knowledge cannot be further applied. All of these directors were assigned an ID of 0.The remaining directors were assigned a unique director ID. This gave the neural net numerical data to work with instead of strings.

8.2 Architecture

The architecture for the meta-data model is below. We merged the embedded director information with the output of the InceptionV3 pre-trained model. These were then fed through two of densely connected layers and a dropout layer. After training the model for 100 epochs, it only achieved an F-score of 0.41. From this, we can infer that adding meta-data did not significantly help performance. This is possibly because of the size of the dataset. If more movies with the same directors already in the metadata were included, perhaps the score would increase even higher.

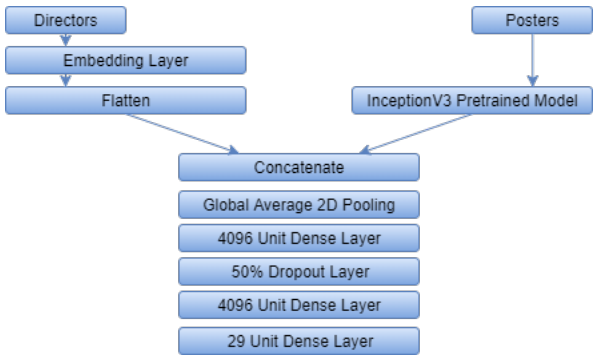


Figure 5: The architecture for the Convolutional Neural Network with additional meta-data

9 Results Analysis

In this paper we explored several different methods of identifying the genre of a movie poster. Our histogram of colors approach was unable to learn how to identify the genre of the movie, and in fact, learned to only predict no genres for any movie. This was a very poor preliminary result, and resulted in an f-score of zero. When we trained a CNN from scratch, we were able to achieve an F-Score of 0.27. In our transfer learning module, we achieve 0.40. Finally, when using meta-data, we were able to learn a network that achieved an F-Score of 0.41 after 100 epochs.

Guessing two genres correct and a third wrong should result in an F-Score of about 0.6. So on average, our best methods were able to guess one genre correct reliably, but a second genre correct about half the time. While this is slightly better than human performance, it is still not a fool-proof result. Clearly much more work must be done for a program of this nature to be of general use.

10 Error Analysis

In general, this was a very difficult problem to complete accurately, as classification of genre is an inherently subjective task anyway. In addition, there were 29 total genres, and a multi-label

classification of up to four categories over these 29 yields over half a million different combinations. This large number over a dataset of only 40,000 examples makes it very hard to learn, even if the problem were entirely learn-able with proper resources. However, even humans with significantly more information about the problem due to context and increased metadata are generally unable to learn to correctly classify, so making any headway over that at all is somewhat of a victory.

11 Difficulties and Challenges

- **Sparse classification**: One of the first major difficulties we encountered was with the nature of genre classification in the dataset. Since each movie is being classified into up to 4 of 29 total generes, most of the classifications are zero. To get around the issues caused by sparseness, we implemented the sparsemax activation and loss function and used the F-Score metric.
- **Unbalanced dataset**: Another issue we encountered is the sheer number of feature films in our dataset classified as "drama." Over half of the dataset is drama, so it tended to be guessed, regardless of the poster's appearance. This is not what we hoped our algorithm to learn. To get around this, we first considered trying to reduce the number of classifications in our dataset by combining certain genres into one. However, this is difficult to do, as classifications are somewhat subjective, and it still doesn't really get around the error. We also considered rewarding the program less for correctly guessing a more popular genre, but this is also an imperfect solution, as it is not actually training the neural net to recognize what makes a movie truly a "drama." This is a problem that we are still struggling with.
- **Lots of atypical movies in data**: A more abstract problem that we are faced with is the wide diversity of movies in our data set. The data contains movies from many different countries, cultures, and stylistic backgrounds. Much of this diversity is also reflected in the posters. Examples could include different architecture styles, different countries having different symbolic meanings for different objects, and in some cases, more stylistically abstract poster art. These inconsistencies hurt the program's ability to learn. A possible solution might be to trim the dataset of some of this diversity by limiting it solely to American-made movies for example. This might help the dataset become more cohesive.

12 Future Work

We do have a few ideas to improve on our model. First, we think it is important to improve on the metadata idea, as this could add lots of additional context to the posters for a net to learn from. We could add additional metadata such as more actors, producers, music composers, and production companies. This would likely improve the results. Another idea is to put more research into solving the sparse multi-classification problem. The sparsemax loss function was a great step in the right direction, but improvement on this model could also lead to additional performance. Assigning a higher weight to classics in a particular genre may also be helpful. Lastly, modifying the dataset could help. Limiting the diversity of the data to only include American films for example could unify the data while still providing a useful result to American advertisers. Otherwise, generally expanding the dataset could be extremely useful.

References

[1] Renuka Joshi. Accuracy, precision, recall & f1 score: Interpretation of performance measures. In *Exsilio Blog*, 2016.

[2] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623, 2016.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.