CS229 Project Milestone Report

Rajarshi Roy
Electrical Engineering
Stanford University
rroy@stanford.edu

Ethan Chan
Computer Science
Stanford University
ethancys@stanford.edu

John Lee
Electrical Engineering
Stanford University
johnwlee@stanford.edu

Abstract

With the rise of movie recommendation systems popularized by Netflix that incorporates reviews, subtitles and movie metadata, research in classification of movies by their frames and colors is lacking. In this paper, we set out to investigate color patterns among movies through unsupervised clustering algorithms as well as using supervised learning algorithms to classify movies according to their genres. We decided to use a color timeline vector, a very intuitive representation that reflects a lot of information about the movie on its colors and time of colors, as our feature in our algorithms.

Current Progress: We have ran k-means algorithm and found that some movies have clustered very well, and we have also implemented a horror movie vs the rest SVM classifier and tuned it to get some initial results on our data.

1 Introduction

We set out to explore how movies genres are related to the color characteristics of a film. The inspiration for this is that as humans, we can intuitively tell what genre a film is just by looking at its color hues with warm red tones for romances, and desaturated colors for post-apocalyptic films to name a few. We want to investigate this intuition in a thorough and scientific way to incorporate machine learning concepts on color characteristics to predict the genre of films, and perhaps find relations between films we did not originally thought was there.

2 Review of Previous Work

Since current methods of clustering movies are based on data tags such as genre, director and year of production, we also want to investigate using unsupervised learning methods to cluster movies based on the vectors of color data. For example, we hope to discover and explore how color is a common motif among movies with similar attributes such as the mood of the film, or animated films, or classics, country of the film.

There has been a paper [1] that investigated investigated color characterization of movies for mood analysis. They used movie color palette, similar to our definition of color vector but discretizing it to 12 colors instead of the whole RGB spectrum and mood dynamics histogram as features for classification and found that the movie dynamics, the way a movie transitions scenes, is very discriminative of the mood of the movie. There are limitations to the findings, as firstly, they only used 15 movies, and they were analyzing how a mood varies within a film. Our research paper aims to find out how such color characteristics distinguish different films, and probably could be used for accurate genre classification in the future.

3 Methodology

3.1 Data Collection

For initial workflow setup and datapoints we have used the ffmpeg library on a set of about 150 movies of various genre that we collectively owned. Thus our second methodology involves a tool that we built to capture keyframes from Netflix. Pressing the right arrow key during netflix movie playback jumps to keyframes at 10 second intervals and displays a small preview of the frame. Our tool generates rightkeypresses, tracks the movement of the preview window and averages the color in the preview window to capture the color timeline of the movie. Keypress generation and screenshot capture is done using the Java robot class. Since the preview window is essentially a downscaled version of the fullframe, each pixel in the preview is already a mean of a larger set of pixels. Thus, we are focussing on the mean pixel color of a frame.

3.2 Feature Extraction

From the 229 movies we have collected, we first sampled 1 frame every 10 seconds using the FFmpeg library. For each frame we sampled, we obtained the average rgb value for all the pixels in the frame and for each movie we obtained a feature vector that looks like this:

$$\begin{bmatrix} r_1 \\ g_1 \\ b_1 \\ r_i \\ g_i \\ b_i \\ \dots \\ \dots \\ \dots \\ r_n \\ g_n \\ b_n \end{bmatrix} \tag{1}$$

Where i is the frame number, n is the total number of frames in the movie, and $0 \geq r, g, b \geq 255$. As different movies had different lengths which were all greater than 700, we normalized the lengths by re-sampling 200 interpolated rgb points from the vectors to obtain vectors of length n = 600.

3.3 Feature Visualization

To assist us in visualizing the data that we've obtained, we plotted the colors as a function of time (or frame number). For example, Figure 1 below is a visualization of the features extracted from "The Lord of the Rings: The Fellowship of The Ring". This way we can visually find patterns and inspect if the results of our algorithms are working very quickly.

4 Results

4.1 Unsupervised Learning - K Means Clustering

To begin exploring the patterns and clusters that can be established by using unsupervised learning methods, we ran the data that we had obtained through a k-means clustering algorithm. Some of the resulting clusters and the movie's corresponding visualizations are presented in the figures section of this report.

We experimented with using different values of k (i.e. the number of clusters) to determine if the k-means algorithm could pick out structure within the feature vectors that we had obtained.Based on the preliminary clusters that we had obtained, we notice that the unsupervised training method successfully clusters movies based on the colors present in the movie. In certain cases, we also observe that certain clusters had more movies of a certain genre than in others. For example, figure

6 has 4 of the 12 animated films that we have in our database, which seems to suggest that certain genres (such as animation) have rather distinctive color schemes.

Based on the the clustering results that we obtained, (some of which are attached in the figures section), we observe that an unsupervised learning algorithm (such as the k-means algorithm) can successfully group movies that have similar color schemes. Specifically, the movies in the cluster depicted in figure 4 have much darker scenes, and consists of movies like the "The Dark Knight" and "Harry Potter and the Deathly Hallows" (both parts).

4.2 Supervised Learning - Support Vector Machines

Having seen some patterns emerge through our K means clustering, we decided to build an SVM classifier that identifies genres from the color vectors. To simplify things, we decided to first use an SVM to build a horror movie classifier on our data-set. We started the supervised learning section by training a SVM on 229 films split into two classes - horror and non-horror. Based on the data that we had gathered up to the writing of the project milestone report, the 229 films were split pretty evenly into those two classes, with 121 horror films and 108 non-horror films.

To evaluate the performance of the SVM using different kernels, we started out by training the SVM on a random permutation of 90% of our entire data-set, and then determined the generalization error based on the remaining 10% of the data that was not used in the training of the model. We repeat this for 100 random permutations, and found that the average generalization error of the SVM using different kernels is as follows:

- Linear: 36.78%
- Polynomial degree 2: 43.78%
- Polynomial degree 3: 45.87%
- Gaussian: 47.13%

Based on the generalization errors above, we notice that as the kernel becomes more "complex", the generalization error increases, which could be because we are over-fitting the data.

We also wanted to determine how the number of training examples that we were training on affected the performance of the SVM, providing some insight on whether we should be spending the time and effort to gather more data. To do this, we trained the SVM using a linear kernel on a random subset of our data ($10\%, 20\%, ..., 90\%$), and tested the generalization error on a separate subset (10%) of our data. Figure 7 shows a plot of the training and generalization error as a function of the size of the training set.

First, we notice that the training error is 0 regardless of the size of the training set we use. This is due to the high dimensionality of our feature vector (length 600), while the largest subset of the data that we train the SVM on only consists of about 200 training examples. Thus we are guaranteed that the data is linearly separable. Second, we notice that the generalization error decreases slightly up till the training set consists of approximately 100 training examples. From that point on, the generalization error does not appear to improve by much despite the increase in the size of the training example.

The second observation seems to indicate that the current features we are using are not ideal for classifying horror vs non-horror using an SVM. In the following section, we shall discuss some of the things that we are exploring to improve the classifier.

5 Appendix: Discussions

5.1 Future Work

5.1.1 Adding New Features

We suspect that our feature of a color timeline might not be sufficient enough to capture the nuances of genres and hence we will try and create a new feature, the mood dynamics vector, a vector to capture how scenes transition, and a difference in their pixels and see if this produces better results.

We also realized that by normalizing our vectors, we might have removed pace from the movies, as we know that pace is a very good indicator of a type of genre, we might reconsider this in rebuilding our next feature.

5.1.2 Collecting More Data

Right now, our data is skewed as the data collection process has been very very tedious. We might need to collect more data as we have 100 horror movies, and 100 of other genres. Given enough time, we should get more data points.

5.1.3 New Supervised Learning Classifiers

We may want to try a multi-layer perceptron to provide classification as we do know that deep learning has worked very well in computer vision problems.

5.1.4 Unsupervised Clustering labelling

We can perhaps find reviews of movies that we have on RottenTomatoes and also subtitles and incorporate that to our movie data so that for each cluster, we can find the top phrases, and that will help us label our clusters better instead of just visual inspection.

5.1.5 Supervised Learning - Consider other class features other than genre classification

Right now we have tried both unsupervised and supervised learning on our movie data. If supervised learning of genre classification does not work out, we might have to realized that perhaps there is no correlation between genre and color directly. We might want to find new ”class” labels for our data. A consideration to this is because artists probably have their own style and flair, and will not like their work boxed up in one genre only.

6 Figures

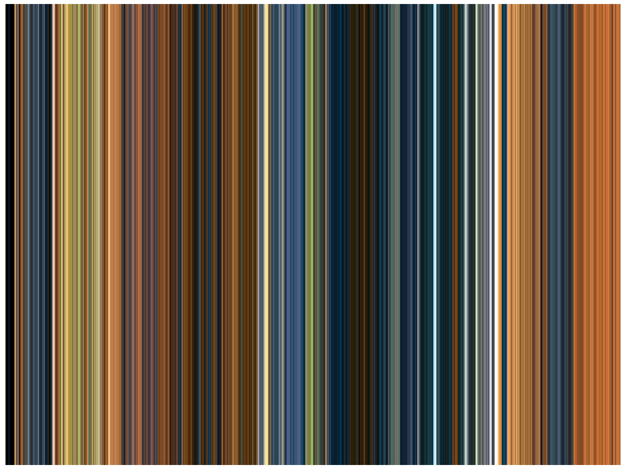

Figure 1: Visualization of ”Lord of The Rings: The Fellowship of The Ring”

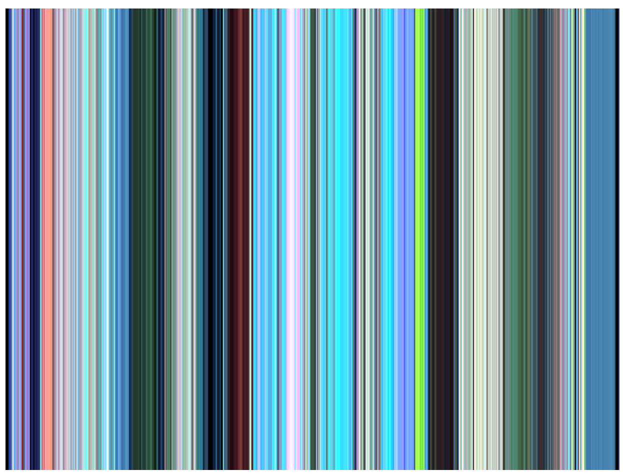

Figure 2: Visualization of ”Finding Nemo”

Figure 3: Visualization of ”Man Of Steel”

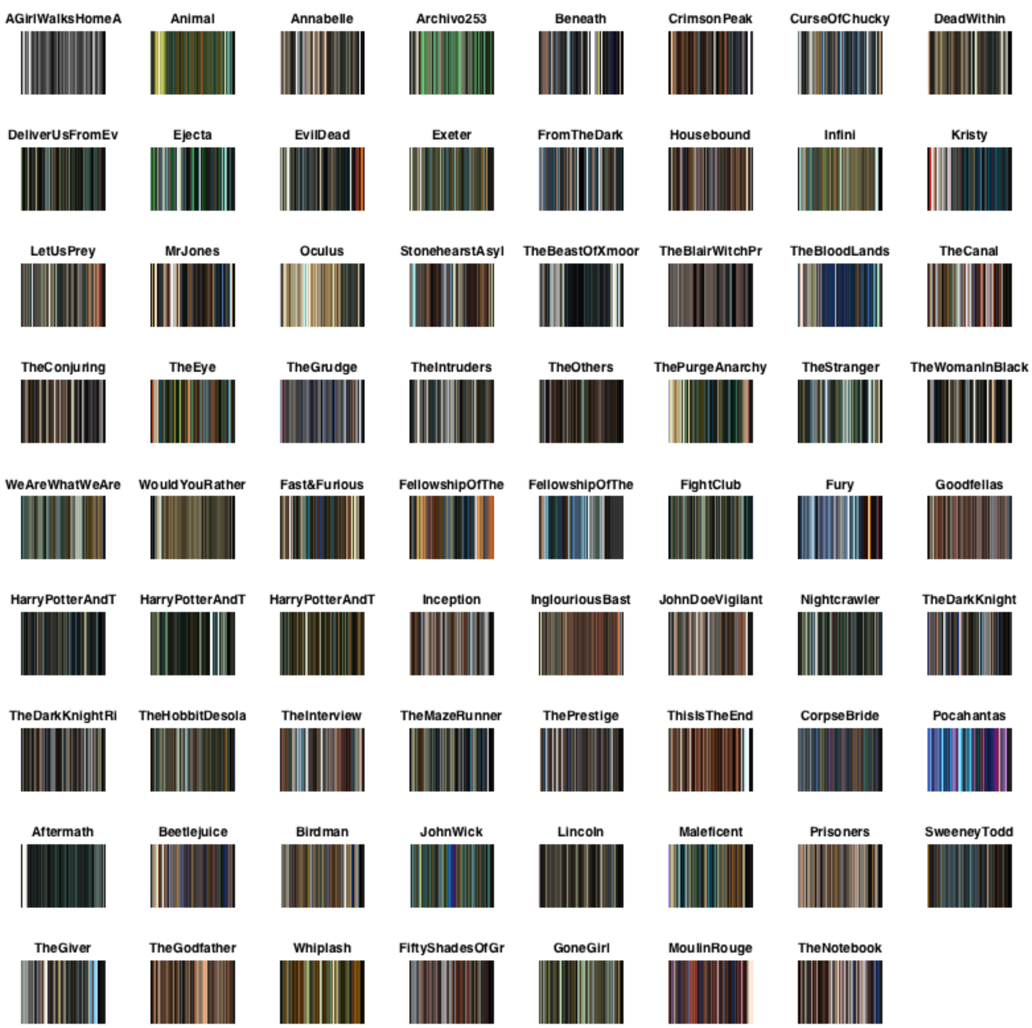

Figure 4: Visualization of K-Means Cluster-1 when k =7

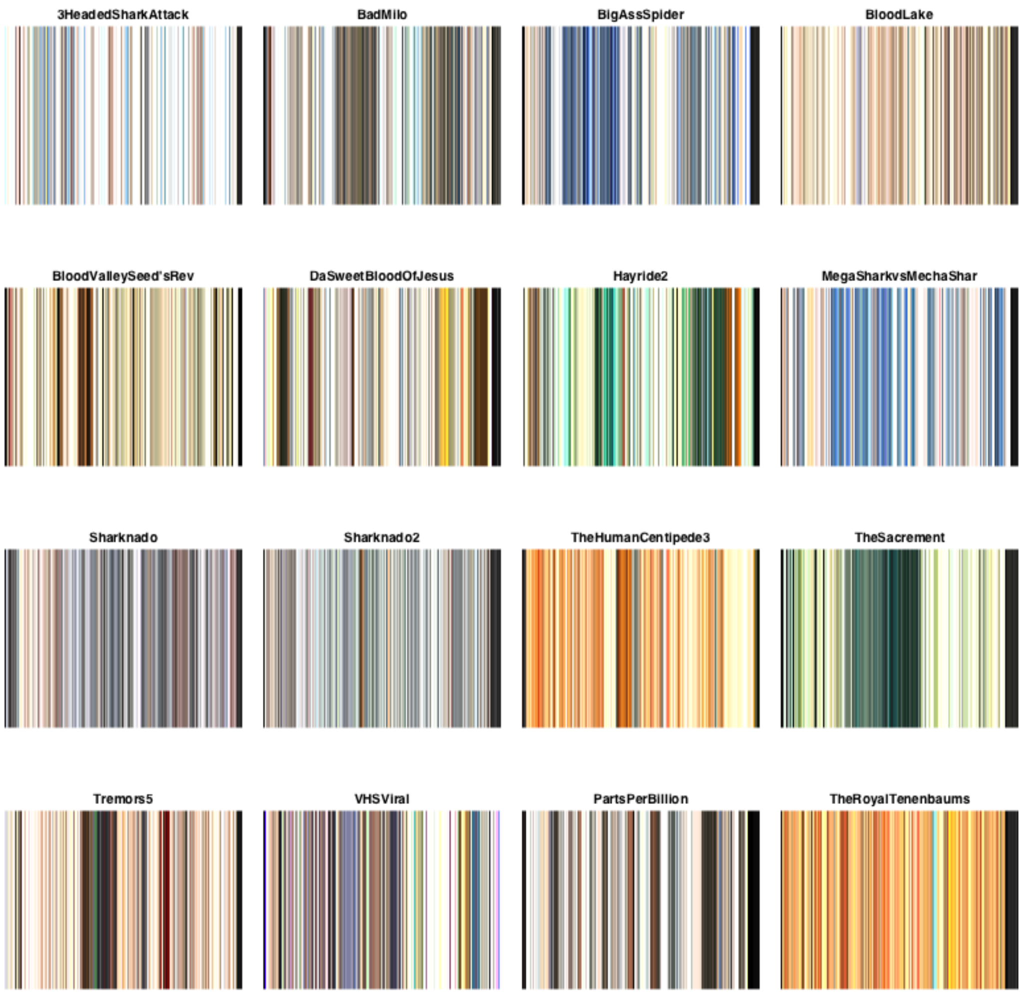

Figure 5: Visualization of K-Means Cluster-2 when k =7

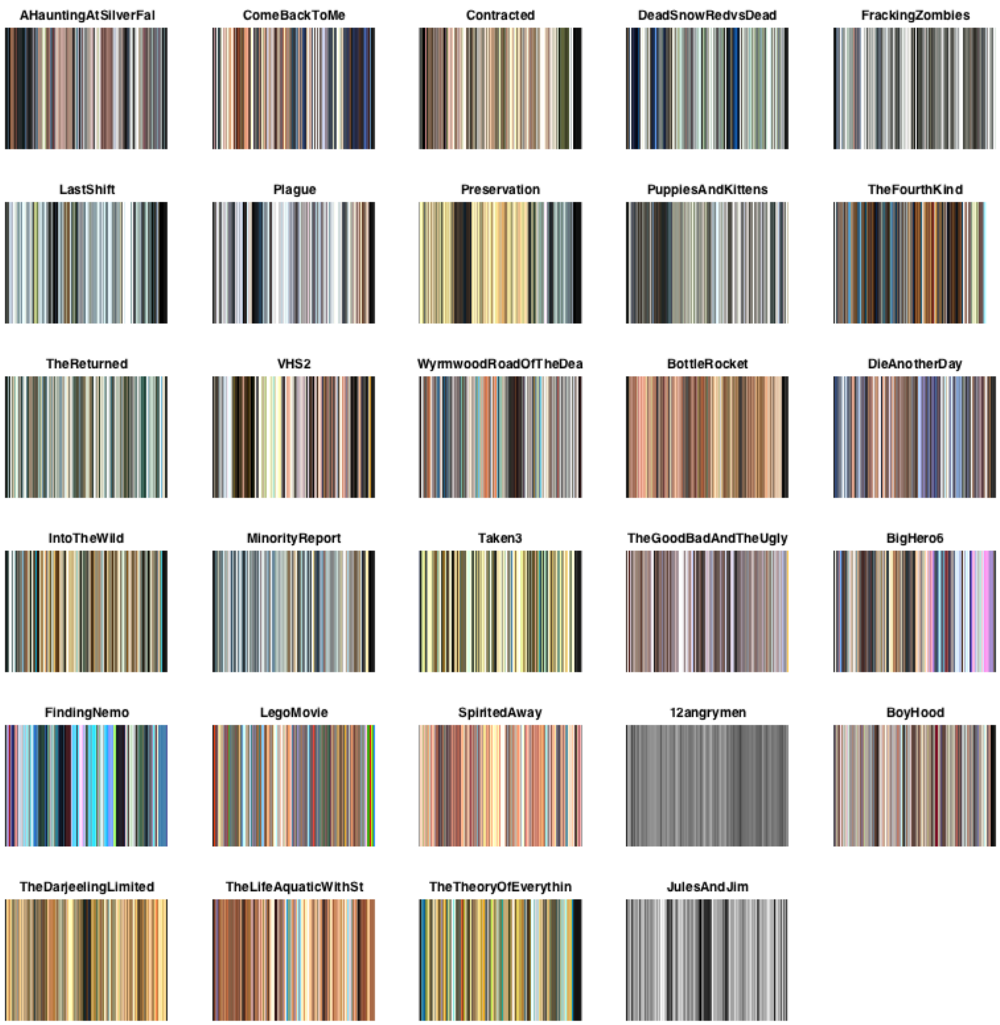

Figure 6: Visualization of K-Means Cluster-6 when k =7

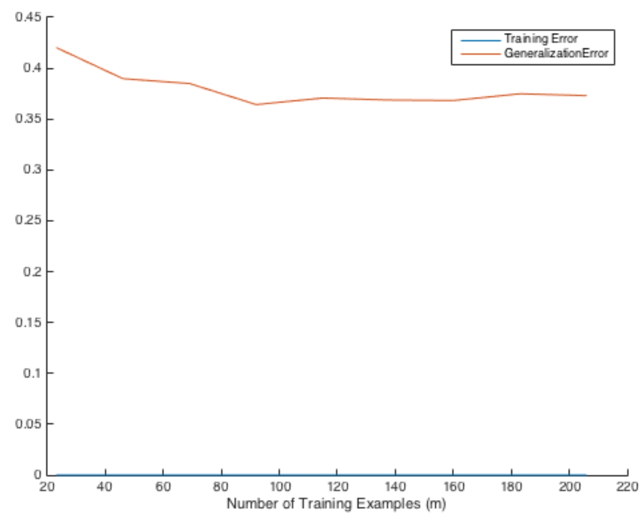

Figure 7: Training & Generalization Error vs Size of Training Set

7 References

References

[1] Cheng-Yu Wei, Nevenka Dimitrova, Shih-Fu Chang. *Color Mood Analysis of Films Based on Syntactic and Psychological Models*. [*http://www.ee.columbia.edu/ln/dvmm/publications/04/ICMEjune04_nelson.pdf*]

[2] ftp://ftp.fu-berlin.de/pub/misc/movies/database/

[3] http://www.smartjava.org/examples/movie-viz/batman.html

[4] https://medium.com/the-outtake/what-does-the-western-look-like-545981d93ae8

[5] http://thecolorsofmotion.com/films

[6] https://publications.theseus.fi/bitstream/handle/10024/2708/Fagerholm_Cecilia.pdf?

[7] https://www.ffmpeg.org/