What Can We Learn From Movie Colors?

Ethan Chan
Computer Science
Stanford University
ethancys@cs.stanford.edu

John Lee
Electrical Engineering
Stanford University
johnwlee@stanford.edu

Rajarshi Roy
Electrical Engineering
Stanford University
rroy@stanford.edu

Abstract

We applied supervised learning methods to classify movies into one of four genres based solely on colors present in the film. Data was obtained from screen captures of the Netflix preview window at 10 second intervals. Our final feature was X = {512-color histogram, avg. color change}. Our genre classes were Y = {Action, Animation, Horror, Romance}. We attempted Multinomial Naive Bayes and 1-v-all SVM to predict the genres of a film based on its color features. We also attempted Unsupervised Learning K-Means Clustering algorithm on our movies to observe patterns across movies.

1. Introduction

It is popular opinion that the mood and the genre of a film is closely related to the color schemes used in the movie. We can intuitively tell what genre a film is just by looking at its color hues; warm red tones for romances, desaturated colors for post-apocalyptic films[3]. We set out to explore how movies genres are related to the color characteristics of a film, and investigated this intuition in a thorough and scientific way by incorporate machine learning concepts to predict the genre of films based on the color characteristics.

2. Related Work

Existing work in the analysis of colors in film can be broadly split into the following three categories. First, work has been done to categorize films into genres based on content [5] such as the scenes present [13], and the movie script [4]. The second group of existing work is the use of computational tools to analyze the color scheme of existing movies, and the subsequent manipulation of the color scheme of films [7] or pictures [12] to match the learnt color scheme. The third category of related work as been to analyze the effect of the color scheme on the mood of film and images[10]. Little work has been done on determining genre from color alone, which is what we set out to investigate in this paper.

The most relevant paper to our problem[11] investigated color characterization of specific scenes in a movie for mood analysis. They analyzed 15 movies and tracked how the mood for each scene transitioned based on color features of the movies. The features that they used consist of two parts. First, they chose 12 basic colors and used them as the basis for a color histogram for each scene in the movie. They then associated each color with a different mood. The second feature they used was a mood dynamics histogram, where they determined the mood of each scene based a mood's associated colors and created a mood dynamic histogram based on every possible combination of transitions between different moods.

The paper identified its features in a very clear and technical way which was thoroughly explained and justified. This proved to be very useful for our own feature extraction choices. However, while the paper concluded that the mood dynamics histogram was the most effective feature to classify a movie, their limited dataset of 15 movies suggests that their results might have been biased.

In this paper, the problem we set out to investigate was predicting the genre of a movie instead of transitions of moods within a movie, we decided to extract features at a more global level and set our class labels to be movie genres instead of moods. By only inputting the genre labels and raw data into our supervised learning algorithms, we are taking a purely data driven approach unlike the paper where they hard assigned moods to their 12 color palette.

3. Dataset and Features

3.1. Data Collection

Since our ownership of movies were limited in numbers, we used the Netflix movie streaming service to gather our movie color data. The Netflix user interface contains a low-resolution preview window that displays keyframes at 10-second intervals which can be navigated by arrow keys. We created a script that generates keypresses and gathers data from the screenshot of the preview window. The advantage of this method as opposed to taking screenshots of the actual movie stream is that the preview window does not require any buffering time. At the same time, this approach restricted the data sample rate to one sample per 10 seconds. The color data that is sampled for each preview frame is the mean pixel red, green, blue (RGB) values of all the pixels in the preview frame.

We chose mean color as opposed to other statistical measures because the Netflix preview window pixels are already down-sampled mean colors of pixels from the actual movie frame. The resulting ”Color Barcode” feature captured is similar to the data presented in the Colors of Motion Visualizations [1]. The color barcode is represented as:

$$\begin{bmatrix} r_1 & r_2 & ... & r_i & ... & r_n \\ g_1 & g_2 & ... & g_i & ... & g_n \\ b_1 & b_2 & ... & b_i & ... & b_n \end{bmatrix}$$

Where $\{r_i, g_i, b_i\}$ are the mean red, green and blue pixel values of the frame captured at the $10 \times i^{th}$ second, n is the total number of frames captured for the movie, and $0 \leq r_i, g_i, b_i \leq 255$. That is,

$$r_i = \frac{1}{m \times n} \sum_{x=1}^{m} \sum_{y=1}^{n} r_{i(x,y)}$$

$$g_i = \frac{1}{m \times n} \sum_{x=1}^{m} \sum_{y=1}^{n} g_{i(x,y)}$$

$$b_i = \frac{1}{m \times n} \sum_{x=1}^{m} \sum_{y=1}^{n} b_{i(x,y)}$$

Where $m \times n$ is the resolution of the preview frame and $\{r_{i(x,y)}, g_{i(x,y)}, b_{i(x,y)}\}$ is the pixel color at pixel coordinate (x,y) of the preview frame.

Our data collection process is outlined in Figure 1 below.

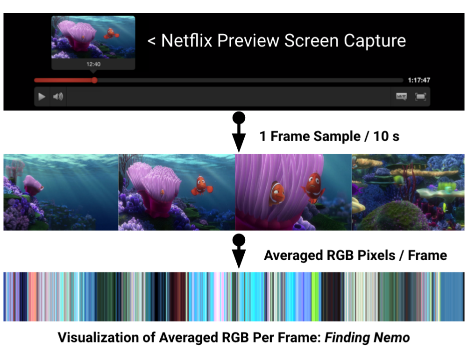

Figure 1. Data Collection process

We collected the color barcode data from 140 movies from each of the following genres: horror, animation, romance and action, making a total of 560 data points.

4. Methods

4.1. Support Vector Machine (SVM)

One of the main algorithms that we use to classify the movies into genres is the SVM. The SVM is an algorithm that finds the optimal decision boundary to separate the data into two partitions based on its labels. That is, it finds the line that maximizes the distance to the points closest to the boundary(these closest points are called support vectors), while trying to match the labels of the data points. Mathematically, this can be expressed as:

$$\min_{\gamma,w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{s.t.} \quad y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i, i = 1, ..., m$$
$$\xi_i \geq 0, i = 1, ..., m$$

In the equation above, the parameter C balances the two objectives of increasing distance between the decision boundary and the support vectors, and correctly labeling the points.

Solving the optimization problem above is difficult, and it is easier to solve the Lagrangian dual optimization problem below:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, i = 1, ..., m$$
$$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$$

One of the distinct advantages of the SVM is the option to use kernels to map the feature vector to a higher dimension space, which has the potential to separate data that falls along curved decision boundaries.

To classify between multiple classes, we used the Winner Takes All (WTA) SVM method as described and empirically supported in papers by Duan and Rifkin [6, 8]. That is, we trained multiple 1-vs-all SVMs, and then to determine the label of an unlabeled test example, we pass the unlabeled test example to each of the SVM models, and obtain the score corresponding to each model. This score is the signed distance from the data point to the decision boundary, hence a large positive score signify that the data point is both a positive example, and is far from the decision boundary. Therefore to assign a test example to the most probable class, we label that test example with the label of the model that had the largest score.

4.2. Multinomial Naive Bayes

The Multinomial Naive Bayes model is another supervised learning method that we used to classify the movies. The Naive Bayes model assumes that observing a certain feature, X_i, is independent of observing another feature, X_j, given the label, Y. That is

$$p(X_i, X_j|Y) = p(X_i|Y)p(X_j|Y) \quad \forall i \neq j$$

With this assumption, we can build a probabilistic model based on the training examples. Specifically, we are finding the parameters

$$\Phi_{j|Y=k} = p(x_j = 1|Y = k)$$
$$\Phi_{Y=k} = p(Y = k)$$

that maximizes the likelihood of the model:

$$\hat{P}(Y = k|X_1, ..., X_P) = \frac{\Phi(Y = k) \prod_{j=1}^{P} p(X_j|Y = k)}{\sum_{k=1}^{K} \Phi(Y = k) \prod_{j=1}^{P} p(X_j|Y = k)}$$

For each testing example, we can then calculate $\hat{P}(Y = 1|X_1, ..., X_P), \hat{P}(Y = 2|X_1, ..., X_P), ...,$ and label the training example with the label that has the largest corresponding probability.

4.3. k-means Clustering

k-means clustering is an unsupervised learning method to identify clusters in a dataset, $x(1), x(2), ..., x(m)$. The algorithm is initialized by randomly selecting k centroids, $\mu_1, \mu_2, ...\mu_k$, in the space of the dataset. It then iteratively labels each point in the dataset with the index of the nearest centroid, $c^{(i)}$, and subsequently moves the centroid to the center of those points. That is, for each iteration,

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2$$

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}$$

We run this until the labels for the data points,$c^{(i)}$, and the location of the centroids,μ_j, converges and stops varying between iterations.

5. Results

5.1. Parameters

5.1.1 SVM Kernels

We chose to use a linear kernel for the SVM because higher order kernels (polynomial, Gaussian, RBF) were over-fitting the data. That is, we observed that while these model consistently had a 0% training error, they also had consistently high generalization errors. For all future discussion of SVM's, a linear kernel is used.

5.1.2 Slack Parameter

To determine the optimal value for C, we tried a range of different values to find the value that would result in the lowest training and generalization error. For example, figure 2 shows the plots of training and generalization error plotted against C for the final feature vector. In this case, we notice that the training error is zero right from when C is small, which implies that the data points are linearly separable, and hence the weighting coefficient for the slack parameter is inconsequential. For the subsequent discussion of SVM's, the value of C used is 10.

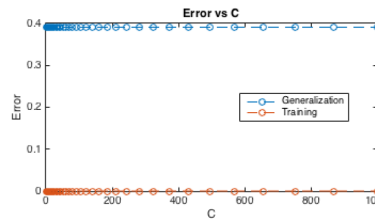

Figure 2. Error for different values of C

5.1.3 Number of Clusters

Part of building our feature vectors involved using k-mean clusters to group similar colors together. To determine the value of k, the number of clusters, that was not too large, and yet results in the lowest error, we ran our movie classification algorithm on different values of k. Figure 3 shows the classification errors for the different genres across a range of values for k.

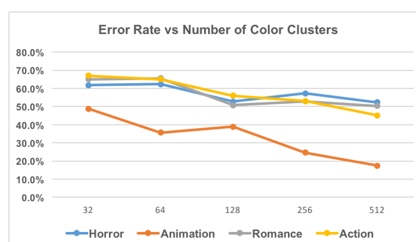

Figure 3. Plot of Generalization error against k

Based on the results presented in figure 3, and limited by the amount of computational power we had available, we used $k = 512$ for the experiments to be discussed below.

5.2. Color Barcode

The first feature vector we tried was the raw sampled "color barcode" data that was resampled and interpolated to be of a fixed length. Table 1 shows the confusion matrix of the Linear SVM classifier tested with 10-fold cross validation over 100 trials. During each trial, a random permutation of our data set was split into the ten equally sized partitions for cross validation. Each partition had 14 movies. Thus the training set is always 136 movies and the test set is always 14 movies. Hence, by the law of large numbers, the average TP+FN for each genre converges to 14 over 100 trials.

	TP	FP	FN	Precision	Recall
Horror	5.33	7.98	8.67	40.3%	38.4%
Animation	7.19	4.36	6.81	62.6%	51.6%
Romance	6.30	10.75	7.70	37.1%	45.3%
Action	4.08	10.01	9.92	29.0%	29.3%

Table 1. Confusion Matrix for Color Barcode Feature

Animation movies had the highest classification accuracy with 62.6% precision and 51.6% accuracy.

5.3. Clustered Color Barcode

Since columnizing the R,G,B channels of the the color barcode conceals and ignores the relationship between the red, green, and blue intensities, we tried clustering all the colors in the training dataset using k-Mean Clustering (Refer to Section 5.1.3). After clustering, the R,G,B values for each in the test dataset color barcodes is replaced by the index of the centroid that it is the nearest to. We then use this time sequence of indexes as our feature vector.

Table 2 shows the confusion matrix of the Linear SVM classifier tested with 10-fold cross validation over 100 trials. The precision and recall values were very close to the random classifier value of 25% and is worse than the unclustered color barcode results.

	TP	FP	FN	Precision	Recall
Horror	3.03	11.57	10.97	21.1%	21.5%
Animation	3.37	10.43	10.63	24.3%	24.5%
Romance	3.62	10.03	10.38	26.7%	26.6%
Action	3.35	10.60	10.65	24.7%	24.0%

Table 2. Confusion Matrix for Clustered Color Barcode Feature using SVM

5.4. Clustered Color Histogram

For the color barcode and clustered color barcode features, the featured dimension is storngly linked to the position of the sample within the movie. This representation may not be ideal since the same movie sampled at slightly different start and end points will have the color information in different dimensions. Thus, we experimented with removing the temporal information and purely looking at color distributions ”palettes” of movies. Since color histograms were used in various works to analyze mood of visual content [11], we tried the color histogram as a feature vector. To generate the clustered color histogram vector, we binned the colors in a movie's color barcode to the nearest of the 512 color cluster centroids by R,G,B distance. Each element of the feature vector corresponds to the number of occurrences of that corresponding color. The clustered color histogram for Finding Nemo (with 32 cluster centroids for visual representation) is shown in Figure 4. We can see that Finding Nemo's colors are predominated by blue hues due to the underwater scenes.

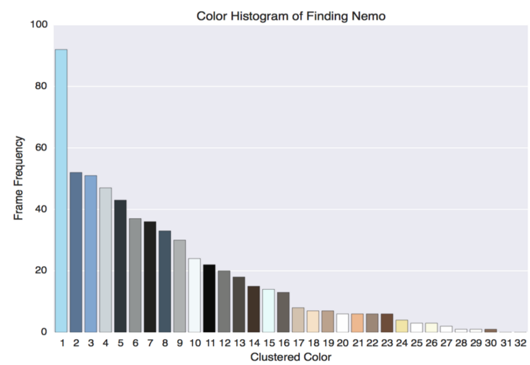

Figure 4. Color Histogram of Finding Nemo

	TP	FP	FN	Precision	Recall
Horror	6.29	7.06	7.71	47.2%	45.0%
Animation	10.94	4.39	3.06	71.5%	78.2%
Romance	5.61	7.85	8.39	41.9%	40.3%
Action	7.18	6.69	6.83	51.9%	51.3%

Table 3. Confusion Matrix for Clustered Color Histogram Feature using SVM

	TP	FP	FN	Precision	Recall
Horror	5.56	5.641	8.44	49.8%	39.8%
Animation	11.92	6.34	2.08	65.5%	85.2%
Romance	7.28	6.68	6.72	52.2%	52.0%
Action	5.32	7.25	8.68	42.6%	38.1%

Table 4. Confusion Matrix for Clustered Color Histogram Feature using NB

The accuracy of the Linear SVM and Naive Bayes classifiers tested with 10-fold cross validation over 100 trials (Table 3 and Table 4) improved for all genres with the color information unlinked from the temporal location of each frame. We used the multinomial Naive-Bayes algorithm for this feature vector because the frequencies are positive integers.

5.5. Clustered Color Histogram with Dynamics

The clustered color histogram feature performed better without temporal information than the color barcode and clustered color barcode features which had temporal information. Thus, we wanted to introduce temporal information in a manner that would not be destructive to the clustered color histogram feature.

We first converted the RGB values to CIELUV[1] format, a format that allows us calculate the difference between two colors as the euclidean distance between color vectors. This allows us to track the change of color between frames. Figure 5 shows the change in CIELUV color distances for Finding Nemo over time.

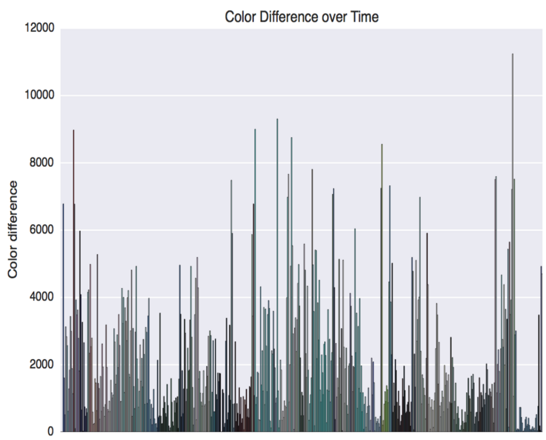

Figure 5. Color Difference over Time for Finding Nemo

We then averaged the color change distances between all frames to get a mean color change value (color dynamics) for each movie. The distributions of movie color dynamics for the four genres in Figure 6.

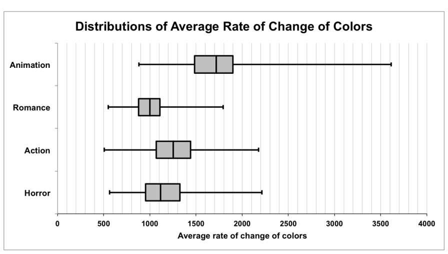

Figure 6. Distributions of Genre Color Differences

From the color dynamics distributions we can observe that romance movies generally have a smaller color dynamics value and animation movies have a larger color dynamics value. This makes sense since romance movies have slower scenes while animated movies are flashier. Based on this data and the usefulness of color transition in movie mood studies [11], we added the color dynamics value as a new element to the clustered color histogram feature vector.

[1]The 1976 CIELUV color space provides a perceptually uniform color space. In this color space, the distance between two points approximately tells how different the colors are in luminance, chroma, and hue[9]

	TP	FP	FN	Precision	Recall
Horror	6.51	7.10	7.49	48.0%	46.5%
Animation	11.67	3.21	2.33	78.5%	83.5%
Romance	7.20	6.93	6.80	51.2%	51.7%
Action	7.56	5.82	6.44	56.7%	54.2%

Table 5. Confusion Matrix for Clustered Color Histogram with Dynamics Feature using SVM

	TP	FP	FN	Precision	Recall
Horror	6.73	5.50	7.27	55.4%	48.2%
Animation	12.98	3.98	1.02	76.6%	92.7%
Romance	7.82	6.13	6.18	56.2%	56.1%
Action	5.59	7.28	8.41	43.7%	40.0%

Table 6. Confusion Matrix for Clustered Color Histogram with Dynamics Feature using NB

The accuracy of the Linear SVM and Naive Bayes classifiers tested with 10-fold cross validation over 100 trials (Table 5 and Table 6) improved for all genres with the color dynamics information.

6. Conclusion

The combined clustered histogram and color dynamics feature processed from the color barcode feature gave us better results than the unprocessed color barcode feature. We could correctly classify 13 out of 14 animated movies among four genres (horror, animation, romance, action) from the average frame colors sampled 6 times a minute using a Naive Bayes classifier with a training set of 126 movies. The classification accuracy (recall) for other genres (horror: 48.2%, romance: 56.1%, action: 54.2%) were better than the 25% random classifier accuracy indicating that there is learnable movie genre information in movie colors.

We did some exploration of our dataset through unsupervised K-Means clustering on the unprocessed color barcode feature. Based on the the clustering results that we obtained, we observe that an unsupervised learning algorithm (such as the k-means algorithm) shows promise to successfully group movies that have similar color schemes.

Clustering movies by combining color cluster histogram and color dynamics feature can be explored and developed in the future. One of the potential applications of such a clustering algorithm would be a movie recommendation system based on the intrinsic properties of movies, similar to the Pandora ”Music Genome Project” [2] for music. It would also be interesting to find common words in the subtitles of the clustered movies to link movie content to movie color.

References

[1] The colors of motion. http://thecolorsofmotion.com/films. [Online; accessed 11-December-2015].

[2] The music genome project. https://www.pandora.com/about/mgp. [Online; accessed 11-December-2015].

[3] P. Bellantoni. *If it's purple someone's gonna die*. Focal Press, 2005.

[4] A. Blackstock and M. Spitz. Classifying movie scripts by genre with a memm using nlp-based features.

[5] C. Dorai. Automatic genre identification for content-based video categorization. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 230–233 vol.4, 2000.

[6] K.-B. Duan and S. S. Keerthi. Which is the best multiclass svm method? an empirical study. In *Multiple Classifier Systems*, pages 278–285. Springer, 2005.

[7] S. P. N. Bonneel, K. Sunkavalli and H. Pfister. Example-based video color grading. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH), vol. 32, no. 4.*, 2013.

[8] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141, 2004.

[9] J. Schwiegerling. Field guide to visual and ophthalmic optics. SPIE Bellingham, WA, USA, 2004.

[10] M. Solli and C. C. M. Solli. Linkping studies in science and technology dissertations, no. 1362 color emotions in large scale content based image indexing.

[11] C.-Y. Wei, N. Dimitrova, and S.-F. Chang. Color-mood analysis of films based on syntactic and psychological models. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 2, pages 831–834. IEEE, 2004.

[12] S. Xue, A. Agarwala, J. Dorsey, and H. Rushmeier. Learning and applying color styles from feature films. *Computer Graphics Forum*, 32(7):255–264, 2013.

[13] H. Zhou, T. Hermans, A. V. Karandikar, and J. M. Rehg. Movie genre classification via scene categorization. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 747–750, New York, NY, USA, 2010. ACM.