

Multimedia Feature Generation of Movie Trailers for Genre Prediction

John Fuini, Nathaniel Guy, and Yong Han Noel Kim

University of Washington, Physics/Aeronautics & Astronautics



MOTIVATION

Movie trailers are one of the most effective advertising tools for movies. Human movie viewers can tell the genre of a movie given the first minute of watching its trailer. Viewers have developed this cognitive ability by watching countless movies of various genres over time, and have subconsciously associated certain cinematic features with specific genres.

We aimed to develop an algorithm that mimics this process for computers. There were two major parts to this endeavor. The first is the process of identifying cinematic features from a large set of trailers of a known genre (as determined by www.movie-list.com); the second is training a machine learning algorithm to develop classification criteria based on these features and genre metadata.

Lastly, the generated classification criteria were tested on a set of trailers to assess the effectiveness of our approach.

MOVIE FEATURE GENERATION

Video Features

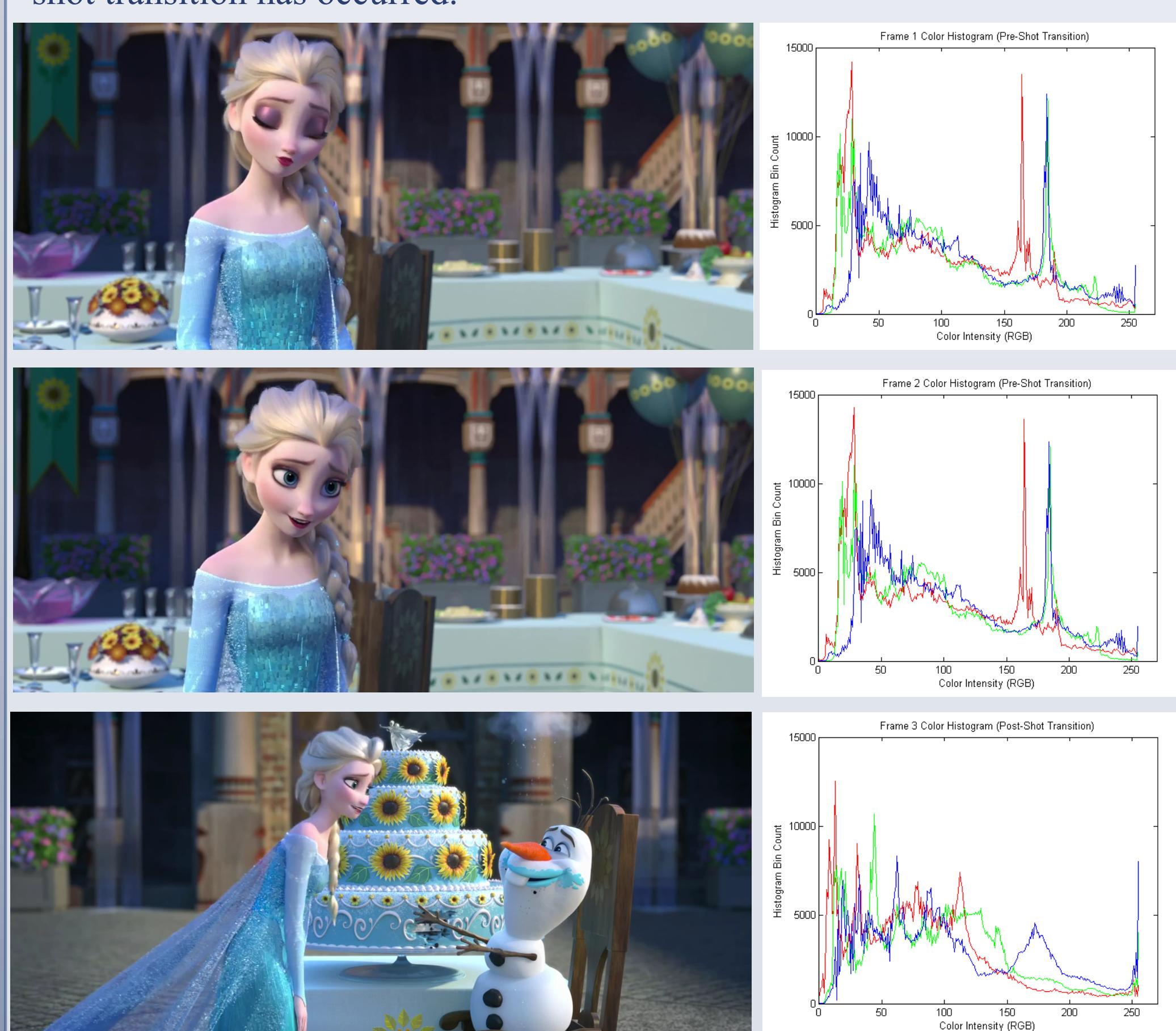
A large part of our features were generated via the “OpenCV” computer vision analysis package, with code written in Python [1].

I. Total Run-time/Number of Frames

Individual frames were counted to get a total run-time of any given trailer in terms of frame count. OpenCV can provide the frames per second (FPS) for a given trailer as well, from which we obtained the run-time of trailers in seconds.

II. Number of shots

We detected changes between shots by examining the color histogram for each frame, and calculating the chi-square difference between adjacent frames’ histograms. When this distance exceeds a certain empirically-determined threshold, we can conclude with reasonable certainty that a shot transition has occurred.



A demonstration of shot transition detection. The histograms pertaining to the same shot (top, middle) share similar color histograms, while the one for a different shot (bottom) displays different color components. (Frozen Fever, animation)

III. Shot length

Once the shot transitions were identified, we were able to obtain the following features: shot length mean, standard deviation, minimum value, and maximum value.

IV. RGB Component Statistics

R, G and B components denote the proportion of red, green and blue colors for a given pixel. We calculated the average intensities along each of these color channels, across pixels and across all frames.

MOVIE FEATURE GENERATION (cont.)

V. Detail Score

We summed Canny edge detection filter [2] data to estimate frame detail.



An edge detection filter was used to obtain different detail scores for frames with different level of detail. (top) Kung Fu Panda 3, Animation-Comedy; (bottom) Interstellar, Adventure.

VI. Dark Scenes

We defined a “dark scene” as consecutive frames of darkness. We recorded the minimum, maximum, mean and standard deviation of the lengths of these scenes, and the total fraction of frames composed of dark scenes.

Audio Features

I. Volume

The mean, standard deviation, minimum and maximum volume features were acquired from the audio waveforms.

II. Sudden rise/fall

We captured the occurrence of sudden increase and decrease of volume events (i.e., changes in volume during a small time period, with a magnitude of change larger than the standard deviation of the volume).



A sample waveform showing a sudden rise (red) and a sudden fall (green)

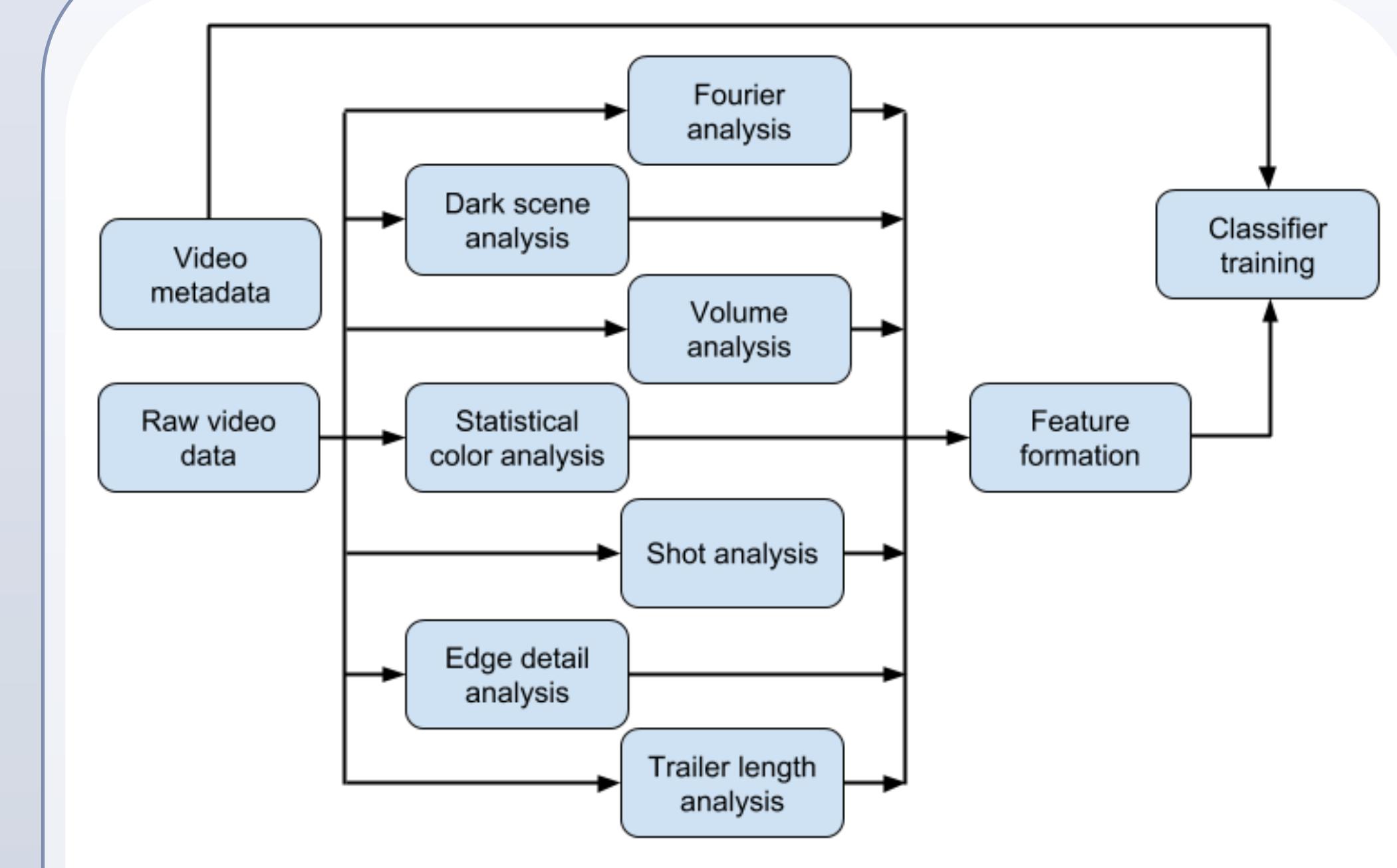
III. Frequency components of audio

The frequencies constituting the audio track were analyzed via Fast Fourier Transform. They were bucketed into eleven standard octave bands, and the percentage of each octave in the given trailer was recorded.

MACHINE LEARNING ALGORITHM

Using supervised learning, we created a decision tree for each genre. Each trailer data entry has Boolean variables for each genre. With 958 movie trailers in total, we trained our decision tree on a randomly selected 80% of our trailers using the MATLAB environment (command: `fittree`). We then used this tree to predict whether or not the remaining 20% of trailers were in a particular genre, and the results were compared to known Boolean values (command: `predict`). We cross-validated this process 40 times for the ten most common genres in our data set. The average success rates are plotted in the Results section of this poster.

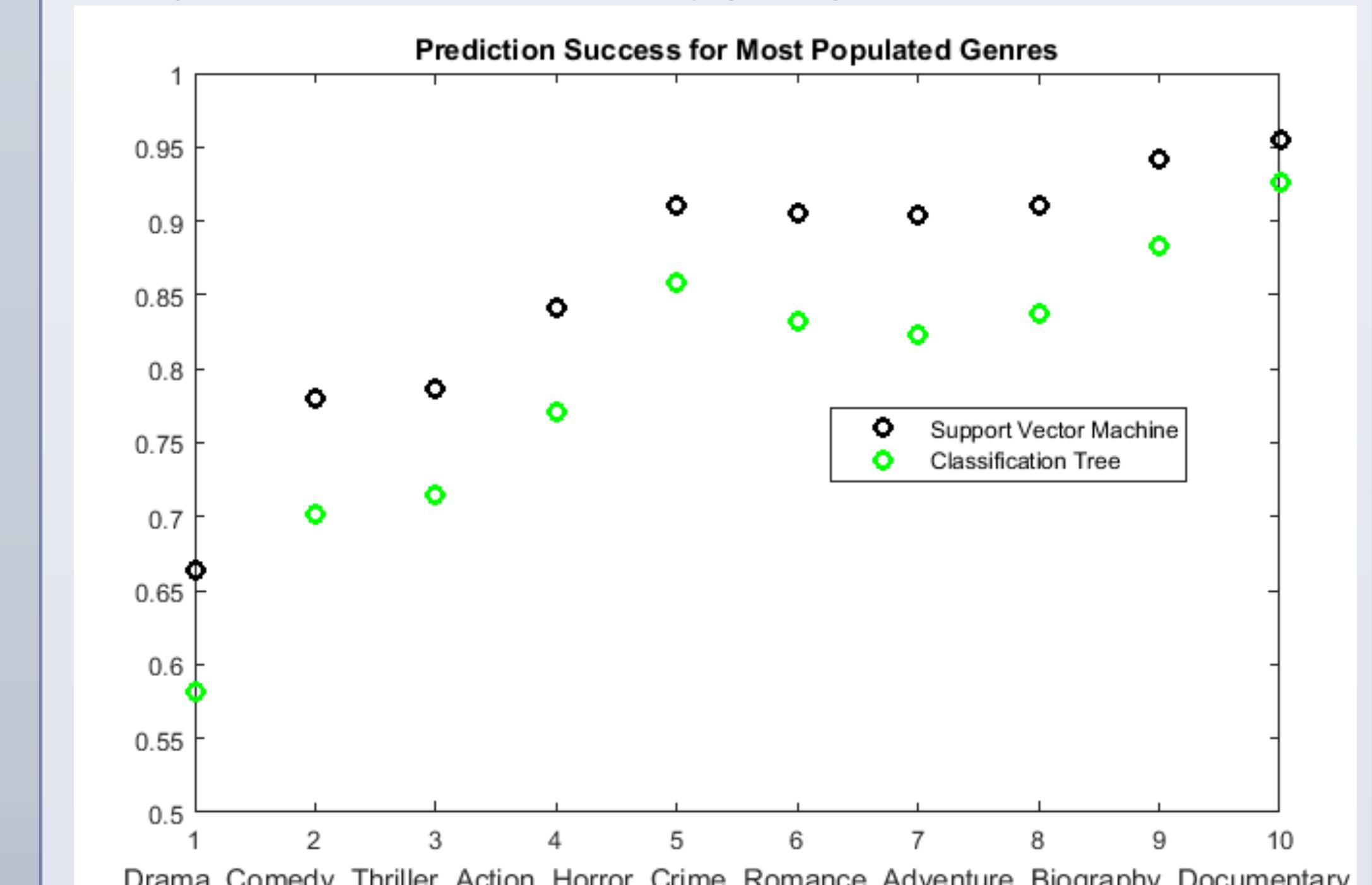
The binary decision tree provides an adequate success rate. Nonetheless, we experimented with the support vector machine (SVM) model as well. We created an SVM classifier (command: `fitesvm`), used it for classification, performed 10-fold cross-validation (command: `crossval`), and noted the obtained accuracy data (command: `kfoldLoss`).



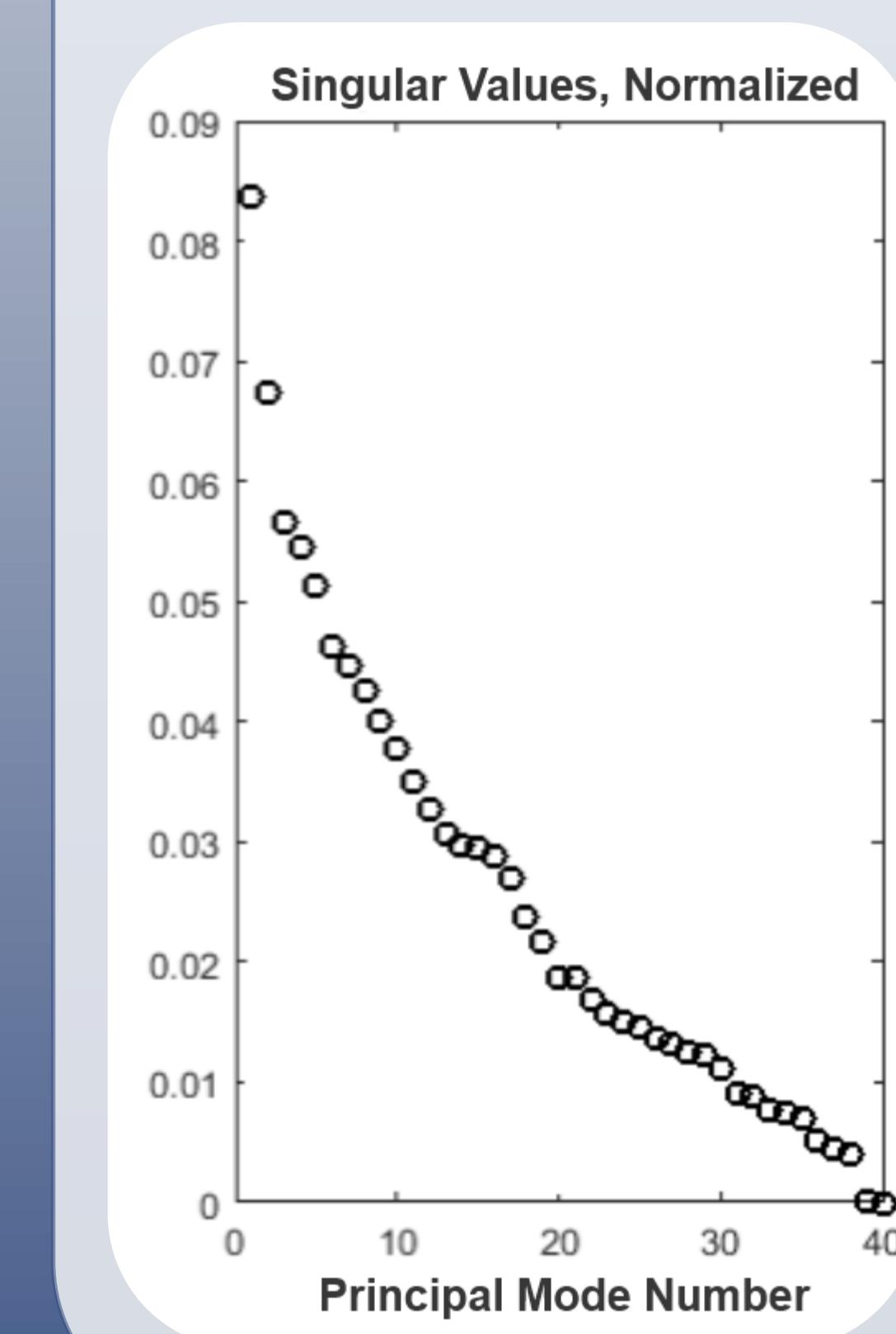
Block diagram illustrating our overall algorithm pipeline

RESULTS

The rates of success of classification by our algorithm for top ten most popular genres are shown below. The Drama genre had the lowest success rate, with 58% and 66% successful classification for binary decision tree and SVM respectively. The rest of the genres registered consistently over a 70% success rate. Compared to binary decision trees, SVM had roughly an 8% higher success rate overall for any given genre.



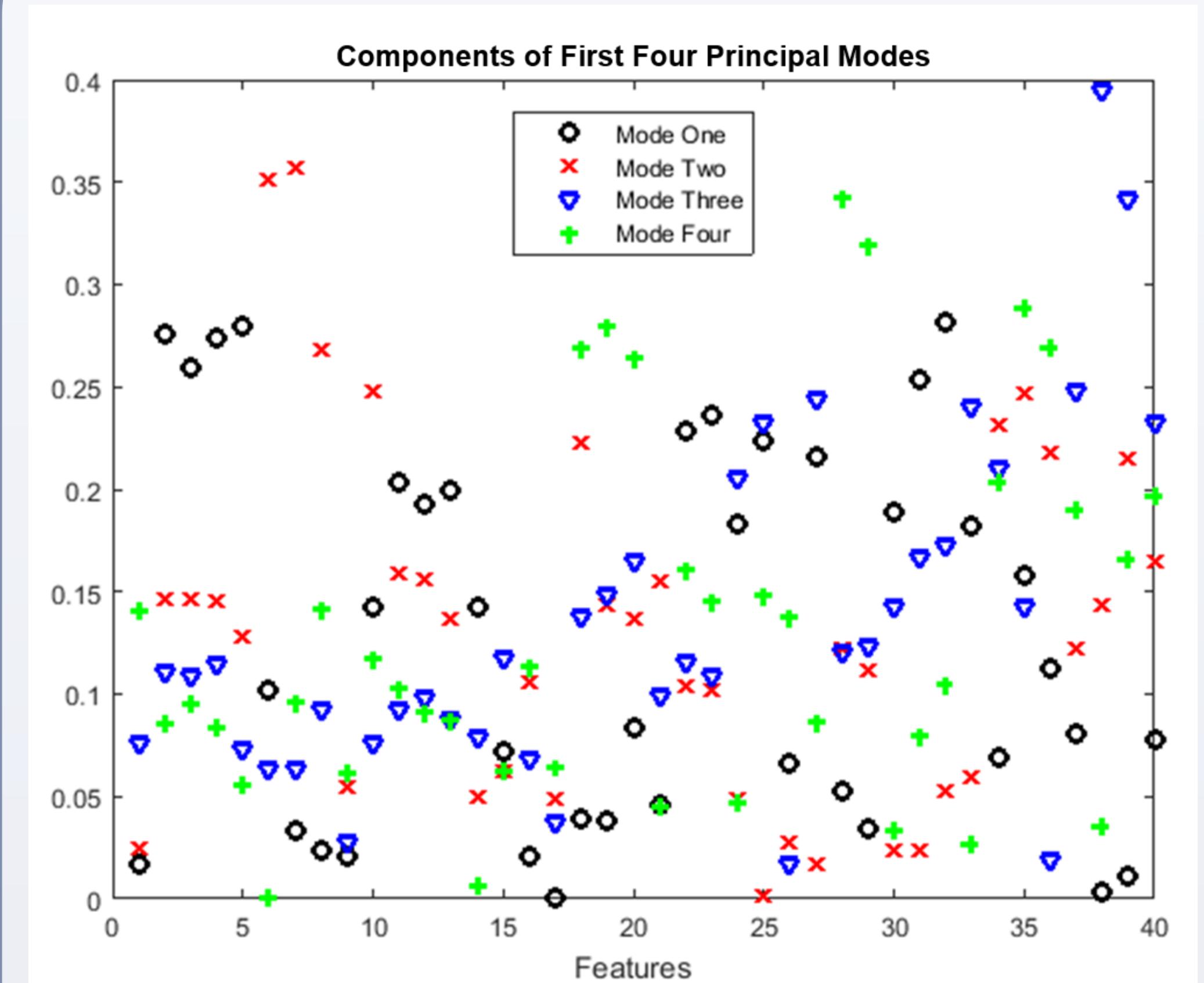
Performing singular value decomposition (SVD) on the feature set can give us the covariance of the principal modes in the set.



Aside from the first two modes, the covariance of the principal modes decreased steadily. Note that these modes are not equivalent to features, but rather, a combination of features present in the set. For example, in the figure on the upper right, one can observe how modes 1 through 4 are each combinations, to a varying degree, of all the features.

The first mode captures approx. 8.5% of the energy in the feature set. We performed a dimensional reduction by making classifications using the first mode only. Surprisingly, the difference of success rate between single-mode and full-mode was less than 10%.

RESULTS (cont.)



CONCLUSIONS AND FUTURE WORK

This work demonstrates the construction of a movie genre classifier using features acquired from video and audio portions of movie trailers. A series of video analyses using a computer vision library generated numerous video features, such as detail scores, dark scenes, color and intensity profiles, etc. Temporal and frequency analysis of audio identified sharp increases and decreases of volume and frequency spectra. MATLAB classifiers trained using these features classified movie genres with success rate ranging from 60% to 95%. Of two classifiers, the binary decision tree and support vector machine, the latter had approx. an 8% higher likelihood of successful classification overall.

It is worth noting that our machine learning algorithm’s pipeline has a minimum amount of required human interaction. The only step within that necessitates human input is providing a genre classification for the initial set of movie trailers used for training. The rest of the pipeline is almost entirely automated. While this project was tailored to movie trailers, it can be adapted more generally to any environment that utilizes computer vision/audio-based machine learning. This can include surveillance, video tracking, and other applications.

Another target of future work is comparing our results with the performance of human subjects. Since the definition of a “genre” is rather loose, not everyone would agree on the genre of a given trailer. Thus, it’s likely that the predictions of a human subject may differ from the genre labels in our metadata. We hypothesize that it may even be possible for machine learning to outperform humans at this task.

REFERENCES

- [1] OpenCV. Itseez. Web. 09 Mar. 2016. <http://opencv.org/>.
- [2] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6): 679–698, 1986

ACKNOWLEDGEMENTS

Thanks to Dr. Nathan Kutz for his assistance with this project.