Analyzing Thumbnail-Category Fit for YouTube Videos

Seth Gregory[*]
University of Maryland
College Park, MD, USA
sethaarongregory@gmail.com

Michael Suehle[*]
University of Maryland
College Park, MD, USA
mike.suehle@gmail.com

Andrew Zhong[*]
University of Maryland
College Park, MD, USA
azhong13@terpmail.umd.edu

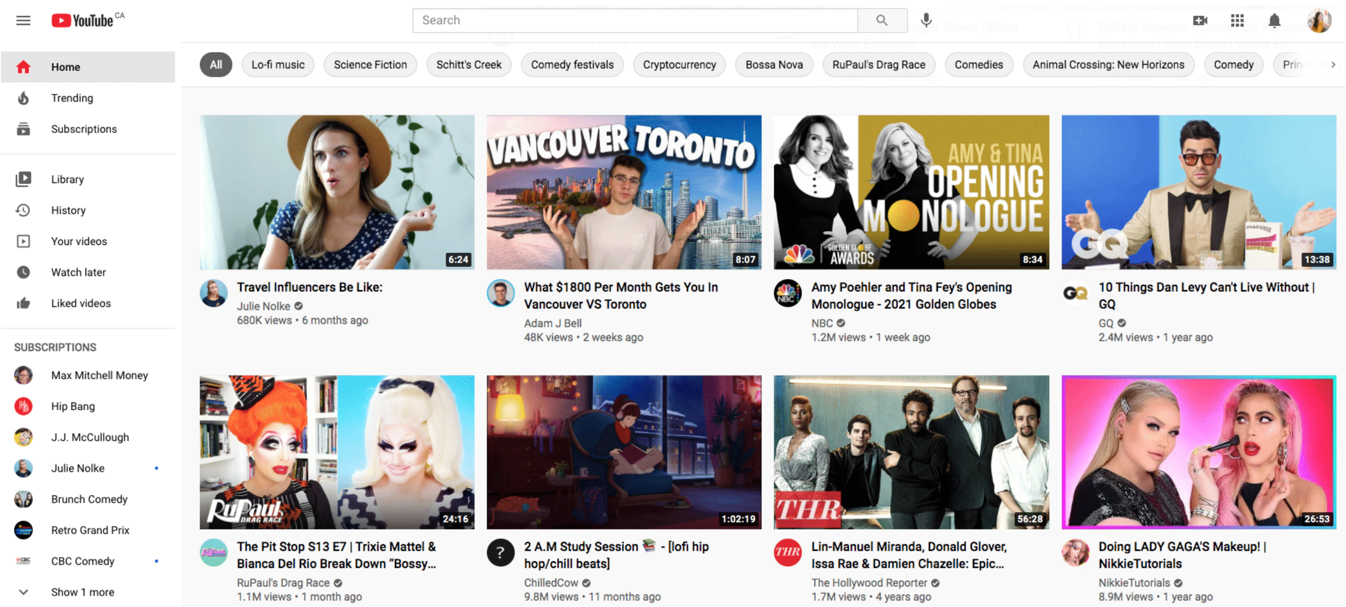

Figure 1. Sample YouTube Home Page with Thumbnails

Abstract

On social media platforms, a significant amount of new content that users consume is discovered through recommendations, which are often presented with limited information per post (On Youtube's Recommendation System). As a result content creators must carefully curate how their content appears to prospective viewers. In the case of Youtube, video suggestions are dominated by their thumbnail and title, which are the primary targets of uploaders seeking to improve video performance.

In this research project, our goal is to determine if there is a trend in thumbnails for high performing Youtube videos from different categories and if we could train a classifier to determine which category a thumbnail belongs to.

Keywords: datasets, convolutional neural networks, support vector machines, sentiment analysis, computer vision

ACM Reference Format:
Seth Gregory, Michael Suehle, and Andrew Zhong. 2021. Analyzing Thumbnail-Category Fit for YouTube Videos. In . ACM, New York, NY, USA, 4 pages. https://doi.org/N/A

1 Introduction

The problem we tried to solve is, "what makes YouTube thumbnails successful for videos in their respective category?" Solving this problem could help advertisers and content creators make more successful thumbnails. Since trends on YouTube are constantly changing, we chose to focus on the most recent data we could find. Exploring successful thumbnails throughout the years may be helpful for predicting thumbnail trends in the future and documenting historical trends, but for this classifier, we focused on predicting what categories recent thumbnails fall into.

One of the main challenges to solving the problem was that it is often difficult to control for factors outside of visual thumbnail appearance. For example, certain videos will significantly outperform others even without using "good" thumbnails due to external factors such as channel, channel history, and outside fame. Another challenge is that videos

[*] All authors contributed equally to this research.

UMD CS Honors Program, Fall 2021, College Park, MD

https://doi.org/N/A

of different genres require different features to succeed, and determining where to split genres is a completely different problem from general thumbnail quality.

We developed a model that can predict what YouTube thumbnail qualities are indicative of success in different video genres, and gained insight into what high level features content creators should pay attention to when designing video thumbnails. After making improvements to this model, we could have it analyze potential thumbnails of content creators and output which genre it will likely perform well in.

2 Related Work

A variety of video quality analysis models have existed, with differing methods of describing quality. The most straightforward is the popularity model, which attempts to predict the future number of views on a video. Support vector regressions techniques have found that features obtained after video release (such as early view counts or other interactions) are able to predict future view counts with correlation coefficients of 0.9 or higher, while visual features that are available to uploaders before release had lesser predictive power with a correlation of 0.23 (Trzcinski.) Visual features analyses included average color, facial recognition, text detection, scene dynamics, clutter, rigidity, and deep features obtained from ResNet-152, and award-winning image identification neural network. Their results suggested that visual features present in Youtube video recommendations were poor predictors of video performance. An analysis on the effects of optimization of YouTube video metadata on video performance similarly found that popularity could be predicted with an R2 coefficient of determination of 0.8 using machine learning methods (Hoiles). However, like in the support vector regression study, the strongest predictors were found to be factors outside of the uploader's control such as subscriber count and first day view count. In its analysis on effects of metadata optimization, Hoiles et al. found that performance was sensitive to metadata changes; in just over half of videos improved titles, keywords, or thumbnails were able to increase video views. Sensitivity to optimization was found to be similar across all levels of popularity, suggesting that optimization methods used by high performing channels were applicable to smaller content creators.

3 Dataset

In choosing a dataset, there were several considerations in mind: for one, due to Youtube's ever-changing algorithm, shifts in trends, and other variable factors, we wanted a dataset as recent as possible, to ensure our results would not be obsolete. Furthermore, we wanted to train our model primarily on high-performing data, under the assumption that these thumbnails would be deliberately-crafted and likely contributed to the video's success. This way, our results might support our claim that there is a meaningful distinction between thumbnails of different categories/genres. To that end, we first attempted to utilize Youtube's API to manually create a dataset of current thumbnails, using it to search and download the top 100 high-performing thumbnails (by view-count) from each category. However, we struggled to obtain suitable data in this way, due to usage limits, storage space, and trouble with the search parameters. So, instead we obtained an existing dataset of about 5,000 thumbnails uniformly selected at random, five different times between December 31, 2020 and January 7, 2021, for a total of about 25,000 different thumbnails. We chose our "high-performance threshold" to be 100,000 views, leaving us with about 15,000 thumbnails to train the model.

4 SVM Model

4.1 Model

We first attempted to use similar models to those from popularity prediction models by using high-level features extracted from recommendation data. Specifically, we trained an SVM to predict video categories based on average thumbnail color, compound sentiment from image text, and compound sentiment from image title. Average color was obtained by computing the mean red, green, and blue color values for all pixels within each image, and was provided to the SVM as three separate parameters. Image text was extracted via OCR (optical character recognition) using Tesseract, a Google-sponsored open source OCR engine. Extracted texts were transformed into numerical inputs for the SVM by determining the compound sentiment. Compound sentiment is a numerical measure from -1 to 1 describing how negative (lower numbers) or positive (higher numbers) the sentiment in text is. Sentiment was computed using the Natural Language Toolkit (NLTK) library in Python. Images without text were encoded as having no text sentiment, or having a compound sentiment value of zero. NLTK was similarly used to encode title text. Scikit-learn was used to split data into training and test sets with an 80/20 split.

4.2 Results

The SVM performed rather poorly, with an accuracy of 0.112, and with an F-Score of 0.097. We believe this is due to the limited nature of the image attributes we considered; given only the average color and sentiment of text in a thumbnail, there is no clear classification into one category, so our model suggests that these factors alone do not differ substantially between categories. Perhaps with other image factors in consideration, an SVM could better classify a thumbnail, but at this point in our experiment we took an alternative approach.

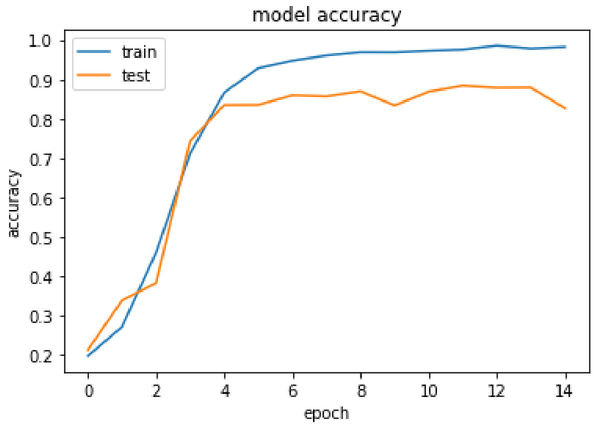

Figure 2. CNN Model Accuracy

5 CNN Model

5.1 Model

We next focused on applying a convolutional neural network (CNN) to the classification problem, due to its holistic approach to considering image data and its wide usage in the realm of computer vision. Our model architecture consisted of 15 layers, a combination of mostly convolutional layers and max pooling, which we performed on the training thumbnails re-sized down to about one-fourth of their original size. These architectural decisions were made primarily based on trial-and-error to achieve optimal performance. Image processing was done primarily using the OpenCV library, and the CNN implementation was performed using Keras and TensorFlow.

5.2 Results

Our CNN model performed much better than the SVM, achieving a testing accuracy of about 0.904. In other words, given a high performing thumbnail, our model should be able to predict its category with about 90 percent accuracy. This supports our claim that there are meaningful distinctions between thumbnails of different categories. We believe the CNN was able to perform better than the SVM due to its more comprehensive structure, working directly on the images as input rather than on more specifically chosen features. Given the nature of our dataset, we have reason to believe our model should generalize fairly well, and thus could be considered as a metric in helping creators decide whether their thumbnail is well-suited for their intended genre, though testing on a larger, more comprehensive dataset is likely necessary, though beyond the scope of this project.

6 Future Work

We identified many ways this experiment can be improved and built upon. The first way to improve this experiment would be to use an alternative to Google Colab. Google Colab is great for developing code that is not computationally intensive, but it had some issues running our code. One issue was the runtime for our image processing. Using the original thumbnails, our image processing took over twelve hours to complete. After twelve hours of processing, Google Colab ended the process due to their twelve hour limit. To address this, we reduced the quality of thumbnails and split up the types of image processing. With the lower quality thumbnails, both the OCR text and Harris corners processing still took a fairly long time to run on every thumbnail, taking around three and eight hours respectively.

Another issue occurred when we tried editing the code at the same time. Whenever we edited the Google Colab notebook at the same time, only one of our changes would get saved. To deal with this, we had to edit the code at separate times. For future work, we would instead use a Github repository. This would allow us to work concurrently and it would provide better and more documented version control.

Some ways we can improve and build upon this experiment involve creating new datasets, trying different classifiers, and analyzing other Youtube video characteristics. Since Youtube trends are always changing, it is worthwhile to try repeating this experiment using datasets from different years. The dataset we used from the user, wchaktse, on kaggle.com was created in the beginning of 2021. We could also try using classifiers other than CNNs and SVMs. Some other classifiers we identified are unsupervised classifiers, artificial neural networks, and random forest algorithms. Finally, there are many other characteristics in both thumbnails and Youtube videos that we do not account for in this experiment. Some characteristics in thumbnails that are worth testing are the presence of faces and their expressions, the presence of animals, the presence of expensive products, and a cartoon artstyle. Some characteristics of Youtube videos that we could explore are their length, title, channel, and view count. Although these characteristics along with thumbnails all have a large influence on the performance of a Youtube video, the Youtube algorithm may be an even bigger influence.

7 Acknowledgments

Special thanks to Dr. Dave Levin and the UMD Computer Science department for their help with this project and this introduction to research through the CS Honors Program!

References

[1] Arthurs, N., Birnbaum, S., and Gruver, N. (2017) Selecting Youtube Video Thumbnails via Convolutional Neural Networks. *Stanford.* http://cs231n.stanford.edu/reports/2017/pdfs/710.pdf

[2] Hoiles, W., Aprem, A., and Krishnamurthy, V. (2017) Engagement and Popularity Dynamics of YouTube Videos and Sensitivity to Meta-Data. *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1426-1437, 1 July 2017, doi: 10.1109/TKDE.2017.2682858.

[3] Lee, M. (n.d.). pytesseract: Python-tesseract is a python wrapper for Google's Tesseract-OCR (0.3.8) [Python]. Retrieved December 18, 2021, from https://github.com/madmaze/pytesseract

[4] Lu, X., Lin, Z., Jin, H., Yang, J., and Wang, J. Z. (2015) Rating Image Aesthetics Using Deep Learning. *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2021-2034, Nov. 2015, doi: 10.1109/TMM.2015.2477040.

[5] On YouTube's recommendation system. (n.d.). Blog.Youtube. Retrieved December 18, 2021, from https://blog.youtube/inside-youtube/on-youtubes-recommendation-system/

[6] Python, R. (n.d.). Sentiment Analysis: First Steps With Python's NLTK Library – Real Python. Retrieved December 18, 2021, from https://realpython.com/python-nltk-sentiment-analysis/

[7] Scikit-learn: Machine learning in Python—Scikit-learn 1.0.1 documentation. (n.d.). Retrieved December 18, 2021, from https://scikit-learn.org/stable/index.html

[8] Song, Y., Redi, M., Vallmitjana, J., Jaimes, A. (2016) To Click or Not To Click:Automatic Selection of Beautiful Thumbnails from Videos. *Yahoo Research*. https://dl.acm.org/doi/epdf/10.1145/2983323.2983349

[9] Tesseract User Manual. (n.d.). Tessdoc. Retrieved December 18, 2021, from https://tesseract-ocr.github.io/tessdoc/Home.html

[10] Trzcinski, T., and Rokita, P. (2017). Predicting popularity of online videos using Support Vector Regression. *arXiv*. https://arxiv.org/pdf/1510.06223.pdf

[11] Xie, T., Le, T., and Lee, D. (2021). Detecting Clickbait Thumbnails with Weak Supervision and Co-teaching. *ECML PKDD 2021*. https://2021.ecmlpkdd.org/wp-content/uploads/2021/07/sub_147.pdf

[12] Zannettou, S., Chatzis, S., Papadamou K., and Sirivianos, M. (2018) The Good, the Bad and the Bait: Detecting and Characterizing Clickbait on YouTube. *IEEE Security and Privacy Workshops (SPW)*, pp. 63-69, doi: 10.1109/SPW.2018.00018.