Automated Thumbnail Selection

Noah Arthurs, Sawyer Birnbaum, Nate Gruver
narthurs@stanford.edu sawyerb@stanford.edu ngruver@stanford.edu

Mission Statement

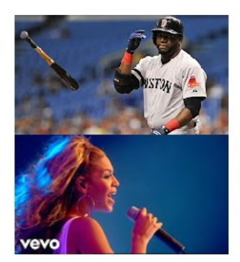
Automatically select an appealing thumbnail from within the frames of a video.

Background

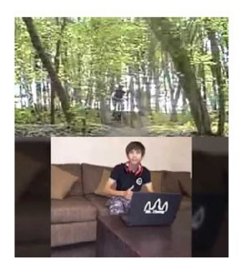
- The thumbnail of a YouTube video is the image that a user sees before clicking on the video. Naturally, this has a large affect on the success of the video.
- Experienced YouTubers often create and upload custom thumbnails, but newer content creators often let YouTube choose a thumbnail for them, in which case it comes from within the video.
- Yang and Tsai, 2015 [1] used CNNs to select good thumbnails.
- Liu et. al., 2015 [2] also investigated thumbnail selection, but focused on thumbnail-query relevance.

Dataset

In order to train a network that could rate the quality of a thumbnail, we put together a dataset labelled with two classes: **good** and **bad**.



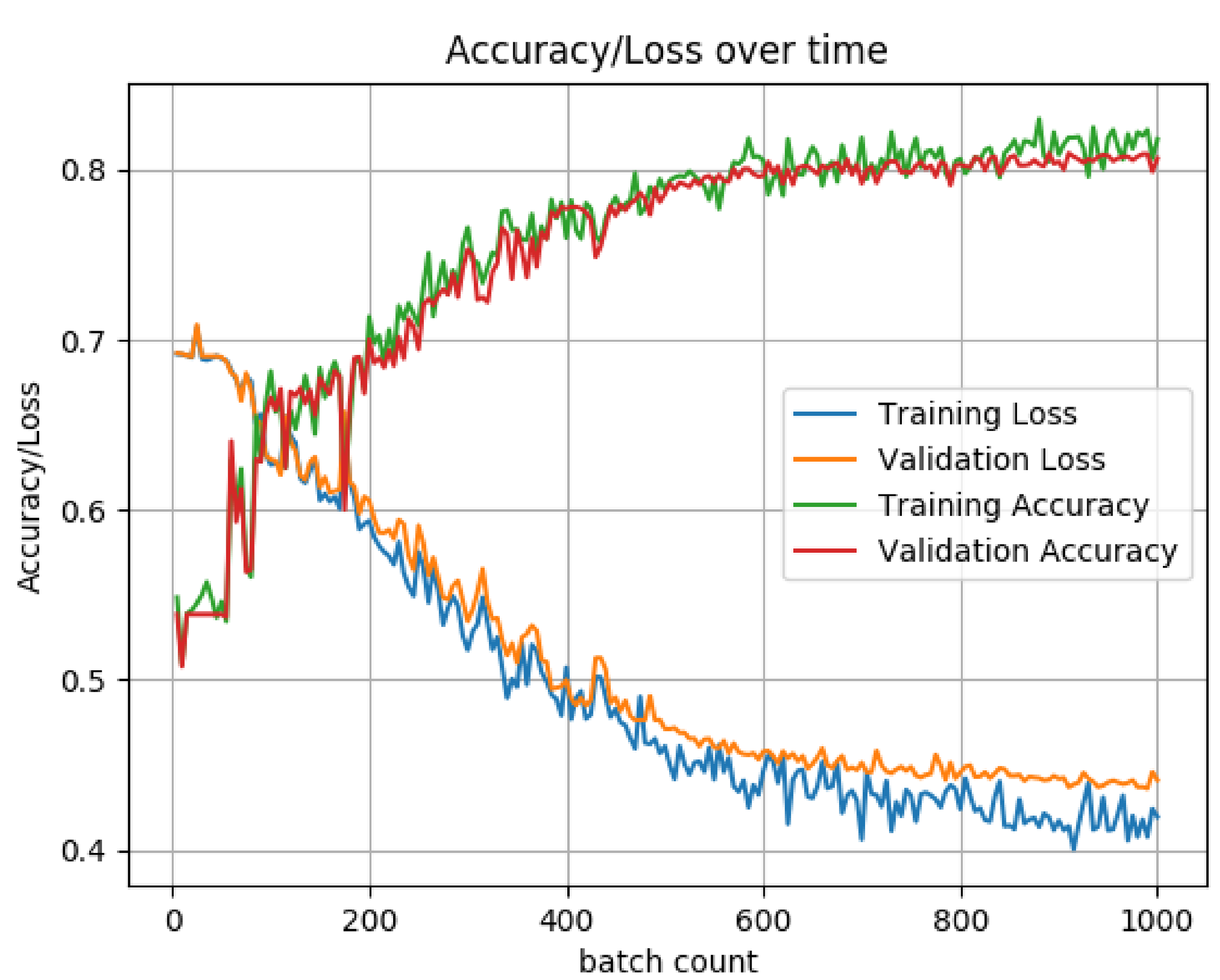
Good: We define a good video as one with 1 million or more views. In order to find these videos, we downloaded (at most) 5 videos with a million or more views from the 2,500 most-subscribed YouTube channels. It is worth noting that this set of channels is skewed towards the categories most popular on YouTube, like music and sports.



Bad: We define a bad video as one with 100 or fewer views. In order to find these, we looked at videos selected by a pseudorandom algorithm [3], of which about half were under 100 views. Unlike the "good" videos, we take these to be a representative sample of what is on YouTube.

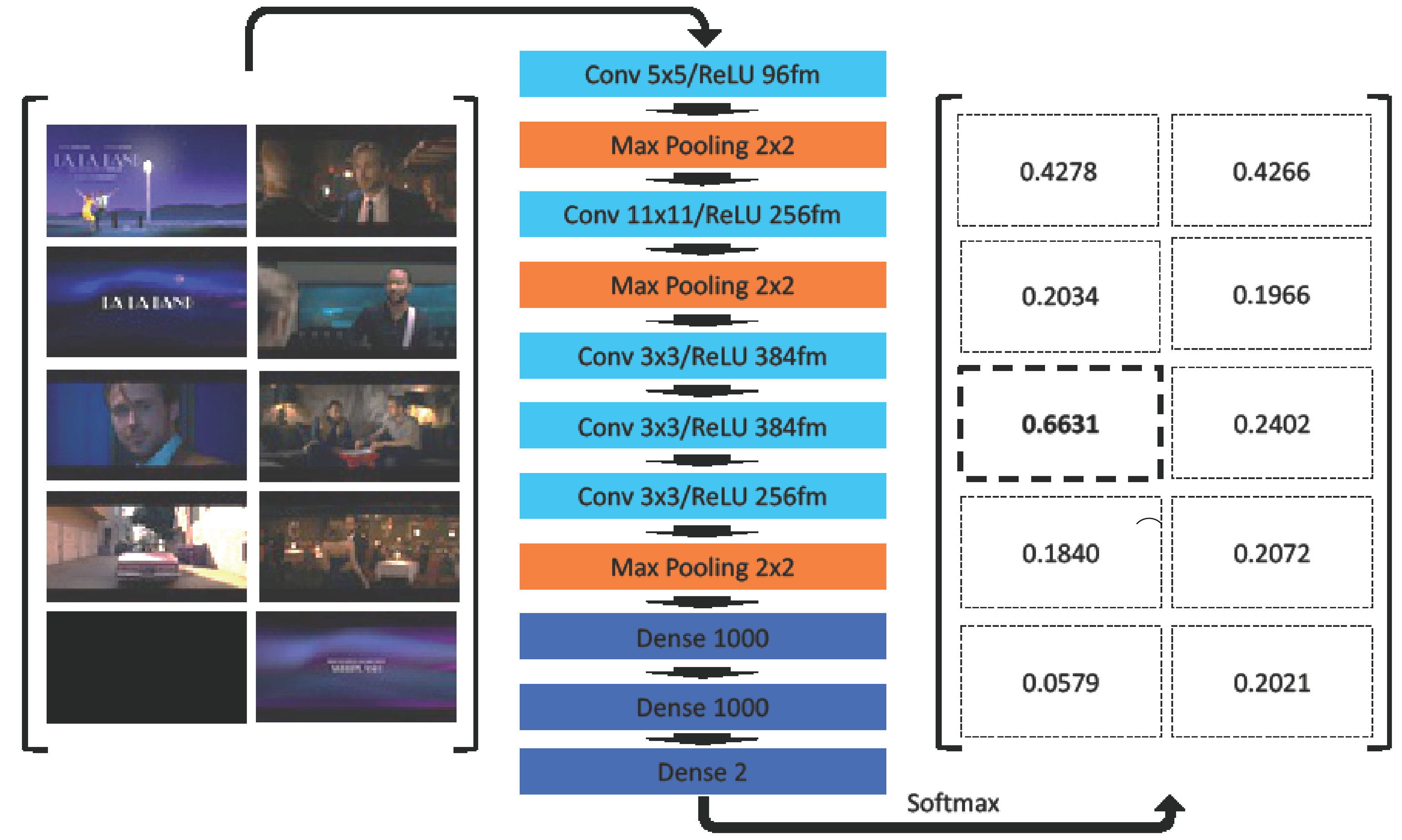
We ended up with ~5000 thumbnails of each class. Every image was cropped and scaled down to 45 pixels by 80 pixels before being fed into our model.

Training Visualization



- Above is the training graph for our full AlexNet with learning rate decay, regularization, and dropout. Learning rate decays from 1e-3 to 1e-5 over the course of 1000 batches.
- The training loss and accuracies are nosier because they were calculated by sampling 2000 points from the training set.

Model



Approach

1. First, we use our dataset of "good" and "bad" thumbnails to train a convolutional 2-class classifier.
2. In order to choose the thumbnail for a video, we push each* frame of the video through the classifier and select the frame that receives the highest probability of being in the "good" class.

Model Architecture

Our best model is based on the AlexNet architecture [4] with the following modifications:

1. We removed the batch normalization layers because they did not help learning.
2. We decreased the filter size in the 1st convolutional layer from 11x11 to 5x5 because our images have about half as many pixels as ImageNet, which AlexNet was trained on.
3. We reduced the size of the dense layers from 4096 to 1000 because we are only performing binary classification.

Training

- To select hyperparameters, we ran the following experiments (which except for the last two were run on a model with half as many filters per layer):

Validation Accuracy	Learning Rate (LR)	LR Decay	Reg	Drop %
0.76	1e-4	0	0	0
0.76	1e-4	0	**1e-2**	0
0.5594	1e-4	0	**1e-1**	0
0.762366	1e-4	0	1e-2	**0.2**
0.763386	1e-4	0	1e-2	**0.4**
0.790413	**1e-3**	**0.631**	1e-2	0.4
0.809	1e-3	**0.79**	1e-2	0.4

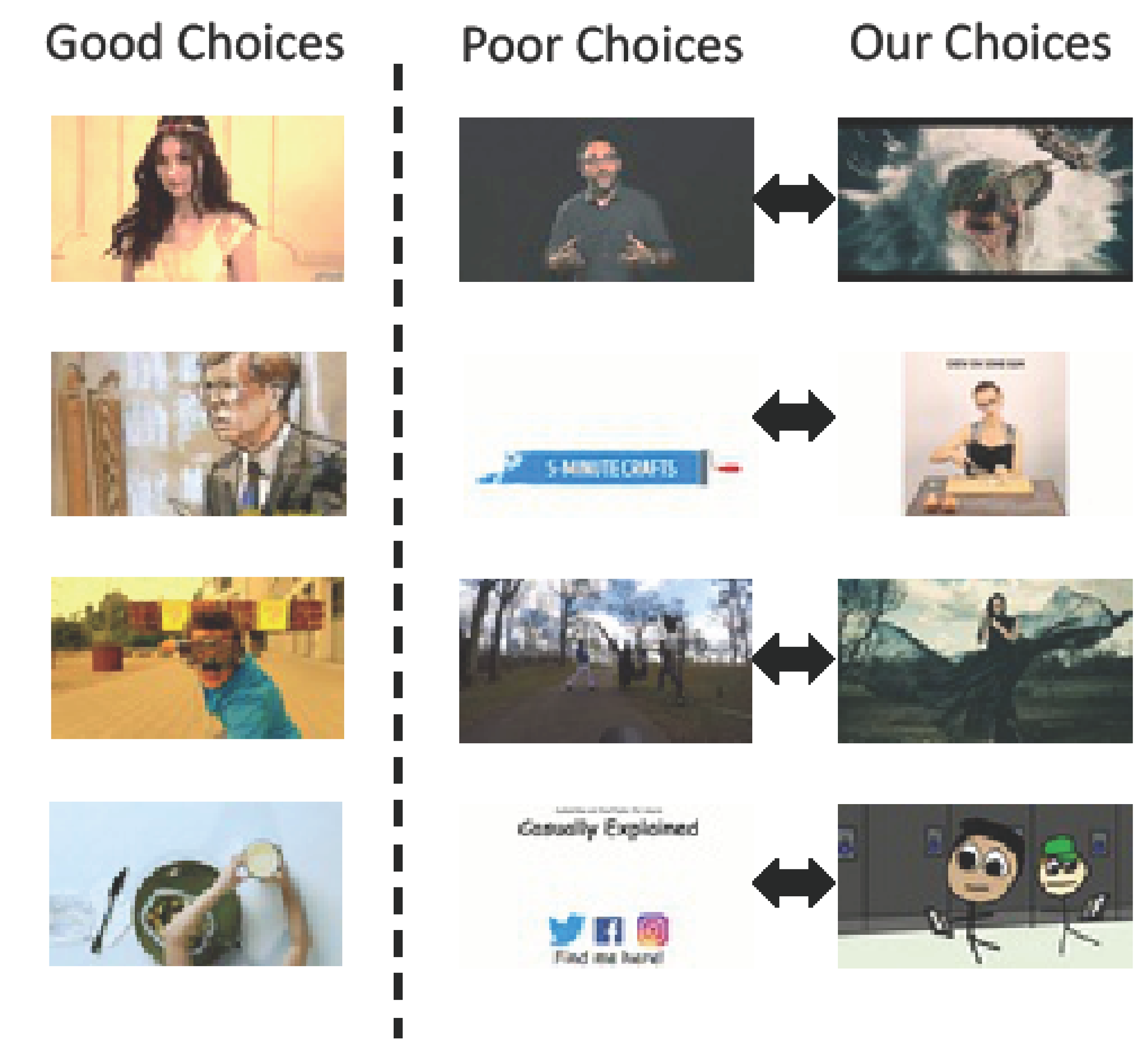
Model Evaluation

- Two of our group members classified 212 thumbnails. Both achieved an accuracy of 81.6%, so our model almost has a human level of accuracy on the classification task.
- We have a 7.7% false positive rate and a 10.7% false negative rate on the validation set. Below are some examples illustrating some forgivable and unforgivable mistakes that our model makes.
- Our saliency maps show some features our model has fit to. The left one shows its preference for hands, which makes sense as something to focus on in a thumbnail. However, the right one shows its focus on logos/labels which appear often in the thumbnails from our "good" set. This is something to mitigate since having a logo does not indicate a high quality image.

False Positives False Negatives Saliency Maps

Thumbnail Picker Results

- After training our model, we had it select thumbnails for 84 videos across 9 different categories on YouTube.
- We evaluated its success by comparing it to our own judgments. Since we could not rate every frame in a video (as the model ultimately would), we evaluated 10 evenly spaced frames for each video.
  - **23.5%** of the time our model agreed with our top choice.
  - **83.9%** of the time our model chose an image that we deemed a reasonable choice given the options. The number of reasonable choices varies per video, but averages to ~5 out of 10.
- Below we have visualized some of the picker's successes and mistakes along with the frames we would have chosen for those mistakes:



Conclusion

- Our model shows human levels of accuracy on the classification task. It may not be possible to go much higher than 81% accuracy since there are plenty of bad videos with good thumbnails and vice versa.
- One limitation is that our model is fitting to certain features common in the thumbnails of popular videos such as having text in the image. Text however is not indicative of a good thumbnail unless it says the right thing.
- Based on our metrics success on the classification task results in success on the frame selection task. However, comparing the model's choices to our human judgments makes the questionable assumption that we are capable of correctly selecting thumbnails.

Future Work

- Perform further hyperparameter turning.
- Experiment with more model architectures, including ResNet.
- It should be possible to make better thumbnail choices by incorporating tags and category information into the network.
- We should be able to prevent the network from overfitting to features like in-image text by performing data augmentation.
- We would like to incorporate text from YouTube titles and descriptions, but this is a challenging NLP problem since these pieces of text are in many languages and tend to include proper nouns and non-words.

References

[1] Yang, Weilong and Tsai, Min-husan. "Improving YouTube video thumbnails with deep neural nets." Google Research Blog, 2015.

[2] Liu, Wu, et al. "Multi-task deep visual-semantic embedding for video thumbnail selection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[3] randomyoutube.net generously provided us with video IDs for random YouTube videos scraped using their pseudorandom algorithm.

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.