CS230

Extracting "Poster-Ready" Images from Videos

Tam Dinh
Graduate School of Business
Stanford University
tamdn@stanford.edu

Nat Gardenswartz
Graduate School of Business
Stanford University
ngardens@stanford.edu

Katie Fo
Stanford University
kfo@stanford.edu

Abstract

The motivation for this project is to develop a mechanism for extracting high-quality static images to represent video content. For the purpose of this project, we define high-quality as containing an image of a person with a specified gender and emotion, with facial features fully in view.

1 Introduction

Media companies frequently need to select a single static image to represent video content for consumer marketing purposes. While companies could select these images manually, machine learning enables them to more efficiently at scale, and, in some cases, more effectively as well. //

The input to our algorithm is a color video. We then process this video by extracting a sequence of images from the video, reducing these images to a 100x100 pixel size, and using a convolutional neural network to identify which images in this sequence contain faces, and the gender and emotion of those faces. The B-IT-BOTS demo code that we use as backbone of this neural network comes with a training data set, which we use to train the model. We also train a different model using the "Fast and the Furious" movie trailer and a handful of modifications to the B-IT-BOTS demo code. These modifications are discussed in greater detail below.

2 Related work

Arriaga et al. describe the neural network architecture that we use in this project [2]. They explore two architectures premised on the idea of omitting fully connected layers from their architecture. One architecture is a CNN with 9 convolution layers, 5 ReLu layers, 7 batch normalization layers, global average pooling, and a softmax activation function. This architecture involves 600,000 parameters. The second architecture further reduces the number of parameters used in this application to 60,000 by adding depth-wise separable convolutions and residual modules. The first architecture yields 96 percent accuracy using the IMDB gender dataset and 66 percent accuracy using the FER-2013 emotion dataset. The second architecture yields nearly identical accuracy (95 percent for gender, and again 66 percent for emotion). Neither architecture uses an LSTM, which we have added to our architecture, and they use an ADAM optimizer, whereas we use RMSProp. Lastly, we have added a dropout factor to our architecture as well.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset and Features

To train our face emotion classifier, we use the FER2013 dataset from the Kaggle competition"Challenges in Representation Learning: Facial Expression Recognition Challenge". The data consists of 48x48 pixel gray-scale images of faces. The image has been preprocessed so that the face roughly occupy the center of the image and take the same amount of space in each image. The training data comes with 7 emotion labels: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral.The dataset of 35,887 examples is splitted 80/20 into training and validation set.

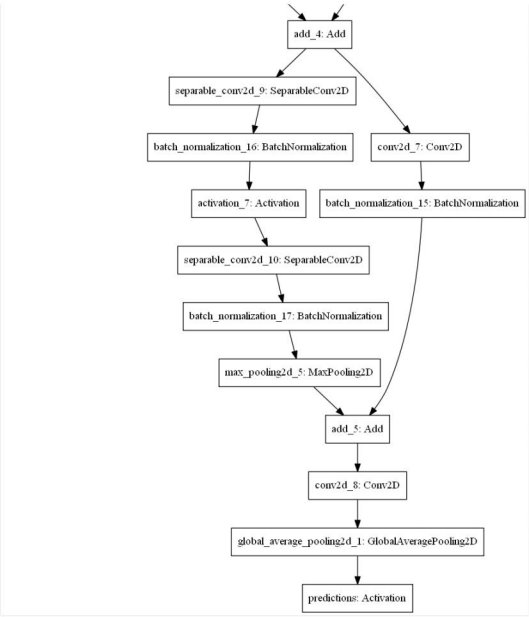
Next, we created a second dataset by using www.onlinevideoconvert.com to download a movie trailer from YouTube as an MP4 file. We then used www.filezigzag.com to convert this MP4 file into an ordered sequence of images. We manually label all images with a binary classification variable: "1" if the image contained a face with recognizable facial features, and "0" otherwise. We converted all images into 100x100 pixel to speed up our model's training

4 Methods

Based on github [2], We use the following Keras layers in our basic CNN architecture:

- *2D Convolutional*: convolves the input with a kernel of size n*n. We use kernel of size 3x3 for the first layer and size 1x1 for the residual layer.[3]
- *Separable 2D Convolutional*: first performs a depthwise spatial convolution (which acts on each input channel separately) then performs a pointwise convolution which mixes together the resulting output channels.[6]
- *Batch Normalization*: normalize the activations of the previous layer at each batch by applying a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.
- *Rectified Linear Activation*: return x if $x > 0$ and 0 if $x < 0$.
- *Max 2D Pooling*: is used with pool size of 3x3, stride of 2x2 and same padding.
- *Residual Layer*: the residual layer copies the identity map from the previous module so that the next module only learns the difference between output of the previous module and the desired output [5]
- *Global Average Pooling 2D*: is used just before the softmax activation layer.
- *Softmax Activation*: is used to produce class probabilities.
- *Categorical Cross Entropy Loss*: to measure the loss of multiple classes classification.

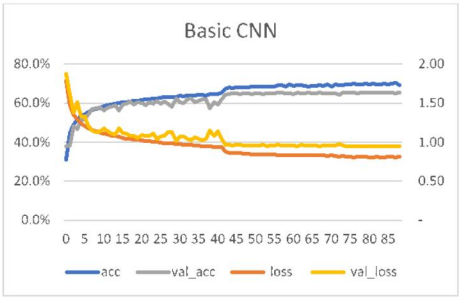
- *Adam Optimizer*: adapt the parameter learning rates based on the average first moment and the average of the second moments of the gradients.

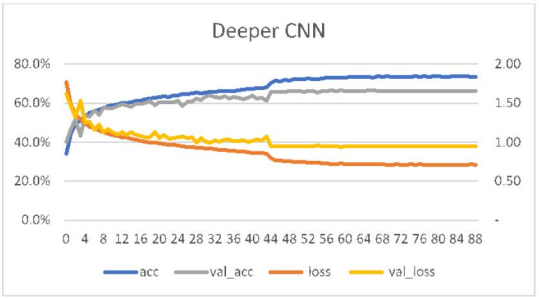


We then consider the following improvements to our basic CNN architecture:

- Deeper CNN: will allow us to reduce the bias and improve the accurary of the model.
- Dropout: Mitigates overfitting and reduces variance of results. We hypothesize that this is important given that our data set has not been vetted to ensure a diversity of image categories and attributes. Implementing a dropout may also help prevent false positives.
- RMSprop: to compare the benefits of different adaptive learning rates.
- LSTM single layer: Though LSTM is not frequently used for image processing because it is primarily used for sequential data, it could be useful for face classification of images from videos given that we present these images in the original sequence of frames. That is, temporal memory can account for faces present over time.
- LSTM stateful: Batches are appended to the end of the previous batch, making this configuration potentially useful for longer-duration, contiguous video data.

5 Experiments/Results/Discussion





The charts above demonstrate how the accuracy and cost of the base face-recognition CNN compares to the modified CNN, which includes a dropout layer and the RMSprop optimizer, both of which contributed to the success of the model in identifying It is likely that the results of this project are limited by the fact that the model is trained on still-image data rather than on video data. However, it still identifies a set of frames from a video input from which the user can identify the qualities of "poster-ready" images that display the most visible faces.

6 Conclusion/Future Work

In conclusion, we have modified the existing face-recognition CNN to demonstrate increased accuracy in identifying the presence, emotion, and gender of faces found in video data, Given more time,

we would implement LSTM layers in order to find additional temporal patterns in the video data that we have tested in this project. The use of LSTM layers is promising for the purpose of finding "poster-ready" images because of the possibility of finding patterns in the "clusters" of images that include one or more faces, given that LSTM layers treat the frames of a video as sequential data.

7 Contributions

Each team member made different and significant contributions to this project. Katie primarily worked on exploring modifications to the model and the benefits of additional layer types, especially LSTM layers. Tam set up the AWS system and handled other logistics of training and testing each model. Nat contributed the initial idea for this project and much of the background research and data presentation.

References

[1] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

[2] Arriaga, O. & Plöger, P. &Valdenegro, M. (2017) Real-time Convolutional Neural Networks for Emotion and Gender Classification. Published on Github

[3] https://keras.io/layers/convolutional/

[4] Kaiming He & Xiangyu Zhang & Shaoqing Ren & Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016

[5] Kaiming He & Xiangyu Zhang & Shaoqing Ren & Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016

[6] Andrew G. Howard et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017

[7]https://www.linkedin.com/pulse/sentiment-analysis-using-recurrent-neural-networklstm-subramanian