Selecting "Poster-Ready Images" from Videos

Context: Media companies frequently need to select a single static image to represent video content for consumer marketing purposes

Example image:

Objective: Leverage neural networks to identify suitable static images for representation of video content.

Unsuitable images | **Suitable image**

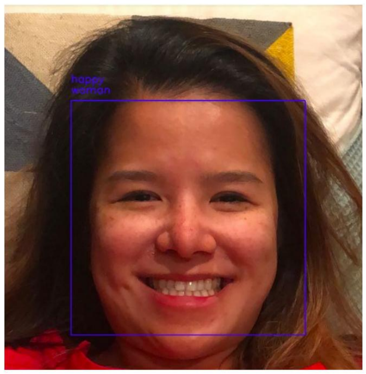
No humans, no plot content

Facial features unrecognizable

Face detectable

Image Selection Criteria: Using demo code from the B-IT-BOTS team[1], we can select images that contain human faces and classify those faces by gender and emotion.

Positive prediction: face, gender, and emotion identified



Negative prediction: no face identified

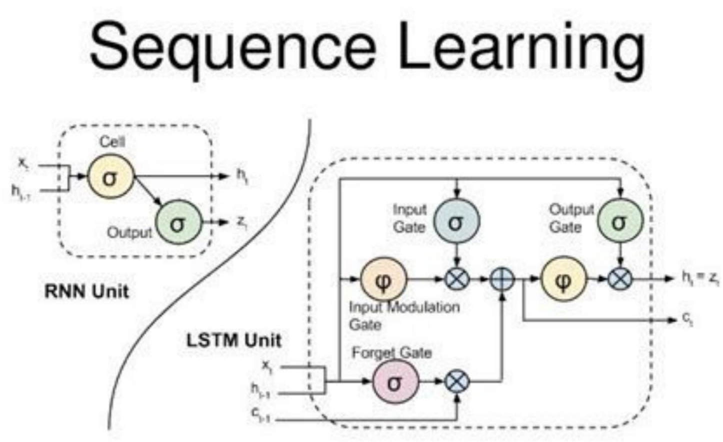
1. From Github user Oarriaga (https://github.com/oarriaga/face_classification)

Architecture modification and hyperparameter tuning:

Modification	Potential advantage
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.
Change optimizer to RMSprop	Possible optimization benefits
Dropout	Potential reduction in variance

Implementing LSTM for Video Data

- Enhances recognition of long-term patterns by appending each batch to the end of the previous batch, allowing for patterns to be recognized across an entire video clip.
- Because images are ordered sequentially, facial classification results in one image are correlated with sequential images



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

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.

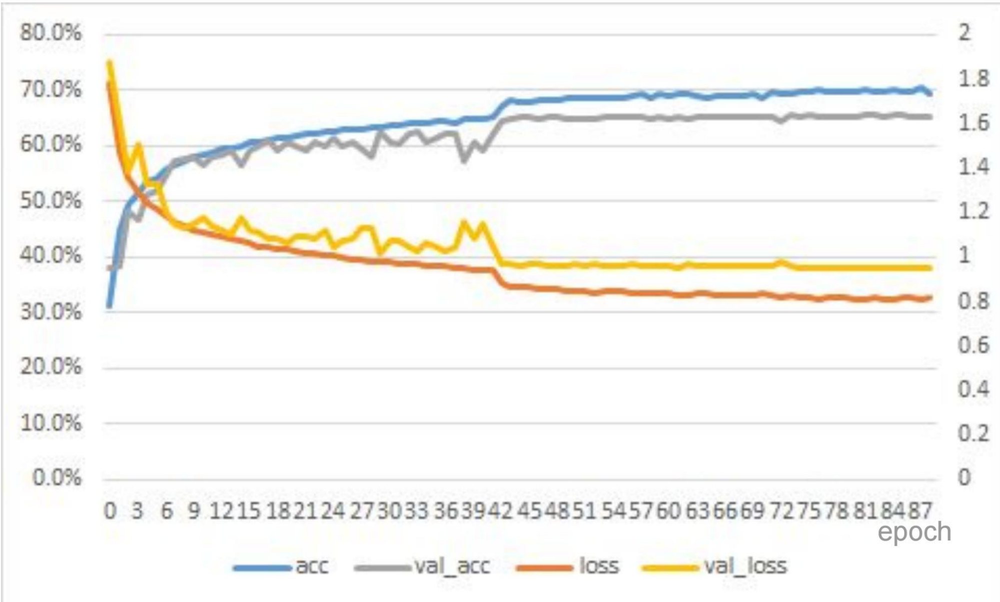
RMSprop

Source code uses an ADAM optimizer. We are experimenting with an RMSprop optimizer in order to compare the benefits of different adaptive learning rates.

Image and Data Preprocessing:

- We used 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

Results of B-IT-BOTS source code on Fer2013 data



Note: The Fer2013 data set contains 48x48 pixel grayscale images of faces. The Fer2013 data set contains 28,709 images and there are two test sets, each containing 3,589 images.

Additional results to be obtained

- Results of modified neural network on FER2013 data
- Results of “BI-IT-BOTS” source code on “Fast and Furious” trailer images
- Results of modified neural network on “Fast and Furious” trailer images