Adrian Marinovich
Springboard Data Science Career Track
Capstone Project #2

Classification of human actions in videos

Milestone Report Slide Set
February 14, 2018

Mentor: Hobson Lane

The problem

Classify human actions as seen in videos.

Why is this interesting?

Potential applications:

- Development of instructional tools for robots
- Accident prevention in industrial sites, pools, recreational facilities
- Monitoring of secure locations
- Predict intentions and possible outcomes of human actions for robots / self-driving cars safety

<u>Description of data</u>

UCF101 Action Recognition Data Set:

- 13320 videos collected from YouTube
  - Diverse set of realistic examples in varied settings
- 101 action categories, of 5 types:
  - human-object interaction
  - body-motion only
  - human-human interaction
  - playing musical instruments
  - sports
- 25 unique groups of videos per action
  - Each group consisting of 4-7 video clips segmented from larger video
- Frame rate: 25
- Length range: 1.06 to 71.04 seconds (mean: 7.21 seconds)
- 240 x 320 pixel dimensions, x 3 colors

Data wrangling

Data split into train and test sets using pre-made lists prepared by the UFC research group

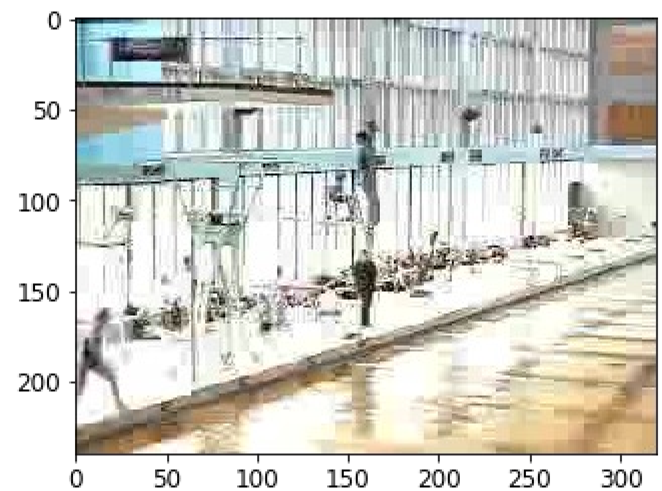
- Designed so groups of video clips obtained from the same original video not split between sets
- Train set: 9537 videos
- Test set: 3783 videos

Extracted frame-by-frame JPEG still image file sequences from each video using FFmpeg

Example images

Biking

Diving



<u>Preliminary analysis</u>

Video classification suing single-stream methods:

- Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) recurrent neural network (RNN) models
- Fed 2048 features extracted from the image sequences using a pretrained convolutional neural network (CNN) InceptionV3 model
- LSTM and GRU models require fixed sequence lengths
    - Downsampling and interpolation of frames to get sequence length desired
- Range of sequence lengths tested for effect on model accuracy
- Range of model architectures and batch sizes tested

Machine learning was performed using Python, primarily in Keras with a TensorFlow backend, on an NVIDIA GTX 1060 GPU with 6 GB memory.

Initial findings from exploratory analysis

LSTM and GRU models gave top-1 validation accuracies between 64% and 74%

- GPU memory constraints limited development of LSTM models
- GRU models used for most hyperparameter tuning

Current highest top-1 validation accuracy: 75%

- Attained using model with 3 GRU layers and a dense layer
- 50-frame sequence length and batch size of 32.

Batch size tuning using the 50-frame sequence reveals improvement in validation accuracy with smaller batch sizes

- This tuning is ongoing for larger sequence length models

Model descriptions:
('seq' = sequence length)

GRU-3 - *top performer*:

Layer (type)	Output Shape	Param #
gru_1 (GRU)	(None, seq, 2048)	25171968
gru_2 (GRU)	(None, seq, 2048)	25171968
gru_3 (GRU)	(None, 2048)	25171968
dense_1 (Dense)	(None, 512)	1049088
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 101)	51813

Total params: 76,616,805

GRU-2:

Layer (type)	Output Shape	Param #
gru_1 (GRU)	(None, seq, 2048)	25171968
gru_2 (GRU)	(None, 4096)	75509760
dense_1 (Dense)	(None, 512)	2097664
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 101)	51813

Total params: 102,831,205

Model descriptions (continued):

GRU-4:

Layer (type)	Output Shape	Param #
gru_1 (GRU)	(None, seq, 2048)	25171968
gru_2 (GRU)	(None, seq, 2048)	25171968
gru_3 (GRU)	(None, seq, 1024)	9440256
gru_4 (GRU)	(None, 1024)	6294528
dense_1 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 101)	51813

Total params: 66,655,333

LSTM-3:

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, seq, 2048)	33562624
lstm_2 (LSTM)	(None, seq, 1024)	12587008
lstm_3 (LSTM)	(None, 1024)	8392704
dense_1 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 101)	51813

Total params: 55,118,949

Table 1. Hyperparameter tuning results.

Model	Sequence length (frames)	Batch size	Number of epochs until validation loss stopped improving	Validation accuracy	Validation loss
GRU-3	50	8	13	0.750	1.027
GRU-3	50	16	10	0.731	0.993
GRU-3	50	32	10	0.696	1.120
GRU-3	50	120	26	0.700	1.090
GRU-3	80	8	13	0.741	1.032
GRU-3	80	32	26	0.741	0.966
GRU-2	80	32	12	0.715	1.052
GRU-4	80	32	20	0.702	1.068
LSTM-3	80	32	14	0.637	1.359
GRU-3	100	32	17	0.709	1.109

Future directions

Additional GRU modeling to determine if the smaller batch sizes also improve performance in models using larger sequence lengths.

Once tuning complete, further items to be produced:

- Machine-readable tuning table
- Tuning plot
- Feeding the top-performing model with variable sequence lengths from original video image sets

Limitations of the above overall RNN approach:

- fixed sequence lengths
- single stream training
- lack of readily available visualization tools for LSTM/GRU models

Further modeling approach to be considered:

- implement two-stream temporal-spatial neural network models
- using video sequences of variable lengths
- end-to-end training framework that containing convolutional layers that can be visualized

Project code

https://github.com/adriatic13/springboard/tree/master/dsct_capstone2

References

https://crcv.ucf.edu/data/UCF101.php

https://github.com/fchollet/deep-learning-with-python-notebooks

https://github.com/harvitronix/five-video-classification-methods