Readme

Introduction:

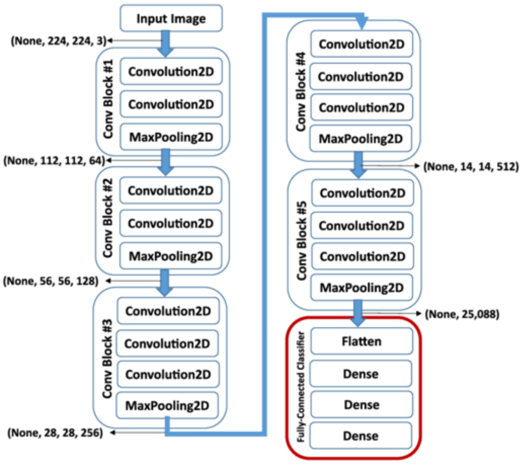
In Q6, to do short video classification, when we take a few frames from the picture, we realize that the differences between frames are little, so we decide to treat this question as the image classification. And we use the VGG as our feature extractor. But to put the time factor into consideration, we choose LSTM to do classification. And detailed information are listed below.

Methods:

1. Data preprocessing and feature engineering.

- Put the train videos into the classifier folder named as 'label' and get 15 folders to train.
- Extract some images for each video by using cv2.VideoCapture(video);
- Convert the images to array by using img_to_array(img);
- Expand the dims and save them as a ".npy" file for each video in separate 'label' folder. If Video Capture failed, skip this video;
- VGG feature engineering: Set the parameter: vgg16_include_top = True, and extract all the layers as features except for the last layer; Optimizer=SGD；

For each group of images, we use VGG16 to extract features and get features of each video.



2. Train LSTM model

- Create LSTM Sequential model to train the extracted features of each image;
- Fit the LSTM model with the trained features and labels, epoch size=21;
- Save the model files for prediction.

3. Predict the test video

Load the pretrained-H5 file and predict the videos.

References:

https://github.com/chen0040/keras-video-classifier