Emotion And Action Recognition

Ziqi Zhao 727003114
(nickname: astrajoan)

Abstract

Video action recognition is one of the representative tasks for video understanding. Models based on deep convolutional networks have dominated recent image interpretation tasks. In this report, I'll mainly detect 6 actions. The conventions in this paper involve pretrained model, data augmentation. The size of the collected dataset for recognition task this final submission is 2598. To view the clips of earlier submission, please go to the links provided for each repository. The test accuracy achieved are $0.88, 0.92, 0.92, 0.92, 0.93, 0.92$.

1 Topic

In this report, the main emotions and actions to be detected include waking up, house work on clothes shoes hats etc, bandaging, cut polish or do nails, sneezing, arrange flowers or water plants etc. The following report will be divided into these topics respectively for discussion.

1.1 Waking up

In this submission, the target action worked on is 'waking up'.

1.2 House work on clothes shoes hats etc

In this submission, the target action worked on is 'house work on clothes shoes hats etc'.

1.3 Bandaging

In this submission, the target action worked on is 'bandaging'.

1.4 Cut polish or do nails

In this submission, the target action worked on is 'cut polish or do nails'.

1.5 Sneezing

In this submission, the target action worked on is 'sneezing'.

1.6 Arrange flowers or water plants etc

In this submission, the target action worked on is 'arrange flowers or water plants etc'.

2 Motivation

Some overview of motivation for action recognition. Human action recognition has a wide range of applications, such as intelligent video surveillance and environmental home monitoring, intelligent human–machine interfaces, and identity recognition. Human action recognition covers many research topics in computer vision, including human detection in video, human pose estimation, human tracking, and analysis and understanding of time series data. It is also challenging in that many key problems in human action recognition that remain unsolved. The key to high performance action recognition is robust human action modeling and feature representation. Unlike feature representation in an image space, the feature representation of human action in video not only describes the appearance of the human(s) in the image space, but must also extract changes in appearance and pose.

An important aspect of the status on human action recognition research is that most studies concentrated on human action feature representations. The processed image sequence data are usually well-segmented and contain only one action event, becomes classification. However, there are two key problems in the actual scenario: interaction recognition and action detection. Interaction refers to actions that involve more than two people or actions between people and objects, such as carrying a knife or playing an instrument. Action detection refers to locating the position at which an action occurs in time and space from image sequence data that have not been segmented. There is a lot of research over these topics, however there is no overarching summary of the methods applicable to these two issues.

2.1 Waking up

The action of waking up can mean a lot of things for many age groups. For baby, it's a sign to alert the adult that the baby may need some care. For adults, it may mean going to work or starting to make breakfast. Sometimes, waking up at unusual hours may mean the person has some sort of mental distress or physical unwell, which requires medical attention. It may also help alert working people to wake up in time if they did not wake at the required time. The smart camera that can detect this will be able to direct the house to react.

2.2 House work on clothes shoes hats etc

This submission is mainly on cleaning and renovating shoes. This process can contain a lot of procedures and can be creative. There may be some special care or cleaning tools or solutions that work for certain type of shoe or dirt, which the camera can detect and help guide the work and creativity. A camera that can detect this can provide inspirations to the owner who is working on the shoes and create better shoes.

2.3 Bandaging

When humans are hurt, they may require bandaging to prevent further infection. Bandaging can be applied to many parts of the human body and to learn which part is being treated can help guide further steps or start an alert to new bandaging after a time that maybe assigned by a doctor. A camera that can detect the action bandaging can be equipped to release warnings to the human for possible medical attention or instruction on how to bandage the hurt human.

2.4 Cut polish or do nails

When cutting nails, sometimes if the person is not focusing, they can cut their fingers, so it would be nice to be alerted to focus on the nails when it is detected that the person may cut himself. Polishing nails usually involve a lot of chemicals and terrible smells. When this action happens, if it is detected, the person can be reminded to open windows to allow air from outside to enter. For doing nails, when trimming and polishing the nails to be more fit to apply fake nails or gel, there can be dusts from the process. It would be nice to warn people nearby who have allergies or sinus conditions to stay away to prevent any discomfort. A camera that can detect the actions cutting and doing nails can alert people to focus when cutting nails, and guide to open windows to reduce chemical harm and to warn people of conditions to stay further to avoid discomfort.

2.5 Sneezing

When humans are sneezing, it can be caused by many reasons. Some human catching a cold or being sick cause them to sneeze. Some people have slight or severe allergies which cause them to sneeze. Bad air condition from pollution will make people sneeze. Some situations there are dust or chemicals that make normal people sneeze. A camera that can detect a human sneezing can run diagnosis to determine the reason and let the human know if it requires medical attention or what action is required to reduce any risk, or if it requires immediate attention to act in a specific way, such as open windows, or it is harmless.

2.6 Arrange flowers or water plants etc

When arranging flowers or watering plants, there sometimes will be a lot of dust and pollen. For humans that have severe or life threatening allergies, sometimes some flowers or certain type of plants may trigger their conditions. This situations can be dangerous for those humans if not taken actions before working on those flowers or plants, so if medical information of those humans are recorded and matched with the action, it maybe able to save lives in certain situations, especially if the person doesn't know that the plant poses danger to their condition. A camera that can detect this can alert the human and raise warnings if necessary to take precautions or medicines before working on the flower or plant, especially if the type of flower or plant is in the category that can severely affects the human.

3 Related Works

Action recognition task involves the identification of different actions from video clips (a sequence of 2D frames) where the action may or may not be performed throughout the entire duration of the video. This seems like a natural extension of image classification tasks to multiple frames and then aggregating the predictions from each frame.

Before deep learning came along, most of the traditional CV algorithm variants for action recognition can be broken down into 3 broad steps. Local high-dimensional visual features that describe a region of the video are extracted either densely [1] or at a sparse set of interest points [2]. The extracted features get combined into a fixed-sized video level description. One popular variant to the step is to bag of visual words (derived using hierarchical or k-means clustering) for encoding features at video-level. A classifier, like SVM or RF, is trained on bag of visual words for final prediction.

The problem of action recognition in videos can vary widely and there's no single approach that suits all the problem statements. Traditional approaches to action recognition rely on object detection, pose detection, dense trajectories, or structural information. Convolutional Neural Networks(CNN) extracts the features from each frame and pool the features from multiple frames to get a video-level prediction. The drawback of this approach is that it fails to capture sufficient motion information. Motion information can be captured by combining optical flow containing short-term motion. In addition to RGB and optical flow, information from other modalities such as audio, pose, and trajectory. Constructing a spatiotemporal representation by fusion motion and appearance information in the way of two streams would work well for short duration clips, but would not be able to capture long-term temporal dynamics. The two-stream network consists of two separate subnetworks, where one is for raw images and the other is for stacked optical flow, respectively, and captures spatiotemporal information by fusing the softmax scores of two streams. Recurrent neural networks (RNNs), especially long short-term memory (LSTM), achieved impressive results in the sequence tasks due to the ability of long-term temporal modeling, so an alternative strategy is to adopt LSTM to model dynamics of frame-level features. However, most existing LSTM-based approaches do not make the distinction between various parts of video frames. More recent architectures have focussed on using attention mechanisms for picking salient parts of the video. This helps in overcoming the limitation of LSTMs which didn't distinguish between various parts of the video.

3D convolutional networks as feature extractors uses 3D convolutions on video frames(where convolution is applied on a spatiotemporal cube) [3]. Their finding was a simple linear classifier like SVM on top of an ensemble of extracted features worked better than the state-of-the-art algorithms. The network focussed on spatial appearance in the first few frames and tracked the motion in the subsequent frames. But long-range temporal modeling is a problem and training such huge networks is computationally a problem in their work. Another approach is to model videos by combining dense sampling with feature tracking [4]. They introduce an efficient solution to remove camera motion by computing the motion boundaries descriptors along the dense trajectories. Local descriptors computed in a 3D video volume around interest points have become a popular way for video representation. To leverage the motion information in dense trajectories, they compute descriptors within a space-time volume around the trajectory. They reduced the problem of trajectories tend to drift from their initial location during tracking. A new spatio-temporal interest point detector was introduced and analyzes various cuboid descriptors to conclude that cuboid prototyping(using K-means clustering) is a good behavior descriptor [5]. The idea of incorporating information across longer video sequences was explored and introduced feature pooling method that processes each frame independently and uses max-pooling on local information to combine frame-level information [6]. They demonstrates the usage of an RNN that uses LSTM cells which are connected to the output of the underlying CNN and validates the effectiveness of using Optical flow for motion information. LRCN(Long term Recurrent Convolutional Networks) which combines convolutional layers with long-range temporal recursion was proposed but had the problem of a single prediction for the entire video [7]. Soft attention-based model was proposed for action recognition [8]. The model learns to focus selectively on the important parts of the video. Initially, the model takes a video frame as input and produces a feature cube. At each time step, the model predicts a softmax over K $\times$ K location(l_t+1) and a softmax over the label classes (y_t). L_t is the probability with which the model believes the corresponding region in the input frame is important.

Action recognition has come a long way in part 5–6 years after the advent of neural networks. Human action recognition is still a very active research area and new approaches are still trying to solve the issues with the current approaches. Some of the existing issues are background clutter or fast irregular motion in videos, occlusion, viewpoint changes, high computational complexity,

and responsiveness to illumination changes

4 Proposed Model

First, introduce some techniques and networks used in this report.

The Xception pretrained network [9]. With a modified depthwise separable convolution, it is even better than Inception-v3 for both ImageNet ILSVRC and JFT datasets. The modified depthwise separable convolution is the pointwise convolution followed by a depthwise convolution. This modification is motivated by the inception module in Inception-v3 that 1×1 convolution is done first before any n×n spatial convolutions. In Xception, the modified depthwise separable convolution, there is NO intermediate ReLU non-linearity. The Xception without any intermediate activation has the highest accuracy compared with the ones using either ELU or ReLU. The Modified Depthwise Separable Convolution and overall architecture of Xception are shown in Fig. 1 and 2.

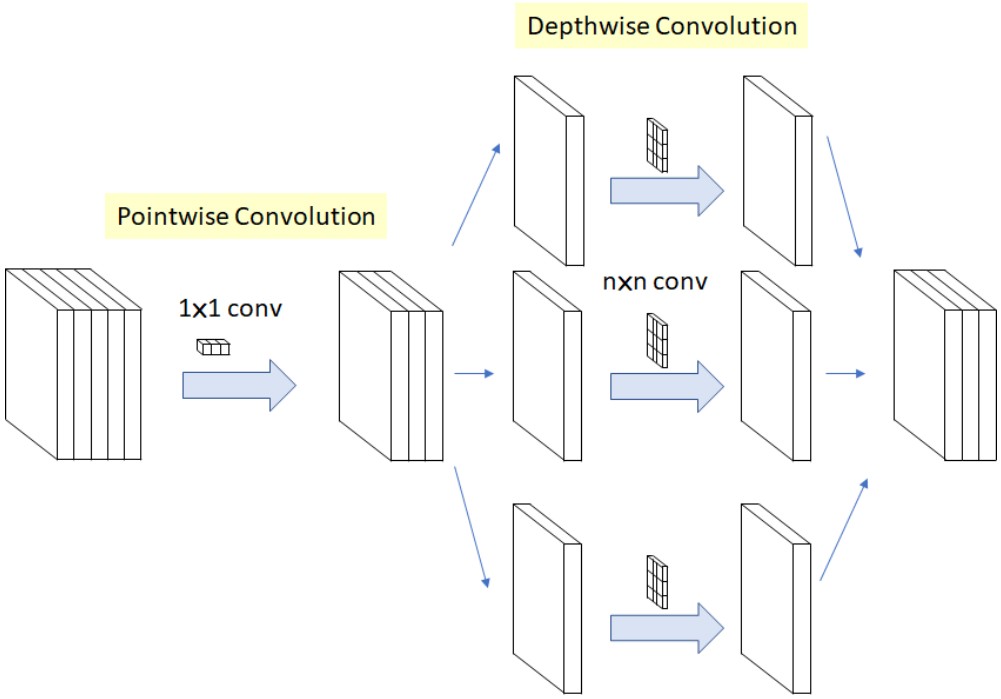

Figure 1: Depthwise Separable Convolution Xception

The Inception V3 pretrained network [10]. The authors noted that the auxiliary classifiers didn't contribute much until near the end of the training process, when accuracies were nearing saturation. They argued that they function as regularizes, especially if they have BatchNorm or Dropout operations. Inception Net v3 incorporated RMSProp Optimizer, factorized 7×7 convolutions, BatchNorm in the Auxillary Classifiers, and label smoothing, which is a type of regularizing component added to the loss formula that prevents the network from becoming too confident about a class, prevents over fitting. Computational efficiency and fewer parameters are realized. With fewer parameters, 42-layer deep learning network, with similar complexity as VGGNet, can be achieved.

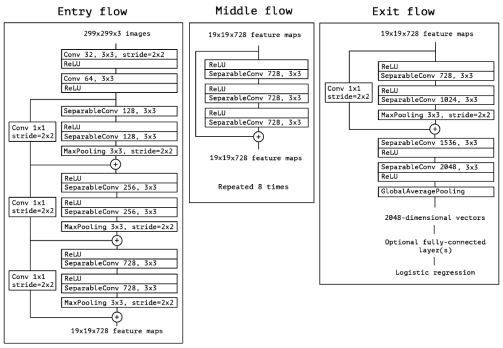

Figure 2: Overall architecture of Xception

The aim of factorizing Convolutions is to reduce the number of connections/parameters without decreasing the network efficiency. Auxiliary Classifiers were already suggested in Inception-v1 [11]. There are some modifications in Inception-v3. Only 1 auxiliary classifier is used on the top of the last 17×17 layer, instead of using 2 auxiliary classifiers. Here, an efficient grid size reduction is proposed. With the efficient grid size reduction, 320 feature maps are done by conv with stride 2. 320 feature maps are obtained by max pooling. And these 2 sets of feature maps are concatenated as 640 feature maps and go to the next level of inception module. Less expensive and still efficient network is achieved. The downsampling example is shown in Fig. 3. The architecture of the Inception-V3 is shown in Fig. 4.

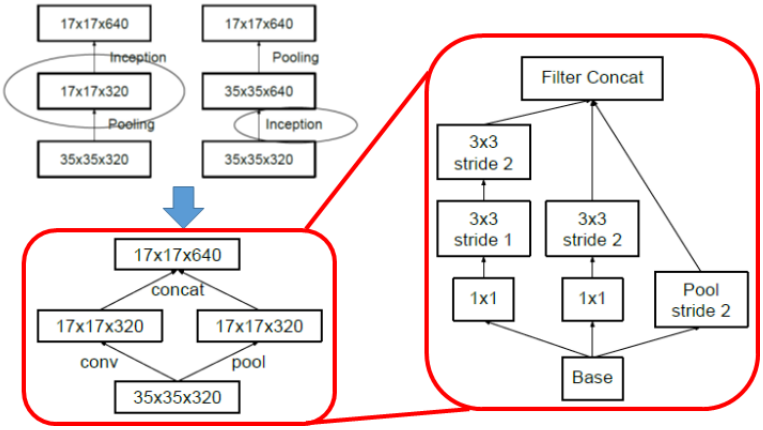

Figure 3: Conventional downsizing (Top Left), Efficient Grid Size Reduction (Bottom Left), Detailed Architecture of Efficient Grid Size Reduction (Right)

Batch Norm [12]. As the distribution of the weights of the network varies much less with this layer (this called internal covariate shift in the paper) we can use higher learning rates. The direction in which we are heading during training is less erratic allowing us to move faster on the direction of the loss. Even though the network will see the same examples on each epoch, the normalization of each mini-batch is different, thus changing the values slightly each time. This improves regularization. It also improves accuracy. BatchNorm ensures that the received input have mean 0 and a standard deviation of 1. On training we keep track of an exponential moving average of the mean and the variance, for later use during inference. The reason for this

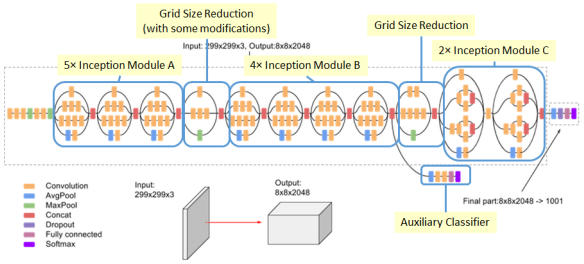

Figure 4: Inception-v3 Architecture. Batch Norm and ReLU are used after Conv

that we can obtain a much better estimation of mean and variance of the input over time while processing the batches during training, then use it on inference. Usually it appears between a fully connected layer / conv layer and an activation layer.

VideoFrameGenerator installed in the Keras video generator was used to preprocess the videos into frames and that are used as input to the network.

4.1 Waking up

The model uses the pretrained Xception network weights in the feature extraction CNN, followed by an LSTM layer and a densely connected classifier on top. The architecture is shown in Fig. 5. The input shape is (15, 112, 112, 3). Shape of the output tensor is (2,), because it's a binary classification.

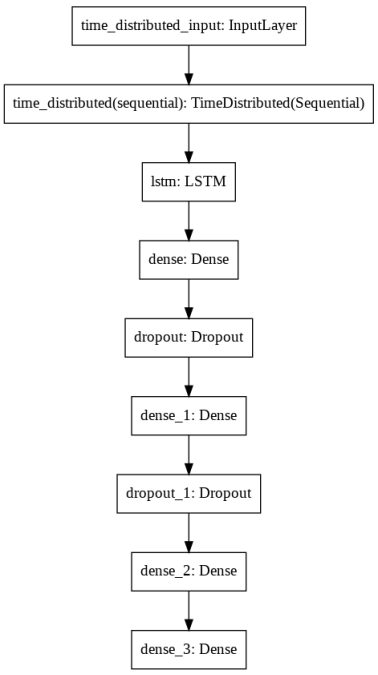

Figure 5: Model

4.2 House work on clothes shoes hats etc

The model uses the pretrained Xception network weights in the feature extraction CNN, followed by an LSTM layer and a densely connected classifier on top. The architecture is shown in Fig. 6. The input shape is (25, 112, 112, 3). Shape of the output tensor is (2,), because it's a binary classification.

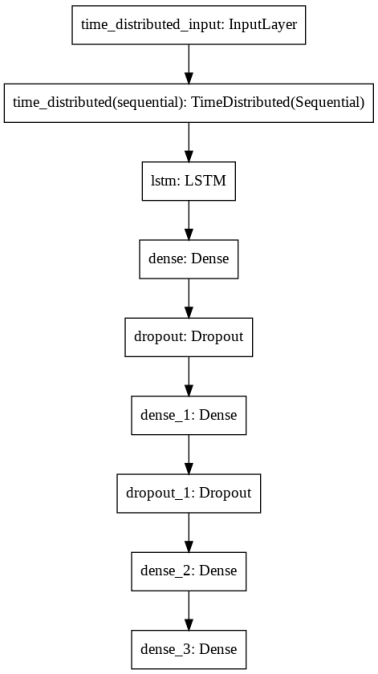

Figure 6: Model

4.3 Bandaging

The model uses the pretrained InceptionV3 network weights in the feature extraction CNN, followed by an LSTM layer and a densely connected classifier on top. The architecture is shown in Fig. 6. The input shape is (25, 112, 112, 3). Shape of the output tensor is (2,), because it's a binary classification.

4.4 Cut polish or do nails

The model uses the pretrained InceptionV3 network weights in the feature extraction CNN, followed by an LSTM layer and a densely connected classifier on top. The architecture is shown in Fig. 7. The input shape is (25, 112, 112, 3). Shape of the output tensor is (2,), because it's a binary classification.

4.5 Sneezing

The model uses the pretrained InceptionV3 network weights in the feature extraction CNN, followed by an LSTM layer and a densely connected classifier on top. The architecture is shown

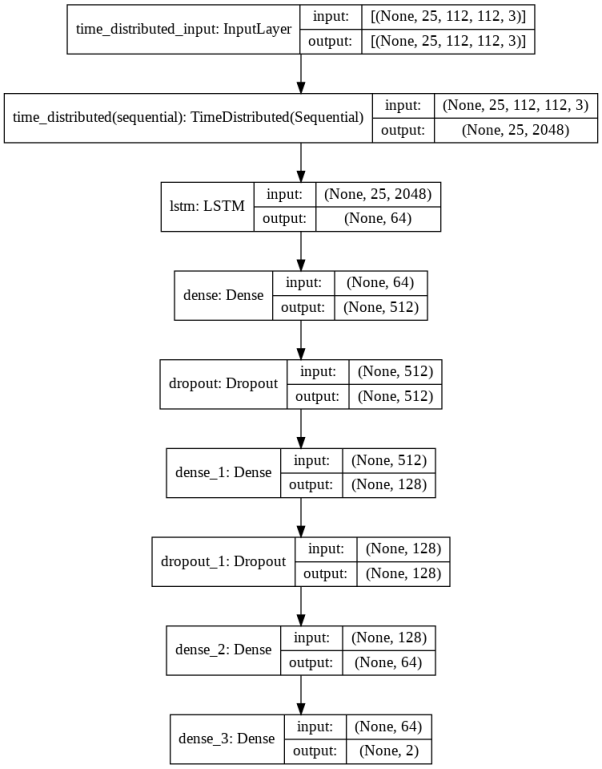

Figure 7: Model

in Fig. 7. The input shape is (25, 112, 112, 3). Shape of the output tensor is (2,), because it's a binary classification.

4.6 Arrange flowers or water plants etc

The model uses the pretrained InceptionV3 network weights in the feature extraction CNN, followed by an LSTM layer and a densely connected classifier on top. The architecture is shown in Fig. 7. The input shape is (25, 112, 112, 3). Shape of the output tensor is (2,), because it's a binary classification.

5 Dataset

5.1 Waking up, House work on clothes shoes hats etc, Bandaging, Cut polish or do nails, Sneezing, Arrange flowers or water plants etc

The dataset I used is Kinetics. The dataset consists of 543,214 training samples, 34,066 validation samples and 67,485 test samples, where test samples are without labels. There are 700 classes in total, and each sample is provided in the form of 10 second video.

6 Model Training and Performance

There is certain overall reason why video action recognition can be difficult. Action recognition involves capturing spatiotemporal context across frames. Additionally, the spatial information captured has to be compensated for camera movement. Even having strong spatial object detection doesn't suffice as the motion information also carries finer details.

6.1 Waking up

1. Hyperparameters: except for the Xception module, the model has one LSTM layer, and three densely connected layers. The densely connected layers were experimented with different combinations and the best performance and more efficient layer combination was used in the model. The optimizer used was Adam with a learning rate of 10^{-4}. 15 frames were sectioned from each video.

2. The model was trained 30 epochs. The model began to overfit at $\sim$24 epochs.

3. Highest training accuracy: 0.892. Highest validation accuracy: 0.9091. Test accuracy: 0.88.

4. In my opinion, the model is subject to further improvement. Although pretrained weights were used from Xception, the architecture of the model is rather simple and may not be able to capture all the motion information.

6.2 House work on clothes shoes hats etc

1. Hyperparameters: except for the Xception module, the model has one LSTM layer, and three densely connected layers. The densely connected layers were experimented with different combinations and the best performance and more efficient layer combination was used in the model. The optimizer used was Adam with a learning rate of 10^{-4}. 25 frames were sectioned from each video.

2. The model was trained 30 epochs. The model began to overfit at $\sim$26 epochs.

3. Highest training accuracy: 0.8872. Highest validation accuracy: 0.9286. Test accuracy: 0.92. Learning curve is shown in Fig. 8.

4. In my opinion, the model is subject to further improvement. Although pretrained weights were used from Xception, the architecture of the model is rather simple and may not be able to capture all the motion information. The frames each video were increased from 15 to 25, as the accuracy was not improving much at 15 when tuning the hyperparameters. After increasing the frames to 25, the accuracy improved at least 1%.

6.3 Bandaging

1. Hyperparameters: except for the InceptionV3 module, the model has one LSTM layer, and three densely connected layers. The optimizer used was Adam with a learning rate of 10^{-4}. A smaller learning rate of 10^{-5} was tried but the performance was worse, so learning rate kept at 10^{-4}. 25 frames were sectioned from each video.

2. The model was trained 30 epochs. The model began to overfit at $\sim$17 epochs.

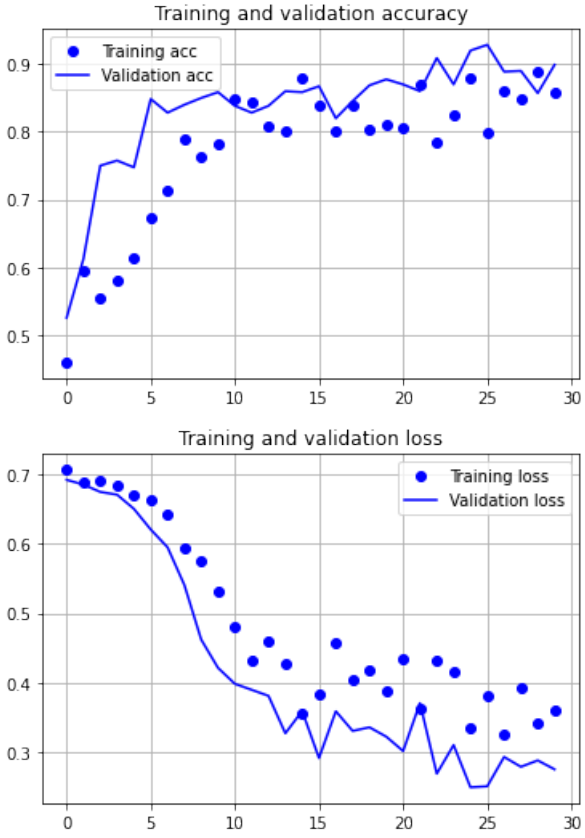

Figure 8: Accuracy and loss curve

3. Highest training accuracy: 0.9296. Highest validation accuracy: 0.96. Test accuracy: 0.92.

4. In my opinion, unlike previous models which used the Xception module, this time InceptionV3 module is used, which provided higher accuracy. The accuracy when training this model using Xception pretrained weights was also tried and was increased 3% after adopting InceptionV3 pretrained weights. However, although pretrained weights were used, the architecture of the model is rather simple and may not be able to capture all the motion information.

6.4 Cut polish or do nails

1. Hyperparameters: except for the InceptionV3 module, the model has one LSTM layer, and four densely connected layers. The optimizer used was Adam with a learning rate of 10^{-4}. 25 frames were sectioned from each video.

2. The model was trained 30 epochs. The model began to overfit at $\sim$22 epochs.

3. Highest training accuracy: 0.9354. Highest validation accuracy: 0.97. Test accuracy: 0.92. Learning curve is shown in Fig. 9.

4. In my opinion, unlike more previous models which used the Xception module, this time InceptionV3 module is used, which provided higher accuracy. The accuracy when training this model using Xception pretrained weights was also tried and was increased 3% after adopting InceptionV3 pretrained weights. However, although pretrained weights were used, the architecture of the model is rather simple and may not be able to capture all the motion information.

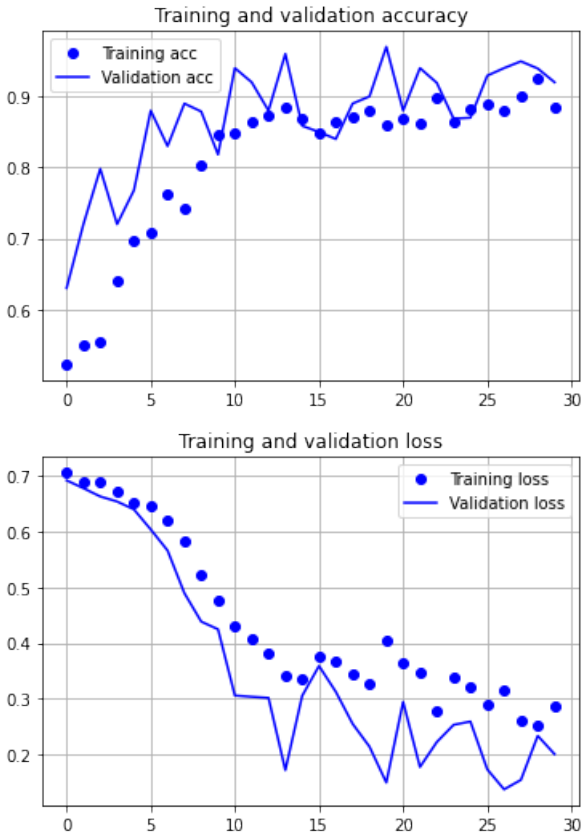

Figure 9: Accuracy and loss curve

6.5 Sneezing

1. Hyperparameters: except for the InceptionV3 module, the model has one LSTM layer, and four densely connected layers. The optimizer used was Adam with a learning rate of 10^{-4}. 25 frames were sectioned from each video.

2. The model was trained 60 epochs. The model began to overfit at ∼40 epochs.

3. Highest training accuracy: 0.9598. Highest validation accuracy: 0.97. Test accuracy: 0.93. Learning curve is shown in Fig. 10.

4. In my opinion, this time the model was trained for a longer time, which provided higher accuracy. The accuracy when training this model using 30 epochs was also tried and was

increased 4% after using the best model by training for 60 epochs. However, although pretrained weights were used, the architecture of the model is rather simple and may not be able to capture all the motion information.

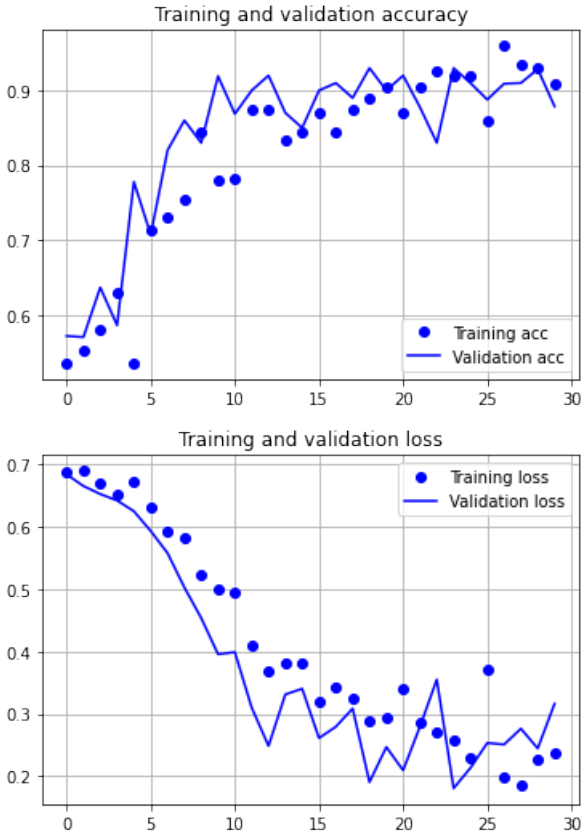

Figure 10: Accuracy and loss curve

6.6 Arrange flowers or water plants etc

1. Hyperparameters: except for the InceptionV3 module, the model has one LSTM layer, and four densely connected layers. The optimizer used was Adam with a learning rate of 10^{-4}. 25 frames were sectioned from each video.
2. The model was trained 30 epochs. The model began to overfit at ∼17 epochs.
3. Highest training accuracy: 0.9410. Highest validation accuracy: 0.9495. Test accuracy: 0.92. Learning curve is shown in Fig. 11.
4. In my opinion, although pretrained weights were used, the architecture of the model is rather simple and may not be able to capture all the motion information. Reduce learning rate on plateau was used in callback with different modification factors. It was kept at original for best performance and efficiency balance.

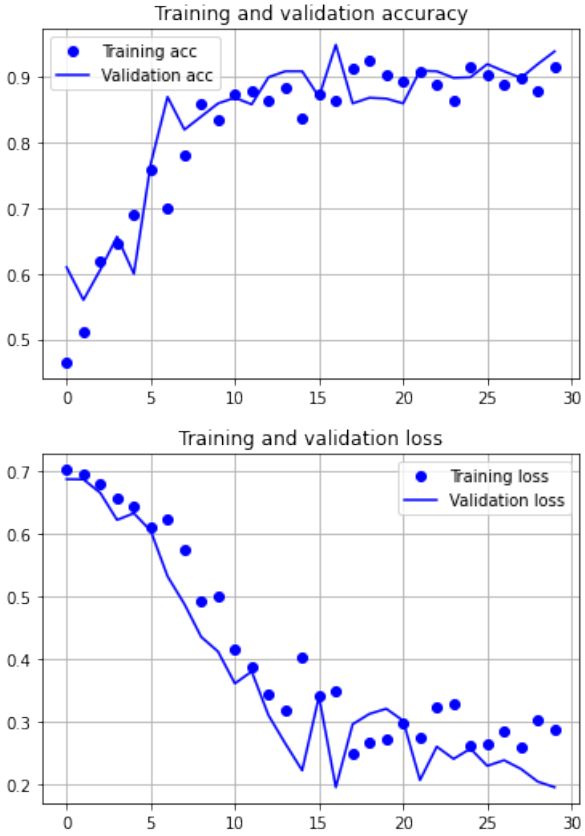

Figure 11: Accuracy and loss curve

7 Performance on YouTube Videos

7.1 How to detect "moments" of target action/emotion

The steps to detect moments of the target action is follows:

1. Use a crawl algorithm to find YouTube videos that matches a certain query, save the video information.
2. Use the scraped YouTube video information to download the videos.
3. Use cv2 to process each video and save frames of each 2 second clip as input to the trained model to predict. For speed up manually delete the obvious non-target segments. Save all the clip information predicted by the model to contain the target action to a .json file.

Crawl algorithm may be further improved using NLP tool to find accurate target action videos.

7.2 View found "moments" in iLab website

7.2.1 Waking up

The label is 'waking up'. 513 clips were uploaded.

7.2.2 House work on clothes shoes hats etc

The label is 'house work on clothes shoes hats etc'. Nearly 1800 clips were uploaded. All clips are labeled as 'House work on clothes shoes hats etc', to search use 'House work on clothes shoes hats etc' in filter.

7.2.3 Bandaging

The label is 'bandaging'. Nearly 1700 clips were uploaded. All clips are labeled as 'Bandaging', to search use 'Bandaging' in filter.

7.2.4 Cut polish or do nails

The label is 'Cut polish or do nails'. Nearly 4000 clips were uploaded. All clips are labeled as 'Cut polish or do nails', to search use 'Cut polish or do nails' in filter.

7.2.5 Sneezing

The label is 'Sneezing'. Nearly 1400 clips were uploaded. All clips are labeled as 'Sneezing', to search use 'Sneezing' in filter.

7.2.6 Arrange flowers or water plants etc

The label is 'Arrange flowers or water plants etc'. 2598 clips were uploaded. All clips are labeled as 'Arrange flowers or water plants etc', to search use 'Arrange flowers or water plants etc' in filter.

7.3 Accuracy

I define False Negative Rate as the clips that include the target action but the model did not predict positive. False Positive Rate is the clips that does not include the target action but the model predicted positive.

7.3.1 Wake up

The false negative rate is low in some videos where adults are involved, but may fail to detect baby waking up occasionally, especially when the situation was not seen by the model before, such as lighting, object blocking the baby, etc.

7.3.2 House work on clothes shoes hats etc

The false negative rate is low in most videos, but occasionally it may not include target actions that involve cleaning and renovating instructions. The false positive rate is very low in videos.

7.3.3 Bandaging

The false negative rate is low in most videos, but occasionally it may not include target actions that involve cleaning and renovating instructions. The False Negative Rate is close to satisfactory. One way to improve is to have more preprocessing steps on the video frames. The false positive rate is very low, with the criteria of clips containing target action set a margin beyond the usual value.

7.3.4 Cut polish or do nails

The false negative rate is low in most videos, but occasionally it may not include target actions that involve when the hand is presented for instructions or being prepared to do nails. The False Negative Rate is close to satisfactory. One way to improve is to have more input data of the type that are not detected and train the network on those new targeted samples along all used data. The false positive rate is very low, with the criteria of clips containing target action set a margin beyond the usual value.

7.3.5 Sneezing

The false negative rate is higher than before in YouTube videos, it may not include target actions that involve very dense and fast sneezing of one person, when the size of the pictures of video is too small, or when the person sneezes and go outside the video window. The reason maybe that this is such a short action and usually happens in ∼1s, the frames are more difficult to represent the whole action and may include more other actions, and make it less easy to recognize. One way to improve is to have more input data frames that captures more of the motion and semantic meaning of the action and train the network on those new targeted samples along all used data. The false positive rate is very low, with the criteria of clips containing target action set a margin beyond the usual value.

7.3.6 Arrange flowers or water plants etc

The false negative rate is low in most of the video, but it may not include target actions that involve when the person in the video is talking and arranging flowers when talking as instructions, when it is a small summary of the video, or when preparing and picking the flowers. The reason maybe that this mixes information too much and very frequently or the action is not very common in the training set, making it more difficult to recognize from just one action, so it will be detected less frequently. One way to improve is to have more input data frames that captures more of the motion and semantic meaning of the action and train the network on those new targeted samples along all used data. The false positive rate is very low, with the criteria of clips containing target action set a margin beyond the usual value.

7.4 Efficiency

The steps to find the target clips is presented in Section 7.1. It takes ∼165 seconds to screen a 11:34 video.

In each video, not all frames of the videos or clips were used. Only 15 frames (in the first project) and 25 frames (in the next projects) are sectioned from the videos and used as input to the model. The total number of frames of one video or clip is very large and most adjacent clips illustrate the same information. Therefore, there's no need to use all of the frames in one video or clip for training or testing, but to use only part of the frames sampled at a reasonable rate.

By using only a part of the frames of each video or clip, the accuracy is high and the speed will be much faster than using all the frames of each video or clip. This will be very useful when the videos are longer, where there will be far too many frames to process and take a lot of computational power. Part of the frames if sampled at a proper rate will pertain sufficient information while improves the efficiently significantly.

Yet, considering the total frame amount of each video, too few frames used may cause the network input to suffer some information loss. It may have been what happened in project 2

with the topic waking up, when only 15 frames were used from each video or clip. But the action is short, so it didn't hurt the performance very much.

In most of the actions I selected, the action isn't very short and may persist for longer. And there are also other actions in between the target actions but do not contain the target action itself. So using more frames per video will ensure more information is collected from the videos of the action and more target action involved features can be learned by the network. This helped increase the accuracy of the network by a reasonable amount.

In my experiments, from project 3, I tried more than 15 frames, and I found out that over 30 frames doesn't improve the accuracy very much. A 25 frame per video is maintaining high accuracy while doesn't slow down the running too much. So 25 frames of each video or clip were used.

8 Improve Accuracy and Efficiency

8.1 Improve accuracy

I used the pretrained network Xception as the feature extraction CNN module. Because it's trained on the ImageNet dataset, so the convolution base already has some level of understanding of input images. In our case, it's the frames of a video.

Next, I tried the pretrained network InceptionV3 as feature extraction which achieved better performance and converged faster compared to Xception. The accuracy increased from 89% to 92%.

As mentioned above, different frames per video or clips used were also experimented with. With longer actions, the frame cannot be too few. When working on short action in project 2, which is waking up, 15 frames per video or clip were used. For longer actions, with experimenting with different frames per video or clips, a 25 frames per video or clip were used for best tradeoff between performance and efficiency.

8.2 Improve efficiency

Only 15 or 25 frames were used in each video or clip as input to the network. There is a huge amount of frames per second in videos or clips, it would be extremely inefficient to use all the frames as input to network. Most adjacent frames and clips contain the same information or very similar information, so only sampling parts from all the frames will be sufficient for achieving a decent accuracy.

In project 2, 15 frames would suffice the need for accuracy, and in projects after it, 25 frames would achieve a satisfactory accuracy while maintaining high efficiency.

There may be other ways to improve efficiency of the network further. I haven't tried many of the ways, but there are some methods that may be promising.

Some preprocessing on the videos and frames may be useful to improve the efficiency more. Such as some feature extraction, including HOG, optical flow, etc. Or extract the pose of the human in the video or frame, using tools such as Open Pose. Use these feature extraction processed tensors as input to the network, it may improve the efficiency or even the accuracy of the network, by removing unnecessary information from the input.

A more efficient network architecture. This can be used to improve training time.

Capture the motion correlation in the frames of a video or clip and remove frames that contain no target action at all before using as input to the network. I'm not exactly sure how to implement it, but a method to capture the motion correlation and remove frames that do not contain target action will speed up the training time and may even achieve higher accuracy. And

with more useful information in the frames, fewer frames may be used as input to the network, which will also improve the efficiency.

9 Code in TAMU GitHub

Links included:
Wake up
House work on clothes shoes hats etc
Bandaging
Cut polish or do nails
Sneezing
Arrange flowers or water plants etc

References

[1] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin, "Action Recognition by Dense Trajectories," in *CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition*, (Colorado Springs, United States), pp. 3169–3176, IEEE, June 2011.

[2] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[3] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," *arXiv preprint arXiv:1708.05038*, 2017.

[4] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558, 2013.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65–72, IEEE, 2005.

[6] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4694–4702, 2015.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015.

[8] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[9] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.