Report

ACTION RECOGNITION BY EGOCENTRIC VISION

Bachelor of Technology
in
Computer Science and Engineering

Submitted by:

Name	Enrollment No.
Parvat Yadav	16114043
Sagar Dhurwe	16114059
Tarun Kumar	16114066
Rishikesh Chaudhary	16114054

Under the guidance of:
Dr. Raman Balasubramanian

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
Roorkee-Uttarakhand

Contents

1 INRODUCTION

This project is about action recognition by egocentric vision [1]. Our task is to recognize action in first person view videos. For this first we used Interactive Museum dataset for recognizing hand gestures. Then we made our own dataset for recognizing outcomes of delivery whether it is run out or catch out or bowled, etc.

2 PROPOSED APPROACH

Here we used Interactive museum dataset. The dataset has been split into 70 % training and 30 % test data.

BY USING MHI

2.1 Data Preprocessing(MHI)

For this we have generated Motion History Images (MHI) of all videos.

2.1.1 Motion History Images

In the MHI[3], the silhouette sequence is condensed into gray scale images, while dominant motion information is preserved. It keeps a history of temporal changes at each pixel location, which then decays over time. The MHI expresses the motion flow or sequence by using the intensity of every pixel in temporal manner. The motion history recognizes general patterns of movement.

2.2 Feature Extraction

For this we have extracted Histogram of Oriented Gradients (HOG) features.

2.2.1 Histogram of Oriented Gradients

The HOG[4] (histogram of oriented gradients) is a feature descriptor used in computer vision and image processing for the purpose of object detection.

The technique counts occurrences of gradient orientation in localized portions of an image.

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

2.3 Models

We have tried mainly two models given below:

2.3.1 Support Vector Machine

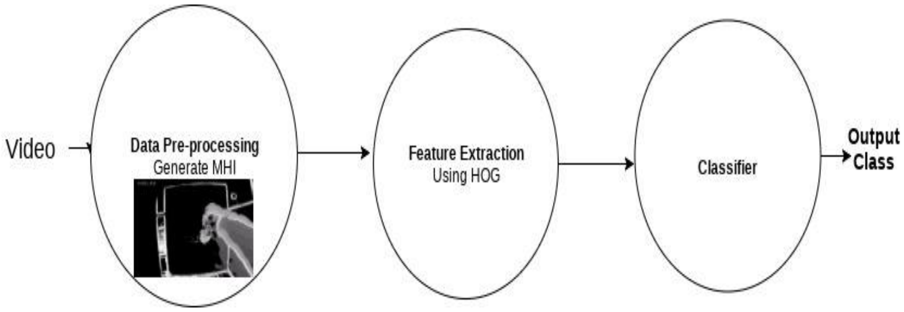
The SVM algorithm predicts by computing the hyperplane with largest margin as the classification separation plane. The SVM algorithm only works with binary classifiers. To allow multi-class classification, we use one-vs-all method.
We have used svm.SVC() classifier from sklearn library. We have used all the default parameters provided by the library.
We achieved **15%** accuracy using this model

2.3.2 Random Forest

It operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

In this we took all parameters as default.
We achieved accuracy **20%** in this model.

Results		
Dataset	Models	Accuracy
Interative Museum	Random Forest	15%
Interactive Museum	SVM	80.95%



MHI is susceptible to noise. Thats why we got such low accuracy. So we tried new approach of dynamic images.

BY USING DYNAMIC IMAGE

2.4 Data Preprocessing(Dynamic Image)

For this we have generated Dynamic Image of all videos.

2.5 Dynamic Image

Dynamic Image[5] is a novel compact representation of videos useful for video analysis especially when convolutional neural networks **(CNNs)**are used. The dynamic image is based on the rank pooling concept and is obtained through the parameters of a ranking machine that encodes the temporal evolution of the frames of the video. Dynamic images are obtained by directly applying rank pooling on the raw image pixels of a video producing a single **RGB** image per video.

2.6 Feature Extraction

We extracted **HOG** features for **SVM** and **Random Forest** classifier. **AlexNet** doesnt requires explicit feature extraction.

2.7 Models

We have used three models given below:

2.7.1 Support Vector Machine

We provided HOG features of dynamic images, and used all default parameters of svm.SVC() classifier of sklearn library.
We achieved accuracy **12.85%** in this model.

2.7.2 Random Forest

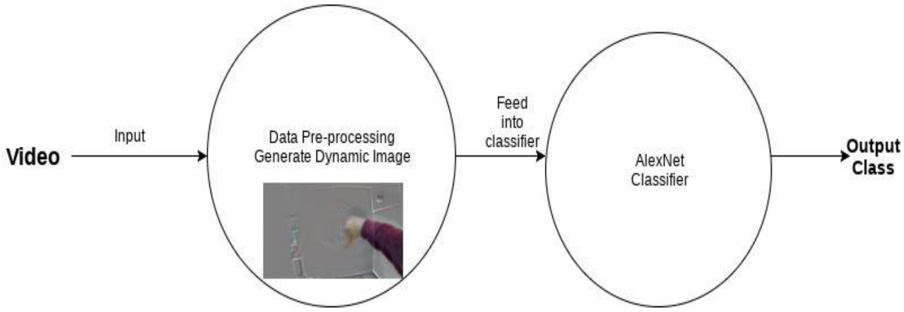
We provided HOG features of dynamic images, we took all parameters as default.
We achieved accuracy **22.85%** in this model.

2.7.3 AlexNet convolutional neural network

AlexNet[6] contains 5 convolutional layers and 3 fully connected layers. Relu is applied after every convolutional and fully connected layer. Dropout is applied before the first and the second fully connected layer. The image size

in the following architecture chart should be 227 * 227.
We achieved accuracy **80.95%** in this model.

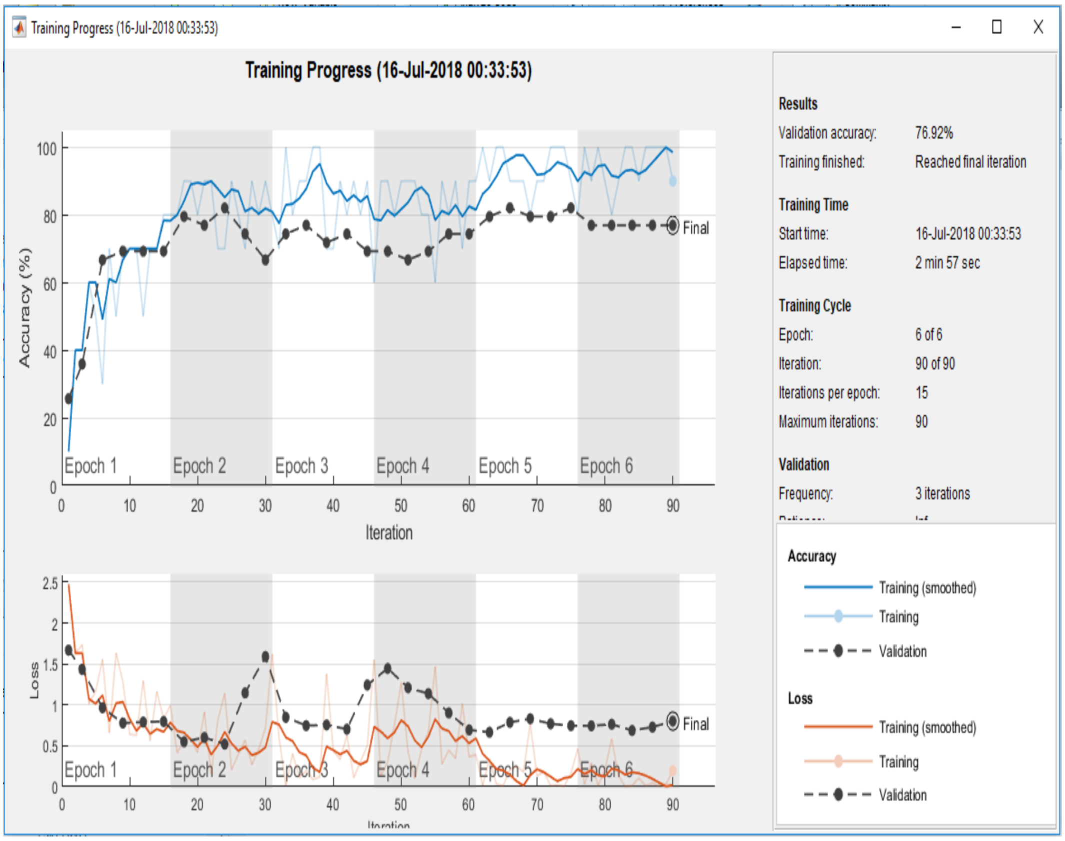
Results		
Dataset	Models	Accuracy
Interactive Museum	Random Forest	22.85%
Interactive Museum	SVM	12.85%
Interactive Museum	AlexNet	80.95%



Among all these models, **AlexNet convolutional neural network** gives the highest accuracy using dynamic images.So we used AlexNet for Cricket dataset.

The dataset has been split into 80% training and 20% test data.Here are the results:

Results		
Dataset	Models	Accuracy
Cricket	Random Forest	51.24%
Cricket	SVM	23.80%
Cricket	AlexNet	76.92%



Results

Validation accuracy: 76.92%

Training finished: Reached final iteration

Training Time

Start time: 16-Jul-2018 00:33:53

Elapsed time: 2 min 57 sec

Training Cycle

Epoch: 6 of 6

Iteration: 90 of 90

Iterations per epoch: 15

Maximum iterations: 90

Validation

Frequency: 3 iterations

Patience: Inf

fig.1: Training and testing with AlexNet

3 DATASET

This is our dataset on which we hava trained our models.

3.1 Interactive Museum dataset

A gesture recognition dataset[2] taken from the ego-centric perspective in a virtual museum environment. It consists of 700 video sequences, all shot with a wearable camera, in an interactive exhibition room, in which paintings and artworks are projected over a wall, in a virtual museum fashion. In this dataset videos of various hand gestures are recorded by means of ego-vision embedded devices. There are 7 classes in this dataset given below:
1. Ok
2. Like
3. Dislike
4. Point

5. Slide left to right
6. Slide right to left
7. Take a picture
Each of these class has 100 videos.

3.2 Cricket Dataset

This dataset contains short clips of cricket. There are 4 classes in this dataset given below:
1. Catch out
2. Run out
3. out Bowled
4. Four
Each of these class has 50 videos.

4 CONCLUSION

We concluded that combination of Dynamic Image as preprocessing tool and **AlexNet convolutional neural network** as our classification algorithm, we obtained high performing model for egocentric activity recognition as compared to other models. We used this model for our Cricket dataset ,and achieved final accuracy of **76.92%.**

Acknowledgement

We respect and thank **Dr. R. Balasubramanian**, for providing us an opportunity to do the project work in **IIT Roorkee** and giving us all support and guidance which made us complete the project duly. We are extremely thankful to him for providing such a nice support and guidance.
The success and final outcome of this project required a lot of guidance and assistance from Mr. Javed Imran and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

1.)Parvat Yadav
2.)Rishikesh chaudhary
3.)Sagar Dhurwe
4.)Tarun Kumar

15 May 2017 to 15 July 2018

Indian Institute of Technology Roorkee

REFERENCES

1.) http://giuseppeserra.com/content/egocentric-vision-cultural-heritage
2.) http://phidiasproject.eu/publications/pdf/J3.pdf
3.) https://www.researchgate.net
4.) https://in.mathworks.com/help/vision/ref/extracthogfeatures.html
5.) http://egavves.com/data/cvpr2016bilen.pdf
6.) https://in.mathworks.com/help/nnet/ref/alexnet.html