Final Project Evaluation Report

Shivank Sharma
2018EEB1180

Indian Institute of Technology Ropar
Rupnagar, Punjab

Abstract

Human Action Recognition is a popular challenge now due to its high requirement in various applications. The objective of this project is to take an RGB video as input and recognize the activity that is being performed in it. The dataset that will be used for this purpose is UCF50 - Action Recognition Data Set which contains videos taken from YouTube under 50 different categories.

The **literature review** summarises and compares some of the leading techniques used for video classification. It includes how standard approaches are used to classify videos but they have drawbacks regarding the performance on large datasets. However the Convolutional Neural Network removes these problems and works even on very large datasets like the Sports-1M.

It also includes the information and results of the model that I trained on the UCF-50 dataset. A video can be assumed as a series of images so the trained model is an CNN image classifier and is used to recognize human actions.

1 Introduction

Human action recognition is a widely researched topic in the past few years due to its very wide range of applications. Its applications include CCTV surveillance systems, human behaviour monitoring and a variety of systems which include a human-computer interface.

Challenges like camera motion, different viewpoints, large inter class variations, cluttered background, occlusions, bad illumination conditions, and poor quality of web videos cause the majority of the state-of-the-art action recognition approaches to fail. The increase in number of categories and in size of dataset also increase the challenges in action recognition.

The challenge of video classification is however very different from image classification. In a video the subsequent frames are correlated with respect to their semantic contents. By taking advantage of the temporal nature of videos we can improve our video classification results.

The dataset used for the video classification is UCF50, which is an extension of YouTube Action data set (UCF11) which has 11 action categories. It contains videos downloaded from YouTube and has 50 categories (figure 1).

Various approaches have been taken for video classification. The standard approach includes the extraction of local visual features that describe a region of the video. Then these features get combined into a fixed-sized video-level description. Later a classifier is trained on these descriptions to classify a video. However we now use Convolutional Neural Network that is trained end to end from raw pixel to classifier outputs.

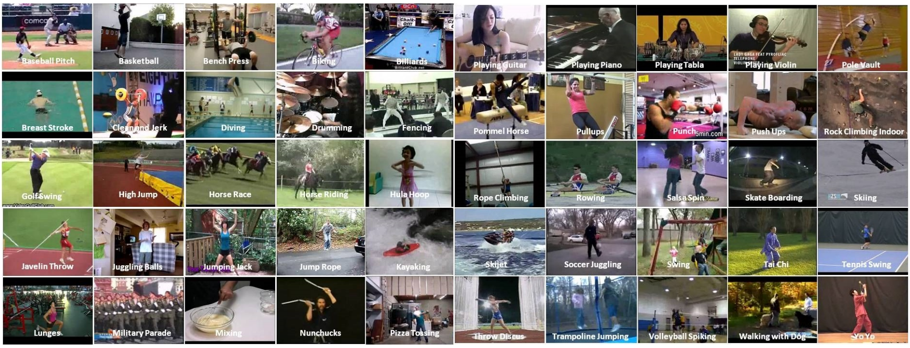


Figure 1: Screenshots from videos in the UCF50 dataset showing the diverse action categories.

2 Related Work

The classification of an activity in video is easy for humans but is challenging for computers. As mentioned earlier, these challenges occur due to varying conditions like pose, light and orientation. Hence so many different approaches have been developed and reported. Some main methods for video classification are mentioned below.

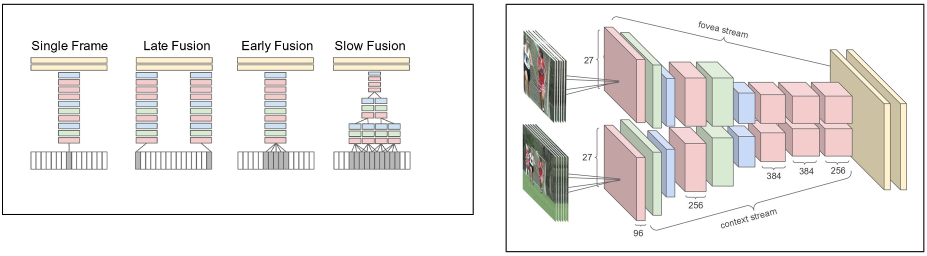
2.1 Recognizing 50 Human Action Categories of Web Videos

The paper "Recognizing 50 Human Action Categories of Web Videos" provides an insight into the challenges of large and complex datasets like UCF50. They propose the use of groups of moving pixels which can be roughly assumed as salient regions and groups of stationary pixels as an approximation of non salient regions in a given video. They use optical flow at each pixel and apply a threshold on the magnitude of the optical flow, to decide if the pixel is moving or stationary.

They also show that as the number of actions to be categorized increases, the scene context plays a more important role in action classification. They use the idea of early fusion schema for descriptors obtained from moving and stationary pixels to understand the scene context, and finally perform a probabilistic fusion of scene context descriptor and motion descriptor.

The classification is done by concatenating different descriptors and then training a classifier, such as support vector machines (SVM), that can provide a probability estimate for all the classes rather than a hard classification decision. The average performance achieved by the method is 68.20%.

However the drawback of this method include its unusability on a large dataset like UCF101 which include 101 classes or Sports-1M with 487 classes. The performance on such datasets is low in comparison to Convolutional Neural Networks. But it is easy to train this shallow algorithm in comparison to deep networks.



(a) (b)

Figure 2: (a) Explored approaches for fusing information over temporal dimension through the network. Red, green and blue boxes indicate convolutional, normalization and pooling layers respectively. In the Slow Fusion model, the depicted columns share parameters. ; (b) Multiresolution CNN architecture. Input frames are fed into two separate streams of processing: a context stream that models low-resolution image and a fovea stream that processes high-resolution center crop. Both streams consist of alternating convolution (red), normalization (green) and pooling (blue) layers. Both streams converge to two fully connected layers (yellow).

2.2 Large-scale Video Classification with Convolutional Neural Networks

The paper “Large-scale Video Classification with Convolutional Neural Networks” studies the performance of CNNs in large scale video classification, where the networks have access to not only the appearance information present in single,static images,but also their complex temporal evolution. They have also introduced a new dataset Sports-1M, which consists of 1 million YouTube videos belonging to a taxonomy of 487 classes of sports.

In this method they treat every video as a bag of short, fixed-sized clips. Since each clip contains several contiguous frames in time dimension to learn spatio-temporal features. They also describe three approaches (Early Fusion, Late Fusion and Slow Fusion) for extending the connectivity of a CNN in time domain to take advantage of local spatio-temporal information (figure 2.a). It suggests a multiresolution, foveated architecture as a promising way of speeding up the training (figure 2.b).

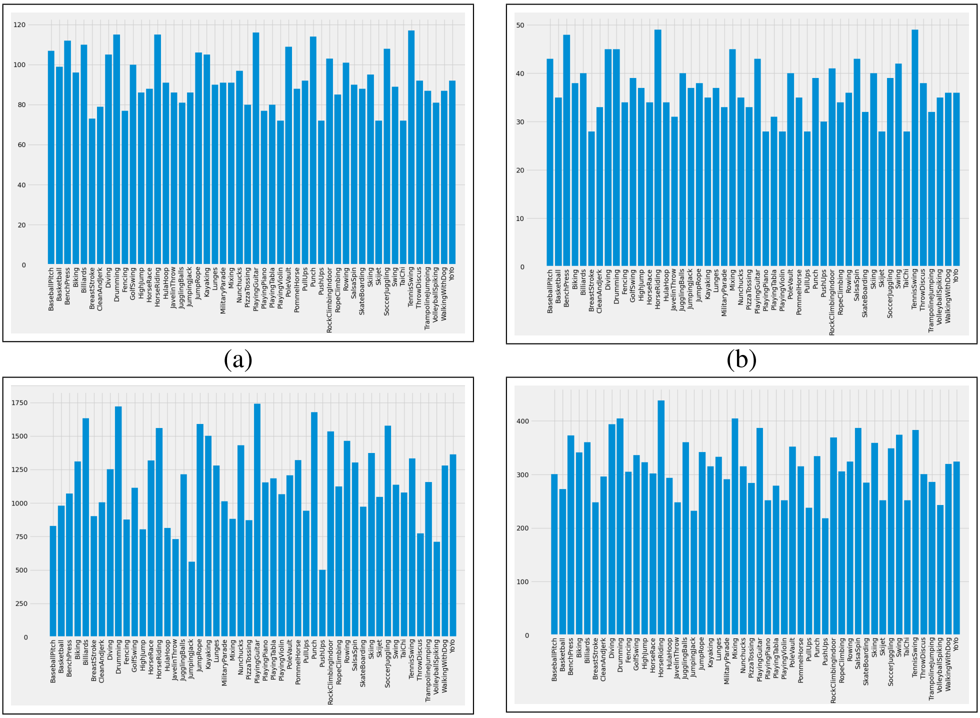
Their best spatio-temporal networks display significant performance improvements compared to strong feature-based baselines (55.3% to 63.9%), but only a surprisingly modest improvement compared to single-frame models (59.3% to 60.9%).

They further study the generalization performance of the best model by retraining the top layers on the UCF-101 Action Recognition dataset and observe significant performance improvements compared to the UCF-101 baseline model (63.3% up from 43.9%). The UCF101 is an extension of UCF50 and the same method can be applied on UCF50.

However this method also has some drawbacks like it takes too long to train such models. It also requires so much computational power for training such models.

3 Approach

Videos can be understood as a series of images and it would an initial approach to treat video classification as performing image classification on the frames of a video. My approach is to



(a) (b)

(c) (d)

Figure 3: (a)(b) Category wise number of videos after splitting in train and test set respectively according to list on official website of dataset. (c)(d) Category wise number of extracted frames from videos in train and test set respectively.

take some frames from the video and classify them individually into the given categories of human actions.

After classifying the frames we can get the maximum occurring prediction among all chosen frames. It is because in all the chosen frames some will be misclassified but a majority of them will be classified correctly and this leads to good performance just by image classification.

The technique used is Transfer Learning. The base model chosen is ResNet-50 along with the trained weights. The top layers were replaced by Dense layers.

4 Learning

Optimization : I have used Adam as the optimizer for the model. I have used mini-batches of 16 examples and a learning rate of 1e-2.

Train-Test Split : The train test split of the videos was done on the basis of list provided on the dataset official website. So by using it, I have extracted the frames from the videos and have arranged them according to their classes. After splitting I had 4645 videos in training set and 1837 videos in test set. The category wise number of videos for both sets is given in figure figure 3(a) and 3(b).

Data Augmentation and Preprocessing : To reduce overfitting in the model I have used data augmentation. Before presenting any example to the network, I processed all images by resizing them to 224 x 224 pixels along with 50% probability of flipping the image horizontally. The other operations applied on the images include rotating, shifting horizontally and vertically, zooming and shearing. For all these operation I have extracted images from training videos with a rate of 2 frames per second. However for the test videos the frame rate was same i.e. 2 frames per second but only for initial 4 seconds of the video. After this I had 58298 images in training set and 15855 images in test set. The category wise number of images for both training and test set are given in figure 3(c) and 3(d).

5 Results

The model was trained on GTX 1050Ti GPU with 4 GB GPU memory. The model was trained for 50 epochs and the results are really surprising for such a simple approach for video classification. The model was validated on test videos and performed well with an accuracy of 64.6%. The official benchmark on the dataset is of 68.2%. Hence the model performed well on the dataset.

From the results it can be clearly seen that model performed worse on some classes. Like in Jumping Jack and Jump Rope, the model got confused and predicted the results interchangeably. Also it got confused between Javelin Throw and Tennis Swing as both the actions require swinging of the hand.

However the performance on the dataset can be increased by using 3D convolutional neural networks or by Sequence Models like LSTMs or RNNs.

The confusion matrix for the model is given in figure . The classification report the model is given in table 1 & 2.

6 References

1. Kishore K. Reddy, and Mubarak Shah, *Recognizing 50 Human Action Categories of Web Videos*, Machine Vision and Applications Journal (MVAP), September, 2012.

2. Andrej Karpathy and George Toderici and Sanketh Shetty and Thomas Leung and Rahul Sukthankar and Li Fei-Fei, *Large-scale Video Classification with Convolutional Neural Networks*, CVPR, 2014.

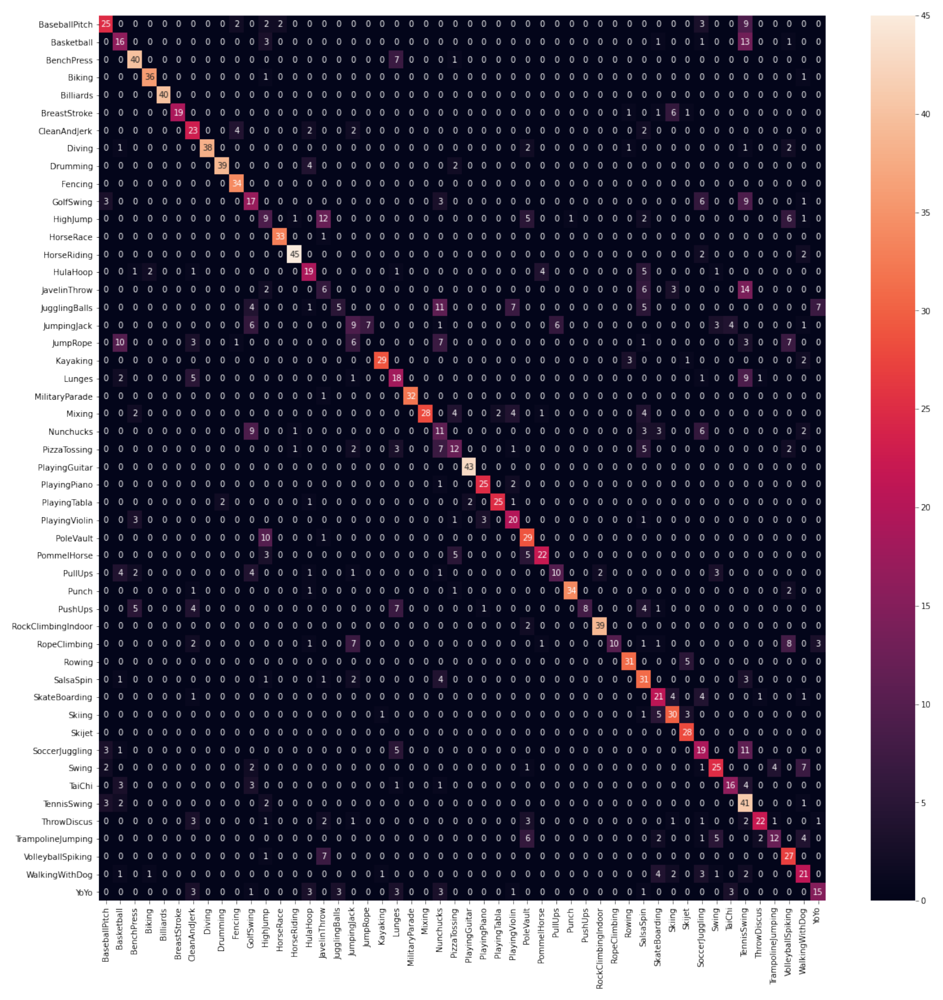


Figure 4: Confusion table for UCF-50 dataset

	precision	recall	f1-score	support
BaseballPitch	0.69	0.58	0.63	43
Basketball	0.39	0.46	0.42	35
BenchPress	0.75	0.83	0.79	48
Biking	0.92	0.95	0.94	38
Billiards	1.00	1.00	1.00	40
BreastStroke	1.00	0.68	0.81	28
CleanAndJerk	0.50	0.70	0.58	33
Diving	1.00	0.84	0.92	45
Drumming	0.95	0.87	0.91	45
Fencing	0.83	1.00	0.91	34
GolfSwing	0.37	0.44	0.40	39
HighJump	0.26	0.24	0.25	37
HorseRace	0.94	0.97	0.96	34
HorseRiding	0.94	0.92	0.93	49
HulaHoop	0.58	0.56	0.57	34
JavelinThrow	0.19	0.19	0.19	31
JugglingBalls	0.62	0.12	0.21	40
JumpingJack	0.29	0.24	0.26	37
JumpRope	0.00	0.00	0.00	38
Kayaking	0.94	0.83	0.88	35
Lunges	0.40	0.49	0.44	37
MilitaryParade	1.00	0.97	0.98	33
Mixing	1.00	0.62	0.77	45
Nunchucks	0.22	0.31	0.26	35
PizzaTossing	0.46	0.36	0.41	33
PlayingGuitar	0.96	1.00	0.98	43
PlayingPiano	0.86	0.89	0.88	28
PlayingTabla	0.93	0.81	0.86	31
PlayingViolin	0.56	0.71	0.63	28
PoleVault	0.55	0.72	0.62	40
PommelHorse	0.79	0.63	0.70	35
PullUps	0.62	0.36	0.45	28
Punch	0.97	0.87	0.92	39
PushUps	1.00	0.27	0.42	30
RockClimbingIndoor	0.95	0.95	0.95	41
RopeClimbing	1.00	0.29	0.45	34
Rowing	0.86	0.86	0.86	36
SalsaSpin	0.43	0.72	0.54	43
SkateBoarding	0.54	0.66	0.59	32
Skiing	0.65	0.75	0.70	40
Skijet	0.74	1.00	0.85	28
SoccerJuggling	0.40	0.49	0.44	39
Swing	0.66	0.60	0.62	42

Table 1: Classification Report - Part 1

	precision	recall	f1-score	support
TaiChi	0.70	0.57	0.63	28
TennisSwing	0.34	0.84	0.48	49
ThrowDiscus	0.85	0.58	0.69	38
TrampolineJumping	0.71	0.38	0.49	32
VolleyballSpiking	0.49	0.77	0.60	35
WalkingWithDog	0.48	0.58	0.53	36
YoYo	0.58	0.42	0.48	36
accuracy			0.65	1837
macro avg	0.68	0.64	0.64	1837
weighted avg	0.68	0.65	0.64	1837

Table 2: Classification Report - Part 2