B.Tech Project Report

on

Unsupervised Action Recognition

Submitted by

Ruchin Kukreja (2009CS10212)

Supervised by

Dr. Parag Singla

Computer Science and Engineering,

Indian Institute of Technology Delhi

May 2013

Declaration of Authorship

I, Ruchin Kukreja, declare that this thesis titled, Unsupervised Video Action Recognition and the work presented in it are entirely my own. I confirm that:

- This work was done wholly or mainly while in candidature for a B.Tech degree at the Indian Institute of Technology, Delhi.
- Where any part of this thesis has been previously submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- Where I have consulted the published works of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed :
Date :

Abstract

A lot of work on action recognition in videos has focused on extending hand designed local features, from static images to the video domain. In this thesis, we explore the use of unsupervised learning of features directly from video data. Supervised learning based algorithms require large amount of labeled training data and don’t work well in deep networks. Many unsupervised learning based algorithms have been implemented and have shown better results. They can also be easily extended to other fields.

We consider an extension of the Independent Subspace Analysis algorithm to learn invariant spatial- temporal features from unlabeled video data and to classify on the basis of these features using SVM (a supervised learning algorithm). We compare it's performance with classifiers using HOG/HOF features at Spatio Temporal Interest Points (STIP) on Hollywood 2 dataset. An alternate algorithm combining unsupervised ISA features with the features derived from object detection on videos (hand designed local features) is proposed and its performance is compared with the rest.

Acknowledgments

I would like to thank my guide and supervisor Prof. Parag Singla for his constant guidance and support. His knowledge and eye for detail have helped me to learn and produce more in the given time duration. I would like to thank all my colleagues who have co-operated with us whenever needed.

Contents

1. Introduction

Analyzing and interpreting video is a growing topic in computer vision and its applications. Video data contains information about changes in the environment and is highly important for various tasks including navigation, surveillance and video indexing.

In this project, we analyze video data to recognize different pre- defined actions in the video using machine learning algorithms. The emphasis of the project is not the classification methods but the features extracted from the videos used for classification. We have compared the performance of unsupervised feature learning method(Stacked Convoluted ISA) to STIP based features keeping the learning method as well as the dataset used for training and testing same. In the end, we present a new feature set which is the combination of unsupervised features with the features derived from object detection in frames of the video.
We use previously trained object recognizers to automatically detect objects in video and use this information to help identify related activities. For example, detecting a car in the image helps classify the activity as driving. So probability of car occurring in a video can be added as a feature in the existing feature set.

We have used the same classification method ie SVM across all these so as to make sensible comparisons. The dataset used is Hollywood2, a collection of small video clips taken from Hollywood movies, and is same across all the methods.

2. Motivation

Supervised feature extraction requires back-propagation which further requires labeled training data and don’t work well in deep networks, also training data may not be available in large amounts. Many unsupervised algorithms have been implemented and have shown better results than different handpicked features used for videos.

Problem Statement: Given a dataset containing labeled training videos and testing videos (Hollywood2), build a feature set so as to find better classification accuracy.

3. Algorithms

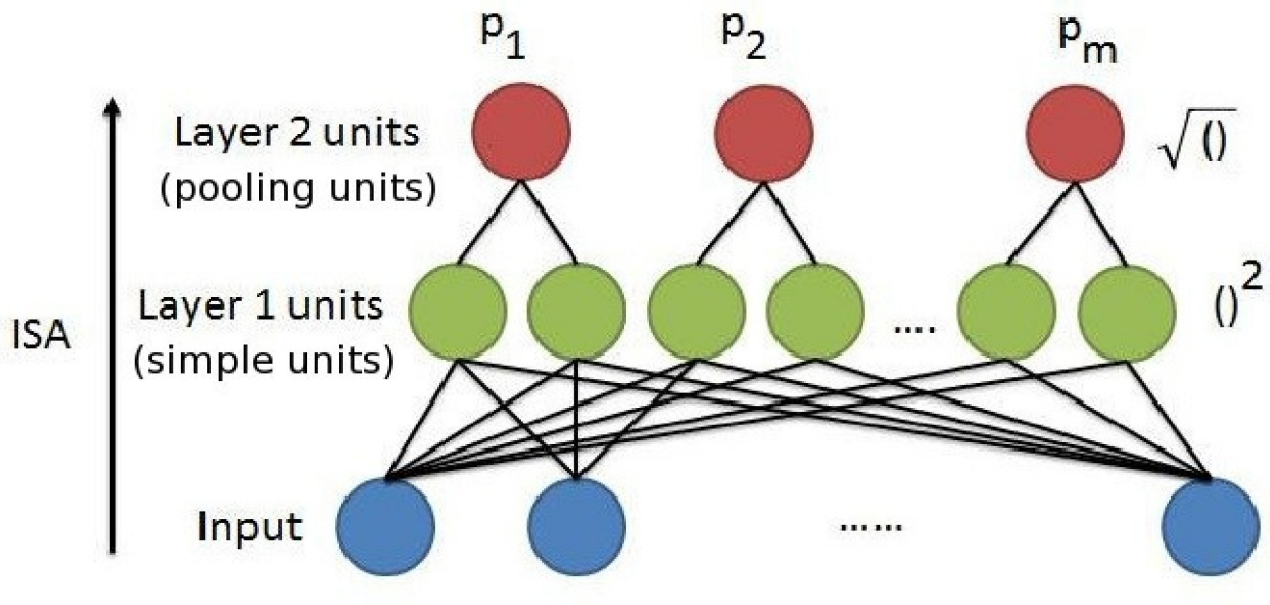
A description of the important algorithms used for action recognition is given below. The references to the algorithms are given in the side of the headings.

3.1 Unsupervised Feature Learning

Unsupervised learning refers to the problem of trying to find hidden structure in unlabeled data. There is a growing interest in unsupervised feature learning methods because they learn features directly from data and are generalizable. The unsupervised algorithm used is Independent Subspace Analysis (ISA) for videos.

3.1.1 Independent Subspace Analysis (ISA) for images [5]

ISA is an unsupervised learning algorithm that learns features from unlabeled image patches. An ISA network can be described as a two-layered network, with square and square-root nonlinearities in the first and second layers respectively. The weights W in the first layer are learned, and the weights V of the second layer are fixed. We will call the first and second layer units simple and pooling units, respectively.



The above image has been taken from Andrew Ng paper.

More precisely, given an input pattern x^t , the activation of each second layer unit is

$$p_i\ (x^t\ ; W, V\ ) \quad = \quad \sqrt{\sum_{k=1}^{m} V_{ik} (\sum_{j=1}^{n} W_{kj} x_j^t)^2}$$

ISA learns parameters W_j through finding sparse feature representations in the second layer, by solving:

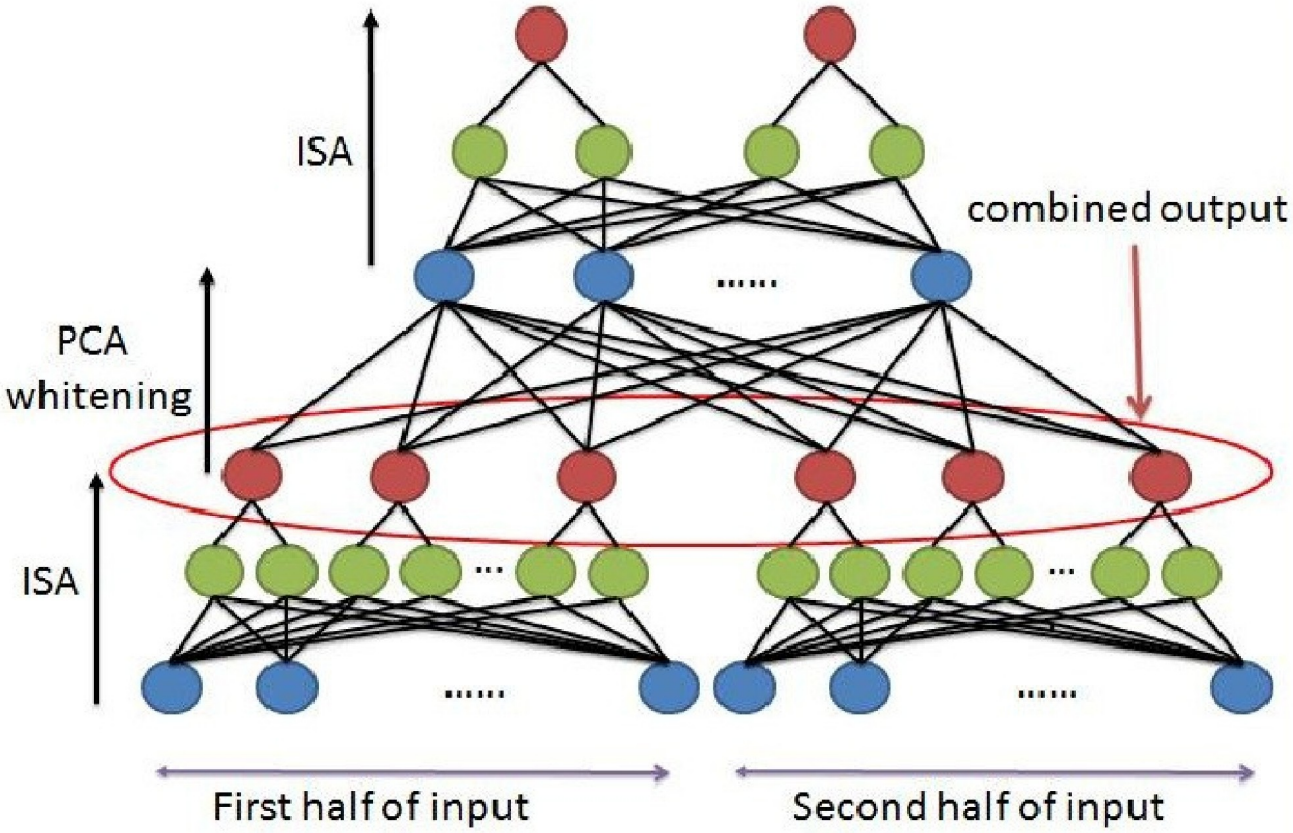
$$\begin{array}{ll} \underset{W}{\text{minimize}} & \sum_{t=1}^{T} \sum_{i=1}^{m} p_i(x^t; W, V), \\ \text{subject to} & WW^T = \mathbf{I} \end{array} \qquad (1)$$

where $\{x^t\}_{t \in T}$ are whitened input examples. Here, $W \in R^{k \times n}$ is the weights connecting the input data to the simple units, $V \in R^{m \times k}$ is the weights connecting the simple units to the pooling units (V is typically fixed); n, k, m are the input dimension, number of simple units and pooling units respectively. The orthonormal constraint is to ensure the features are diverse.

One property of the learned ISA pooling units is that they are invariant and hence suitable for recognition tasks. The standard ISA training algorithm becomes less efficient when input is large. Thus, training this algorithm with high dimensional data, especially video data, takes days to complete.

3.1.2 Stacked Convoluted ISA [1]

The key ideas of this approach are as follows. We first train the ISA algorithm on small input patches. We then take this learned network and convolve with a larger region of the input image. The combined responses of the convolution step are then given as input to the next layer which is also implemented by another ISA algorithm with PCA as a prepossessing step. Similar to the first layer, we use PCA to whiten the data and reduce their dimensions such that the next layer of the ISA algorithm only works with low dimensional inputs. In our experiments, the stacked model is trained greedily layer wise. More specifically, we train layer 1 until convergence before training layer 2. Using this idea, the training time requirement is reduced to 1-2 hours.



The above image has been taken from Andrew Ng paper

3.1.3 Batch Projected Gradient Descent [1]

Compared to other feature learning methods, the gradient of the objective function in Eq. 1 is tractable. The orthonormal constraint is ensured by projection with symmetric orthogonalization. In detail, during optimization, projected gradient descent requires us to project W to the constraint set by computing

$$(W W^T)^{-1/2} W$$

Note that the inverse square root of the matrix usually involves solving an eigenvector problem, which requires cubic time. Therefore, this algorithm is expensive when the input dimension is large. The convolution and stacking ideas address this problem by slowly expanding the receptive fields via convolution.

3.2 Spatial-temporal Interest Points (STIP) [2]

In the spatial domain, points with a significant variation of image intensities are frequently to as “interest points” and are important due to their high information content and relative stability with respect to transformations of the data. Extension of the notion of interest points into the spatial temporal domain(image to video) results in local space-time features which correspond to interesting events in video data These features can be used for sparse coding of videoinformation that in turn can be used for interpreting videos.

To capture spatially and temporally interesting movements, we use the motion descriptors developed by Laptev. These features have been shown to work well for human-activity recognition in real-world videos. In addition, this approach is easy to apply to new problems since it does not use any domain-specific features or prior domain knowledge. First, a set of spatial temporal interest points (STIP) are extracted from a video clip. At each interest point, we extract a HoG (Histograms of oriented Gradients) feature and a HoF (Histograms of optical Flow) feature computed on the 3D video space-time volume.

The patch is partitioned into a grid with 3x3x2 spatio-temporal blocks. Four-bin HoG and five-bin HoF descriptors are then computed for all blocks and concatenated into 72-element and 90- element descriptors, respectively. We then concatenate these vectors to form a 162-element descriptor. A randomly sampled set of 500,000 motion descriptors from all video clips is then clustered using K-means(k=200) to form a vocabulary.

Finally, a video clip is represented as a histogram over this vocabulary. The final “bag of visual words” representing a video clip consists of a vector of k values, where the i^{th} value represents the number of motion descriptors in the video that belong to the i^{th} cluster.

3.3 Object Detection Algorithm for Videos [4]

We used an off-the -shelf pre-trained object detector based on Discriminatingly Trained Deformable Part Models to detect objects in videos [6]. We used pretrained models for 19 objects. The 19 objects detected are: airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, potted plant, sheep, sofa, train, TV monitor. This approach can be easily extended to videos.

First, we extracted one frame per second from each video in the test set, and represented each video by a set of frames. Then we ran the object detector on each resulting frame to produce bounding -boxes with scores for each of the 19 objects. To produce a probability, P (Oi |Vj), for object Oi appearing in video Vj , we took the maximum probability assigned to any detection of object Oi in any of the frames for video Vj . In this way, we computed a probability of each of the 19 objects occurring in each video.

3.4 Support Vector Classifier

We use Support Vector Machines (SVM) to learn actions from unsupervised features. The decision function of a binary SVM classifier takes the following form:

$$g(\boldsymbol{x}) = \sum_i \alpha_i y_i K(\boldsymbol{x_i}, \boldsymbol{x}) - b$$

where K(xi,x) is the value of a kernel function for the training sample xi and the test sample x, yi ∈ {+1, −1} is the class label (positive/negative) of xi , αi is a learned weight of the training sample xi , and b is a learned threshold. We use the values of the decision function as a confidence score and plot recall-precision curves for evaluation. We then compute the average precision (AP), which approximates the area under a recall-precision curve. The simplest kernel possible is a linear kernel. The decision function can then be rewritten as a weighted sum of sample components as:

K(xi , x j) = xi · xj g(x) = w·x − b .

To classify feature distributions compared with the χ2 distance, we use the multi-channel Gaussian kernel:

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\big(-\sum_c \frac{1}{\Omega_c} D_c(\boldsymbol{x_i}, \boldsymbol{x_j})\big)$$

where Dc (xi , xj) is the χ2 distance for channel c, and Ωc is the average channel distance. To build a multi-class classifier one might combine binary classifiers using one-against-rest or one -against-one strategies. Note, however, that in our setup all problems are binary, i.e., we recognize each action independently and concurrent presence of multiple concepts (mainly multiple actions) is possible. To compare the overall system performance, we compute an average AP over a set of binary classification problems.

4) Related Work

We know enlist some of the important work from which we have taken some ideas for the project.

- “Improving Video Activity Recognition using Object Recognition and Text Mining” Tanvi S. Motwani and Raymond J. Mooney. They have presented a combination of standard activity classification, object recognition, and text mining to learn effective activity recognizers. They use labeled data to train an activity classifier based on spatio temporal features. Next, text mining is employed to learn the correlations between these verbs and related objects. This knowledge is then used together with the outputs of an object recognizer and the trained activity classifier to produce an improved activity recognizer.

- “Learning multiple layers of representation” Geoffrey E. Hinton who proposed using multilayer neural networks that contain top-down connections and training them to generate sensory data rather than to classify it.

- “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis” Quoc V. Le, Will Y. Zou, Serena Y. Yeung, Andrew Y. Ng who have proposed using unsupervised feature learning as a way to learn features directly from video data specifically an extension of the Independent Subspace Analysis algorithm to learn invariant spatio-temporal features from unlabeled video data has been given along with the code.

5. Proposed Framework

In this section we describe the experiments we undertook and the key results along with what we learned from them. We first describe the dataset followed by the various experiments we carried out.

5.1 Dataset

The Hollywood2 human actions dataset (http://pascal.inrialpes.fr/hollywood2/) containing 823 train and 872 test video clips. The dataset consists of small video clips of length 6 – 20 seconds taken from different famous movies in Hollywood. There are in total 12 action classes and each video clip may have more than one action label.

The 12 action classes are :

- Answer Phone
- Drive Car
- Eat
- Fight person
- Get out of car
- Hand shake
- Hug person
- Kiss
- Run
- Sit down
- Sit up
- Stand up

5.2 Methodology

Across all the features, we use the same pipeline with Wang et. al [5]. This pipeline first extracts features from videos on a dense grid in which cube samples overlap 50% in x, y and t dimensions. K-means vector quantization is applied on the extracted features and each video is histogramed to form a bag-of-words representation. Finally, the bag-of-words representation is L1-normalized and a $\chi 2$ kernel SVM is used to classify human action. To make our experiments comparable to earlier work, we apply the same evaluation setting. Detailed results, such as average precision/accuracy per action class and confusion matrices, are also provided.

5.3 Learning using ISA Features

To arrive at the results, we first train the SVM using features extracted from the training subset of video clips of the Hollywood2 dataset by the Stacked Convoluted ISA algorithm. Then the SVM is applied to the test subset of the clips to identify various actions defined in the hollywood2 dataset. The classification is compared to the manually prepared list of actions for various videos and the result are arrived at. The code for extracting the features and applying SVM has been provided at http://ai.stanford.edu/~wzou/ by the research team behind the development of Stacked Convoluted ISA.

5.4 Learning using STIP features

Here we have represented the training and testing video clips from Hollywood2 dataset using HoG and HoF features at the Spatial-temporal Interest Points and run SVM for training and classification. The code for the extraction of features has was taken from Prof. Ivan Laptev's Homepage.

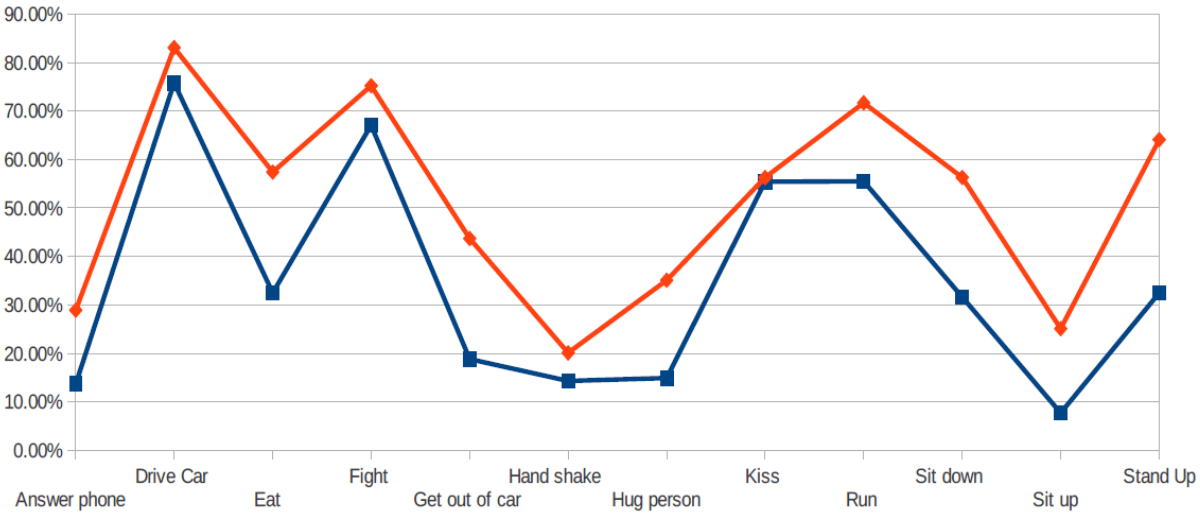
5.5 A New Feature Set

The pipeline for testing is same as that of the previously discussed cases, the only difference being the feature set. The new feature set consists of two components Features derived from stacked convoluted ISA, Features obtained by executing Object Recognition on videos. These features are added as a column to the ISA features where each entry of the column indicates the probability of an object occurring during a video.

5.6 Combined Results

Action	Accuracy (STIP)	Accuracy (ISA)	Accuracy (New Features)
Answer Phone	13.7%	28.9%	29.4%
Drive Car	75.8%	83.1%	85.1%
Eat	32.5%	57.4%	57.8%
Fight	67.1%	75.2%	75.7%
Get out of car	18.8%	43.7%	44.8%
Hand shake	14.3%	20.1%	20.1%
Hug	14.9%	35.1%	35.4%
Kiss	55.4%	56.3%	56.4%
Run	55.5%	71.7%	71.8%
Sit Down	31.6%	56.3%	56.5%
Sit Up	7.7%	25.1%	25.3%
Stand Up	32.5%	64.1%	64.5%
Average	34.5%	51.4%	51.9%

The average accuracy is 1.9% which is .5 % more than the accuracy observed in ISA features. Hence adding object recognition features to the ISA feature set helps improve classification accuracy but not by much. The increase in accuracy in the case where objects identified had better correlation to the activity was more. For eg. We detect persons using object recognition but activities like fight, hand shake, kiss, hug, run all involve the same object, hence the new features
can't distinguish between them, but activities such as getting out of car which has two objects person and a car shows much more improvement.
There is a huge improvement in the case of ISA features over STIP features which has been demonstrated by the graph below.



Here the Red line is for ISA while the blue line is for STIP features.

Future Work

The improvements in the current feature set and the experimentation techniques are:-

- Comparisons using different datasets like KTH, Youtube etc as well across the feature sets to compare their performances.

- Including more models for object detection so that correlation of objects with action is captured.

- Instead of just adding a row corresponding to the object detector features, one can do classification separately on the basis of these two features and then run a separate SVM on the results predicted by them.

References

1) Quoc V. Le, Will Y. Zou, Serena Y. Yeung, Andrew Y. Ng “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis” in CVPR, 2011

2) Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld, ‘Learning realistic human actions from movies’, in CVPR,(2008).

3) Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh. “A Fast Learning Algorithm for Deep Belief Nets” 2006

4) Tanvi S. Motwani and Raymond J. Mooney “Improving Video Activity Recognition using Object Recognition and Text Mining” ECAI-2012

5) Neu. Comp., A. Hyvarinen and P. Hoyer. “Emergence of phase- and shift invariant features by decomposition of natural images into independent feature subspaces”.,2000

6) Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan, ‘Object detection with discriminatively trained partbased models’, IEEE Trans. Pattern Anal. Mach. Intell., (2010).