Nasri Chiheb

# AVPR Lab: Assignment 2

# Recognizing Human Actions in Still Images using Support Vector Machine

## Introduction:

The purpose of this second assignment is to elaborate a human action recognition algorithm. The algorithm is implemented with MATLAB using mainly the Image processing toolbox library. The algorithm follows four steps preprocessing, feature extraction, training, and testing. The algorithm is to compare different methods for feature extraction and pattern classification.

## Method description

The algorithm follows four steps. The algorithm needs an input image then follows this scheme: pre-processing / Feature extraction / Train the algorithm / Test the algorithm.

- **Pre-processing :** In this part we will work on the image to prepare it for the feature extraction.
  - We start by smoothing the image an apply a filter on it. I chose the gaussian filter because I get best results with it (I also tried median filter).
  - We resize the image, so all the images have the same size 256 x 256
  - Normalize effect of illumination by applying histogram equalization

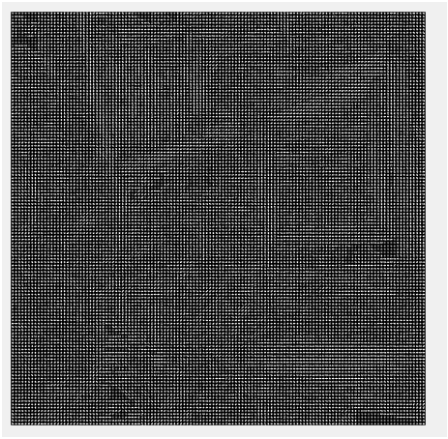
At the end of the process we get this result:

Original Image

Processed image

- **Feature extraction :** We apply two types of feature extraction HOG and LBP in order to compare them later
    - We apply Histogram of oriented Gradient with a cell size of [2 2], set the Block size at [4 4] and Block overlap at [2 2]. These parameters were chosen to get best results
    - For Local Binary Pattern, no special modifications were applied.

Here a visualization of the HOG feature on the image.

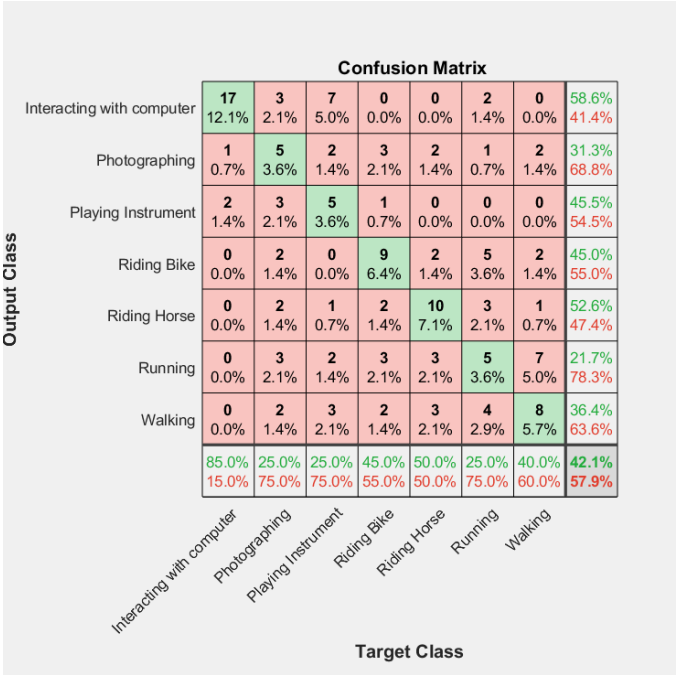


- **Train :** In this part we train KNN and SVM algorithms with train data
    - **KNN :** We choose 10 as number of neighbors and set standardize feature on
    - **SVM :** We use fitcecoc function which is applied for multi class classification. We choose a coding of one vs all that gets best use result .
- **Test :** Test the algorithms by predicting results of test dataset. Analyze the result with the confusion matrix.

## Results:

This section is dedicated for results discussion. As we have a lot of parameters multiple combination were made.

First result is by following all the steps described above with HOG features and SVM. I get 42,1% of accuracy. We can see that running images are the hardest images to classify.




Now we try LBP features with SVM. I get an accuracy of 42.1% which the same as HOG method.

For HOG and KNN. We get 22.9% of accuracy which is very low. It is explained because this type of data is not suitable for KNN.

Finally, we have LBP with KNN we get 26.4%. KNN has better results with LBP method

## Conclusion:

Overall, the results are quite low under 50%. HOG + SVM and LBP+SVM have the same results but the main difference is the execution time (as we can see in fig1). HOG feature is more time consuming than LBP method. This can be explained by the fact that HOG method extracts more values 571 3356 compared to 59 of LBP which takes more time to create the model. Results can be optimized but due the amount of parameters it is hard to do it automatically with our machines because it will take too much time. KNN is not suited for this problem as results are too low. A solution can be to reduce the dimension of input data to make faster the processes and get better results.

| Function Name | Calls | Total Time (s) |
|---|---|---|
| Action_Recognition | 1 | 10.069 |

Execution time LBP + SVM

| Function Name | Calls | Total Time (s) |
|---|---|---|
| Action_Recognition | 1 | 237.680 |

Execution time HOG + SVM

Fig1: Execution time comparison