Final Project: Activity Classification using MHI

## Description

For this topic, you will implement the methods presented in the lectures to create Motion History Images (MHIs) and use these images to perform activity classification in video. Your algorithm must take in a video of a human walking, jogging, running, boxing, waving or clapping, and correctly classify the behavior. Intermediate steps in your classifier might include generating features from motion history images and some form of machine learning on those features to classify the activity.  
  
Sample Dataset: <http://www.nada.kth.se/cvap/actions/>   
Related Lectures (not exhaustive): 8A-8D, 9A-9B

## Problem Overview

**Methods to be used:** Implement methods based on Motion History Images to recognize and classify human motion behavior. These methods must not contain pre compiled functions that create MHIs or compute Hu moments. Instead, you are required to apply the knowledge acquired from your research. You can use machine learning functions available only in OpenCV and Scikit-Learn.

**RULES**:

* **Don’t use external libraries for core functionality**You are encouraged to use libraries while writing code for your final report. However you will receive a low score if the main functionality of your code is provided via an external library. Don’t use functions that perform Motion History operations in images and Hu moments functions available in OpenCV or any other library.
* **Don’t copy code from the internet**The course honor code is still in effect during the final project. All of the code you submit must be your own. You may consult tutorials for libraries you are unfamiliar with, but your final project submission must be your own work. Any instance that does not follow the Honor Code and the class rules will be directly reported to the Office of Student Integrity.
* **Don’t use pre-trained machine learning models**If you choose a topic that requires the use of machine learning techniques, you are expected to do your own training. Downloading and submitting a pre-trained model is not acceptable for any project topic.
* **Don’t rely on a single source**We want to see that you performed research on your chosen topic and incorporated ideas from multiple sources in your final results. Your project must not be based on a single research paper and definitely must not be based on a single online tutorial.

**Please do not use absolute paths in your submission code. All paths must be relative to the submission directory. Any submissions with absolute paths are in danger of receiving a penalty!**

## Programming Instructions

You may use the python 2 environment that you have been using for all the assignments or the [python 3 environment provided for the project](https://gatech.instructure.com/courses/26111/files/folder/Project Files?cv_proj.yml). This new environment is simply a list of the versions of libraries that will be used during grading. You may install them however you wish. We recommend conda. Include a README.md file with usage instructions that are clear for the grader to run your code. Remember to specify what version of python you are using. Notice that despite having Tensorflow and Pytorch in the environment, you are not allowed to use them.

Windows Users Warning:

Be warned that TA’s grade exclusively on linux machines. Thus, it is your responsibility to make sure that your code is platform independent. This is particularly important when using paths to files. If your code doesn’t run during grading due to some incompatibility you will incur a heavy penalty.

## Write-up instructions

The report must be a PDF of 3-6 pages including images and references. Not following this requirement will incur in a significant penalty and the content will be graded up to page 6. **Note that the report will be graded subject to a working code.** There will be no report templates provided with the project materials.

The report must contain:

1. A clear and concise description of the algorithms you implemented. This description must include references to recently published computer vision research and show a deep understanding of your chosen topic.
2. Results from applying your algorithm to images or video. Both positive and negative results must be shown in the report and you must explain why your algorithm works on some images, but not others.
3. Performance statistics obtained by applying your algorithm to a public imagery or video database. You are expected to determine appropriate quantitative performance metrics based on your own research.
4. A technical discussion of how your results compare to the state of the art and how your results could be improved.

You report must be written to show off your work and demonstrate a deep understanding or your chosen topic. The discussion in your report must be technical and quantitative wherever possible.

## Video presentation:

Present your work in a video (screen, camera / cellphone recording) showcasing your work and results. This video must not be more than 3.5 minutes long (3:31 will not be accepted). **Videos longer than this will not be watched and therefore not graded.** Submit a link to your video hosted somewhere the grader can access it. We recommend providing multiple links in case one of them does not work or is removed. We will only use the link you provide at the time of submission. **Please do not send private posts with updated links as these will not be used.**

## How to submit

Unlike the class assignments, **you must submit your project files in one compressed (.zip) folder via Canvas.** Find the assignment labeled “Final Project”, attach all the required files, click on submit. Late submissions will not be graded so plan accordingly.

**Important: Submissions sent to Bonnie, Piazza or anything that is not Canvas will not be graded..**

## Grading

The report will be graded following the scheme below:

* Code (20%): We will verify that the methods and rules indicated above have been followed.
* Report (70%): Subject to a working code.
  + Description of existing methods published in recent computer vision research.
  + Description of the method you implemented.
  + Results obtained from applying your algorithms to images or videos.
  + Analysis on why your method works on some images and not on others.
  + Performance statistics analysis.
  + Discussion on how your results compare to the state of the art methods.
  + Proposals on how your methods can be improved.
  + References and citations.
* Video Presentation (10%)

## Assignment Overview

Use the following information as an introduction to the theory behind Motion History Images. Keep in mind that you will need to read more about this topic in order to achieve an activity recognition classifier. Part of the research requirement is to find how each parameter works and apply this knowledge accordingly.

### 1. Method requirements

Create a method to obtain a binary motion signal to analyze over time. Extract the person from the background and use the area the person occupies as a signal. To create an MHI you first need to compute the frame difference sequence. The binary image Bt (x, y) is defined simply as:  
Screen Shot 2015-04-06 at 3.11.29 PM.png  
θ needs to be chosen such that it tends to capture the motion in the right places. You might want to “clean up” the binary images using a morphological OPEN operator to remove noise.   
  
Given the sequence Bt we can construct the MHIs. The Motion History Image M at time t is defined as:  
Screen Shot 2015-04-06 at 3.16.20 PM.png

τ needs to be chosen such that it captures the full extent of the action. Actions can be defined as lifting the arms from bottom to top, wave back-and-forth or similar. τ can be different for each action. And it is up to you to find this value and how to define each action.

Use Hu moments to characterize the MHI. Use both the unscaled and scaled central moments µpq  and νpq . The equations for µpq  are defined with respect to the regular moments:  
Screen Shot 2015-04-06 at 3.26.58 PM.png  
Given this definition, the average *x* and *y* are:  
Screen Shot 2015-04-06 at 3.27.54 PM.png  
and the central moments are defined as:  
Screen Shot 2015-04-06 at 3.29.08 PM.png  
Under this definition, note that µ10  and µ01  are both zero.  
  
To achieve scale invariance we define the scale invariant moments:  
  
  
Using all the moments for <*pq*> ∈ {20, 11, 02, 30, 21, 12, 03, 22} as features it is possible to find the “nearest” action that is recognized using methods like nearest neighbor or other machine learning method of your choice.

### 2. Input Requirements

Use images from videos found in datasets such as [Recognition of human actions](http://www.nada.kth.se/cvap/actions/). The amount of video frames needed to train your model is up to you. The activity classification method must take in a video of a human subject walking, jogging, running, boxing, waving or clapping, and correctly classify the behavior as they happen. The input must also include videos where the subject is performing multiple actions.

### 3. Final Results

Present your results using sample frames including their action label. Include a link where the output video can be viewed.

For each frame, present the predicted label using image annotations. This means your method must predict what action the person is performing during the entire video.

#### 3.1 Video Results

Sample video containing multiple actions where each is labeled. Include the link to this video in your report and in the presentation. This means you will provide two video links: one for the action classification and one for the video presentation.

#### 3.2 Image Results

Include sample images in your report that showcase your method’s performance. Make sure you present all cases mentioned in the project instructions. As part of your analysis present your results using confusion matrices and classification error plots.