Activity Classification Using Motion History Images

Yang Hu
whoyoung99@gmail.com

1 Introduction

The report re-constructs the method firstly proposed by Dr. Bobick [1], and attempts to involve other state of the art machine learning classification models, such as support vector machine (SVM) and convolution neural network (CNN), in order to build a scalable architecture which could be applied to real life. To be more specific, the system presented in this report isolates the "classification part" from "MHI frame processing part", so the model can quickly adapt to a new field of application by feeding a different classifier without other major changes to the code.

This report is organized as follows: Section 2 introduces the overall structure of my implementation of MHI motion recognition system. Section 3 presents the result of the system with two different classification model: SVM and CNN, followed by explanations about the positive/negative classified cases. Section 4 sums up the limitations of the MHI-based system.

The training dataset being used within the report is from Nada University in Swedish, called "Recognition of human actions". The database contains 25 different subjects performing 6 types of motions, e.g., running, jogging, walking, hand clapping, hand waving, and boxing, from 4 different shooting angles of view. Ideally this should make up a total of 25 x 6 x 4 = 600 videos, but there's 1 video missing from "hand clapping" class (person #13, viewing angle d3 from class "hand clapping" to be precise). And for the sake of consistency, I duplicate the video of same person from angle d2 within same class to make up the deficiency.

2 Implementation Structure

This section presents the technical details in the submitted source code. Effectiveness of various parameters is analyzed. Research on alternatives of update function is surveyed. Finally, different feature extraction methods are pointed out.

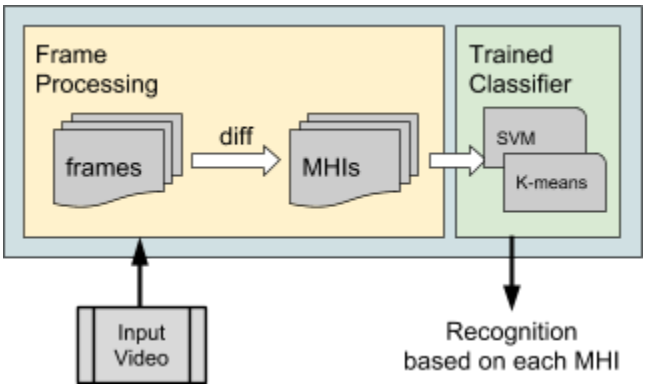


Fig. 1 Components of recognition system

Figure 1 shows the general structure of the basic MHI approach recognition system. Since frame processing is independent to the latter classification part, isolating these two makes whole system become highly adaptive to different use cases. Users can focus on perfecting the classification model while the rest of the part remains still.

2.1 MHI templates

Note: Since this report aims for an in-depth illustration to the submitted source code, only the actual implementations are covered instead of mathematical expression. For mathematical formulae, please refer to the original paper [1].

The MHI method is a template matching approach [5], that being said, a motion in the video (or a series of consecutive images) is first converted into one single static shape pattern, and then it is compared with other known action patterns during recognition. As a result, the first part of MHI-based recognition system is how to compute MHIs from videos.

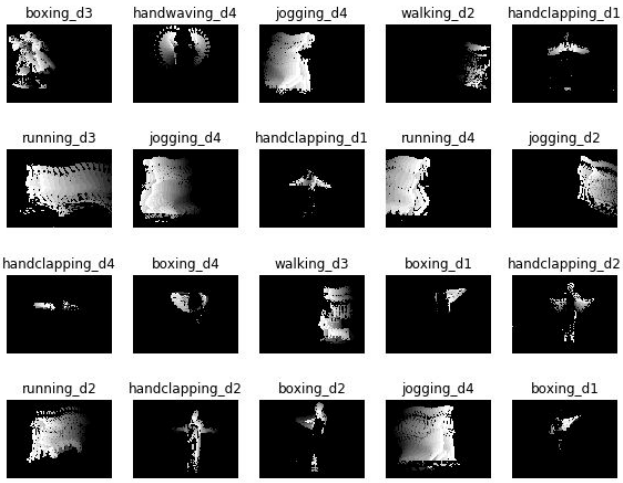
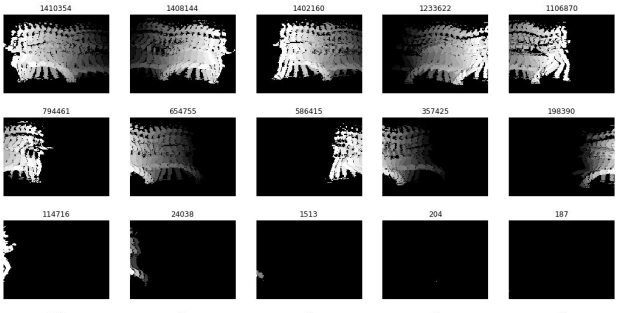


Fig. 2 MHIs of different motions (τ=15)

A motion-history image represents the moving area in the frame within a period of time. At the beginning, t=1, no motion occurs since there's only one image, so a MHI is initiated as all zeros with the shape of the first frame dimension. At t=2, I need an "update function" D(x, y, t) to let me know which pixels are moving. The value of D function is binary, either "1" indicating the pixel is moving or "0" representing the still pixel. Then I can compute the MHI based on the value of D; if D equals "1" then marked the corresponding position on MHI as τ, otherwise subtract δ (normally be 1).

Figure 2 shows the MHIs of different motions. You can find the code generating this figure in the provided ipython notebook. The next step is to extract some major MHIs from a video. By saying "some", I am implying that a video should contain more than one MHI due to the characteristic of MHI's self-overwritten behavior. The previous motion information is overwritten or deleted by later actions along with the time passing on. Imaging a video with a man walking from right through the scene and coming back into the scene again from left side. Then for this video, it should have at least



2 MHIs (walking toward left and toward right). And here comes the question: How to automatically find the MHIs that is representative?

Fig. 3 Auto search the dominated MHIs

In the report I propose a simple but effective way to search the major/dominated MHIs within a video without human intervention. Figure 3 shows a series of MHIs generated from person09_running_d3_uncomp.avi. It is an excellent example for illustrating the idea because the subject (person) passes the scene within a second, so most of the time the scene on the screen is nothing but still background. In order to get the non-black MHIs, intuitively we can compute the sum the each MHIs and assuming that the greater the value, the more likely it is a good MHI. Figure 3 shows the the top-15 high summention MHIs computed from different time steps. As you can see, by sorting a series of MHIs with their image sum and select top-n items, I am able to automatically extract MHIs that are representative to the motion itself.

2.2 Update function *D(x, y, t)*

In this report, the update function D is simply the difference of two consecutive images (frames). Due to the data type of images are unsigned, extra correction for subtraction underflow is required.

Apart from "frame-to-frame differencing", another way to determine whether a pixel is moving is by using the "optical flow fields". It is especially effective in the scenario when background is moving as well. Naive frame difference method usually don't perform well on target segmentation if the background is not still. However, the dataset used for this report has relatively low fps rate (25 fps), the movement of pixels between two frames is too big for constructing a promising workable optical flow.

2.3 Features and classification

In the original paper [1], 7 Hu moments were used for feature extracting. Although the vector is known for the feature of invariant, the

process of condensing (120, 160) pixel value into just 7 numbers will inevitably lose information. Not to mention after plotting the log value of Hu moments, there's no single feature that is significant enough to distinguish between different motions. As a result, the SVM and CNN model was trained using the whole MHI images, flatten to 1-D as training data.

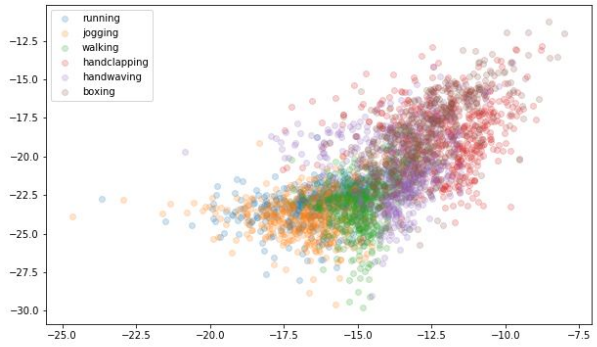


Fig. 4 Projection of Hu moments (#1, #2)

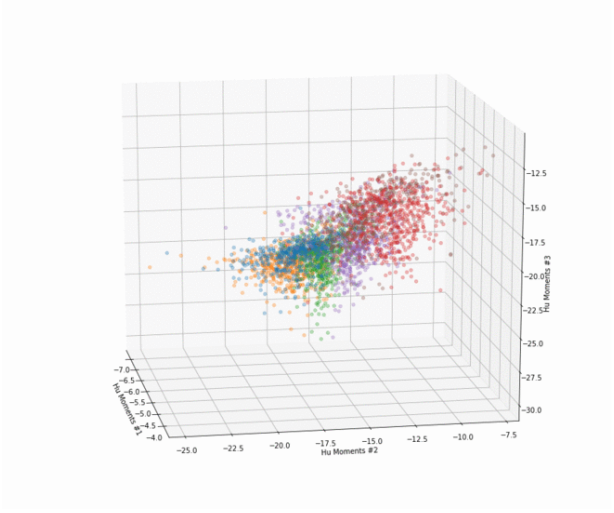


Fig. 5 Projection of Hu moments (#0, #1, #2)

2.4 System demonstration

The motion recognition system in file "MotionDetector.py" will load a sklearn SVM model (pre-trained by myself) before starting motion recognition. As a result, you need to have scikit-learn libraries installed on your environment before executing.

As shown in figure 6, the script will use cv2.imshow() to "play" the video. There will be two screens; on top the screen shows the original video frame with labeled motion and a rectangle to indicate the interest of region, on the bottom shows the MHI images at that specific time step. For each frame, the system will take the current MHI to "ask" the trained SVM model, to see if there's a match. The predicted result return from the SVM model will be printed out on top-left corner with "recognized motion" and its probability (confidence level).

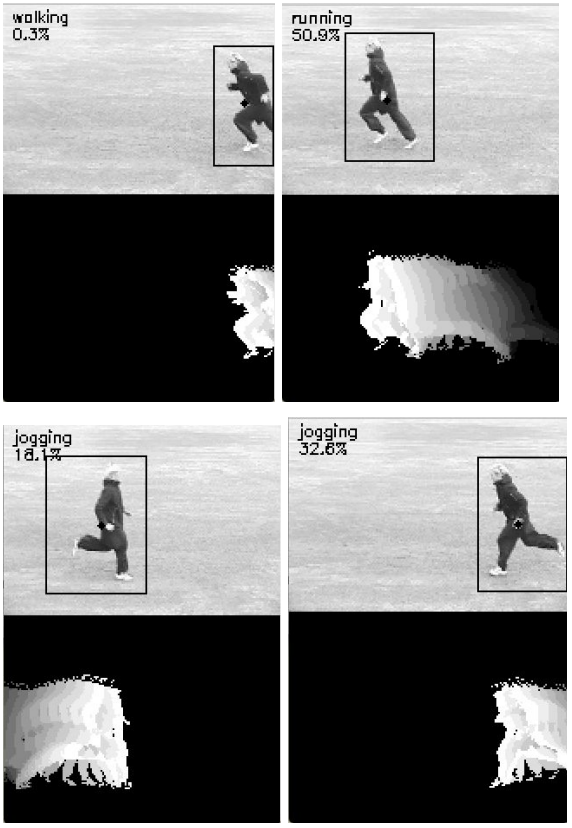


Fig. 6 Motion recognition system output

Although all four images is from same input video, one may notice that at different time step, the recognized motion will vary. Take figure 6 as example, at the very beginning, the subject is first classified as "walking", then with a longer time frames as "running", sometimes "jogging". I will not consider this as false recognition or wrong prediction. On the contrary, this reflects the nature of ambiguity of some motion naming, and the deficiency of the training dataset. How to quantify the difference between jogging and running? How could one assume only single motion was conducted throughout the entire video?

3 Classification model

Three classification model were experimented and the SVM model is chosen for demo purpose. Both SVM and CNN model will be attached for submission.

3.1 Motion Detection

From previous Section, we notice that (i) the system points out the limitation of MHI-based recognition system, and also (ii) the potential defects in training dataset. First, one major limitation of such system lacks the ability to discriminate among similar motions, e.g. running versus jogging versus walking. Secondly, in the given dataset, some subjects perform different motion pattern than the majority although they are still performing the motion they believe they are, e.g. running while keeping the upper body straight (like how jogging does) instead of leaning forward.

Based on the believe that training data is questionable, I choose the most generalized model as the final classifier instead of the one with highest accuracy in training stage since it is certainly overfitting. The overall accuracy is acceptable: 79% accuracy on general model, and the accuracy gets higher if assuming the angle of view is known to each input videos.

```
Accuracy: 0.788
Confusion matrix:
[[18  5 10  3  0  0  2  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 3 19  3  0  2  2  0  0  2  1  0  0  0  0  0  0  0  0  1  0  0  0  0]
 [ 4  0 30  4  1  1  3  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 3  0  2 24  3  0  4  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  5  3  6 13  2  6  1  3  1  2  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 16  0  0  0 18  2  0  3  9  0  0  0  0  0  0  0  0  3  0  0  0  0]
 [ 2  4  5  5  8  0 15  5  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  6  2  0  0 23  0  1  0  7  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  2  0  2 15  3  3  6  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  2 20  0  0  0  0  0  1  0  0  0  2  0  0  0]
 [ 0  2  0  1  6  1  3  0 14  1 10  1  0  0  0  0  0  0  0  0  1  0  0]
 [ 0  1  0  0  0  1  0  3  5  1  1 27  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 33  1  6  4  0  0  0  0  2  0  0  1]
 [ 0  0  0  0  0  0  0  0  0  1  0  0  0  1 28  2  1  0  0  1  1  0  3  0  1]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  9  2 32  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  8  1  1 37  0  0  0  1  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  1  1 27  5  4  5  0  1  0  0]
 [ 0  0  0  0  0  0  1  0  0  0  0  0  0  1  4  0  2  4 15  1  2  0  7  0  1]
 [ 0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  1  2  8  3 26  4  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  3  1 38  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  3  0  0  0  0  0 30  1  3  2]
 [ 0  0  0  0  0  0  0  0  1  1  0  1  0  2  0  0  0  2  0  0  4 25  2  4]
 [ 0  1  0  0  0  1  0  0  0  0  1  0  2  0  1  0  0  0  2  2  4  0 25  4]
 [ 0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  3  6  2  1 26]]
```

Fig. 7 Confusion matrix of finalized SVM model

```
Direction: d1 Accuracy: 0.845
Confusion matrix:
[[40 16  1  0  0  0]
 [ 1 43 14  0  0  0]
 [ 0  8 47  0  0  2]
 [ 0  0  1 66  0  3]
 [ 0  0  0  4 60  2]
 [ 0  0  0  6  0 61]]
Direction: d2 Accuracy: 0.731
Confusion matrix:
[[30 18  7  1  1  0]
 [16 25 15  0  0  2]
 [ 1  9 46  0  0  1]
 [ 0  1  1 60  5  3]
 [ 2  3  1  6 52  2]
 [ 0  0  2  4  0 61]]
Direction: d3 Accuracy: 0.789
Confusion matrix:
[[40 10  7  0  0  0]
 [11 31 16  0  0  0]
 [ 0  6 48  1  1  1]
 [ 0  0  0 60  2  8]
 [ 0  0  0  4 60  2]
 [ 0  0  1  5  4 57]]
Direction: d4 Accuracy: 0.864
Confusion matrix:
[[46 11  0  0  0  0]
 [ 5 36 17  0  0  0]
 [ 0  9 48  0  0  0]
 [ 0  0  0 68  0  2]
 [ 0  0  0  0 64  2]
 [ 0  0  2  3  0 62]]
```

Fig. 8 Confusion matrix of separate angle of view

One may notice that videos shooting from direction d2 have the worst accuracy, even worse than general model. This results from the continuously zoom-in and zoom-out of the scene in those d2 videos. For motions that actually travel a distance like running, jogging, and walking, the d2 was made by asking subjects to run diagonally instead of horizontally; for motions that stay at where they suppose to be like hand waving, hand clapping, and boxing, instead of recording from 45 degree angle, the creator of the dataset use zoom-in and zoom-out to imitate the changes of viewing angle. The zoom-in/out will create a series of weird MHIs that focused on the change of body size rather than what they are actually doing (motion). That's the main reason for direction d2's poor performance.

```
model = Sequential()
model.add(Conv2D(16, (3, 3), padding='same', input_shape=(120, 160, 1)))
model.add(Activation('relu'))
model.add(Conv2D(16, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
# model.add(Dropout(0.05))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dense(512))
model.add(Activation('relu'))
# model.add(Dropout(0.05))
model.add(Dense(6))
model.add(Activation('softmax'))
```

```
learning_rate = 0.001
learning_decay = learning_rate/32

early_stop = EarlyStopping(monitor='val_acc', patience=3)
sgd = SGD(lr=learning_rate, decay=learning_decay, momentum=0.)
adam = Adam(lr=learning_rate, decay=learning_decay)
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])

model.summary()
```

Fig. 9 Structure of CNN model (overall Acc: 42%)

3.2 Tracking window

In this section, a skill taught in lecture is performed to find the rectangle contour of the moving subject. The tracking window greatly help visualize the area of "differencing", i.e. the difference of t=n and t=n+1. It is computed as shown below in figure 10: for each MHI, select the highest value region, which indicates the motion of the latest motion, then a 20x20 kernel is used for blurring to create a mosaic-like effect, then take the binary of it will render an area of interest. Finally, use the moments to compute the center coordinates of the bounded contour in order to get the height and width of the tracking window.

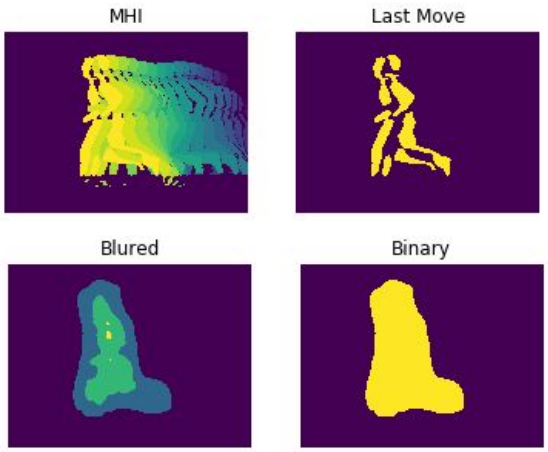


Fig. 10 Steps of creating detecting window

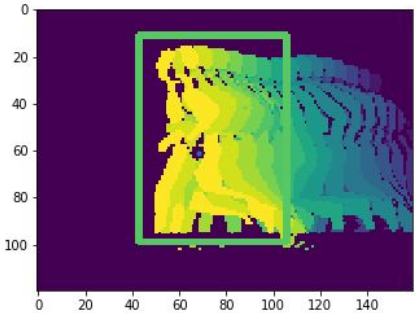


Fig. 11 Draw rectangles based on blurred blob

4 Limitations of MHI-based System

A key element of MHIs based motion system is segmentation. In order to find the region of interests, the system needs to have the capability of sementing regions corresponding to the subject from the rest of the image. It is vital for such system because the subsequent processes such as tracking and action recognition are greatly dependent on the performance and proper segmentation of the region of interest [3]. As a result, the system is suited the best for scenarios without any background movement; in that case, the background subtraction would be trivial. However, even if the we could have a certain still background, segmentation could still be tricky sometimes.

As mentioned, when differencing frames to determine where do motions happen, only diff values higher than a threshold would be considered as moving. This intents to filter out unwanted noise from the background. However, this procedure will certain erase some information from the target of interest. Figure 12 shows the scenario when it is hard to distinguish the background from the region of interests after thresholding the diff image. In this case, the woman is wearing a white cloak which happens to have similar pixel value with the background. It renders a relatively bad motion pattern, comparing with other shown in figure 13. Therefore, the selection of both update function and its threshold are very crucial for calculating motion history templates.

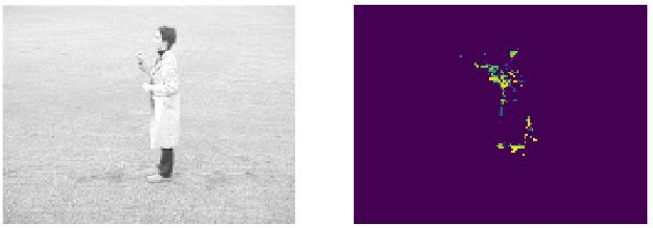


Fig. 12 Woman with white cloak boxing

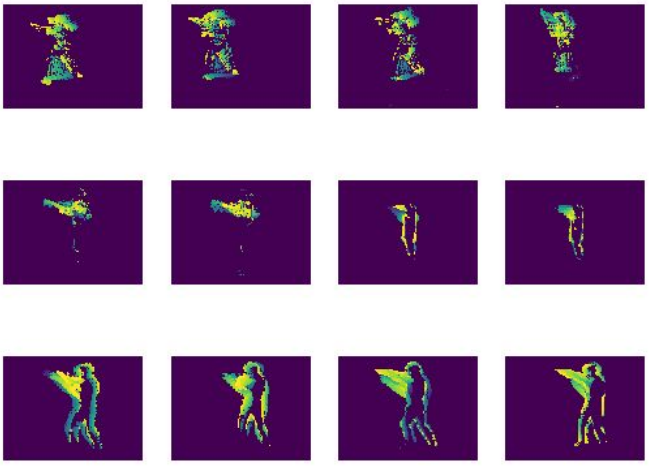


Fig. 13 Representative boxing MHIs

References

[1] J.Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proc. CVPR*, pages 928-934, 1997.

[2] C. Schuldt, I. Laptev and B. Caputo. Recognizing Human Actions: A Local SVM Approach. In *International Conference on Pattern Recognition*, 2004.

[3] Md. Atiqur Rahman Ahad, J. K. Tan and H. Kim. Motion history image: its variants and applications. In *Machine Vision and Applications 23*, pages 255-281, 2012.
[4] S. Arseneau, J.R. Cooperstock. Real-time image segmentation for action recognition. In *Proc. IEEE Pacific Rim Conf.* on *Communications, Computers and Signal Processing*, page 86-89, 1999.
[5] J.K. Aggarwal, Q. Cai. Human motion analysis: a review. In *Computer Vision Image Underst.* 73, page 428-440, 1999.