Table of Contents

Human Activity Recognition

ABSTRACT

HAR stands for Human Activity Recognition. Local space-time features capture local events in video and can be adapted to the size, the frequency and the velocity of moving patterns. In this project, we demonstrate how such features can be used for recognizing complex motion patterns. We construct video representations in terms of local space-time features and integrate such representations with KNN classification schemes for recognition. For the purpose of evaluation we introduce a new video database containing 2391 sequences of six human actions namely boxing, walking, running, hand clapping, jogging and hand waving. The presented results of action recognition justify the proposed method and demonstrate its advantage compared to other relative approaches for action recognition.

1. Introduction

Applications such as surveillance, video retrieval and human-computer interaction require methods for recognizing human actions in various scenarios. Typical scenarios include scenes with cluttered, moving backgrounds, non- stationary camera, scale variations, individual variations in appearance and cloth of people, changes in light and view point and so forth.

In this project, we demonstrate that action recognition can be achieved using local measurements in terms of Spatio Temporal Interest Points (STIP). Such features capture local motion events in video and can be adapted to the size, the frequency and the velocity of moving patterns, hence, resulting in video representations that are stable with respect to corresponding transformations.

1.1 Purpose

Activity recognition aims to recognize the actions and goals of one or more agents from a series of observations on the agents' actions and the environmental conditions. Since the 1980s, this research field has captured the attention of several computer science communities due to its strength in providing personalized support for many different applications and its connection to many different fields of study such as medicine, human-computer interaction, or sociology.
Many different applications have been studied by researchers in activity recognition; examples include assisting the sick and disabled. For example, by automatically monitoring human activities, home-based rehabilitation can be provided for people suffering from traumatic brain injuries. One can find applications ranging from security-related applications and logistics support to location-based services. Due to its many-faceted nature, different fields may refer to activity recognition as plan

recognition, goal recognition, intent recognition, behaviour recognition, location estimation and location-based services.

1.2 AIM AND OBJECTIVE

To recognize the actions and goals of one or more agents from a series of observations on the agents' actions and the environmental conditions. Detection of human activities through video processing can solve numerous real life problems. To solve these, a technology that can prove essential in solving problems related to this field must be developed**.**

2. Review of Literature

2.1 Introduction to MATLAB

MATLAB (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages,
including C, C++, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPad symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

2.2 MATLAB structures

MATLAB has structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations, etc.). Unfortunately, MATLAB JIT does not support MATLAB structures; therefore just a simple bundling of various variables into a structure will come at a cost.

2.3 MATLAB functions

When creating a MATLAB function, the name of the file should match the name of the first function in the file. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores.

3. Proposed approach and system architecture

3.1 Proposed System

To develop an automated mechanism and interpretation that detects activities and positional data from live video footages, recognizes and labels them for further analysis. Evaluating human activity recognition systems usually implies following expensive and time consuming methodologies. We propose an evaluation methodology to overcome the enumerated problems.

3.2Work Done

3.2.1 Data Collection

To test and evaluate the proposed algorithms and methods for automated activity analysis and summarization in eldercare, we need to build a large dataset of activity monitoring videos.This provides a sufficiently large amount of data for algorithm testing and performance evaluation. Once the algorithms and technologies in this research project are proven to be effective and robust, as our final objective, we can deploy cameras to recognize activities.

3.2.2 SILHOUETTE EXTRACTION AND HUMANTRACKING

Silhouette extraction, namely, segmenting a human body or objects from a background, is the first and enabling step for many high-level vision analysis tasks, such as video surveillance, people tracking and activity recognition.For effective privacy protection and accurate activity analysis, we need the silhouette extraction to be accurate and robust. A number of efficient silhouette extraction algorithms have been developed in the literature

3.2.3 Adaptive Background Update

In silhouette extraction within a dynamic video scene, we need to continuously update the background model by incorporating background changes. A commonly used method to update background is that, if an object or image area remains stationary for a certain period of time, it is considered to be background.To solve this problem, we propose to utilize high-level knowledge about human motion as a guideline to perform adaptive update of the background model.

3.2.4 HUMAN ACTION RECOGNITION ANDACTIVITY ANALYSIS

Due to the non-rigidness of human body, the wide variety of possible human actions, as well as the deep structure of human activities, human action recognition remains a challenging task.In this work, we aim to develop a low-complexity, efficient, and robust scheme for action recognition. Our objective is to convert the input activity monitoring video stream into meaningful information, such as ADL (Activity of Daily Living) statistics , for functional assessment. To extract this type of information, we must first identify the actions being performed in the input videos, including walking, sitting on the couch, standing up, sitting at the dining table, preparing meals in the kitchen, visiting the bathroom, going outdoors, etc.

3.3 Architecture of Proposed System

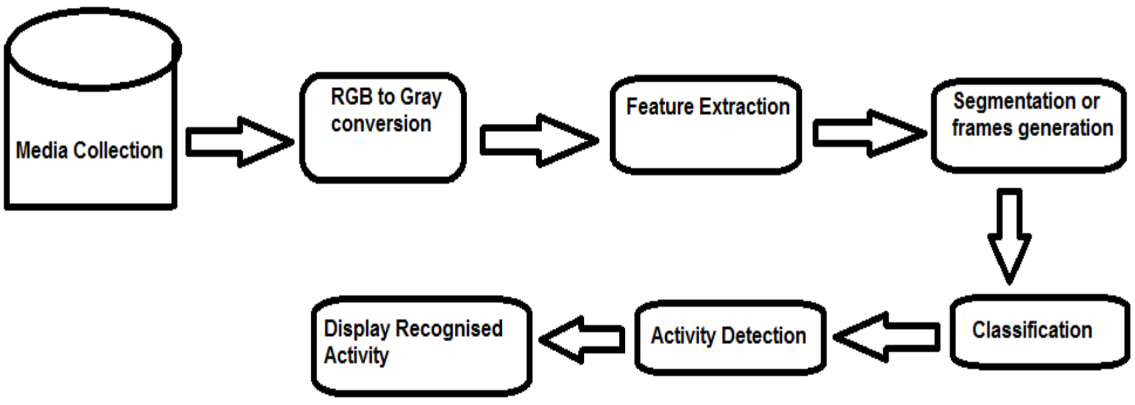


Fig (i)- Pipeline of HAR

3.3.1Architecture Explained

All the videos are collected and stored in a folder (dataset) which has six videos namely jogging, walking, running, hand clapping, hand waving and boxing. Any video can be selected by the user for activity detection. Selected video is converted into black and white (grey scale) format so that segmentation can be done. Features of that video are extracted and 50 intermediate frames are generated. These frames are nothing but the still images of the video.

Then, using KNN (K-nearest neighbour) classification algorithm, sample datasets are classified according to frequency, velocity, etc of videos. Initially K=1, which means that distance from Kth position is calculated and shortest distance between kth point and sample point is returned as the result i.e. the activity performed. This detected activity is then displayed on the screen.

3.3.2 Future Applications

1.Surveillance

Ubiquitous cameras in public places (e.g. CCTVs).

Goal:

- Monitor suspicious activitiesfor real-time reactions.
- ‘Fighting’, ‘stealing’.
- Currently, surveillance systems are mainly for recording

Activity recognitionis essential for surveillance and other monitoring systems in public places.

2.Intelligent environments (HCI)

Intelligent home, office, and workspace

- Monitoring of elderly people and children.
- Support one’s quality of life.
- Recognition of ongoing activitiesand understanding of current context is essential.

3.Sports play analysis

Analysing the play and deducing the actions in the sport.

Eg. Actions of an Umpire(like out,four,six etc.)

4.A system to enable autistic children

Many times,autistic children cannot communicate their feelings and basic needs.Theirbehaviour seems to be logical but in their own criteria,not typical for people around them**.**

5.YouTube

- 20 hours of videos uploaded every minute.
- Search based on contents of the video, instead of user-attached keywords .
- Content-based search.

4. System Description

4.1 KNN

In pattern recognition the ***k*-Nearest Neighbors algorithm** (or ***k*-NN** for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- In *k-NN classification*, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.
- In *k-NN regression*, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbors.

.

4.1.1 ALGORITHM FOR KNN

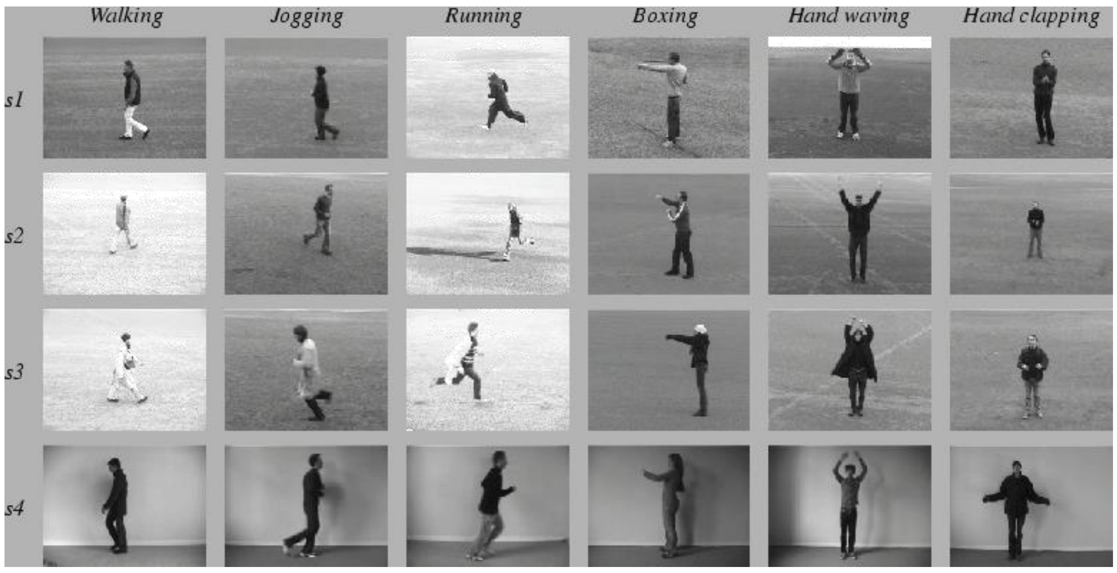
- The training examples are vectors in a multidimensional feature space, each with a class label.
- The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.
- In the classification phase, *k* is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the *k* training samples nearest to that query point.
- A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the **overlap metric** (or Hamming distance).
- In the context of gene expression microarray data, for example, *k*-NN has also been employed with correlation coefficients such as Pearson and Spearman.

- Often, the classification accuracy of *k*-NN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbour or Neighbourhood components analysis.

4.1.2 KTH DATA SET

The current video database containing six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors *s1*, outdoors with scale variation *s2*, outdoors with different clothes *s3* and indoors *s4* as illustrated below. Currently the database contains 2391 sequences. All sequences were taken over homogeneous backgrounds with a static camera with *25*fps frame rate. The sequences were downsampled to the spatial resolution of *160x120* pixels and have a length of four seconds in average.

In our experiments reported in ICPR'04 all sequences were divided with respect to the subjects into a training set (8 persons), a validation set (8 persons) and a test set (9 persons). The classifiers were trained on a training set while the validation set was used to optimize the parameters of each method. The recognition results were obtained on the test set.



All sequences are stored using AVI file format and are available on-line (DIVX-compressed version). Uncompressed version is available on demand. There are *25x6x4=600* video files for each combination of 25 subjects, 6 actions and 4 scenarios. Each file contains about four subsequence's used as a *sequence* in our experiments. The subdivision of each file into sequences in terms *start_frame* and *end_frame* as well as the list of all sequences is given in 00sequences.txt.

4.2 Algorithm :

In computer vision, **maximally stable extremal regions** (**MSER**) are used as a method of blob detection in images. This technique was proposed by Matas et al. to find correspondences between image elements from two images with different viewpoints. This method of extracting a comprehensive number of corresponding image elements contributes to the wide-baseline matching, and it has led to better stereo matching and object recognition algorithms. Consists of 5 steps:

- Step 1: Detect Candidate Text Regions Using MSER
- Step 2: Remove Non-Text Regions Based On Basic Geometric Properties
- Step 3: Remove Non-Text Regions Based On Stroke Width Variation
- Step 4: Merge Text Regions For Final Detection Result
- Step 5: Recognize Detected Text Using OCR

Step 1: Detect Candidate Text Regions Using MSER

The MSER feature detector works well for finding text regions [1]. It works well for text because the consistent colour and high contrast of text leads to stable intensity profiles.

Use **the *detectMSERFeatures*** () function to find all the regions within the image and plot these results. Notice that there are many non-text regions detected alongside the text.

Step 2: Remove Non-Text Regions Based On Basic Geometric Properties

Although the MSER algorithm picks out most of the text, it also detects many other stable regions in the image that are not text. You can use a rule-based approach to remove non-text regions. For example, geometric properties of text can be used to filter out non-text regions using simple thresholds. Alternatively, you can use a machine learning approach to train a text vs. non-text classifier. Typically, a combination of the two approaches produces better results.Use ***regionprops*** to measure a few of these properties and then remove regions based on their property values.

Step 3: Remove Non-Text Regions Based On Stroke Width Variation

Another common metric used to discriminate between text and non-text is stroke width. ***Stroke width*** is a measure of the width of the curves and lines that make up a character. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations.

To help understand how the stroke width can be used to remove non-text regions, estimate the stroke width of one of the detected MSER regions. You can do this by using a distance transform and binary thinning operation.

Step 4: Merge Text Regions for Final Detection Result

At this point, all the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carry more meaningful information than just the individual characters. For example, recognizing the string 'EXIT' vs. the set of individual characters {'X','E','T','I'}, where the meaning of the word is lost without the correct ordering.

One approach for merging individual text regions into words or text lines is to first find neighbouring text regions and then form a bounding box around these regions. To find neighbouring regions, expand the bounding boxes computed earlier with ***regionprops***. This makes the bounding boxes of neighbouring text regions overlap such that text regions that are part of the same word or text line form a chain of overlapping bounding boxes.

Step 5: Recognize Detected Text Using OCR

After detecting the text regions, use the OCR function to recognize the text within each bounding box. Note that without first finding the text regions, the output of the OCR function would be considerably noisier.

5. Results and Discussions

We demonstrated how local spatio-temporal features can be used for representing and recognizing motion patterns such as human actions. By combining local features with K nearest neighbors classification we developed a method for activity recognition that gives high recognition performance compared to other relative approaches.

5.1 Software and Hardware considerations

Technology used: - Machine learning

Tools: - MATLABs.

Operating system: -works on Windows OS (7 or higher).

Hardware: - Processor having minimum 512 GB of RAM, 300 MHz

5.2Future Scope

The Human Activity Recognition software can be enhanced in the future in different kinds of ways such as:

1. Training and recognition speeds can be increased greater and greater by making it more user-friendly.

2. Many applications exist where it would be desirable to classify a larger variation of activities. This would result in many solving many applications with increased efficiencies.

5.3CONCLUSION

Representations of motion patterns in terms of local features have advantages of being robust to variations in the scale, the frequency and the velocity of the pattern. We also have indications that local features give better recognition performance in scenes with complex non-stationary backgrounds and plan to investigate this matter in future work. Using these features, the KNN classifier is able to distinguish all activities with good accuracy. Thus to classify data given as video format it proved beneficial to use the spatio-temporal features that can effectively classify the variations in the different actions.

5.4 References

Under this references section, we have mentioned various references from which we collected our problem and several others that supported us to design the solution for our problem. These references include either books, papers published through some standards and several websites:-

- Algorithm steps from MathWorks.
- IEEE Research paper : Recognizing Human Actions: A Local SVM Approach∗ By Christian Schuldt, Ivan Laptev and Barbara Caputo.
- Wikipedia.com for concepts.
- Matlab documentation.

6. APPENDIX

6.1 Glossary

TERMS

All the terms and abbreviations in the project are specified clearly. For further development of project evolved definitions will be specified.

ACRONYMS

KNN:	*K*-Nearest Neighbours
HAR:	Human Activity Recognizer