Eigenvector-based Dimensionality Reduction for Human Activity Recognition and Data Classification

by

Abdunnaser Abdulhamid Diaf

A Dissertation
Submitted to the Faculty of Graduate Studies
through Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

2012

Your file Votre référence
ISBN: 978-0-494-79307-7

Our file Notre référence
ISBN: 978-0-494-79307-7

Eigenvector-based Dimensionality Reduction for Human Activity Recognition and Data Classification

by

Abdunnaser Abdulhamid Diaf

APPROVED BY:

Dr. Q. Gao, External Examiner
Faculty of Computer Science, Dalhousie University

Dr. I. Ahmad, Outside Reader
School of Computer Science

Dr. M. Ahmadi, Dept Reader
Electrical & Computer Engineering

Dr. J. Wu, Dept Reader
Electrical & Computer Engineering

Dr. B. Boufama, Advisor
Electrical & Computer Engineering

Dr. R. Benlamri, Co-Advisor
Software Engineering, Lakehead University

Dr. F. Simpson, Chair of Defense
Earth & Environmental Sciences

October, 1, 2012

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates materials that is result of joint research, as follows: The entire thesis and all six papers listed below in part II were written with the guidance of my supervisors Dr. Boufama and Dr. Benlamri who provided valuable feedback and editorial input during the writing process. In addition, Section 3.2 is in collaboration with Dr. Ksantini under the supervision of Dr. Boufama and Dr. Benlamri.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publications

This thesis includes parts of 6 papers throughout it that have been previously published in peer reviewed journals and conference proceedings. These papers are totally from my own work during my PhD studies in terms of concepts, design, implementations, and writing. In addition to what is mentioned in part I, my supervisors helped me choosing what research

problems I have to work on, evaluating and giving their feedbacks about my outcomes, reviewing my papers and thesis. The above mentioned papers are as follows:

1. A. Diaf and R. Benlamri. An effective view-based motion representation for human motion recognition. In *Modeling and Implementing Complex Systems, International Symposium on*, pages 57-64, May 2010.

2. A. Diaf, R. Ksantini, B. Boufama, and R. Benlamri. A novel human motion recognition method based on eigenspace. In *Image Analysis and Recognition, ICIAR-2010*, volume 6111, pages 167-175. Springer, 2010.

3. A. Diaf, R. Benlamri, B. Boufama, and R. Ksantini, A Novel Eigenspace-based Method for Human Action Recognition, Proc. of the 5th Int. Conf. on *Digital Information Management - ICDIM2010*, Thunder Bay, Canada, pp.182–187, 2010.

4. A. Diaf, R. Benlamri, and B. Boufama. Nonlinear-based human activity recognition using the kernel technique. In Rachid Benlamri, editor, *Networked Digital Technologies*, volume 294 of *Communications in Computer and Information Science*, pages 342-355. Springer Berlin Heidelberg, 2012.

5. A. Diaf, B. Boufama, and R. Benlamri. A compound eigenspace for recognizing directed human activities. In Aurlio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, volume 7325 of *Lecture Notes in Computer Science*, pages 122-129. Springer Berlin / Heidelberg, 2012.

6. A. Diaf, B. Boufama, and R. Benlamri. Non-parametric fishers discriminant analysis with kernels for data classification. *Pattern Recognition Letters*, (In press), 2012.

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyoneś copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

In the context of appearance-based human motion compression, representation, and recognition, we have proposed a robust framework based on the eigenspace technique. First, the new appearance-based template matching approach which we named "*Motion Intensity Image*" for compressing a human motion video into a simple and concise, yet very expressive representation [23]. Second, a learning strategy based on the eigenspace technique is employed for dimensionality reduction using each of PCA and FDA, while providing maximum data variance and maximum class separability, respectively. Third, a new compound eigenspace is introduced for multiple directed motion recognition that takes care also of the possible changes in scale. This method extracts two more features that are used to control the recognition process. A similarity measure, based on Euclidean distance, has been employed for matching dimensionally-reduced testing templates against a projected set of known motions templates.

In the stream of nonlinear classification, we have introduced a new eigenvector-based recognition model, built upon the idea of the kernel technique. A practical study on the use of the kernel technique with 18 different functions has been carried out. We have shown in this study how crucial choosing the right kernel function is, for the success of the subsequent linear discrimination in the feature space for a particular problem. Second, building upon the theory of reproducing kernels, we have proposed a new robust nonparametric discriminant analysis approach with kernels. Our proposed technique can efficiently find a nonparametric kernel representation where linear discriminants can perform better. Data

classification is achieved by integrating the linear version of the NDA with the kernel mapping. Based on the kernel trick, we have provided a new formulation for Fisher's criterion, defined in terms of the Gram matrix only.

Keywords: Learning; Classification; Human Motion Recognition; Video Compression; Kernel Trick; Eigenproblems; Principal Component Analysis; Fisher's Discriminant Analysis; Nonparametric Discriminant Analysis.

To the martyrs of Feb 17's Revolution who sacrificed their lives and wealth for the sake of ALLAH to rid the people of Libya from one of the worst regimes in the twenty-first century. I ask ALLAH to accommodate their immaculate souls in the elevated rooms of Paradise, Amen.

Acknowledgement

In the beginning I would like to express my gratitude and thanks to my supervisors Dr. Bubakeur Boufama and Dr. Rachid Benlamri. They gave their best in providing me with help, guidance, and a stimulating and relaxed environment throughout the years of my studies. The discussions that I had with Boufama and Benlamri helped me having and developing many ideas in this thesis. However, their support was much more than just scientific.

I gratefully acknowledge the valuable financial support from my beloved country, Libya, via the Ministry of Higher Education and Scientific Research for the period of Fall-2007 to Summer-2011 without which this work would not have been possible.

My greatest thanks to my wife Sana for her understanding, love, support, and encouragement during the past few years without whom I would most certainly be lost.

Last but not least two people I will certainly never forget: All my love to my parents, Abdulhamid and Njaima who dedicated their lives, health, and wealth "just as scented candles" for nothing but to see me and my siblings prosperous and healthy.

Contents

List of Figures

List of Tables

List of Abbreviations

AB	AdaBoost
AB_R	Regularized AdaBoost
FDA	Fisher's Discriminant Analysis
HFDA	Heteroscedastic Fisher's Discriminant Analysis
KDA	Kernel Discriminant Analysis (or LDA)
KFDA	Kernel Fisher's Discriminant Analysis
KNDA	Kernel Non-parametric Discriminant Analysis
KNPDA	Kernel Non-Parametric Discriminant Analysis
KPCA	Kernel Principal Component Analysis
KSDA	Kernel Subclass Discriminant Analysis
KSVM	Kernel Support Vector Machines
LDA	Linear Discriminant Analysis (or FDA)
MEI	Motion Energy Image
MHI	Motion History Image
MII	Motion Intensity Image
MoG	Mixture of Gaussians
NDA	Non-parametric Discriminant Analysis
PCA	Principal Component Analysis
RBF	Radial Basis Function
SVM	Support Vector Machines

Chapter 1

Introduction

This introductory chapter contains a presentation of the background and motivation for the work documented in this thesis. It defines the problem, outlines the objectives and achievements of this work, and provides an overview of the thesis. The presentation is brief in nature as the details of the background material are placed in the next chapter.

1.1 Computer Vision and Machine Learning

As an area of artificial intelligence, machine learning is a procedural fashion of computation that aims to provide computers the ability to predict and behave automatically based on learning a large set of observed examples and with no need for case-programming. Pattern recognition is an active subfield of machine learning and is concerned with determining labels or classes that best define unseen data samples. Data classification in computer vision represents a primary goal in the areas of pattern analysis and recognition, and statistical machine learning. Being humans, we have no ability to recognize a new object or a new

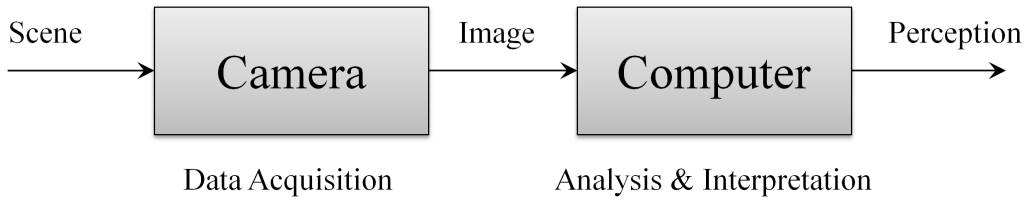


Figure 1.1: Computer Vision

action unless we have seen, trained, or experienced it. Intelligent recognition systems in computer vision must have such experience as well. This can be achieved through two basic procedures, learning and recognition. As machine learning has nothing to do with how to acquire data from sensors, computer vision comprises more comprehensive tasks than just learning and recognition. Computer Vision is the science of analysis of an image or image sequence of a real 3-D scene and development of the theoretical and algorithmic basis in order to achieve results similar to those as by humans. Computer vision has emerged as the discipline that focuses on the following issues:

- How data is obtained from the outputs of visual sensors?
- What information should be extracted from this data?
- How is this information extracted?
- How should this information be represented?
- How must the information be analyzed to allow an automatic system to perform its task?
- How are the analysis results used to provide the final interpretation?

1.2 Human Motion Recognition

This work focuses on the problem of *human motion analysis and recognition* in videos. The main goals of human motion recognition are centered on analyzing and interpreting video images to give machines the ability to tell what action is being performed by one or more humans in the camera field of view. It can be thought of as subfield of pattern recognition because it classifies a new human-action video clip into one of predetermined categories. During the last three decades, recognizing human motions in video, as it provides useful meta-data to numerous applications, has gained great attention by researchers mainly from the computer vision, image processing, and pattern recognition communities [63]. Its popularity has been and is still growing as it has the potential to provide solution for many related applications such as security, indexing and searching of large-scale video archives, surveillance and protection systems (e.g., abnormal behaviour detection), robotics, automated sport analyzers and commentators, and interactive environments. Given the current and future increasing interest in this field, the development of robust human motion recognition systems has become a key issue [30]. The terms “activity”, “action”, “motion”, and “behavior” are exchangeably used in the literature of computer vision for closely similar purposes.

1.3 Problem Statement

Developing reliable intelligent vision systems capable to understand what a human is doing in the camera scene is an interesting and ultimate goal that comes with very high complexity [95]. Researchers are working on several fronts to solve the many challenges in the human motion recognition area, making it one of the most current active research fields. These challenges include, but not limited to, complex appearances of humans, changes in body sizes and shapes, clothing variety, self and temporal occlusions, difficulty in identifying individual limbs, abundant number of body poses, motion overlapping with common

poses, fast movement and motion blur, shadows and camera noise, camera view-point, and cluttered and moving backgrounds. Furthermore, ambiguous vocabularies, coming from natural language, are used for precisely describing human motions [34]. This thesis investigates the following issues:

1. development of a robust appearance-based template matching approach which is able to compress a motion video into an expressive representation and facilitates subsequent training and recognition tasks.

2. employment of dimensionality reduction as a learning strategy in the context of human motion recognition for classification aiming to achieve maximum separability between classes.

3. building a recognition system for directed human motions with scale variation, tackling different types of human motions in terms of direction, speed, and location.

4. employment of the kernel technique for non-linear human motion recognition and study of the effect of using various kernel functions.

5. enhancement of data classification in general through integrating the linear version of the nonparametric discriminant analysis with the kernel mapping.

1.4 Original Contributions

The original contributions offered by this thesis correspond directly to the issues and objectives that are identified in the problem statement.

To address some drawbacks associated with common used appearance-based human motion representation, we first propose a novel representation method, named Motion Intensity Image, ”MII” for short, that can efficiently compress a human motion video clip into a simple and concise image that is remarkably expressive for the whole video. The

main clue used in this proposed method are the centroids of the sequence of binary silhouettes. It aligns them to a reference point and then compresses the sequence using a certain aggregation procedure into a single image. Each "MII" takes very small amount of storage space compared to related methods such as Motion History Images (MHI) and Motion Energy Images (MEI) [11]. Second, we took advantage of the discriminatory power of FDA by employing it in the situations where the number of training samples is large enough for the process. Third, we overcome the problems associated with directed human motions by proposing a robust framework based on a compound eigenspace. In this context, and in addition to the conventional data available in each video frame, this method extracts two additional features to control the recognition process, i.e., the silhouette relative speed and the linear displacement vector. This compound eigenspace model for human motion recognition is built based on the idea of partitioning the motions into subgroups according to the two extracted features. The implementation results show clear improvement of recognition (see Chapter 3).

With respect to non-linear classification using the kernel technique, we have proposed a new kernel-based human motion recognition works on the idea of dimensionality reduction. We have also provided an empirical study on the use the kernel technique to show how crucial choosing the right kernel function is for the success of the subsequent linear discrimination in the feature space. We have investigated eighteen different kernel functions. Our "MII"-based compression method has been employed for video-compression before the technique of kernel PCA was used to reduce image dimensionality and generate an eigenspace that contains all training motions. Finally, a database is built to store the projected samples in order to be used for recognizing test motions by seeking the best matching based on the Euclidean distance. The rich implementation results provided in this part were obtained by applying the model on two of the most common used benchmark datasets in the field of human motion recognition, KTH and Weizmann (see Chapter 4).

The final contribution is made to improve data classification in general by innovating an algorithm that is able to deal with non-normal data distributions, works non-linearly,

and provides maximum separability between classes. We have proposed a new supervised kernel-based approach that defines a non-linear generalization of the non-parametric discriminant analysis (NDA) technique to perform in kernel feature space.The proposed approach relaxes the normality assumption of FDA using the nonparametric form of the between-class scatter matrix introduced in [38] and extended in [54]. At the same time, it behaves nonlinearly with the help of the kernel technique. This algorithm focuses directly on multi-class problems as it is more general than the usual two-class problems. Two other kernel-based non-parametric discriminant analysis methods have been recently proposed in the literature, [102, 104]. The model in [102] computes the nearest neighbors (NNs) for each data sample based on the Euclidean distance amongst original data samples in input space instead of the mapped data samples in the feature space. However, it does not define the weighting function in terms of kernels, which is responsible for capturing the classification boundary structure between classes. The other model in [104] was built based on [13] where the scatter matrices are defined non-parametrically based on *extra-class* NNs and *intra-class* NNs, respectively. When computing their intra-class and extra-class matrices, their algorithm addresses only one NN for each data sample instead of the mean vector of its k NNs. Finding an NN in computing the intra-class, in particular, is done based on the *most distant* sample. For the above mentioned reasons, our proposed model turns out to be superior than these two models as it is shown in the results obtained in the experimental tests (see Chapter 4).

1.5 Thesis Overview

The remaining of the thesis gives more detailed description for each process in the contributions that have been raised in the previous section.

Chapter 2 provides valuable literature review that brings forth and synthesizes what is currently known about the topic. we also provide in this same chapter a general taxonomy of the different methods used for human motion recognition.

Chapter 3 deals with human motion representation and linear dimensionality reduction for classification. It first describes the required preprocessing steps we have used and introduces the proposed template matching approach, the ”MII”, emphasizing its strengths compared to related methods. Second, a linear eigenvector-based framework for learning and recognizing human motions in videos using the ”MII” is covered. Third, it presents the idea of using a compound eigenspace to enhance human motion recognition by making use of the two extracted features, *relative speed* and *displacement vector*.

Chapter 4 covers the proposed nonlinear classification approaches using the combination of kernel trick and linear dimensionality reduction techniques. It first describes the framework of employing the kernel technique for human motion recognition and then presents a novel kernel-based non-parametric method for data classification in general where a new derived expression of the objective function is presented. Implementation details, experimental results, and comparisons are given along with each model in its chapter. Finally, conclusions are drawn and further research work is suggested.

Chapter 2

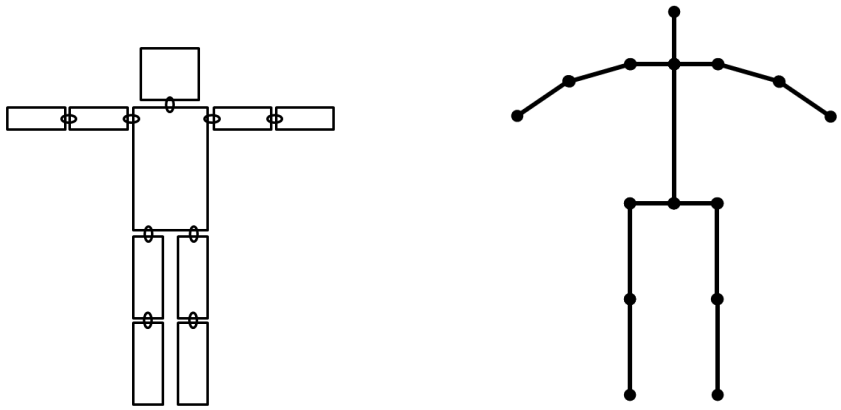
Literature Review and Previous Work

2.1 General Taxonomy

In the literature and based on our point of view, most of the proposed human motion recognition algorithms fall generally into two main types. These two types are model-based methods, also called configuration-based or part-based, and appearance-based methods, also called view-based or motion-based, [67, 73].

2.1.1 Model-based Methods

A model-based method relies on 2D or 3D human body representations. These representations must have the ability to describe basic variations in both space and time based on a set of interest points [71]. Human body is commonly represented by a kinematic model with either a skeleton-like structure or a pictorial structure. The latter has shown more appropriateness in characterizing spatial variations in geometric structures [7, 94]. As shown in Fig. 2.1, the skeleton-like structure consists of joints linked by segments [43]. The pictorial



(a) Pictorial Structure. (b) Skeleton Structure.

Figure 2.1: Some model-based human pose representations.

structure consists of a group of rectangle-like or ellipse-like pieces connected pair-wise by spring-like connections, where each piece symbolizes a certain part of the human body [35].

In general, this type of algorithms achieve human action recognition in two levels of processing: low level and high level. The low-level process is a prerequisite for the high level process and is responsible for identifying the 2D/3D human poses at every time instant, whereas the high-level process recognizes the human action [43]. Each of these processes comprises a number of sub-processes for the purpose of carrying out its function. For instance, identifying human body pose involves three main operation: initialization, tracking, and pose estimation [67]. The initialization step is to capture priori knowledge of the subject in order to be used to constraint the subsequent processes, i.e., tracking and human pose estimation. This would guarantee that the system starts to work with the right explanation of the current scene. It requires an accurate human model description that defines the shape, appearance, and the initial pose of the person which involves body parts detection. A set of determined parameters is then used, as structural information, to describe the human body [43]. Pose estimation is done by either direct inverse kinematics or numerical optimization over those explicit parameters [2]. Many learning-based body parts detectors have been proposed to identify and locate possible parts and limbs. These

pieces of information are then probabilistically fused to locate the human body and defines its pose [67]. Initialization of this kind of models requires a large training set of observations including both positive and negative for every body part [67]. However, most of the model-based methods for 3D use manual initialization with body part dimensions and shape. With the availability of various types of sensors, accurate 3D details about the entire human body surface and shape can be obtained. As identifying human poses still remains a open problem, it has caught more attention from researchers compared to action recognition [43]. The action recognition stage in model based algorithms is done by interpreting a given set of subsequent human body poses and other parameters. Note that in order for action recognition to be successful, we need to have enough rich descriptors for action representations [94].

In the following are some model-based methods that have been recently proposed. Noorit et. al., in [72] used a simple 2D parametric model to describe human body poses. Their model consists of three basic parts which are head, torso, and legs, while neglecting representing hands. These three parts are embodied using elliptical shapes where both legs are merged into one part. This method relies on motion and texture for detecting, tracking, and extracting human body from video. It first extracts the moving areas using the background subtraction technique, and then homogenous areas are gathered into groups to form the three major parts of the human body model. The homogeneity between foreground regions is measured based on a number of aspects, such as texture, color, and continuity or proximity (closeness). Human body poses and movements are described in every video frame through two sets of parameters, internal and external. Internal parameters are used to define temporal body poses by describing the relations between the three model parts. External parameters, on the other hand, are used to define body movement over time by describing the relations between corresponding model parts w.r.t. the preceding video frame. In this method, human actions can considered to be divided into two groups: static and dynamic. Static actions are identified based on the internal parameters when the external parameters show no change over a certain period of time. Dynamic actions are recognized

based on a predefined set of transitions that depend on both internal and external parameters.

This method, however, has many shortcomings. First, the 3-part model is very simple and hence unable to describe human poses and actions with even modest complication. Second, hands are denied and legs are modeled by a single component, therefore, actions that are based on hands and/or legs movement are not addressed. Third, the method is sensitive to light condition such as shadow, contrast changing, and sudden changes of brightness [72]. Fourth, this 3-part model can identify only 5 actions: "standing", "bending", "sitting", "walking", and "laying". Fifth, although legs movement plays the most important role in identifying "walking", this paper relies only on global body movement. Sixth, the algorithm used for recognition is trivial as it works based on simple thresholding and does not involve any kind of training.

Htike et al., in [43], proposed a model-based human action recognition method that aims to be invariant to camera viewpoint, where the human body is described through a 3D configuration model. This model is represented by a skeleton-like kinematic tree, similar to the one shown in Fig. 2.1b, which consists of 16 segments and 17 nodes. Twelve joints of the 17 nodes are used to describe human poses where each joint connects two or more segments. The configuration of each joint is given by two angles and represents the 3D relation between its associated segments. At every video frame, Htike et al. identify a 2D human body pose using a simple search-based and tracking-based technique [43]. Because extracting 3D information from a single uncalibrated camera is an ill-posed problem, it extracts a 2D human body model and then searches in a database for a 3D model with the best matching. To derive a 3D human body pose from a 2D view, the method uses a lookup table that maps 50 different 2D poses from 13 different viewpoints to their corresponding 3D configurations. A simple Euclidean distance is employed here for finding the best matching. For action classification, this approach first compiles those poses into a multivariate time series matrix. Second, matches the computed time series against a large set of labeled time series computed for known human actions. The unknown action is classified as the

one with the minimum global distance which is computed based on all the involved poses. This method, however, has also some drawbacks. The authors assume that body part detection has already been performed, whereas it represents one of the major challenges in human body pose recovery. It is well known that a single 2D human body pose can represent many 3D poses, therefore, the idea of mapping 2D poses to 3D poses is not effective and may mislead the performance.

Tran et. al., in [94], proposed an approach for human action recognition that describes human body poses using a 2D part-based model similar to the one shown in Fig. 2.1a. To define the human pose at any time, it uses the location information of the body parts relative to a certain reference point. This is done by first mapping the locations from their original image coordinate space to a new polar coordinate space where its origin c is the centroid of the torso part and its polar axis A is the vertical direction. Each part location p_i is then given as a vector of two coordinates which are the radial coordinate and the angular coordinate in polar form (r_i, θ_i), where r_i represents the distance between p_i and c, and θ_i is the direction of $\overrightarrow{cp_i}$ w.r.t. A. Global human body movement can then be detected and described through the variation of the corresponding body parts' coordinates over time. This is done by linearly combining movement of all body parts over all subsequent video frames, where each part movement is represented by a single motion descriptor as a polar histogram that contains its coordinates over the entire video. To test a new action video for classification, it is, first, represented by a set of body part motion descriptors, and then, matched, at the level of part descriptors, against a stored set of motion descriptors for known human actions. It is then classified/recognized as the one with the minimum total residual.

In general, most of the model-based human action recognition methods suffer from some issues that need to be seriously addressed. Their performances mainly rely on how accurate identifying human body parts is, which requires careful tracking, detection, and segmentation. These processes are sensitive to noise and blur [95], and subject to fall in the problem of having too many candidates for each human body part [68]. Many of these

methods require large amount of computation power [73] as they use bottom-up and top-down scenarios during the involved low-level processes for identifying human body poses. This is also due to performing recognition more than once at both levels when recognizing human body parts and when recognizing human actions. New human poses that are not involved in the training process may lead to unreliable recognition results [79].

2.1.2 Appearance-based Methods

An appearance-based method does not need to have any structural information about the human body. Instead, it summarizes a video clip into a representation which can be used for recognition. In particular, discriminative features are extracted from each video frame based on motion information [5]. These features are then processed in some way to form a single static shape pattern, or template. This template of motion is then matched against a set of pre-stored templates of known actions [5].

The most commonly used action representations are the MHI and MEI proposed in [11], where motion in videos is compressed 2D images, called templates. More details about these two templates are provided in Section 2.2. Extensions of these templates to the three-dimensional case have been also proposed and are called Motion History Volumes (MHVs) [99]. This method uses multiple calibrated cameras to capture a set of 2D images from multiple viewpoints for the same human pose at any time. The alignment and comparison are performed in this method using Fourier transforms in a coordinate system known as cylindrical coordinate system. This method is able to efficiently model human actions as Fourier magnitudes. However, its applicability is very limited as it needs multiple calibrated cameras and the human must be at the center. Hence, this method's representation is restricted for single location motions only.

Some methods in this stream rely on optical flow information to extract motion features to represent human actions in video. Ali et. al., in [5] proposed a method that aims to represent complex human actions by extracting kinematic features from videos based on

optical flow. This method employed two main features, divergence and vorticity, where the first feature is used to identify the regions where optical flow is being expanded due to the movements of limbs, whereas the other feature is used to represent regions where circular motion is appearing. The algorithm of PCA is used here to utilize the extracted kinematic features in order to derive dominant dynamics. For high-level human action recognition, using the approach of multiple-instance learning [17], the derived dominant dynamics are represented by a bag of kinematic modes. Then, classification is done based on the nearest neighbor algorithm.

Generally, appearance-based techniques have been widely used and generally preferred over the model-based methods as they have the advantage of being easy to implement, require less computational complexity, and are generally more robust [28]. However, Htike et. al., in [43] pointed to three drawbacks about appearance-based methods: they are unable to handle self-occlusion, require a large and diverse set of training samples, and they are more prone to overfitting.

2.2 Human Motion Representation

In the context of appearance-based methods, their primitive component is the input image sequence, which is in turn transformed into a certain representation or template for the purpose of learning and recognition. The main concept of recognition is matching a temporal template against a stored set of templates of known human actions [22]. The most commonly used templates are MHIs and MEIs [11, 73, 99]. The latter is also called Exclusive-OR (XOR) image representation. Yet, these two models suffer from certain drawbacks such as, great and unexpected variation of their sizes, extreme sensitivity to temporal occlusions, and the possibility of producing various final images for the same human motion due to some of their computational parameters. Using the MEI/XOR technique may lead to a strong overlap between different motions, especially, when the human moves in different locations in the scene. Moreover, human motions with high level of intricacy are

prone to overwrite themselves, so the two action representations, MHI and MEI, perform improperly, and therefore a more robust action representation is required [10].

2.2.1 The Human Motion Representations: MHIs and MEIs

Many approaches have been proposed in the literature for encapsulating human actions in videos into a single two dimensional template. The templates MEI and MHI, which were proposed by Bobick et. al., in [11], are used for compressing human actions in videos. Let us go through these two models showing how they work and underlining their shortages for the purpose of proposing a more robust approach that is able to perform efficiently.

Generally and according to [11], both MEI and MHI are computed based on a sequence of related binary images. Each of which contains only the moving portions at its time instant compared to its preceding frames. These sequences can then be compressed to produce each of MEI and HHI based on their corresponding formulas. Two ways of weight-based aggregation methods are used here, *equal weights* and *decaying weights*. The first method of aggregation results in a binary template called the "MEI", whereas the other was built based on the principle of assigning high values of the recently changing pixels and reducing these values over time until vanishing, yielding a representation called "MHI". This can be simply noticed through the distinction between pixels according to the intensity of brightness in the resulting image (see Figure 2.2). An MHI $H_\tau(x,y,t)$ at time t and position (x,y) is computed by Eq. (2.1) from [11].

$$H_\tau(x,y,t) = \begin{cases} \tau, & \text{if } D(x,y,t) = 1 \\ \max(0, H_\tau(x,y,t-1) - 1), & \text{otherwise} \end{cases} \tag{2.1}$$

Here, x, y and t represent the 3D information (position and time or frame number) of an individual pixel, the motion mask $D(x,y,t)$ is the binary difference of a pixel from two subsequent frames, τ is the maximum number of frames a motion is kept, and δ is the decay parameter.

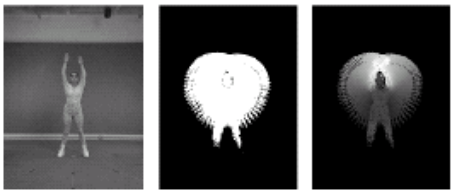


Figure 2.2: Temporal templates as shown in [11]. Left: a key frame of a video clip of an actor waving both hands. Middle and right: MEI and MHI images of the same motion.

An MEI, in turn, is a monochrome image that activates pixels where there is motion over the entire set of image sequence. The MEI, therefore, can be computed thought either Eq. (2.2) from [11] or by thresholding the MHI above 0 based on Eq. (2.3) from [64].

$$E_\tau(x,y,t) = \bigcup_{i=0}^{\tau-1}(x,y,t-1) \tag{2.2}$$

$$E_\tau(x,y,t) = \begin{cases} 1, & \text{if } H_\tau(x,y,t) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

In fact, the integration of these two templates together, the MEI and MHI, can represent both temporal and spacial aspects of the human action [11]. The MHI is to tell how the motion is performed over time whereas its corresponding MEI provides information about motion occurrence, region, shape, and viewing angle.

Although Turaga et al., in [95] stated "that MEI and MHI have sufficient discriminating ability for several simple action classes such as 'sitting down', 'bending', 'crouching' and other aerobic postures", Bobick pointed out in [10], however, that these two models perform improperly in the case of human motions with high level of complication as they are prone to overwrite themselves.

Because of the dependency on δ in computing MHI templates based on the definition of MHI in Eq. (2.1), changes in δ can give different MHIs for the same input motion video. This can be seen clearly in Fig. 2.3.

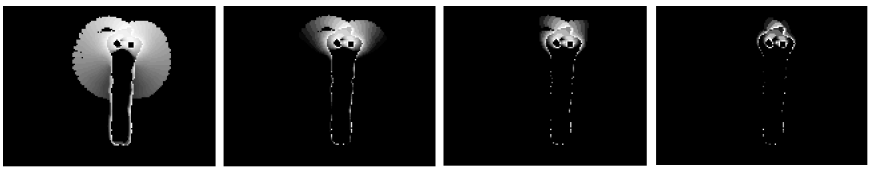


Figure 2.3: Four *different* final MHIs of the same motion video clip of an actor waving both hands. Each of which uses different value of δ.

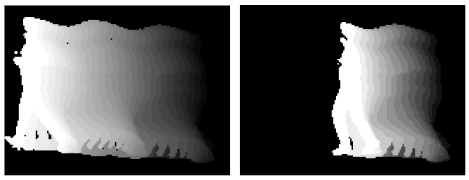


Figure 2.4: Two final MHIs of the same motion, *jogging*, computed at different time instances, t_1 and t_2. Right: $t_1 = 14$. Left: $t_2 = 27$.

Time instance, t, where an MHI is being computed can make a big difference in the final MHI too. Figure 2.4 illustrates the difference of two final MHIs for the same video of 28 frames for a jogging actor generated in different time instances, t_1 and t_2. The MHI on the right side of the same figure is computed mid-way of the input image sequence whereas the other MHI is computed at the end of the sequence.

It is noticed that both MHI and MEI are sensitive to temporal occlusions. As shown in Fig 2.5, the two final templates contain no data in the area where temporal occlusion occurred.

2.2.2 Directed Human Motions

Although some of the existing algorithms were proposed for recognizing directed human motions, but they are limited to single-location motions (such as sitting, bending, etc.) [31] or multi-location motions with displacement vector, *linear displacement vector*, relatively parallel to the image plane [14]. It can be noticed that they do not take into account directed multi-location human motions with displacement vectors unparallel to the image plane. As

Figure 2.5: Temporal templates. From left to right: three key frames of a video clip of a walking actor showing a hand-made occlusion (the vertical red bar), and motion history and energy images of the same motion. They show no tolerance against the effect of temporal occlusion.

an example of this is when a human moves away or toward the camera with some degree of deviation. Two main issues associated with this kind of motions need to be addressed. First, the variation in human silhouette sizes as a result of object-camera distance changes. Second, the insufficient information of shape and speed of the limbs due to self occlusion.

2.3 Linear Eigenvector-based Techniques

Numerous appearance-based techniques were proposed for human motion analysis and recognition such as, hidden Markov model [76, 79, 101], dynamic time warping [9, 81], Support Vector Machines (SVM) [12, 19], and many others. However, most of these techniques are too much dependent on structural models, probability factors, time constraints, iterations, etc. Hence, these techniques are computationally expensive, and restricted to specific applications in some cases.

Comparatively to the techniques mentioned above, the eigenvector-based techniques such as FDA [36, 39] and PCA [46], used in [29, 69, 73, 77], do not require geometrical calculation or partial segmentation of the models. Thus, they are easily adaptable and computationally cost-effective [77]. These two algorithms are the most commonly used algorithms for extracting features and reducing data dimensionality. Mapping data into a lower-dimensionality eigenspace provides significant improvement in recognition and

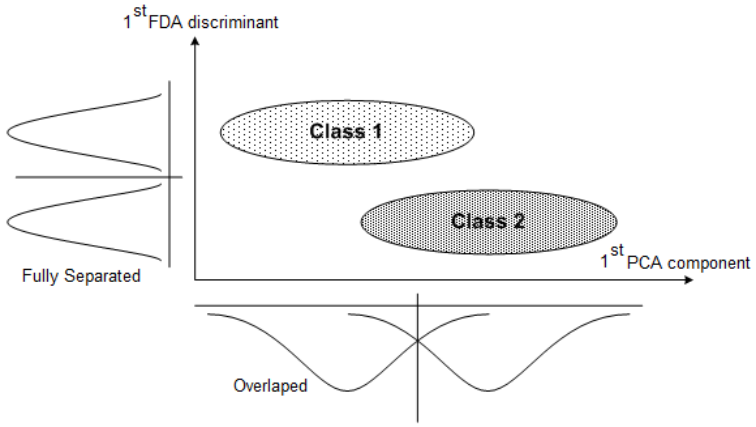


Figure 2.6: FDA versus PCA in terms of class separability, adapted from [91].

computational aspects [96]. Feature extraction and dimensionality reduction offered by the eigenvector-based techniques is one of the commonly used pre-processing procedures for classification. Linear algorithms for describing data seek a feature or a set of features where data points vary the most regardless of their affiliations or classes. On the other hand, algorithms for classification work mainly based on data points affiliations and aimed to find a feature or a set of features where maximum class separability can be achieved. For successful classification, supervised learning techniques are preferred compared to the unsupervised ones. However, the existing eigenvector-based methods for human motion recognition rely on either MHI or MEI/XOR compression techniques for motion discrimination and use PCA for dimensionality reduction and eigenspace generation. The PCA and FDA algorithms share one goal which is reducing data dimensionality, but each of which based on different criterion.

For instance, the unsupervised technique PCA extracts directions where data variance reaches its most. Based on this criterion, it minimizes the reconstruction error [66]. FDA, on the other hand, preserves as much of the class separability as possible by finding the best feature or feature set that maximizes the ratio of between-class scatter to within-class scatter [15]. Figure 2.6 shows that PCA projects data onto the direction with the maximum variance whereas FDA projects data onto the direction with maximum class-mean variance.

From different point of view, the dimensionality of FDA is upper-bounded by $C-1$ as it relies on class means, where C is the number of classes. Moreover, FDA requires at least $C+L$ samples to avoid falling into the singularity problem, where L is the data sample dimensionality. It is generally believed that applications based on FDA are superior to those based on PCA in the case when the number of training samples is large and representative for each class [61]. It is reported in [103] that, in some cases, the PCA technique may discard dimensions that contain important discriminative information between classes.

2.3.1 The algorithm of FDA

This is a brief review of the classical FDA as described in [39]. Let us consider having a set of observations where each observation is described by d features. These samples are divided into C classes. FDA Looks for a direction, or a set of directions, which ensures the highest level of separability between the class centers. To achieve the best class separation, it looks for a projection W, from d down to $\acute{d}$ dimensions, maximizing the ratio of between-classes matrix (S_b) to within-classes matrix (S_w) using the following objective:

$$J(W) = \frac{W^T S_b W}{W^T S_w W} \tag{2.4}$$

The main concept behind this criterion is that the best class separation can be achieved by maximizing the distance between class means while bringing the overall class variance, computed amongst each class, to the unity. This is precisely what we want. The optimization of Eq. (2.4) leads to the following generalized eigenproblem:

$$S_b w_i = \lambda_i S_w w_i \Rightarrow S_w^{-1} S_b w_i = \lambda_i w_i \tag{2.5}$$

Eigenspace projection matrix can then be constructed by choosing the eigenvectors with the largest $\acute{d}$ eigenvalues of $S_w^{-1} S_b$, where $\acute{d}$ is the desired reduced dimensionality, where $0 < \acute{d} < C$. These eigenvectors provide the directions of the maximum discrimination.

Because of the rank limitation of S_b, the number of desired coordinates $\acute{d}$ is upper bounded by $C-1$ [86]..

The Methodology of FDA

Let $x_n = \{x_1, x_2, ..., x_d\}$ be a training data sample where d is the total number of elements in each data sample, C be the number of classes, μ_c be the mean feature vector for class c, μ be the mean vector of the entire set of training samples, N_c be the number of training samples from class c, and N be the size of the entire set of training samples from all classes such that

$$N = \sum_{c=1}^{C} N_c, \tag{2.6}$$

then,

1. Data normalization, $\|x_n\| = 1$.

2. A super matrix $X_{c\in\{1,2,...,C\}}$ is constructed for each class from all of its training vectors,

$$X_c = \{x_1, x_2, ..., x_{N_c}\}. \tag{2.7}$$

3. A mean vector μ_c is computed for each super matrix,

$$\mu_c = \frac{1}{N_c} \sum_{n=1}^{N_c} x_n, \tag{2.8}$$

and then subtracted from its super matrix,

$$X_c = X_c - \mu_c. \tag{2.9}$$

4. A super mean matrix is constructed from all class means,

$$M = \{\mu_1, \mu_2, ..., \mu_C\} \tag{2.10}$$

5. The mean vector of all classes μ is computed out from all μ_cs,

$$\mu = \frac{1}{N}\sum_{c=1}^{C} N_c \mu_c, \tag{2.11}$$

and then subtracted from the super mean matrix,

$$M = M - \mu. \tag{2.12}$$

6. The scatter within-classes and scatter between-classes matrices, S_w and S_b, are computed using Eq. (2.13) and Eq. (2.14), respectively.

$$S_w = \sum_{c=1}^{C} X_c^T X_c \tag{2.13}$$

$$S_b = M^T M \tag{2.14}$$

7. Once S_w and S_b matrices are computed and before the projection matrix P can be constructed, S_w^{-1} needs to be found. To guarantee the existence of S_w^{-1}, we need at least $N = d + C$ training samples. Classic inversion methods cannot be employed here because of the largeness of S_w. Hence, we use the singular value decomposition (SVD) as follows:

$$\begin{aligned} S_w &= U \cdot \Lambda \cdot V^T \\ S_w^{-1} &= (U \cdot \Lambda \cdot V^T)^{-1} \\ &= (V^T)^{-1} \cdot \Lambda^{-1} \cdot U^{-1} \\ &= V \cdot \Lambda^{-1} \cdot U^T \end{aligned} \tag{2.15}$$

where the values of the principal diagonal of Λ^{-1} equal $\frac{1}{\lambda_i}$. The inverse of each of U and V are their corresponding transposes as they are orthogonal. Thus,

$$U^{-1} = U^T \text{ and } V^{-1} = V^T \Rightarrow (V^T)^{-1} = V \tag{2.16}$$

8. Using SVD again, the transformation matrix P is constructed by choosing the eigenvectors, $v_{r\in\{1..\acute{d}\}} \in V$, with the largest $\acute{d}$ eigenvalues $\lambda_{r\in\{1..\acute{d}\}} \in \Lambda$, of $S_w^{-1} S_b$.

9. Finally, the dimensionality of each training sample can be drastically reduced and projected as a point on the eigenspace E using Eq. (2.17).

$$e_n = (x_n - \mu)^T \cdot P \tag{2.17}$$

By labeling and projecting all training data samples into the eigenspace E, the system becomes ready for matching use.

2.3.2 The Algorithm of PCA

The algorithm of PCA is one of the commonly used approaches for reducing data dimensionality and describing data distributions. It seeks directions where data variation in the original data distribution reaches the most [56]. Preserving maximum variance between data points provides PCA the ability to describe high dimensional data distributions with a new small set of coordinates [90]. Dimensionality reduction in PCA is done, generally, by eliminating data dimensions or features that do not contribute in discriminating between data points and preserving only the ones with high level of contribution. The eigenvalues that are computed by solving the eigenvalue problem of the covariance matrix of the entire data distribution are used as a set of contribution indicators for their corresponding features or eigenvectors. The lowest eigenvalue corresponds to the least significant feature whereas the highest eigenvalue corresponds to the most significant feature. PCA provides an optimal basis for least squared reconstruction of data by minimizing the mean squared reconstruction error [92]. As shown in Fig. 2.7, c_1 represents the best linear fitting for the data spread which minimizes the mean squared distance between the data samples and their corresponding projections on c_1. In PCA, the resultant principal components are ranging in power of preserving data variance from high to low.

As PCA is an unsupervised technique, the resulting PCs may not be good for discrimination purpose. For example, suppose that we have two classes in a 2D space represented by the two Gaussian-like densities in Fig. 2.6. They look well separable in their original

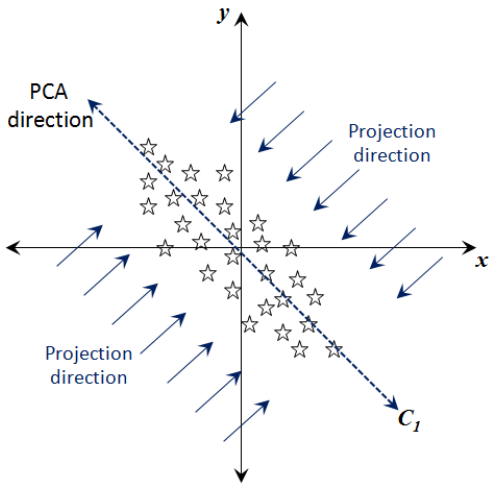


Figure 2.7: The geometric aspects of PCA.

space. But if we project the data from 2D down to the longest 1D axis using the first PC (x-axis), then they become inseparable with a significant interference. The best projection is the short axis (y-axis).

The Methodology of PCA

With respect to human motion recognition, the objective of PCA is to project the entire set of training samples, or human motions, onto an eigenspace using a relatively small number of coordinates [23]. Each training sample, $x_{n\in\{1,2,..,N\}}$, is formed as a data vector $x_n = \{x_1, x_2, ..., x_d\}$ and then normalized as $\|x_n\| = 1$. The whole set of normalized data vectors are then centered based on the mean vector μ and a super training matrix X is then constructed as in Eq. 2.18.

$$X = (x_1 - \mu, x_2 - \mu, .., x_N - \mu) \tag{2.18}$$

where the data mean vector μ is defined by

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{2.19}$$

The super image matrix X is $d \times d$, where N is the total number of training observations, and d is the size of the sample. To compute the eigenvectors of the given training set, the covariance matrix Σ is computed using Eq. 2.20:

$$\Sigma = XX^T \tag{2.20}$$

By solving the eigenvalue problem of Σ using the method of Singular Value Decomposition (SVD), we extract the most overriding eigenvectors. The first $\acute{d}$ eigenvectors $(v_1, v_2, ..., v_{\acute{d}})$, whose eigenvalues are the highest, are chosen to create the transformation matrix P. An eigenspace E is then built by projecting the entire training set using P. Each data sample x_n is then projected into E as a point e_n based on Eq. 2.17.

2.4 Nonlinear Classification using Kernel Technique

Based on the fact that no real data distribution is truly linearly separable, the demand for efficient non-linear classifiers has been growing. Many researchers have shown that kernel-based methods are computationally efficient and robust, and provide significant improvement in pattern recognition and data classification [51, 62, 85, 97, 102]. Instead of directly applying a linear classification model on the original data points, these data points are first mapped into a higher-dimensional feature space where classification can be performed linearly. Such mapping is performed by a nonlinear function Φ implicitly through a mathematical process called the "kernel trick" [3, 12]. Hence, when using any kernel function with any linear classification model, the originally linear operations are done in a reproducing Hilbert space, obtained through an implicit non-linear mapping [82]. Many existing linear algorithms have been extended to work with the kernel technique. Some of those are Kernel Support Vector Machines (KSVM) [12], Kernel Principal Component Analysis (KPCA) [82], Kernel Fisher's Discriminant Analysis (KFDA or KDA) [66], and Kernel Canonical Correlation Analysis (KCCA) [4]. All these kernel-based algorithms are

considered to be extensions of their linear versions to non-linear distributions. By applying the kernel trick using some kernel function $k(\mathbf{x},\mathbf{y})$, the data is transferred from its original space $\mathcal{X}$ to an inner dot-product space $\mathcal{K}$, where mapping features is implicitly performed, $k(\mathbf{x},\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$. Choosing the right kernel function is very important to make linear separability performs properly in the feature space [6, 42].

2.4.1 The Kernel Technique

If data classes cannot be separated linearly in their original space $\mathcal{X}$ (called *input space*), data can be mapped into a higher dimensional Hilbert space $\mathcal{H}$, (called *feature space* $\mathcal{F}$) where data can be linearly classified. This mapping can be done through a non-linear transformation using an appropriate and carefully selected function Φ, (called *mapping function*) [84]. Then, a linear model can be used in the new space.

$$\Phi : \mathbf{x} \to \phi(\mathbf{x}),\ \mathcal{X} \to \mathcal{F},\ \mathbb{R}^{d_1} \to \mathbb{R}^{d_2} \qquad (2.21)$$
$$\text{where } (d_1 < d_2 \leq \infty)$$

Thus, linear classification in the feature space can be thought of as non-linear classification in the input space. This process is called "kernel technique", where the term "kern" was first used by Hilbert in [41]. The choice of the mapping function is very critical for the success of the linear discrimination in the feature space. Fig. 2.8, adapted from [33], illustrates a particularly simple example where the data is mapped from $\mathbb{R}^2$ into $\mathbb{R}^3$:

$$\phi(\mathbf{v} : \mathbb{R}^2) = \phi(x,y) = (x^2, \sqrt{2}xy, y^2)$$

Dealing with the mapped data explicitly in a very high or infinite dimensional feature space is intractable in terms of possibilities and computational cost. This can be overcome by an idea called "kernel trick".

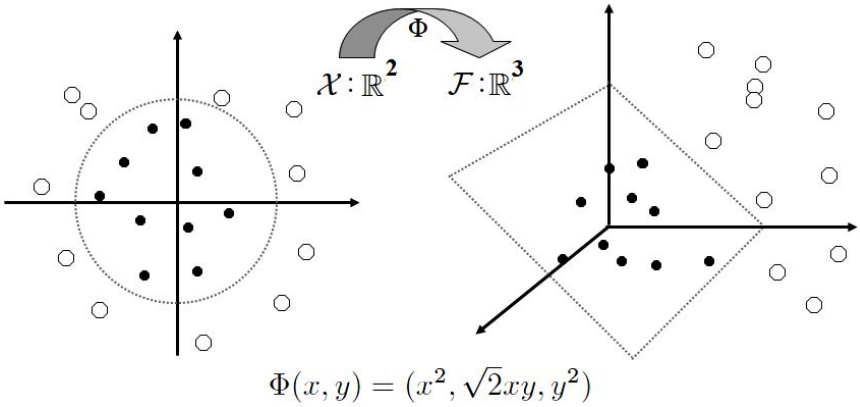


Figure 2.8: An example of mapping data from a 2D input space into a 3D feature space using a simple mapping function $\Phi(x,y)$.

The Kernel Trick

A trick, known as the *kernel trick*, introduced in [3] and re-introduced into the context of machine learning in [12], has eliminated those obstacles making it computationally possible, and even easier. The kernel trick solves the intractability problem by replacing a point $\mathbf{x}$ from the input space $\mathcal{X}$ with its corresponding point $\phi(\mathbf{x})$ from the feature space $\mathcal{F}$ producing an inner dot-product space $\mathcal{K}$ represented by a gram matrix G (also called kernel matrix K),

$$\begin{array}{ccc} & \mathbf{x} \rightarrow \phi(\mathbf{x}) & \\ G_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle & \longrightarrow & G^{\Phi}_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j) \end{array} \tag{2.22}$$

where the notation $\langle\ ,\ \rangle$ represents the inner dot-product between two vectors.

The kernel trick introduced a new function $k()$ called "kernel", or "kernel function", that eliminates the use of the mapping function $\phi()$ in a way that $\phi()$ is done implicitly in it. Based on Mercer's theorem [3, 65], not every function can be considered as a kernel. For $k()$ to be a valid kernel function, it must meet some conditions.

`Mercer's theorem` [3, 65]: "any continuous, symmetric, positive semi-definite kernel function $k(\mathbf{x},\mathbf{y})$ can be expressed as a dot product in a high-dimensional space, $k(\mathbf{x},\mathbf{y}) = \langle\phi(\mathbf{x}),\phi(\mathbf{y})\rangle$" [1] (p. 15). Thus, the mapping function ϕ is implicitly fused in the kernel function. To avoid explicitly computing the actual mappings Φ in the feature space, we need to employ linear algorithms that only depend on G. This is the core idea behind kernel methods.

The kernel trick has many advantages. Based on the principle of the inner dot-product, it defines a similarity measure between any two normalized data samples, or vectors, by reaching 1 for identical vectors and going down as the discrepancy increases [16]. This allows us to integrate prior knowledge of the problem domain [84]. Having such trick makes us able to avoid explicitly mapping data samples into a feature space and its associated difficulties, obstacles, and computational cost. As the kernel gram matrix is computed directly from the original data samples in input space, its computational complexity and the number of operations depend only on input space instead of the number of features in feature space [21].

2.5 Normality Restriction

Although the kernel-based extension of FDA is applicable for many areas where it can nonlinearly deal with normal structures, the normality assumption of FDA limits its performance in a way that it cannot easily capture a non-normal data structures.

Numerous extensions for FDA have been proposed to relax its normality restriction. In [37, 38], Fukunaga et al. introduced the Non-parametric Discriminant Analysis that has overcome the parametric form of FDA by extending the commonly used scatter matrices. As NDA deactivates the normality assumption of the FDA, it has gained the power to model non-Gaussian distributions. This is by incorporating data direction and boundary structure in its within-class scatter matrix S_W and between-class scatter matrix S_b, respectively. The latter is computed based on all data points w.r.t. their k-nearest neighbors for each point,

instead of relying on class means only as in FDA. This has made the scatter matrices generally of full rank, and therefore, the method can work well even for non-Gaussian datasets, as well as, the ability to specify the number of desired extracted features. A weighting function was proposed, for each data sample, to preserve classification structure by deemphasizing samples far from the classification boundary. Loog et al. in [59] introduced a modified Fisher's criterion based on a new weighting function that uses the Mahanalobis distance to evaluate the contribution of each class pair. They call it "approximate pairwise accuracy criterion" and it aims mainly to minimize the classification error. Li et al. in [54] provided new formulation for each of the within-class scatter matrix and the between-class scatter matrix to make the two-class NDA applicable for multi-class cases. They also developed two more improved multi-class NDA-based algorithms called Nonparametric Subspace Analysis (NSA) and Nonparametric Feature Analysis (NFA). Each of which has two interdependent methods where one relies on the principal space and the other relies on the null space of the intra-class scatter matrix. A combined model for NDA and SVM was introduced by Ksantini et al. in [48]. It aims to separate data spreads by incorporating the structure information of the decision boundary using the overriding eigenvectors of NDA, and maximizing the relative margin separating data classes using SVM. However, these extensions behave linearly w.r.t. class separation and fall into the problem of underfitting and misclassification when data classes are not linearly separable.

2.5.1 The Algorithm of NDA

Let $X = \{x_1, \ldots, x_N\}$ be a super sample set from C different classes in the input space $\mathcal{X}$, where $X = \bigcup_{i=1}^{C} X_i$, $X_i = \{x_1^i, \ldots, x_{N_i}^i\}$, and $N = \sum_{i=1}^{C} N_i$. The NDA is performed by finding the vector w that best separates classes though maximizing the ratio of variance between the classes S_B to the variance within the classes S_W nonparametrically using Eq. (2.23). This maximization aims to find the best feature or feature set to discriminate between the classes. This translates into finding the overriding eigenvectors of S_B while bringing S_W to

be an identity matrix. This ratio is given by

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{2.23}$$

where

$$S_W = \frac{1}{N} \sum_{i=1}^{C} \sum_{l=1}^{N_i} (x_l - \mu_i)(x_l - \mu_i)^T, \text{ and} \tag{2.24}$$

$$S_B = \frac{1}{N} \sum_{i=1}^{C} \sum_{\substack{j=1 \\ j \neq i}}^{C} \sum_{l=1}^{N_i} \omega(i,j,l)(x_l^i - m_j(x_l^i))(x_l^i - m_j(x_l^i))^T, \tag{2.25}$$

where μ_i is the mean vector of class i and $m_j(x_l^i)$ represents the mean vector of the k-nearest neighbors of vector $x_l \in X_i$ from class j.

Since capturing classification structure represents a primary concern, a weighting function $\omega(\ldots)$ is used to de-emphasize the effect of samples with large magnitudes which are far away from the decision boundary [39, 50, 54]. This function emphasizes the boundary information contained in the training data by approaching 0.5 for samples near the classification boundary and decaying until vanishing as samples turned away from the boundary between classes. Li et al., in [54] stated that by employing the entire set of training samples and using such a weighting function to weight the sample pairs based to their participation in class separability, a non-parametric model more adapted to problem at hand can be generated. The value of the weighting function, denoted as $\omega(i,j,l)$, is defined as

$$\omega(i,j,l) = \frac{\min\left\{d^\alpha(x_l^i, nn(x_l^i,i,k)), d^\alpha(x_l^i, nn(x_l^i,j,k))\right\}}{d^\alpha(x_l^i, nn(x_l^i,i,k)) + d^\alpha(x_l^i, nn(x_l^i,j,k))}, \tag{2.26}$$

where α represents a control parameter, ranging from zero to infinity, that controls how rapidly ω falls to zero relative to the distance ratio; $d(x_l^i, nn(x_l^i,j,k))$ is the Euclidean distance between $x_l \in X_i$ and its k^{th}-nearest neighbor in class j, and $nn(x_l^i,j,k)$ is the k^{th}-nearest neighbor in class j w.r.t. the sample $x_l \in X_i$.

Note that the mean vectors, $m_j(x_l^i)$ in Eq. (2.25) , are used to represent non-parametric global information about each class. They are given by

$$m_j(x_l^i) = \frac{1}{k} \sum_{p=1}^{k} nn(x_l^i, j, p) \tag{2.27}$$

where each $m_j(x_l^i)$ is the mean vector of the k-nearest neighbors to the sample $x_l \in X_i$, from class j.

Chapter 3

Human Motion Recognition using MIIs and Linear Classification Methods

In the context of appearance-based human motion compression, representation, and recognition, this chapter proposes a novel and robust framework based on the eigenspace technique. This framework is characterized by three main advantages. First, the new appearance-based template matching approach, the Motion Intensity Image, that is used for compressing human motions in videos is simple, concise, and expressive in representing human movements [23]. It aligns the human silhouette of each background-subtracted binary image to the frame center and then form a single intensity image by taking into account the difference between each subsequent silhouettes. Second, a learning strategy based on the eigenspace technique is employed for dimensionality reduction using each of PCA and FDA, while providing maximum data variance and maximum class separability, respectively. Third, a compound eigenspace is introduced to recognize directed human motions with scale variation [25]. It addresses two main issues associated with directed motions

when a human is relatively approaching or moving away from the camera. The first issue is the variation in human silhouette sizes as a result of object-camera distance changes. The second issue is the insufficient information of shape and speed of the limbs due to self occlusions. This method extracts two more pieces of information that are used to control the recognition process. In particular, the use of a compound eigenspace, controlled by the silhouette's relative speed and linear displacement vector, has clearly improved the recognition. To show the robustness of the proposed method, the system has been rigorously trained and tested using one of the benchmark human motion datasets in the literature. A similarity measure based on Euclidean distance has been used for matching reduced testing templates against a projected set of known motions templates. The experimental results are very encouraging, reflecting a high level of satisfactory performance.

3.1 A Proposed Appearance-based Human Motion Representation

In the context of appearance-based human motion representation and recognition, this section describes a novel and robust model for representing and compressing a human motion in video. The model is named Motion Intensity Image (MII). It is simple, concise, and expressive in representing human motions. The MII can efficiently compress a motion video clip in a very small amount of storage space compared to related methods such as MHI and MEI [11,99]. We found a method shares the same name and abbreviation (Motion Intensity Image, MII) proposed in [8].

However, it is limited to representing human gait by generating a single MII template along with three motion direction images (MDIs). Moreover, the representation is different from ours w.r.t. computation and appearance. It incorporates both human shape and motion which makes it inadequate for action classification and in need of more descriptors represented by the three MDIs.

Another method in [57] proposed a representation called motion impression image that shares the abbreviation of MII. This method generates two impression images computed based on each of the characteristics of the motion frequency and the optical flow information. The final template is then computed by combining these two representations.

In the context of our MII and as a result of having a small-size template, the number of data samples required for training in FDA is decreased. The method uses the centroid of each binary silhouette as a clue. It aligns the extracted human silhouette of each background-subtracted binary image to a reference point. It, then, compresses the sequence into a single intensity image based on a simple aggregation procedure that takes into account the differences between each consecutive silhouettes. This process helps us to eliminate the static portions of the human-body and preserving only portions where the motion is concentrated.

3.1.1 The Proposed MII

In this proposed approach, the computation of the MII is done based on the centroid of every extracted human silhouette. As shown in Fig. 3.1 (b), a binary human silhouette is first drawn out from every frame of the image sequence and then the human silhouette centroid is estimated in each background-subtracted binary frame. In our point of view, and as pointed in [40], the appearance and speed of the limbs relative to the torso are really informative for human action recognition compared to the speed of generic displacement of the human body. To deemphasize the generic displacement of human body and emphasize local changes, we translate the entire set of extracted silhouettes to the image center based on their centroids. Dealing only with a sequence of centered silhouettes helps to extract and restrict motion features in a very limited and expressive area. The process of centering human silhouettes represents one of the main clues. It plays the most important role in generating the MII and facilitating subsequent training and recognition tasks.

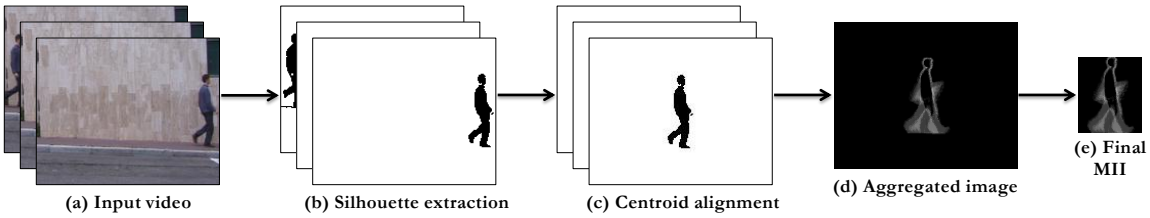


Figure 3.1: The computation steps of human motion compression using the method of MII.

Generating The MII

Once a sequence of aligned-silhouette binary images is computed, the process of compressing it into a single motion representation, MII, begins. Therefore, an MII encapsulates past silhouette changes in itself. In this work, we use a simple aggregation procedure for image compression to obtain the MII. It takes into account only the difference between each two consecutive binary images in the sequence as in Eq. (3.1).

$$I_\tau(x,y,t) = I_\tau(x,y,t-1) + \begin{cases} \delta, & \text{if } \Psi(x,y,t) \neq 0 \\ 0, & \text{otherwise} \end{cases}, \text{ where } \delta = \frac{C}{\tau} \tag{3.1}$$

Here, x and y represent pixel location, t is the time instant or the frame number, $\Psi(x,y,t)$ is the binary difference between the two consecutive frames t and $t-1$ at location (x,y), τ is the maximum number of frames a motion is kept, δ is the increment parameter, and C is the ceiling value a pixel can have. This makes any pixel value in the MII varies from 0 to C.

This kind of compression, which is demonstrated in Fig. 3.1, provides many advantages. These advantages are: (i) obtaining concise and expressive motion representations, (ii) having more details about slow and fast moving portions or limbs of the human silhouette, (iii) throwing off the static portions of the human-silhouette and preserving only portions where the human-body motion is concentrated, and (iv) as a result of the preceding advantages, succeeding processes can work faster and more reliably. Motion concentration varies from light to intensive and can easily be noticed through colors (from dark-gray to

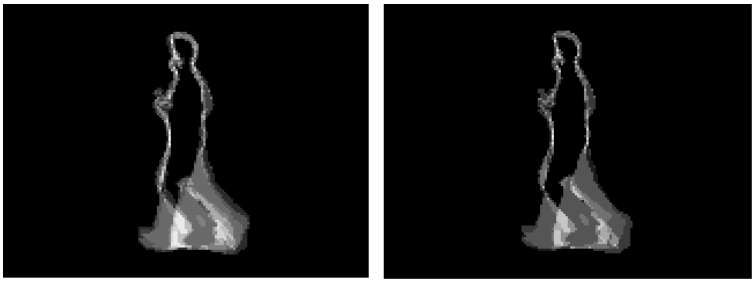


Figure 3.2: Two final MIIs of the same motion, *jogging*, computed at different time instances, t_1 and t_2. Right: $t_1 = 14$. Left: $t_2 = 27$.

white in Fig. 3.1).

Robustness of The MII

Recall that the MHI is sensitive to many parameters such as the decaying parameter, δ, the time instance, t, and temporal occlusions, as shown in Section 2.2.1. The MII, on the other hand, overcomes these shortcomings. It represents the relative value of motion change to the number of frames, τ. Due to the relation between τ and δ, the out put MII won't be affected. Based on the same fact of the MII, the time instance t has almost no effect on the resulting MII. Figure 3.2 is actually comparable to Fig. 2.4 and shows two MIIs of the same motion, *jogging*, generated at different time instances. The left MII was computed based on the whole set of video clip, 27 frames, whereas the other MII used only the first 14 frames of the clip. It can be clearly seen that the two MIIs are almost identical.
With respect to temporal occlusions, the MII on the fourth column in Fig. 3.3 reflects the robustness of our approach. It shows a very little effect of the temporal occlusion compared to its corresponding MHI shown previously in Fig. 2.5. Figure 3.3 shows clearly how the MII is capable to represent a human motion in a detailed way compared to the MHI. It is even easy to recognize the motion that corresponds to any of the resulting MIIs.

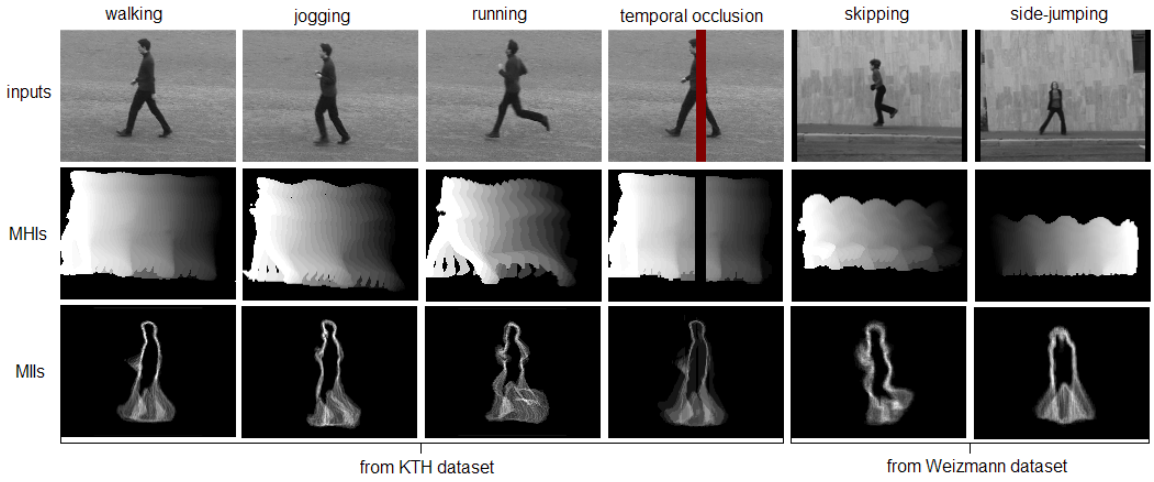


Figure 3.3: Columns 1, 2, 3, 5 and 6 show how MIIs can represent human motions in a clear and expressive way compared to their corresponding MHIs. The fourth column shows the robustness of the MII against temporal occlusions.

3.2 An Eigenvector-based Framework for Human Motion Recognition

As humans, we have no ability to recognize a new object or a new action unless we have seen, trained, or experienced it. Intelligent recognition systems must have such experience as well. This can be achieved through two basic procedures, learning and recognition. In this proposal, the learning procedure aims to build a database that contains an eigenspace with certain dimensionality along with the reduction key that guarantee the best discrimination between classes. Generally, the higher the variance between classes is, the better recognition accuracy is achieved. Each data entity in this database represents a training sample with reduced dimensionality. Using an eigenvector-based technique such as PCA or FDA, the learning stage analyzes the entire set of training data and produces a transformation matrix which represents the turning point from the original input space to the reduced eigenspace. Figure 3.4 shows a block diagram illustrating the main processes of the training procedure.

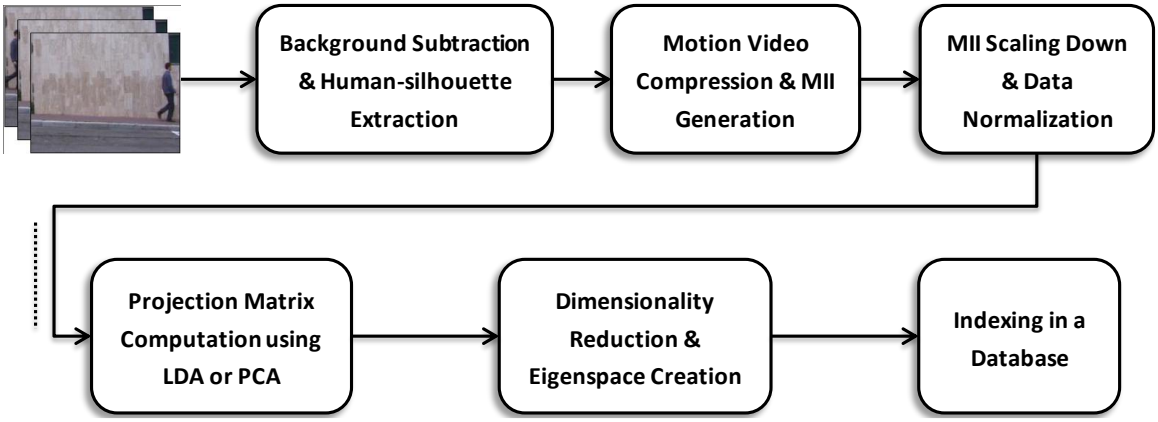


Figure 3.4: The training stage.

Since the main purpose of any intelligent recognition system is to recognize new or unknown entities, the recognition process represents the primary task that is frequently performed whereas the learning process will only be performed once. In the recognition stage, the testing sample is first projected into an existing eigenspace, built during the training stage, using the reduction key (transformation matrix). A process of similarity measure is then performed to find the nearest neighbor among the projected training samples w.r.t. the one being tested. Finally, the testing sample will be classified, or labeled, as its best matching's class.

Both stages are built upon data samples provided in the form of MII. In order for each original data sample, given as a short video clip, to be ready for usage and reduction, it has to be subjected to some preprocessing operations. These operations are binary human-silhouette extraction, MII generation, and MII scale normalization.

3.2.1 Preprocessing Operations

Preprocessing operations are crucial for achieving efficient high level processing such as training and recognition tasks [27]. In appearance-based human motion recognition and many other vision applications, identifying moving objects and extracting human silhou-

ettes from an image sequence is a fundamental and critical preprocessing task [89]. The technique of background subtraction is commonly used for extracting moving objects from videos. This process, however, remains a challenge for computer vision applications [58]. In this approach, extracting human silhouettes is required for the computation of the MII. As background subtraction is the most common approach for identifying the moving objects, we have used this technique for obtaining moving silhouettes. In this technique, each image in a given video is compared against a reference image or background model. A moving object, or foreground, can be identified by the pixels in the current image that deviate significantly from the background model [18]. In this study, these foreground pixels represent the human body, and are further processed for motion recognition. Background subtraction in a video consisting of a sequence of still images is more efficient and productive than in a single image. The main reasons behind this are: (i) in most cases background is already known, (ii) cameras are assumed to be static, and (iii) foreground objects are active since they move inside the camera scene.

Several background subtraction algorithms have been presented in the literature for identifying moving objects in environments that vary from simplistic to complex [18, 20, 32, 93, 105]. In this study, we use the improved adaptive Gaussian mixture model (AGMM) algorithm for background subtraction proposed by Zivkovic [105]. It improved the original GMM approach proposed by Stauffer [88] that models each pixel using a constant number of Gaussians. The AGMM automatically adapt the number of Gaussians that best defines the pixels of the background image. It results in more flexible and efficient performance even with the existence of motion in the background [45]. This extension reduces the algorithms memory requirements, increases its computational efficiency, and performs extremely well compared to other algorithms [74]. The extracted foreground of each video frame is then converted into a binary image and further processed using the techniques of Erosion and Dilation for enhancement.

As CPU-time and constraints and requirements of algorithms are always important issues, the size of samples (MIIs) are usually reduced. In our case, each MII image is a

$W \times H$ matrix that can be reduced to a new $w \times h$ MII, where $w << W$, $h << H$, and the aspect ratio is preserved constant. We have scaled down the MIIs by a factor of n using the cubic interpolation method as it is often preferred over the other two interpolation methods, nearest neighbor and linear, when quality is important [55]. Hence reducing data size n^2 times.

3.2.2 Dimensionality Reduction

As shown in section 3.1, the MII method has helped a great deal in obtaining small and expressive motion representations. Hence, the FDA algorithm is preferred for dimensionality reduction and class discrimination in this case. It seeks the best separation between classes by achieving the largest mean differences between the given classes [61]. To provide the best class separation, it attempts to preserve maximum variance among class means and minimum variance within data distribution using the following objective:

$$J(w) = \frac{w^T S_b w}{w^T S_w w} \tag{3.2}$$

where the definitions of the scatter matrices are:

$$S_b = \sum_{c=1}^{C} N_c(\mu_c - \mu)(\mu_c - \mu)^T \tag{3.3}$$

$$S_w = \sum_{c=1}^{C} \sum_{i=1}^{N_c} (x_i - \mu_c)(x_i - \mu_c)^T \tag{3.4}$$

where μ is the mean vector of the entire set of training MIIs, μ_c is the center vector of class c, C is the number of classes or motion types, N_c is the number of training MIIs from class c, and N is the size of the entire set of training MIIs, $N = \sum_{c=1}^{C} N_c$. To guarantee the existence of S_w^{-1}, we need at least $N = d + C$ training samples, where d is the size of the MII in pixels. Eigenspace projection matrix, P, can then be constructed by choosing the eigenvectors, $v_{i \in \{1,2,..,\acute{d}\}}$, with the largest $\acute{d}$ eigenvalues of Eq. (3.2), where $\acute{d}$ is the desired reduced dimensionality. These eigenvectors provide the directions of the maximum

discrimination. By labeling and projecting all the training MIIs into the eigenspace E using Eq. (3.5), the system is ready for matching use.

$$e_n = (x_n - \mu)^T \cdot P \tag{3.5}$$

As applying the FDA algorithm requires a large number of training samples and upper-bounds the resultant dimensionality compared to PCA, the latter is employed when any of these two factors negatively affects. PCA aims to reduce data dimensionality while achieving maximum variation [56]. In PCA, the best $\acute{d}$ eigenvectors that construct the projection matrix are extracted from the $d \times d$ covariance matrix of the entire set of MIIs by solving the eigenvalue problem (for more information about dimensionality reduction using FDA and PCA, see Chapter 2).

3.2.3 Recognizing New Motion-Videos

For the purpose of recognizing new or unknown motion sequences, we store the generated eigenspace into a database for matching use. By storing the projected training data in a database, the system is trained and ready for recognition use. The recognition stage is the most important component of any recognition system. It represents the system interface as it is responsible for giving final decisions. The accuracy level of those decisions plays the most important role in evaluating recognition systems. This processing stage is much similar and much faster than the learning stage, see Fig. 3.5. Recognizing an unclassified human motion video is simply done by seeking its most similar motion that is stored in the database. Then, the new motion is classified as the best matching motion's class. Two main parts are involved in this stage, dimensionality reduction and nearest neighbor retrieval. Once an unknown video comes as a sequence of related still images, the following steps will take place one after another to reduce the dimensionality:

1. Background subtraction and binary human-silhouette extraction from each image in the sequence.

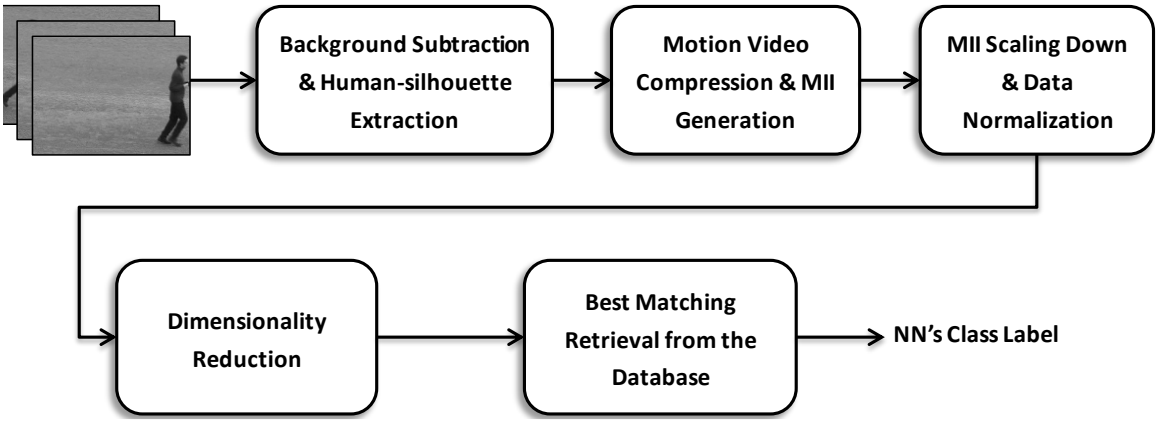


Figure 3.5: The recognition stage.

2. Applying the method of MII to compress the image set into a single model.

3. Scale-normalizing the MII to a suitable dimensions (height and width) using a certain scaling factor.

4. Forming the reduced MII into a vector $u = (p_1, p_2, .., p_d)$, normalizing it, and then projecting it as a point $\bar{u}$ into the existing eigenspace E using Eq. 3.5 and based on the transformation matrix P that has been generated by the learning stage.

5. Performing a similarity measure in the database based on Euclidean distance to retrieve the nearest neighbor's label w.r.t. $\bar{u}$.

3.3 A Compound Eigenspace for Recognizing Directed Human Motions

In this section, we present a new framework for recognizing human motions based on the idea of compound eigenspace. We have tackled different types of human motions in terms of direction, speed, and location. An eigenvector-based technique is employed to build a compound eigenspace that consists of a number of sub-eigenspaces. It reduces

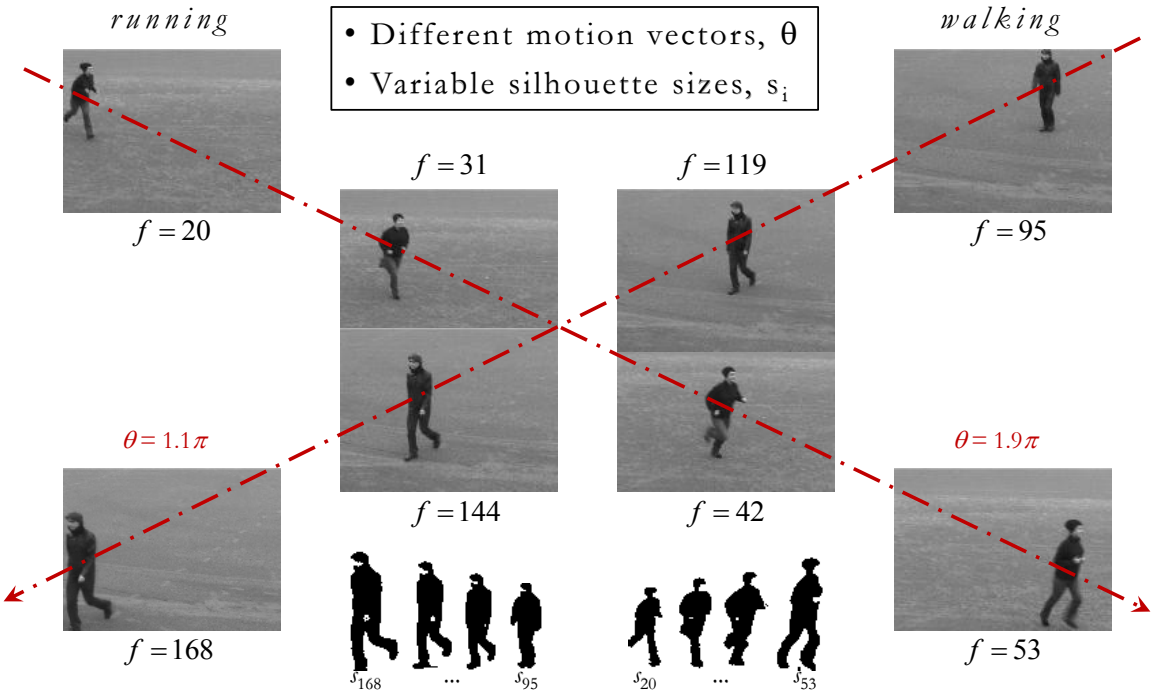


Figure 3.6: Multi-location motions that are unparallel to the image plane cause the silhouette size to have some important variations.

image dimensionality and recognizes new videos of human actions. Motions are partitioned and projected into sub-eigenspaces depending on certain criteria. The method of MII has been employed for compression as it can effectively compress a motion video clip into a simple, expressive, and concise image. It can be noticed that the proposed human motion representation in Section 3.1 does not take into account the variation in silhouette sizes caused by multi-location motions whose global displacement directions are not parallel to the image plane, see Fig. 3.6.

In order for the MII to address this issue, an extra step of silhouette size normalization is added to the five steps shown in Fig. 3.1. This step takes place just before aggregation, so it is between (c) and (d) in the same figure. The most popular technique that has been used for image scaling is interpolation. There are three common interpolation-based methods: *Nearest Neighbor*, *Linear Interpolation* and *Cubic Interpolation*. Choosing the most

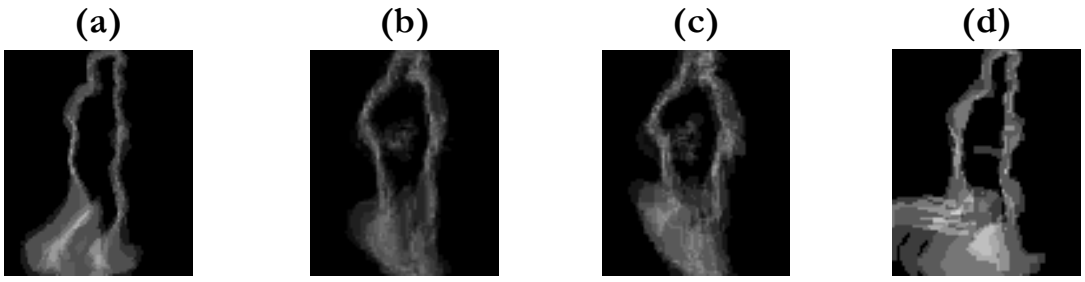


Figure 3.7: Displacement vector (θ): the two MIIs in (b) and (c) are more similar to each other than to their peers shown in (a) and (d), respectively.

appropriate interpolation algorithm for image size reduction relies on the require level of smoothness w.r.t. the resulting images. As Cubic interpolation is often chosen over other techniques in image scaling when quality is the biggest issue [52], we made use of it for silhouette size normalization.

Finally, recognizing test motions can be done by seeking the best matching sample in the activated sub-eigenspace based on Euclidean distance. This method is only applicable to fixed-camera sequences where the camera is static and does not pan, tilt, or move.

As linear algorithms, such as PCA, are too simplistic, they are prone to underfitting problem. When data samples belonging to a certain class do not bear common characteristics, using a simple linear eigenspace for data classification is not recommended. In the stream of human motion recognition and based on our experiments, it was found that motions affiliated to different classes, but share the same motion direction, for example, are more similar to each other than to their original classes, see Fig. 3.7.

In appearance-based methods, extracting more motion features from the input image sequence leads to better classification. Hence, we have introduced the extraction of two valuable features that can be used for better classification. These two features are *Relative speed s* and *displacement vector* θ, or *motion direction*. By having the object's relative speed, we can easily differentiate between *single-location* and *multi-location* motions. It

has been observed that the best recognition for multi-location motions can be obtained when the displacement vector is parallel to the image plane, see (a) and (d) in Fig 3.7. This is mainly because most of the shape information and speed of the limbs are preserved. The compound eigenspace model for classifying human motions is built based on the idea of partitioning the motions into subgroups according to both aspects, s and θ. As in Fig. 3.8, the compound model consists of two or more sub-eigenspaces with only one being active at any recognition time. Each sub-eigenspace is designated for recognizing a subgroup of human motions. The first sub-eigenspace recognizes single-location motions while the others are responsible for multi-location ones. The number of sub-eigenspaces required for recognizing multi-location motions relies on how θ is partitioned. More explanation about how this model work is covered in the following.

3.3.1 Training

Given a set of short human motion videos for training $v_1, v_2, ..., v_N$ along with their affiliations $\mathbf{Y} = \{y_1, y_2, ..., y_N\}$, the following steps are taken in sequence for building a compound eigenspace that is able to recognize new videos:

1. For each video clip v_i, a corresponding human motion representation $\mathbf{x}_i$ is computed based on Section 3.1, so that $\mathbf{x}_i = MII(v_i),\ i \in \{1...N\}$.

2. For each binary image sequence, as in Fig 3.1(b), a linear displacement vector θ_i is approximated using the technique of linear least squares, and a relative speed s_i is computed by:

$$s_i = \frac{\left|\sum_{f=2}^{F}(h_f - h_{f-1})\right| + \sum_{f=2}^{F}\left(\frac{d(c_f, c_{f-1})}{\text{avg}(h_f, h_{f-1})}\right)}{F-1},\ i \in \{1...N\}, \tag{3.6}$$

   where h represents silhouette height, F is the number of frames, c is the silhouette centroid, and $d(,)$ is a function to compute Euclidean distance.

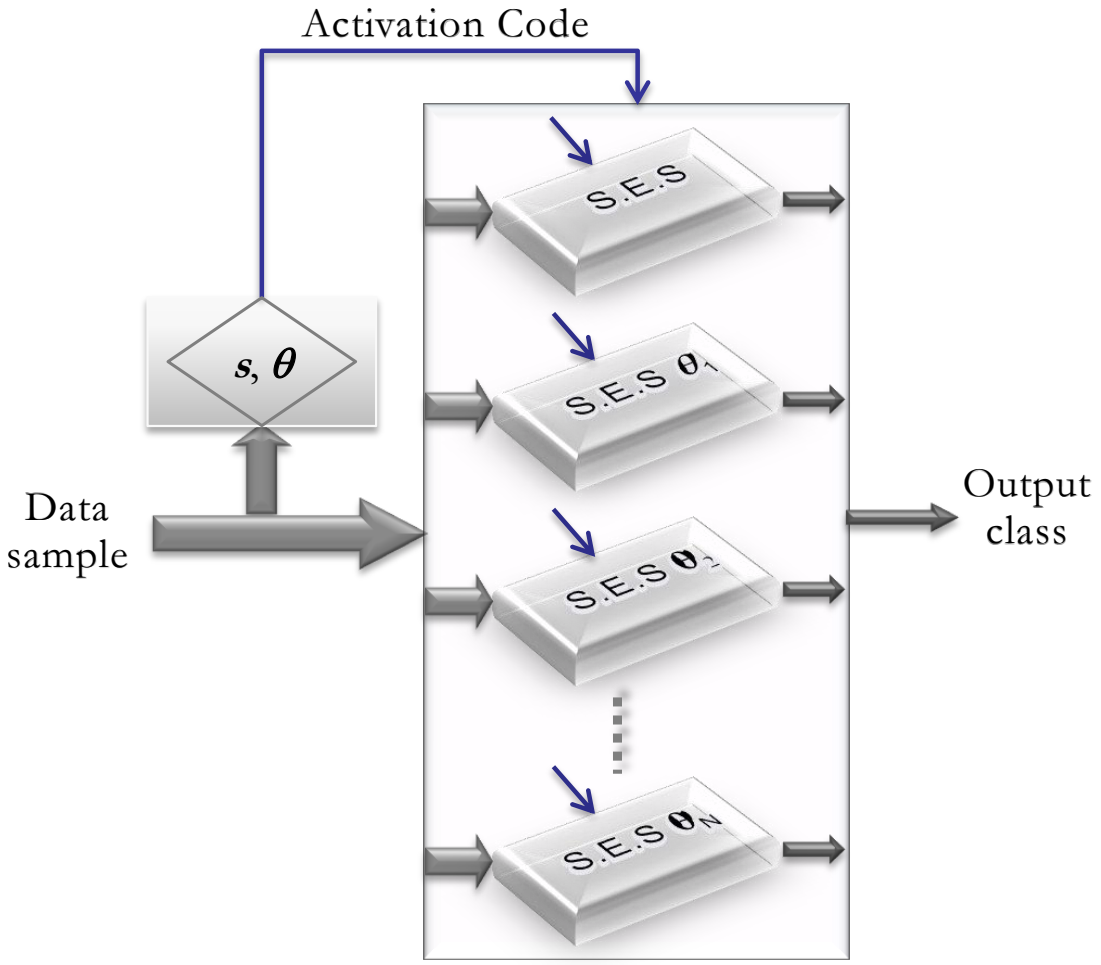


Figure 3.8: A compound eigenspace that partitions human motions into subgroups based on some aspects. It consists of one single-location sub-eigenspace and several multi-location sub-eigenspaces. It is controlled by both s and θ

.

3. Data samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ along with their class labels are then partitioned into M subsets based on their corresponding s_i and θ_i

$$\mathbf{X} = \cup_{m=1}^{M} \mathbf{X}_m, \begin{cases} \mathbf{X}_1 = (\mathbf{X}_1 \cup \mathbf{x}_i), & \text{if } s_i < s_{thr} \\ \mathbf{X}_p = (\mathbf{X}_p \cup \mathbf{x}_i), & \text{otherwise } (p = 2 + \frac{\theta_i}{\lambda}) \end{cases} \tag{3.7}$$

where $p > 1$ is the partition number and λ is the partition size such that $(2\pi/\lambda = M - 1)$.

4. Using each partition, $\mathbf{X}_i = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_i}\}$, $(i \in \{1, ..., M\}$ and $N = \sum_{i=1}^{M} n_i)$, a single eigenspace, $S.E.S._i$, is trained and built individually based on the methodology of PCA. Each data vector $\mathbf{x}_j \in \mathbf{X}_i$ is normalized and centered using (3.8) and (3.9), respectively. A square covariance matrix C_i from the outer product of matrix $\mathbf{X}_i$ with itself as in (3.10).

$$\| \mathbf{x}_j \| = 1 \tag{3.8}$$

$$\mathbf{X}_i = \{\mathbf{x}_1 - \mathbf{c}_i, \mathbf{x}_2 - \mathbf{c}_i, ..., \mathbf{x}_{n_i} - \mathbf{c}_i\}, \mathbf{c}_i = \frac{1}{N} \sum_{j \in n_i} \mathbf{x}_j. \tag{3.9}$$

$$\mathbf{C}_i = \frac{1}{n_i} \sum \mathbf{X}_i . \mathbf{X}_i^T. \tag{3.10}$$

By solving the eigenvalue problem of $\mathbf{C}_i$, D dominant vectors $\{\mathbf{v}_1, \mathbf{v}_2, .., \mathbf{v}_D\}$ corresponding to the largest D eigenvalues are chosen as a transformation matrix $\mathbf{T}_i$ to create the eigenspace $S.E.S._i$. Each training motion $\mathbf{x} \in \mathbf{X}_i$ is then projected into $S.E.S._i$ as a point e using (3.11).

$$e = (\mathbf{v}_1, \mathbf{v}_2, .., \mathbf{v}_D)^T (\mathbf{x} - \mathbf{c}_i) \tag{3.11}$$

5. Now we have M sub-eigenspaces of D dimensions each, built and ready to work.

3.3.2 Testing

Testing a video v for human motion recognition consists of the following steps:

1. A corresponding human motion representation $\mathbf{x} = MII(v)$ is computed based on Section 3.1.

2. A linear displacement vector θ is estimated using the technique of linear least squares, and a relative speed s is computed by Eq. (3.6).

3. A sub-eigenspace is activated based on s and θ

$$E_i = \begin{cases} S.E.S_1, & \text{if } s < s_{thr} \\ S.E.S_p, & \text{otherwise } (p = 2 + \frac{\theta}{\lambda}) \end{cases} \tag{3.12}$$

4. The motion being tested, $\mathbf{x}$, is then projected into E_i as a point e using Eq. (3.11).

5. Searching for the nearest neighbor point $\mathbf{e}_j \in E_i$ to e and v's class is, therefore, equal to $y_j \in \mathbf{Y}_i$.

3.4 Experimental Results

The proposed models have been implemented in C++, and experimental results obtained using a 1.6 GHz PC. In this section, we demonstrate and evaluate the feasibility of our eigenvector-based motion recognition approach, where we ran several experiments on a commonly used benchmark human actions dataset, KTH.

KTH Dataset. This motions dataset introduced in [83] is the most commonly used dataset in evaluating human motion recognition. It consists of 600 (160×120) videos, with 25 actors performing six motions: *boxing*, *hand-clapping*, *2-hand-waving*, *walking*, *jogging*, and *running* under 4 scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4). Each sequence is further divided into shorter clips. Illustrative examples for each of these motions are shown in Fig. 3.9. This dataset is more challenging than other datasets such as the Weizmann dataset [40]. That is due to

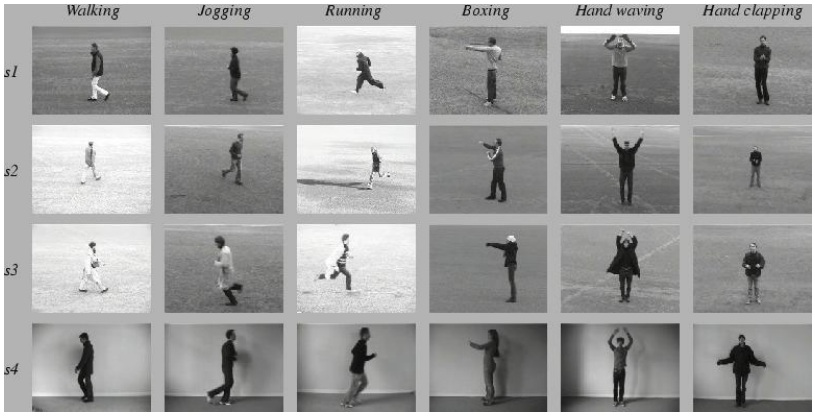


Figure 3.9: Representative frames from the KTH motion dataset.

the existence of two very similar motions, *jogging* and *running*, in the KTH dataset. Additionally, extracting silhouettes in this dataset is not straightforward due to the instability of the recording conditions of the videos, as well as the existence of a significant amount of camera shaking in many cases.

3.4.1 Simple-structure PCA-based and FDA-based Eigenspaces

The proposed eigenvector-based motion recognition model is trained using 625 short motion video clips taken from the three scenarios of the KTH dataset excluding the *outdoors with scale variation* scenario (s2). In the implementations, we align the human silhouettes to the image centre where each primary MII is a 160×120 matrix yielding 19 200 elements. A motion window of 88×108 is then used to capture the new set of MIIs from the centers the primary ones yielding smaller MIIs of 9 504 elements. This window size has been chosen so that it is large enough to capture the motion within every single MII in the KTH dataset video clips. For the purpose of computation, the MIIs are further scaled down to 22×27, reducing the original data size about 32 times. It should be noted that these reductions are performed while preserving the characteristics of the MIIs. The training set of MIIs was then analyzed using each of PCA and FDA algorithms, and two transformation matrices were computed. Two separate eigenspaces were then built by projecting the

input→	***box***		***clap***		***wave***		***walk***		***jog***		***run***	
output↓	PCA	FDA	PCA	FDA	PCA	FDA	PCA	FDA	PCA	FDA	PCA	FDA
box	85	100	5	0	7.5	0	0	0	0	0	0	0
clap	10	0	90	100	0	0	0	0	0	0	0	0
wave	5	0	5	0	92.5	100	0	0	0	0	0	0
walk	0	0	0	0	0	0	97.5	97.5	10	5	0	0
jog	0	0	0	0	0	0	0	2.5	77.5	80	17.5	12.5
run	0	0	0	0	0	0	2.5	0	12.5	15	82.5	87.5

Table 3.1: Confusion matrix using our methods of (MII+PCA) and (MII+FDA) on the KTH motions dataset based on a simple eigenspace structure. Average accuracies are 87.5% and 94.2%, respectively.

set of training MIIs on the dominant eigenvectors in each transformation matrix. As the dimensionality in FDA is upper-bounded by $C-1$, the FDA-based eigenspace has 5 axes, whereas 10 axes were used to build the PCA-based one.

The proposed recognition system has been tested using 150 short motion video clips (i.e. other than those used in the training stage). The obtained recognition results, which represent the rate of successful matching and mismatching, are reported in Table 3.1. It shows the recognition rates obtained by applying each of PCA and FDA. The results clearly show that FDA is about 6.7% superior over PCA although the latter has double the dimensionality of the prior. This superiority is due to the class-discriminatory power of FDA relative to PCA which is unsupervised and more oriented to feature extraction. By taking into account the FDA columns in Table 3.1, the results show perfect motion recognition for *boxing*, *hand-clapping* and *hand-waving*, with no mismatches. This demonstrates the robustness of the proposed recognition technique which is based on variances that discriminate between motion classes. The *walking* motion, however, has been recognized with a success rate of 97.5% against a very small mismatch rate of 2.5% for the *jogging* motion. Although this can be seen as a shortcoming, one has to realize that this mismatch rate is reasonable and could be attributed to some fast-motion *walking*, encountered in some test

Related work	Accuracy (%)
Schuldt et al. in [83]	71.7
Niebles et al. in [70]	81.5
Jiang et al. in [44]	84.4
Wong et al. in [100]	91.6
Our method	**94.2**

Table 3.2: Comparison of our method (MII+FDA) to other methods that have reported results over the KTH dataset.

videos, which were confused with *jogging*. The *jogging* motion has also been recognized with a success rate of 80% against mismatch rates of 5% and 15% for *waking* and *running* respectively. These results can be explained by the similarities that exist between these three human motions at their borderlines, and the fact that *jogging* is closer to *running* than *walking*. In the test video clips, few cases of slow-motion and fast-motion *jogging* were confused with *walking* and *running* respectively. Finally, the *running* motion has been recognized with a success rate of 87.5% against a mismatch rate of 12.5% for *jogging* which could be attributed to some cases of slow-motion *running* that were confused with *jogging*. It should be noted here that the mismatch rates, across the 150 testing video clips, for *walking-jogging* (2.5% and 5%) and *jogging-running* (15% and 12.5%) are remarkably consistent, which confirms the robustness of the proposed recognition technique.
The obtained results emphasize the effectiveness of the proposed human behavior recognition system when compared to methods that used the same dataset and listed in Table 3.2.

The experimental results given in Section 3.1 and shown in Fig. 3.3 illustrate the benefits that have been gained by applying the human-silhouette centering process. The latter represents one of the main pre-processing stages and has a valuable influence on the entire recognition process. It does not only help producing expressive compressed images, as shown in Fig.3.1.d, but also allows for a huge reduction in the MII window which is in turn

Recognition sub-process	Time (msec)	(%)
BG subtraction (per frame)	14.6	72.6
Silhouette centering (per frame)	0.6	3.0
Compression (per frame)	0.4	2.0
Scale normalization (per MII)	2.6	12.9
Eigenspace proj. (per sample)	1.8	9.0
NN retrieving (per sample)	0.1	0.5
Total time at every new frame	20.1	100

Table 3.3: The execution time for each sub-process in the proposed recognition system..

used in the recognition process (Fig.3.1.e). Thus, yielding more efficient recognition.

We conducted one further experiment to demonstrate that the recognition process of the proposed model can be considered for realtime systems. Table 3.3 shows the execution time for each sub-process involved in the proposed recognition system. It has been found that the overall recognition time per image frame is 20.1 milliseconds. It can also be observed that 72.6% of this execution time is consumed by the background subtraction and human silhouette extraction, while only 5% of this time is spent in silhouette centering and image compression. The remaining 22.4% of the execution time is the actual time for high level recognition. To evaluate the gain, in terms of execution time, achieved from applying the silhouette centering process, we ran the recognition algorithm without centering human silhouettes. We noticed an increase of 22% in the execution time, yielding a recognition time per image of 24.5 milliseconds. Therefore, it can be concluded that aligning human silhouettes to a reference point significantly enhances both recognition and execution time. This is mainly because this process restricts motion features in a compact area, making data more concise and expressive. Hence, subsequent processes deal only with a specific window size where the motion is located and concentrated. It can be concluded that this recognition system can be deployed in real-time applications using, for instance,

NTSC video where the video frame rate is $30 fps$, and a single frame is released every 33.3 milliseconds.

3.4.2 A Compound-structure Eigenspace

To evaluate the compound-structured eigenspace model, we compared it with the simple-structured model whose implementation details are demonstrated in Section 3.4.1. The obtained results when testing the simple model are shown in Table 3.1. By including scenario (s2) to it, the model was clearly undermined by the underfitting problem, yielding very poor recognition results. This is because this model cannot generalize well due to the lack of valuable discriminating features and the high level of similarity between different motions as shown in Fig. 3.7. As multiple sub-eigenspaces were used and the training samples had been divided into smaller groups, FDA is no longer the dominant technique, thus, we used PCA instead. We ran 100 experiments each of which with different combination of training-testing samples to show how useful and efficient using the compound eigenspace is to improve the classification results.

When implementing the idea of compound eigenspace and for every experiment using the four scenarios of the KTH dataset, we found that first, all training samples of *boxing*, *hand waving*, and *hand clapping* are partitioned as single-location motions. Second, six displacement vectors $\theta_{i=1,\ldots,6}$ are found for the other three motions, $\{0\pm5, 10\pm5, 170\pm5, 180\pm5, 190\pm5, 250\pm5\}$. By flipping the final MIIs of the multi-location motions with $\frac{\pi}{2} < \theta < \frac{3\pi}{2}$ around Y-axis, we decrease the required number of sub-eigenspaces to the half. A compound of four S.E.S.s is trained and built for every experiment such that 1 for single-location motions and 3 for multi-location motions ($\theta \in \{0\pm5, 10\pm5, 250\pm5\}$). The obtained results are averaged and reported as a confusion matrix shown in Table 3.4. Compared to Table 3.1(PCA columns), clear improvement has been achieved over using a simple-structured eigenspace at all levels. They are +7.5%, +5.0%, +3.8%, 0.0%, +1.3%, and +2.0% for *boxing*, *hand clapping*, *hand waving*, *walking*, *jogging*, and *running*, re-

input→ / **output↓**	***box***	***clap***	***wave***	***walk***			***jog***			***run***		
				↔	↖↗	↙↘	↔	↖↗	↙↘	↔	↖↗	↙↘
box	92.5	2.5	3.7	0	0	0	0	0	0	0	0	0
clap	5.0	95.0	0	0	0	0	0	0	0	0	0	0
wave	2.5	2.5	96.3	0	0	0	0	0	0	0	0	0
walk	0	0	0	97.5	100	100	7.1	14.3	21.4	0	0	0
jog	0	0	0	0	0	0	78.8	85.7	71.5	15.5	42.8	35.7
run	0	0	0	2.5	0	0	14.1	0	7.1	84.5	57.2	64.3

Table 3.4: Confusion matrix using compound eigenspace structure on the KTH motions dataset.

spectively. With respect to recognizing motions with scale variation, the compound model successfully recognizes *walking* motions and provides good accuracy of 78.6% for *jogging* and modest for *running*.

Chapter 4

Kernel Techniques for Human Motion Recognition and Data Classification

Mapping data from a linearly inseparable space to a higher dimensional Hilbert space where data classes can linearly be separable has attracted a great deal of attention from researchers for the purpose of data classification in general [75]. Kernel-based approaches are the better choice whenever a non-linear classification model is needed. Many researchers have shown that methods that are employing the kernel technique are computationally efficient and robust, and provide significant improvement in pattern analysis and data classification [49, 62, 85, 97, 102]. In these methods, the original data points are first mapped into a higher dimensional feature space, then, performing classification using a linear method. Such mapping is performed implicitly using a function with certain properties through a mathematical shortcut called the "kernel trick" [3, 12, 80] (see Chapter 2). Hence, when using any kernel function with any linear classification model, the originally linear operations are done in a reproducing Hilbert space, obtained through an implicit non-linear

mapping [82].

The objective of this chapter is to present two proposed nonlinear classification approaches using the combination of kernel trick and linear dimensionality reduction techniques. It, first, describes the framework of employing the kernel technique for human motion recognition. It, then, presents a novel kernel-based non-parametric method for general data classification, where a new derived expression of the objective function is presented. Implementation details, experimental results, and comparisons are given together with each model.

4.1 A Kernel-based Framework for Human Motion Recognition

It has been observed that the literature lacks the use of the kernel technique in the context of human motion recognition [24]. In this context, this section introduces a non-linear eigenvector-based recognition model that is built upon the idea of the kernel technique. It gives a practical study of using the kernel technique to show how crucial choosing the right kernel function for a specific application is, for the success of the linear discrimination in the feature space. We investigate about eighteen different kernels. The rich implementation results provided in this section were obtained by applying the model on two of the most common used benchmark datasets in the field of human motion recognition, KTH and Weizmann.

In our kernel-based model, and in order for each human motion video clip to be in the form of learning or recognition, it goes through a compression procedure that converts it into a single appearance-based template. The method of MII has been employed for this purpose as it can effectively compress a motion video clip into a simple, expressive, and concise form. Then, the technique of kernel PCA is employed to work in a non-linear feature space, reduces the dimensionality, and generates an eigenspace that contains all the

training motions. Finally, a database is used to store the projected samples that being used for recognizing new motions by seeking the best matching based on Euclidean distance.

4.1.1 Recognizing Human Motions using Kernel PCA

For motion representation purposes, an eigenspace is easy to apply [23]. Mapping data into a lower-dimensionality eigenspace provides significant improvement in recognition and computational aspects [96]. Many recognition systems have employed the eigenvector-based technique for dimensionality reduction. Here, we use the kernel-based version of PCA, KPCA. This algorithm is considered as an extension of its linear version to non-linear distributions. The algorithm of KPCA is to perform linear PCA in a high dimensional feature space $\mathcal{F}$ for the same purpose which is dimensionality reduction while guaranteeing maximum data variance. Similar to classical PCA, it seeks directions where data variation reaches the most. The best low-dimensional space that guarantees maximum data variance can be computed using the most dominant eigenvectors of the covariance matrix. These eigenvectors represent our new coordinate values and are called "kernel-based principal components" or "nonlinear principal components" [82]. They are corresponding to the largest nonzero eigenvalues.

With respect to human motion recognition, the objective of PCA is to project the entire set of training samples, or human motions, onto an eigenspace using a relatively small number of coordinates [23]

At this stage, a training set of human motions is used to generate a linear model that preserves large data variance and to build a low dimensional eigenspace. Similar to conventional PCA described in Section 2.3.2, each final MII as a training sample, $I_{n\in\{1,2,..,N\}}$, is formed as an image matrix $x_i = \{p_1, p_2, .., p_d\}$ and then normalized as $\|x_i\| = 1$. The whole set of normalized MIIs as data vectors are then centered based on the mean image μ and a super training matrix X is then constructed as in Eq. 4.1.

$$X = (x_1 - \mu, x_2 - \mu, .., x_N - \mu) \tag{4.1}$$

where the average image μ is defined by,

$$\mu = \frac{1}{N} \sum_{n=1}^{N} x_n \tag{4.2}$$

The super data matrix X is $d \times N$, where N is the total number of training MIIs, and d is the length of sample vector in the input space $\mathcal{X}$. In order for employing the kernel technique, data has to be transferred into an inner dot-product kernel space generating the Kernel matrix K as in Eq. 4.3. This operation depends mainly on the chosen kernel function $k(.,.)$ that dictates how data be non-linearly transferred to the feature space $\mathcal{F}$, implicitly. Crucially, K is $N \times N$ here and does not depend on d^{Φ}. Therefore it can be computed in a run time that depends only on N.

$$K = \Phi(X)^T \Phi(X) = \sum_{i,j}^{N} k(x_i, x_j) \tag{4.3}$$

To compute the non-linear eigenvectors of the input data, we calculate the eigenvalues of the $N \times N$ kernel matrix K by solving its eigenvalue problem using the SVD method. From N eigenvectors, the first $\acute{d}$ eigenvectors, $(v_1, v_2, ..., v_{\acute{d}})$, corresponding to the top $\acute{d}$ eigenvalues are chosen to represent the projection matrix. An eigenspace E is then built by projecting the entire training set using the extracted eigenvectors. Each data sample x is then projected into E as a point e based on Eq. 4.4.

$$e = (v_1, v_2, .., v_{\acute{d}})^T k(x, x_i) \tag{4.4}$$

For the purpose of recognizing new or unknown motion sequences, we store the generated eigenspace into a database for the purpose of matching.

In eigenspace-based techniques, recognizing a new motion is simply performed by seeking the most similar one in the produced eigenspace E. Then the new motion is classified as the best matching motion's class, or label.

To reduce the dimensionality of the new motion image and prepare it for the sake of search, it needs to be normalized and transferred into the inner-dot product space by kernelizing it using the same kernel function $k(x,y)$ that was used for training. Once it is formed as the ones in the database using Eq. 4.4, classification is simply done by seeking the most similar motion that is stored in the database. This similarity measure, or search, is done by finding the point with the minimum Euclidian distance D w.r.t the first.

4.1.2 Kernel Functions

One of the most important issues that contributed to the success of any kernel-based approach is selecting the most suitable kernel function. This task is highly depending on the problem at hand because it relies on what we are attempting to model. In contrast to the linear kernel, for example, that allows us only to choose lines or hyperplanes, radial basis functions (RBF) such as Gaussian, Exponential, or Laplacian give us the ability to pick out circles or hyperspheres [60]. On the other hand, Polynomial kernel allows us to model feature conjunctions up to the order of the polynomial [60]. This task involves one exhausting and tedious step which is the fine tuning of the kernel function's parameters [87]. Eighteen different kernel functions are deployed individually in the experiments in order to find the most appropriate one for the problem. In the following is a list of the 18 kernels and their formulas in details [87]:

(a) Linear:

$$k(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{y} + c \tag{4.5}$$

(b) Polynomial:

$$k(\mathbf{x},\mathbf{y}) = (\mathbf{x}^T\mathbf{y} + c)^d \tag{4.6}$$

(c) Gaussian:

$$k(\mathbf{x},\mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}) \tag{4.7}$$

(d) Exponential:

$$k(\mathbf{x},\mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|}{2\sigma^2}) \tag{4.8}$$

(e) Laplacian:

$$k(\mathbf{x},\mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|}{\sigma}) \tag{4.9}$$

(f) Sigmoid:

$$k(\mathbf{x},\mathbf{y}) = \tanh(s\mathbf{x}^T\mathbf{y}+c) \tag{4.10}$$

(g) ANOVA:

$$k(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{n} \exp(-\sigma(x^k-y^k)^2)^d \tag{4.11}$$

(h) Multiquadric:

$$k(\mathbf{x},\mathbf{y}) = \sqrt{\|x-y\|^2+c^2} \tag{4.12}$$

(i) Circular:

$$k(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{2}{\pi}\arccos(-\frac{\|x-y\|}{\sigma}) - \frac{2}{\pi}\frac{\|x-y\|}{\sigma}\sqrt{1-(\frac{\|x-y\|}{\sigma})^2}, & \text{if } \|x-y\| < \sigma; \\ 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

(j) Spherical:

$$k(\mathbf{x},\mathbf{y}) = \begin{cases} 1-\frac{3}{2}\frac{\|x-y\|}{\sigma}+\frac{1}{2}(\frac{\|x-y\|}{\sigma})^3, & \text{if } \|x-y\| < \sigma; \\ 0, & \text{otherwise.} \end{cases} \tag{4.14}$$

(k) Wave:

$$k(\mathbf{x},\mathbf{y}) = \frac{\theta}{\|x-y\|}\sin\frac{\|x-y\|}{\theta} \tag{4.15}$$

(l) Power:

$$k(\mathbf{x},\mathbf{y}) = -\|x-y\|^d \tag{4.16}$$

(m) Log:

$$k(\mathbf{x},\mathbf{y}) = -\log(\|x-y\|^d+1) \tag{4.17}$$

(n) Spline:

$$k(\mathbf{x},\mathbf{y}) = 1 + xy + xy\min(x,y) - \frac{x+y}{2}\min(x,y)^2 + \frac{1}{3}\min(x,y)^3 \tag{4.18}$$

(o) Cauchy:

$$k(\mathbf{x},\mathbf{y}) = \frac{1}{1+\frac{\|x-y\|^d}{\sigma}} \tag{4.19}$$

(p) Histogram Intersection:

$$k(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n}\min(x_i,y_i) \tag{4.20}$$

(q) Generalized T-Student:

$$k(\mathbf{x},\mathbf{y}) = \frac{a}{1+\|x-y\|^d} \tag{4.21}$$

(r) Wavelet:

$$k(\mathbf{x},\mathbf{y}) = \prod_{i=1}^{N} w(\frac{x_i - t}{d}) w(\frac{y_i - t}{d}) \tag{4.22}$$

4.1.3 Implementation Results

This model has been implemented in C++, and works based on the one-against-one voting algorithm. In this section, we demonstrate and evaluate the feasibility and the efficiency of our kernel-based approach for classifying human motions, where we ran several experiments on commonly used benchmark human actions datasets, the KTH and Weizmann. We first combined the two datasets into a single dataset with 13 different motions to show how useful and efficient using the kernel technique is to achieve better recognition results. We ran several experiments, each of which with different kernel function and parameters. Then, we give an evaluation and detailed comparison between the 18 kernel functions' influence and performance. Cross-validation method was used for refining the parameter values in each experiment.

Weizmann Dataset. A benchmark human motions dataset introduced in [40] and commonly used in evaluating human motion recognition approaches. It consists of 93 ($180 \times$

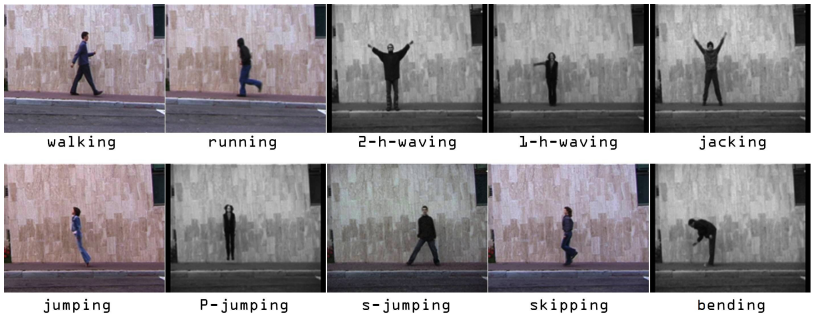


Figure 4.1: Representative frames from the Weizmann motion dataset.

144) short video sequences taken for nine actors performing 10 different motions each: *walking*, *running*, *2-hand waving*, *1-hand waving*, *jacking*, *jumping*, *in-place jumping*, *sideways jumping*, *skipping*, and *bending*. Representative examples for each of these motions are shown in Fig. 4.1.

Results and Analysis

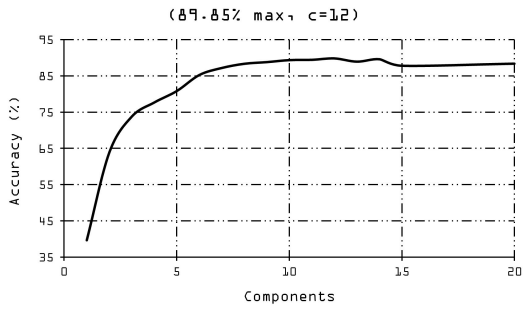
Fig. 4.2 shows the effectiveness of applying each individual kernel function on the combined KTH-Weizmann dataset. It consists of 18 sub-figures each of which is related to its corresponding kernel function in Section 4.1.2. It can be seen clearly that the recognition accuracy varies from one to another. Some of them are some how similar such as Linear, Circular, and Cauchy in figures 4.2a, 4.2i , and 4.2o, respectively. On the contrary, some others are dissimilar such as Sigmoid, Wave, and Spline in figures 4.2f, 4.2k , and 4.2n, respectively. Fig. 4.2 describes also the effect of the number of used kernel features, or components, on the achieved recognition rate. Despite the availability of a large number of kernel features, the experiments are content with 20 components at most. That is because it is reasonable and sufficient for dimensionality reduction and for building the eigenspace that stores the training samples and affects the recognition time. It can be noticed that the relationship between the number kernel-based principal components and the recognition accuracy varies from a kernel to another. In most kernels in the figure, the recognition

accuracy rises in some way as the number of kernel components increases until it reaches some point of stability. But we found that some kernel functions behave differently. In Wave kernel shown in Fig. 4.2k, for example, recognition accuracy starts going down after reaching a particular number of kernel components (4 features). This means that the kernel features beyond that limit are not useful for classification even if their eigenvalues are relatively high.

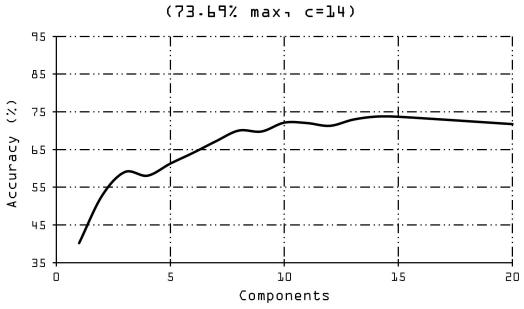
Relating to our kernel-based model using the combined KTH-Weizmann dataset, if we suppose that the number of kernel features does not matter as long as it is within a reasonable range, we look at Fig. 4.3 and conclude that Sigmoid, with 92.92%, is the most appropriate kernel function among the 18 kernels.

Since Linear kernel in Fig. 4.2a, with 89.85%, works exactly as linear classification in the input space $\mathcal{X}$, employing the kernel technique allows us to have better recognition accuracy. There might be some kernel function or a combination of more than a kernel can enhance the recognition accuracy even more.

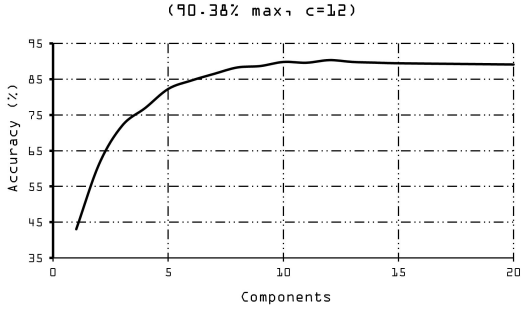
Fig. 4.4 describes how recognition accuracy is strongly influenced by changing the parameters values, upon which the kernel function relies on. The method of cross-validation was used in the experiments to fine tuning the kernel functions' parameters in order to find best combinations for achieving the highest accuracies. Figures 4.4b, 4.4c, 4.4d, 4.4h, 4.4j, 4.4k, and 4.4o show how σ effects the classification accuracy of each of the seven kernels. It also shows that each kernel function has distinct response of σ which reflects the difference of their data spreads in the implicit feature space F. Based on our observation, seeking a suitable value of the parameter σ is crucial when transforming data from its input space to the inner dot-product kernel space. This comment applies also on the other parameters such as the degree d in Figures 4.4a, 4.4g, 4.4m, 4.4n, and 4.4p.



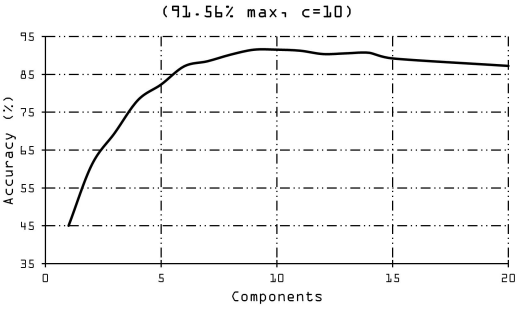
(a) Linear



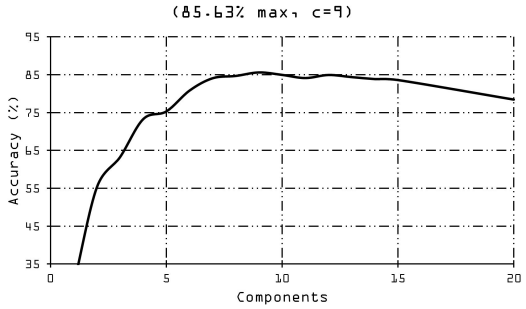
(b) Polynomial



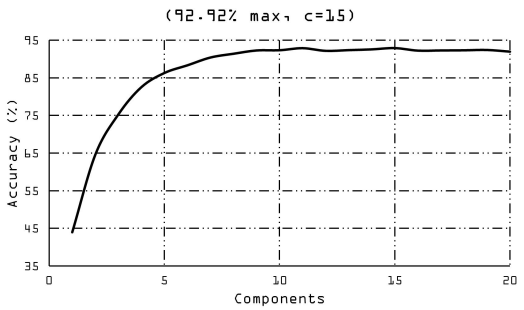
(c) Gaussian



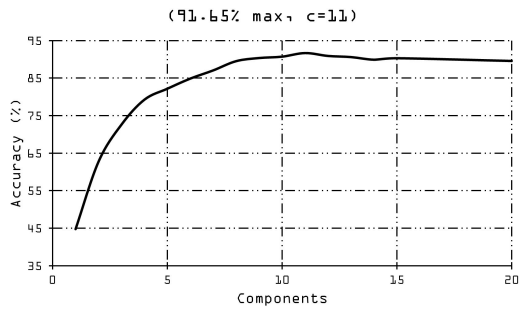
(d) Exponential



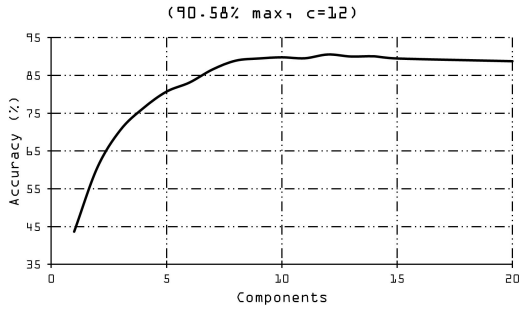
(e) Laplacian



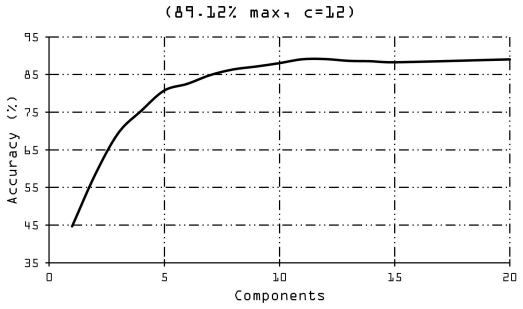
(f) Sigmoid



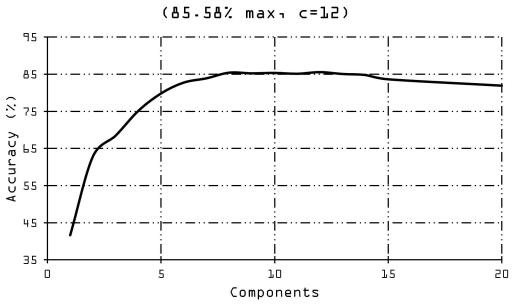
(g) ANOVA



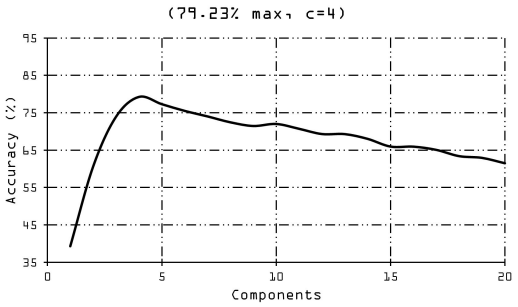
(h) Multiquadric



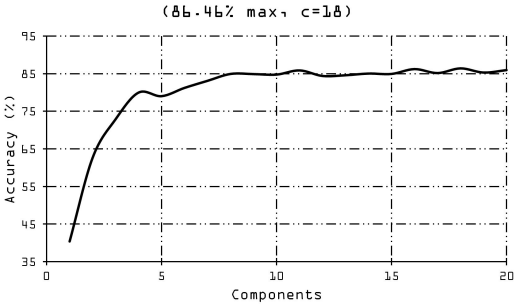
(i) Circular



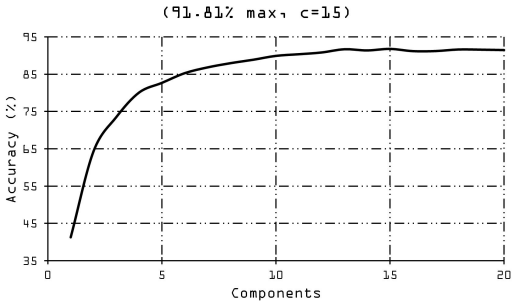
(j) Spherical



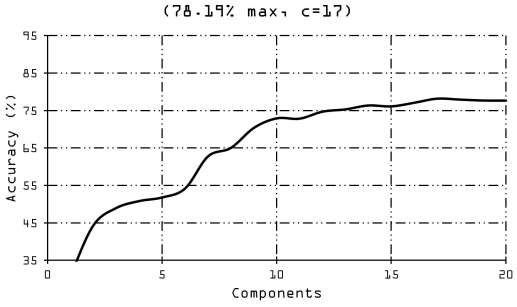
(k) Wave



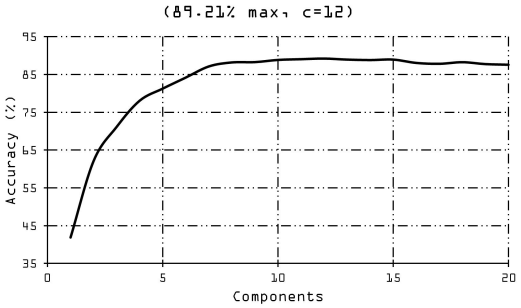
(l) Power



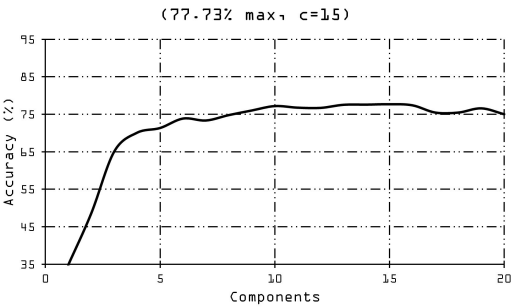
(m) Log



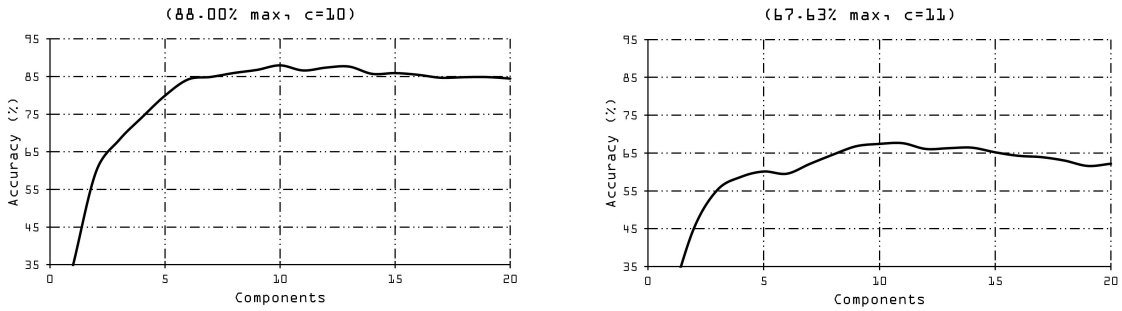
(n) Spline



(o) Cauchy



(p) Histogram Intersection



(q) T-student (r) Wavelet

Figure 4.2: The effect of the number of nonlinear principal components on the recognition accuracy.

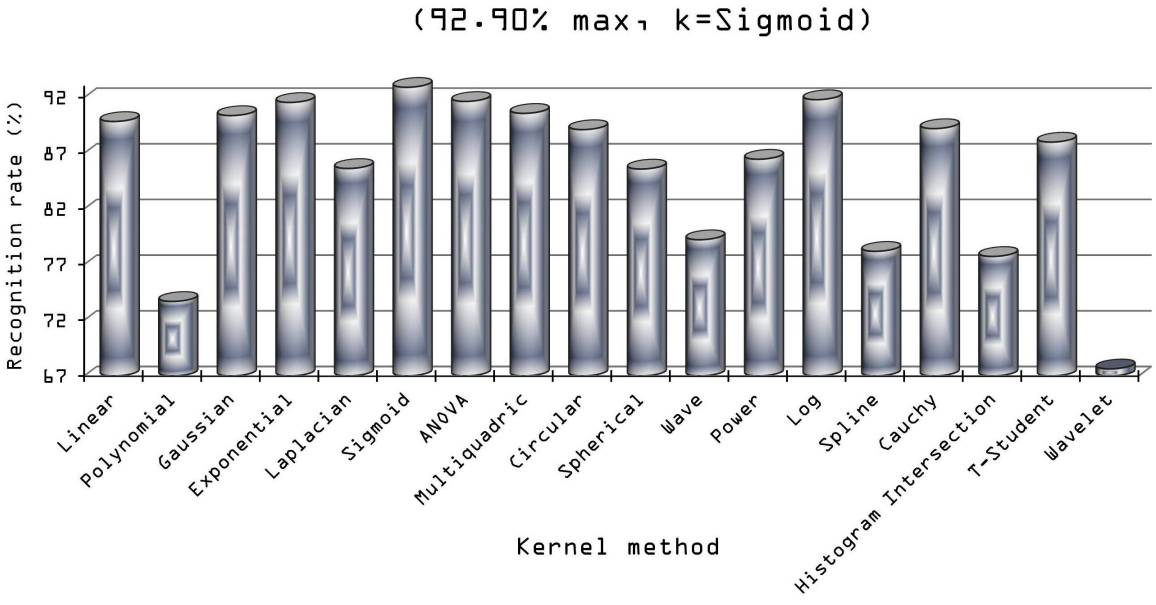
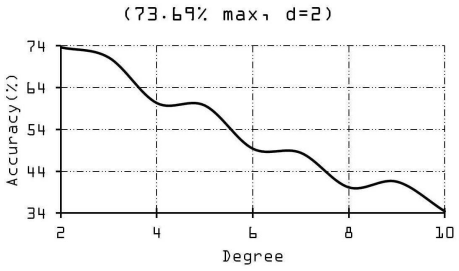
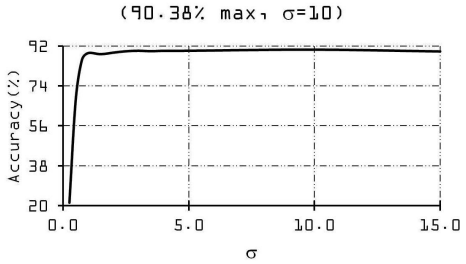


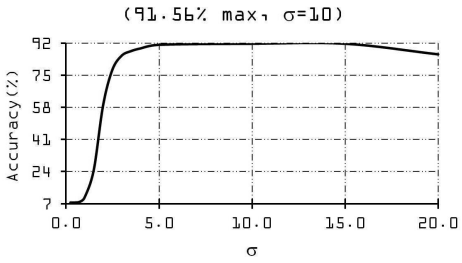
Figure 4.3: The highest achieved recognition accuracies for the 18 kernels.



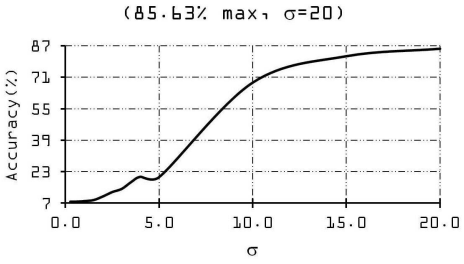
(a) Polynomial



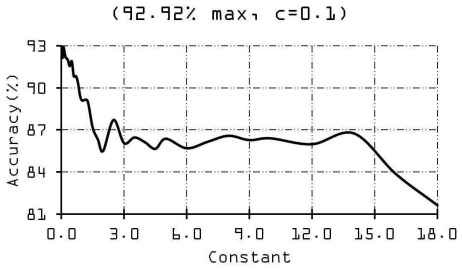
(b) Gaussian



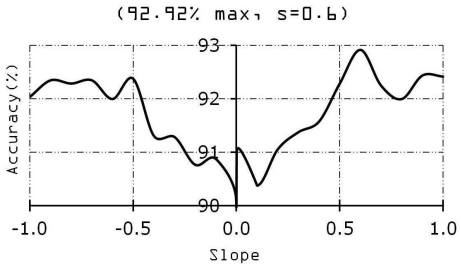
(c) Exponential



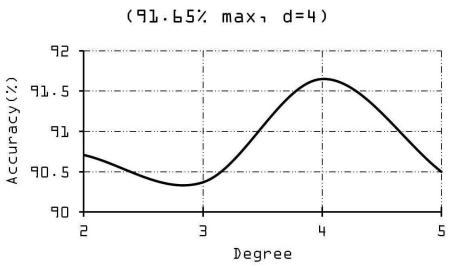
(d) Laplacian



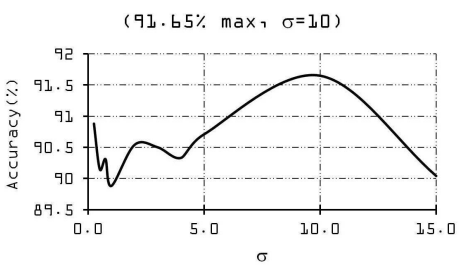
(e) Sigmoid



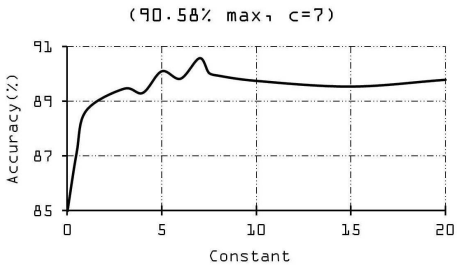
(f) Sigmoid



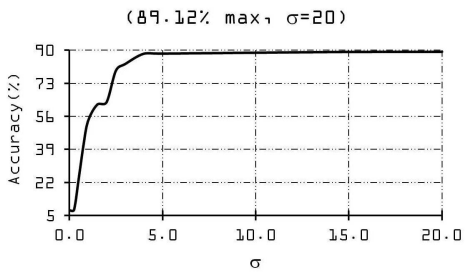
(g) ANOVA



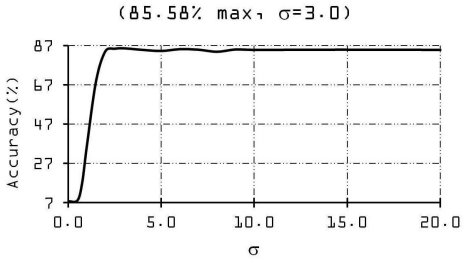
(h) ANOVA



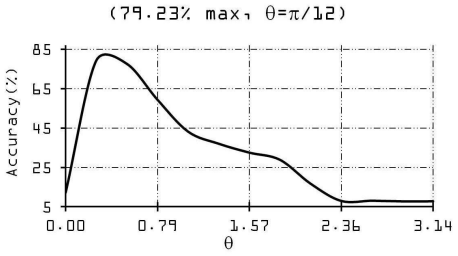
(i) Multiquadric



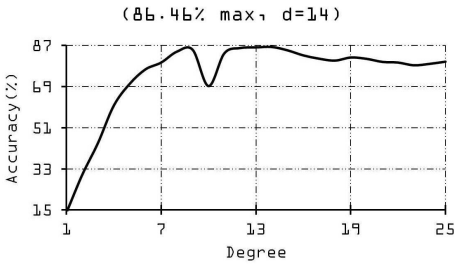
(j) Circular



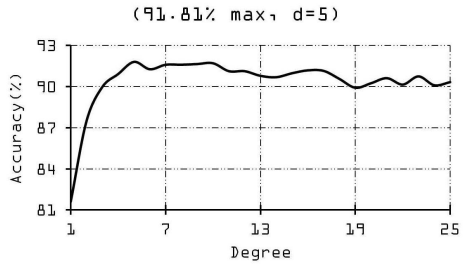
(k) Spherical



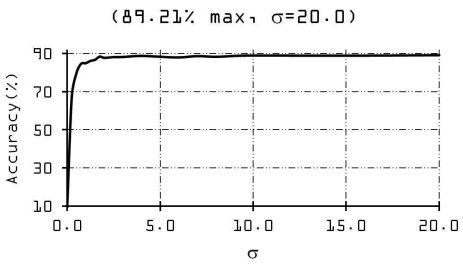
(l) Wave



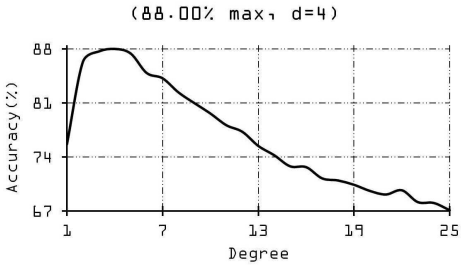
(m) Power



(n) Log



(o) Cauchy



(p) T-student

Figure 4.4: The refinement of the kernel functions' parameters for achieving highest accuracies.

4.2 The Proposed Non-parametric FDA with Kernels

As real data distribution is generally nonlinear, the demand for efficient non-linear classifiers has been growing. This section proposes a nonlinear classification approach based on the NDA (for more information about the latter see Chapter 2). This technique can efficiently find a nonparametric kernel representation where linear discriminants perform better. Data classification is achieved by integrating the linear version of the NDA with the kernel mapping. Based on the kernel trick, we provide a new formulation for Fisher's criterion, defined only in terms of the inner dot-product of the original data samples in their input data. The obtained classification accuracies have proven the competitiveness of our approach compared to other state-of-the-art approaches. An article out from this work has been accepted by the journal of Pattern Recognition Letters and will appear soon [26].

4.2.1 The KNPDA

In this section, we derived a supervised kernel-based approach. Its main goal is to map the data distributions from their input space $\mathcal{X}$ into an implicit feature space $\mathcal{F}$ using the kernel technique where they can be nonparametrically and linearly discriminated by a hyperplane. Thereby, yielding a non-linear discrimination in $\mathcal{X}$. The idea of this kernel non-parametric Discriminant Analysis approach is to adapt the powerful NDA technique to function in the kernel feature space $\mathcal{F}$. The proposed approach aims at relaxing the normality assumption of FDA using the nonparametric form of the between-class scatter matrix introduced in [38] and extended in [54], and behaves nonlinearly with the help of the kernel technique. This algorithm focuses directly on multi-class problems as it is more general than two-class problems.

Since the idea of linear discrimination still applies, even in $\mathcal{F}$, and because $\mathcal{F}$ cannot always be explicitly computed, we have derived a new objective expression that makes use of data samples in terms of only inner-dot products (or Gram matrix K) based on the *kernel trick*. The mapping of data from a space into a higher dimensional Hilbert space $\mathcal{H}$ is

achieved through a mapping function Φ as follows :

$$\Phi : \begin{cases} \mathbb{R}^n \to \mathbb{R}^m (n < m \leq \infty) \\ \mathcal{X} \to \mathcal{F} \\ X \mapsto \phi(X) \end{cases} \tag{4.23}$$

In the linear case, nonparametric Fisher's discriminants are computed by maximizing $S_W^{-1} S_B$ as in Eq. (2.23). This criterion can be defined in $\mathcal{F}$ using the mapped training patterns Φ as:

$$J(w_\Phi) = \frac{w_\Phi^T S_B^\Phi w_\Phi}{w_\Phi^T S_W^\Phi w_\Phi} \tag{4.24}$$

where any solution of $w_\Phi \in \mathcal{F}$ lies in the extent of all mapped training observations in $\mathcal{F}$ based on the theory of reproducing kernels [66]. Thus, w_Φ can be expressed in the expansion form:

$$w_\Phi = \sum_{i=1}^{N} \alpha_i \phi(x_i) = \phi(X)\alpha, \tag{4.25}$$

where α_i are called the expansion coefficients and they are different from α in Eq. (2.26). Therefore, we redefine Eq. (4.24) in terms of α as:

$$J(\alpha) = \frac{\alpha^T B \alpha}{\alpha^T A \alpha} \tag{4.26}$$

where

$$A = \phi(X)^T S_W^\Phi \phi(X), \text{ and} \tag{4.27}$$

$$B = \phi(X)^T S_B^\Phi \phi(X) \tag{4.28}$$

Similar to the definitions of S_W and S_B in Eq. (2.24) and Eq. (2.25), respectively, S_W^ϕ and S_B^Φ in $\mathcal{F}$ can be defined as follows:

$$S_W^{\Phi} = \frac{1}{N}\sum_{i=1}^{C}\sum_{l=1}^{N_i}(\phi(x_l^i)-\mu_i^{\Phi})(\phi(x_l^i)-\mu_i^{\Phi})^T, \tag{4.29}$$

$$S_B^{\Phi} = \frac{1}{N}\sum_{i=1}^{C}\sum_{\substack{j=1\\ j\neq i}}^{C}\sum_{l=1}^{N_i}\omega^{\Phi}(i,j,l)(\phi(x_l^i)-m_j^{\Phi}(\phi(x_l^i)))(\phi(x_l^i)-m_j^{\Phi}(\phi(x_l^i)))^T \tag{4.30}$$

where $\omega^{\Phi}(...)$ is a weighting function similar to the one in Eq. (2.26) but works in $\mathcal{F}$. It plays an important role to preserve the boundary structure between mapped classes by approaching 0.5 for samples near the classification boundary and dropping off to zero as data samples move away from the boundary. Our kernel-based definition for ω^{Φ} is given in Section 4.2.1. Unlike the model in [102] which computes non-parametric global information (*kNN*) in input space $\mathcal{X}$ amongst the data samples of X, we do it in the feature space $\mathcal{F}$ amongst the mapped data $\phi(X)$ as it is the case in the kernel-based approaches. That is because, when working in $\mathcal{F}$, border structure in this space is not necessarily similar to the one in $\mathcal{X}$, as data spreads change according to the mapping function Φ.

The mean vectors, $m_j^{\Phi}(\phi(x_l^i))$ in Eq. (4.30), are used to represent non-parametric global information about each class in $\mathcal{F}$, and can be defined as:

$$m_j^{\Phi}(\phi(x_l^i)) = \frac{1}{k}\sum_{p=1}^{k} nn(\phi(x_l^i),j,p) = NN_{il}^j 1_{\frac{1}{k}} \tag{4.31}$$

where each NN_{il}^j is an $(L^{\Phi}\times k)$ matrix that holds the k-nearest neighbors of $\phi(x_l^i)$ from class j in $\mathcal{F}$, and $1_{\frac{1}{k}}$ is a $(k\times 1)$ vector with all elements equal $\frac{1}{k}$.

To avoid carrying out this task using the high, or infinite, dimensional mapped data $\phi(X)$ in $\mathcal{F}$, we rely on the Gram matrix K instead, since K represents the inner-dot product of $\phi(X)$.

Sections 4.2.1 and 4.2.1 explain how we define each of the numerator and denominator of Eq. (4.26) in terms of K, using the *kernel trick*, and how we end up with Eq. (4.37) and Eq. (4.45), respectively.

Combining these two definitions as in Eq. (4.26), we can determine Fisher's linear discriminants in $\mathcal{F}$ through maximizing Eq. (4.26) and finding the leading eigenvectors where their eigenvalues of $A^{-1}B$ are the highest. An eigenvalue in discriminant analysis is a strength measure of discrimination for each eigenvector. It is an indication of how well that eigenvector differentiates the classes, where the larger the eigenvalue, the better the eigenvector differentiates.

For any data sample x_s to be projected into a one dimensional eigenspace, we use the best eigenvector α with the highest eigenvalue λ_1, as in the following equation:

$$w_{\Phi}^{T}.\phi(x_s) = \alpha^{T}.k(X,x_s) = \sum_{i=1}^{N} \alpha_i k(x_i,x_s) \tag{4.32}$$

In general, to compute $\acute{L}$ non-linear discriminant features of the input data, we calculate the top eigenvalues of $A^{-1}B$ and their corresponding eigenvectors by solving the eigenvalue problem using SVD. From N eigenvectors, the best $\acute{L}$ eigenvectors $(\alpha_1,\alpha_2,..,\alpha_{\acute{L}})$, whose eigenvalues are the highest, are chosen to represent the projection matrix. An eigenspace E is then built by projecting the entire training set of samples, such that each sample x_i is represented as a point e based on the following equation:

$$e = (\alpha_1,\alpha_2,..,\alpha_{\acute{L}})^{T}.k(X,x_i) \tag{4.33}$$

In eigenspace-based techniques, recognizing a new sample is simply done by seeking the most similar one in the produced eigenspace E using the Euclidean metric. For the purpose of classifying a test sample x_t, we project it into E using Eq. (4.33) and it is then classified as the class of its best matching training sample in E.

The Between-class Scatter Matrix S_B^{Φ} in $\mathcal{F}$

Using the definition of $m_j^{\Phi}(\phi(x_l^i))$ in Eq. (4.31), we can rewrite the between-class scatter matrix of the mapped data S_B^{Φ} in Eq. (4.30) as:

$$S_B^{\Phi} = \frac{1}{N}\sum_{i=1}^{C}\sum_{\substack{j=1\\ j\neq i}}^{C}\sum_{l=1}^{N_i}\omega^{\Phi}(i,j,l)(\phi(x_l^i) - NN_{il}^j 1_{\frac{1}{k}})(\phi(x_l^i) - NN_{il}^j 1_{\frac{1}{k}})^T \tag{4.34}$$

Therefore B in Eq. (4.28) becomes:

$$B = \frac{1}{N}\sum_{i=1}^{C}\sum_{\substack{j=1\\ j\neq i}}^{C}\sum_{l=1}^{N_i}\omega^{\Phi}(i,j,l)\underbrace{\phi(X)^T(\phi(x_l^i) - NN_{il}^j 1_{\frac{1}{k}})}_{\text{P1}}\underbrace{(\phi(x_l^i) - NN_{il}^j 1_{\frac{1}{k}})^T\phi(X)}_{\text{P2}} \tag{4.35}$$

Let's consider the left part of the dot product (P1) along with the definitions of the kernel matrix and kernel function ($K = \langle\phi(X),\phi(X)\rangle$ and $k(x_i,x_j) = K_{ij} = \langle\phi(x_i),\phi(x_j)\rangle$),

$$\begin{aligned}\phi(X)^T(\phi(x_l^i) - NN_{il}^j 1_{\frac{1}{k}}) &= \phi(X)^T\phi(x_l^i) - \phi(X)^T NN_{il}^j 1_{\frac{1}{k}} \\ &= K_l^i - NNK_{il}^j 1_{\frac{1}{k}},\ \text{where } NNK_{il}^j \subset K\end{aligned} \tag{4.36}$$

where K_l^i is an $(N\times 1)$ kernel vector of data sample $x_l \in X_i$ and NNK_{il}^j is an $(N\times k)$ matrix which holds the kernel vectors of NN_{il}^j. Equation (4.36) represents the transition point of using the inner-dot product kernel space $\mathcal{K}$ instead of the feature space $\mathcal{F}$. Hence, we are able to represent B as:

$$B = \frac{1}{N}\sum_{i=1}^{C}\sum_{\substack{j=1\\ j\neq i}}^{C}\sum_{l=1}^{N_i}\omega^{\Phi}(i,j,l)(K_l^i - NNK_{il}^j 1_{\frac{1}{k}})(K_l^i - NNK_{il}^j 1_{\frac{1}{k}})^T \tag{4.37}$$

In order for B in Eq. (4.37) to be fully defined in $\mathcal{K}$, the weighting function $\omega^{\Phi}(i,j,l)$ needs to be defined based on K. We define the value of the weighting function in feature space, denoted as $\omega^{\Phi}(i,j,l)$, as:

$$\omega^{\Phi}(i,j,l) = \frac{\min\left\{d^{\alpha}(\phi(x_l^i), nn_k(\phi(x_l^i),i)), d^{\alpha}(\phi(x_l^i), nn_k(\phi(x_l^i),j))\right\}}{d^{\alpha}(\phi(x_l^i), nn_k(\phi(x_l^i),i)) + d^{\alpha}(\phi(x_l^i), nn_k(\phi(x_l^i),j))} \tag{4.38}$$

Using NN_{il}^j, defined in Eq. (4.31), we rewrite Eq. (4.38) as:

$$\omega^{\Phi}(i,j,l) = \frac{\min\left\{d^{\alpha}(\phi(x_l^i), NN_{il}^i[k]), d^{\alpha}(\phi(x_l^i), NN_{il}^j[k])\right\}}{d^{\alpha}(\phi(x_l^i), NN_{il}^i[k]) + d^{\alpha}(\phi(x_l^i), NN_{il}^j[k])} \tag{4.39}$$

where $d(\phi(p),\phi(q))$ is the Euclidean distance between two mapped points, $\phi(p)$ and $\phi(q)$, in $\mathcal{F}$. The kernel-based formula of d that computes the distance in $\mathcal{F}$ using the original points p and q in $\mathcal{X}$, $d_{\Phi}(p,q)$, can simply be derived as in [98] and given by Eq. (4.40).

$$\begin{aligned} d_{\Phi}^2(x_p,x_q) &= d^2(\phi(x_p),\phi(x_q)) = \sum_{l=0}^{L}(\phi_l(x_p)-\phi_l(x_q))^2 = (\phi(x_p)-\phi(x_q))^2 \\ &= (\phi(x_p)-\phi(x_q))^T(\phi(x_p)-\phi(x_q)) \\ &= \phi(x_p)^T\phi(x_p) - \phi(x_p)^T\phi(x_q) - \phi(x_q)^T\phi(x_p) + \phi(x_q)^T\phi(x_p) \\ &= k(x_p,x_q) - 2k(x_p,x_q) + k(x_q,x_q) \end{aligned} \tag{4.40}$$

NN_{il}^j in Eq. (4.39) then can be determined in terms of the original dimensionality in $\mathcal{X}$ using Eq. (4.40). Using the latter, we now able to define the weighting function ω^{Φ} in terms of the kernel matrix K as:

$$\omega^{\Phi}(i,j,l) = \frac{\min\left\{(K_{pp}-2K_{pq}+K_{qq})^{\alpha}, (K_{pp}-2K_{pr}+K_{rr})^{\alpha}\right\}}{(K_{pp}-2K_{pq}+K_{qq})^{\alpha} + (K_{pp}-2K_{pr}+K_{rr})^{\alpha}}, \tag{4.41}$$

where p, q, and r are indices to data samples in input space $\mathcal{X}$ corresponding to $\phi(x_l^i)$, $NN_{il}^i[k]$, and $NN_{il}^j[k]$ in Eq. (4.39), respectively. α is a parameter, varies from 0 to ∞, that controls how fast ω^{Φ} decays until vanishing relative to the distance ratio.

The Within-class Scatter Matrix S_W^{Φ} in $\mathcal{F}$

Since $\sum_{i=1}^{N}\phi(x_i) - \mu^{\Phi} = \phi(X) - \mu^{\Phi}1$, the within-class scatter matrix S_W^{Φ} of the mapped data in Eq. (4.29) can be rewritten as:

$$\begin{aligned} S_W^{\Phi} &= \frac{1}{N}\sum_{i=1}^{C}(\phi(X_i)-\mu_i^{\Phi}1)(\phi(X_i)-\mu_i^{\Phi}1)^T, \\ &= \frac{1}{N}\sum_{i=1}^{C}\phi(X_i)\phi(X_i)^T-\phi(X_i)1^T\mu_i^{\Phi T}-\mu_i^{\Phi}1\phi(X_i)^T+\mu_i^{\Phi}11^T\mu_i^{\Phi T} \end{aligned} \tag{4.42}$$

where 1 is a ones vector with N_i entries. Since,

$$\mu_i^{\Phi}=\frac{1}{N_i}\sum_{j=1}^{N_i}\phi(x_j)=\frac{1}{N_i}\phi(X_i)1 \Rightarrow \mu_i^{\Phi T}=\frac{1}{N_i}1^T\phi(X_i)^T, \tag{4.43}$$

S_W^{Φ} in Eq. (4.42) becomes:

$$\begin{aligned} S_W^{\Phi} &= \frac{1}{N}\sum_{i=1}^{C}\phi(X_i)\phi(X_i)^T-\frac{1}{N_i}\phi(X_i)1\phi(X_i)^T \\ &= \frac{1}{N}\sum_{i=1}^{C}\phi(X_i)(I-1_{\frac{1}{N_i}})\phi(X_i)^T \end{aligned} \tag{4.44}$$

where I is an N_i-squared identity matrix and $1_{\frac{1}{N_i}}$ is an N_i-squared matrix with all entries equal $1/N_i$. Recalling the definition of A in Eq. (4.27), we obtain:

$$\begin{aligned} A &= \frac{1}{N}\sum_{i=1}^{C}\phi(X)^T\phi(X_i)(I-1_{\frac{1}{N_i}})\phi(X_i)^T\phi(X) \\ &= \frac{1}{N}\sum_{i=1}^{C}K_i(I-1_{\frac{1}{N_i}})K_i^T \end{aligned} \tag{4.45}$$

where $K_i \subset K$ is the $N \times N_i$ sub-kernel matrix of class i. Note that this expression is similar to the one in [66] as the S_W matrix of NDA in [37] has the same form as the one of FDA in [36].

4.2.2 Implementation and Experimental Results

To demonstrate and evaluate the robustness of the KNPDA approach for data classification in general and human motion recognition in particular, we ran several experiments on publicly available benchmark datasets. As a kernel function, we have used the Gaussian kernel,

with different parameters for different tests. We have also provided a detailed comparison between our approach and several state-of-the-art methods reported in [66, 102, 104]. Cross-validation method was used for refining the parameter values in each experiment.

Implementation Requirements

To have the ability to implement this approach, five main formulas need to be considered which are Eq. (4.26), Eq. (4.37), Eq. (4.39), Eq. (4.40), and Eq. (4.45), in addition to Eq. (4.33) for projecting new patterns. Let's consider that

$$X = \{X_1 = \{x_1, ..., x_{N_1}\}, ..., X_C = \{x_1, ..., x_{N_C}\}\}$$

is an $(L \times N)$ matrix representing a super training set in the input space $\mathcal{X}$, where L represents the dimensionality of $\mathcal{X}$ (or sample length), N is the total number of samples, C is the number of classes, and each X_i is an $(L \times N_i)$ matrix representing the training set of class i.

Prior to applying the algorithm, a number of components need to be computed as prerequisites. These are the super kernel matrix $K(N \times N)$, the kernel matrix of each class $K_i(N \times N_i) \forall_{i \in C}$, and k-nearest neighbor matrices $NN_{il}^{j} \forall_{(i,j) \in C, l \in N_i}$, and their corresponding kernel matrices $NNK_{il}^{j}(N \times k)$. Although $NN_{il}^{j}(L^{\Phi} \times k)$ matrices used in Eq. (4.39) are represented in $\mathcal{F}$, but Eq. (4.40), subsequently, relies only on their corresponding original data in $\mathcal{X}$. Therefore, we need these matrices to be defined in terms of input data in $\mathcal{X}$ and determined based on Euclidean distances between the mapped data in $\mathcal{F}$ using the kernel trick. Finding an $NN_{il}^{j}(L \times k)$ based on Φ is given by Eq. (4.48).

$$K = k(X, X), \text{ where } K_{ij} = k(x_i, x_j), \tag{4.46}$$

$$K_i \subset K, \quad K_i = k(X, X_i) = \left\{ K\left[1 + \sum_{j=1}^{i-1} N_j\right], ..., K\left[\sum_{j=1}^{i} N_j\right] \right\}, \tag{4.47}$$

$$NN_{il}^{j} = \left\{ x_p^j, k\_\min d_{\Phi}(x_l^i, x_p^j) \; \forall_{p \in N_j}, (i,l) \neq (j,p) \right\} \forall_{(i,j) \in C} \tag{4.48}$$

$$NNK_{il}^{j} = k(X, NN_{il}^{j}) \tag{4.49}$$

Benchmark Classification Datasets

To demonstrate and evaluate the performance of our KNPDA model, we have compared the performance of this method to the results obtained in [66], [102], and [104]. In particular, we have tested our approach with 13 standard benchmark datasets, available from different repositories. Table 4.1 gives details about each of the 13 datasets which include the name, number of attributes L (or $\mathcal{X}$'s dimensionality), total number of instances (or samples), number of categories C (or classes), and source where the dataset is available. The three sources are IDA Benchmark Repository, UCI Machine Learning Repository, and Max Planck Institut Informatik (MPII) [53].

These datasets have been selected for performance evaluation and fair comparison with three references. Each of "breast-cancer", "heart", "image", "ringnorm", "thyroid", and "twonorm" were used in [66], "eth-80", "monk1", "monk2", "monk3", "pima", and "ionosphere" were used in [102], and "ionosphere" and "liver-disorders" were used in [104].

With respect to the main objective of this study which is enhancing human motion recognition, we apply the proposed KNPDA algorithm on each of the two benchmark motions datasets, KTH and Weizmann. Data samples are prepared using the MII template presented in section 3.1.

Note that two kernel-based non-parametric discriminant analysis methods have been recently proposed in the literature, [102, 104]. The model in [102] computes the nearest neighbors for each data sample based on the Euclidean distance amongst original data samples X in input space $\mathcal{X}$ instead of the mapped data samples $\phi(X)$ in the feature space $\mathcal{F}$. Furthermore, the proposed method does not define the weighting function in terms of inner dot-product, which represents a cornerstone in the NDA for capturing the classification boundary structure between mapped classes. The other model in [104] was built based

Dataset	Attributes	Instances	Classes	Source
breast-cancer	9	277	2	IDA
heart	13	270	2	UCI
image	19	2310	2	IDA
ringnorm	20	7400	2	IDA
thyroid	5	215	2	IDA
twonorm	20	7400	2	IDA
eth-80	625	3280	8	MPII
monk1	6	432	2	UCI
monk2	6	432	2	UCI
monk3	6	432	2	UCI
pima	8	768	2	UCI
ionosphere	34	351	2	UCI
liver-disorders	6	345	2	UCI

Table 4.1: Description of 13 benchmark datasets.

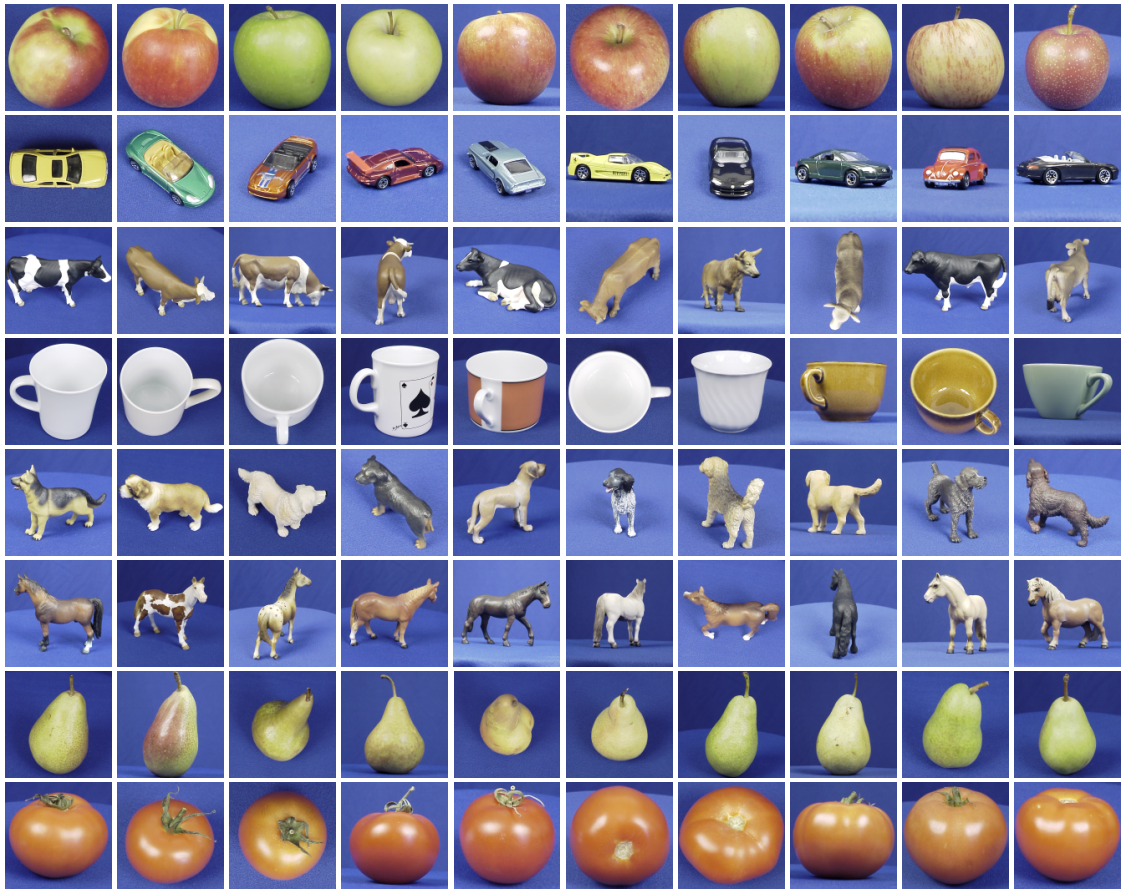
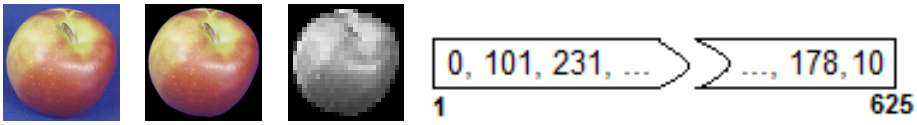


Figure 4.5: Illustrative examples of the 8 classes of the ETH-80 database [53].

on [13] where the scatter matrices, S_B and S_W, are defined non-parametrically based on *extra-class* NNs and *intra-class* NNs, respectively. In computing the intra-class and extra-class matrices, the algorithm addresses only one NN for each data sample instead of the mean vector of its k-nearest neighbors. Finding an NN in computing the intra-class, in particular, is done based on the *most distant* sample. For the above mentioned reasons, our proposed model turns out to be superior to these two models as it is shown in the results obtained in the tests.

Most of those datasets are, to some extent, ready for usage except "eth-80" as it consists of 3,280 (128×128) color images yielding an input space of 16,384 dimensions $\mathcal{X}(\mathbb{R}^{16384})$.



(a) (b) (c) (d)

Figure 4.6: Converting ETH-80's instances from color images to value vectors. (a) original 128×128 image; (b) object image; (c) grayscale 25×25 image; (d) value vector.

This dataset has been specifically built for the purpose of object recognition and classification. It consists of 80 objects divided into 8 classes: *apples*, *cars*, *cows*, *cups*, *dogs*, *horses*, *pears* and *tomatoes* (see Fig. 4.5). Each object is represented by 41 images each of which is captured from different view.

In order for "eth-80" to be ready for usage with reasonable dimensionality, we scaled down all the images to 25×25 pixels and converted their color maps from RGB to grayscale based on the three components, R, G, and B, using the following well known equation:

$$Gray(RGB) = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B.$$

By choosing 25×25, the original aspect ratio is preserved and each sample can be represented as a vector of 625 values yielding a new input space $\mathcal{X}(\mathbb{R}^{625})$. For more information about any of these datasets, please refer to their sources.

Experimental Results

In this study we used the most employed statistical technique for evaluating and comparing learning algorithms called k-fold cross-validation [78, 102]. It divides available data samples into k equally, or approximately equally, subsets. The process of training and validation is then performed k times. During each iteration, different combination of training and testing subsets of samples is used, such that a single subset is held-out for testing while the other $k-1$ subsets are combined together and used for training. It guarantees that every subset is used once in the validation process whereas it is used 9 times, combined

	Mika et al. in [66]					Our approach	
Dataset	RBF	AB	AB_R	SVM	KDA	KNPDA	δ
breast-cancer	72.4	69.6	73.5	74.0	74.2	**81.29**	$\uparrow$7.09
heart	82.4	79.7	83.5	84.0	83.9	**87.54**	$\uparrow$3.54
image	96.7	**97.3**	**97.3**	97.0	95.2	96.80	$\downarrow$0.50
ringnorm	98.3	98.1	98.4	98.3	98.5	**98.54**	$\uparrow$0.04
thyroid	95.5	95.6	95.4	95.2	95.8	**99.15**	$\uparrow$3.35
twonorm	97.1	97.0	97.3	97.0	97.4	**97.72**	$\uparrow$0.32

Table 4.2: Recognition Results of our KNPDA approach compared to 5 other methods reported in [66].

with others, in the training process [78]. We have used a *stratified* 10-fold cross-validation to evaluate the proposed approach. Performing the evaluation this way is highly recommended and is considered to be the best model selection method, as it tends to provide a non-biased estimation of the accuracy [47, 78]. We rearranged each dataset based on its categories prior to being divided into 10 subsets to ensure each subset is a good representative of the whole dataset. During each iteration, parameters' values are tuned and those yielding the highest recognition rates are kept.

A commonly used Gaussian kernel function Eq. (4.50) with a single parameter σ is used in this study, although the proposed method is applicable to any other valid kernel function.

$$k(\mathbf{x},\mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}) \tag{4.50}$$

Four parameters are involved in our algorithm: σ in Eq. (4.50) where $\sigma > 0$, k in Eq. (4.48) where $0 < k < \min(N_i)\forall_{i\in C}$, α in Eq. (4.41) where $\alpha \geq 0$, and $\acute{L}$ in Eq. (4.33) where $\sum_{i=1}^{\acute{L}} \lambda_i \leq 0.9 \sum_{i=1}^{N} \lambda_i$.

The average recognition rate $\bar{r}$ for each dataset which is given by:

	You et al. in [102]									Our approach	
Dataset	KSDA	KDA	KNDA	MoG	KSVM	KPCA	PCA	FDA	HFDA	KNPDA	δ
ETH-80	83.5	83.5	76.2	69.2	81.8	65.3	67.1	65.3	68.4	**86.25**	↑2.75
monks1	90.2	89.7	78.2	80.3	83.6	90.3	81.3	69.0	84.2	**100.0**	↑9.70
monks2	86.1	82.9	85.0	75.9	82.6	68.3	67.1	70.6	83.6	**97.93**	↑11.83
monks3	96.3	93.5	85.4	89.4	93.5	94.4	89.7	70.8	93.8	**100.0**	↑3.70
pima	80.4	78.6	72.0	75.0	79.2	64.3	70.2	64.9	77.4	**81.52**	↑1.12
ionosphere	96.7	94.4	90.1	82.1	96.0	89.4	92.1	84.8	94.0	**98.00**	↑1.30

Table 4.3: Recognition Results of our KNPDA approach compared to 9 other methods reported in [102].

	Zhan et al. in [104]		Our approach	
Dataset	KFDA	KNDA	KNPDA	δ
ionosphere	90.06	92.57	**98.00**	↑5.43
liver-disorders	63.71	66.51	**73.30**	↑6.79

Table 4.4: Recognition Results of our KNPDA approach compared to 2 other methods reported in [104].

$$\bar{r} = \frac{\sum_1^{10} r}{10}, \quad with \quad r = \frac{N_{corr}}{N_{ts}}, \tag{4.51}$$

where N_{corr} is the number of test samples that are successfully recognized and N_{ts} is the total number of testing samples (or 10% of the dataset).

Recognition results obtained with this method clearly reflect a high level of performance compared to other state-of-the-art methods.

The significance of the achieved improvement can be evaluated by comparing it with its peers reported in the three references [66], [102], and [104], and shown in tables 4.2, 4.3, and 4.4, respectively.

The results given in Table 4.2 show mostly significant improvement over each of the RBF classifier (RBF), AdaBoost (AB), regularized AdaBoost (AB_R), SVM, and KDA proposed in [66]. Table 4.3 clearly shows superiority of our method over each of kernel subclass discriminant analysis proposed in [102](KSDA). These are KDA, kernel nonparametric discriminant analysis proposed in [102] (KNDA), mixture of Gaussians with maximal likelihood (MoG), KSVM, KPCA, PCA, FDA, and heteroscedastic FDA with Chernoff distance (HFDA). All reported in [102]. Table 4.4 shows how our model achieves valuable recognition accuracies compared to a KNDA proposed in [104]. The combination of (average enhancement, maximum enhancement) achieved by Mika et al. in [66] is (-0.27, 0.2) whereas ours is (2.3, 7.09). You et al. in [102] achieved (0.8, 1.9) whereas ours is (5.07, 11.83). Zhan et al. in [104] achieved (2.65, 2.8) whereas ours is (6.11, 6.79). It is clear that our algorithm gives really significant improvement in the stream of data classification. The model in [102] determines the nearest neighbors based on the original input space whereas it is working on the mapped data in the feature space. Adjacent data samples in input space are not necessarily to be adjacent in the feature space because this is based on what kernel function is being used. This may result in misleading the method's performance.

Note that choosing the number of nearest neighbors, k, can influence the efficiency of the KNPDA in some how. From Eq.(4.30), we observed that selecting large values for

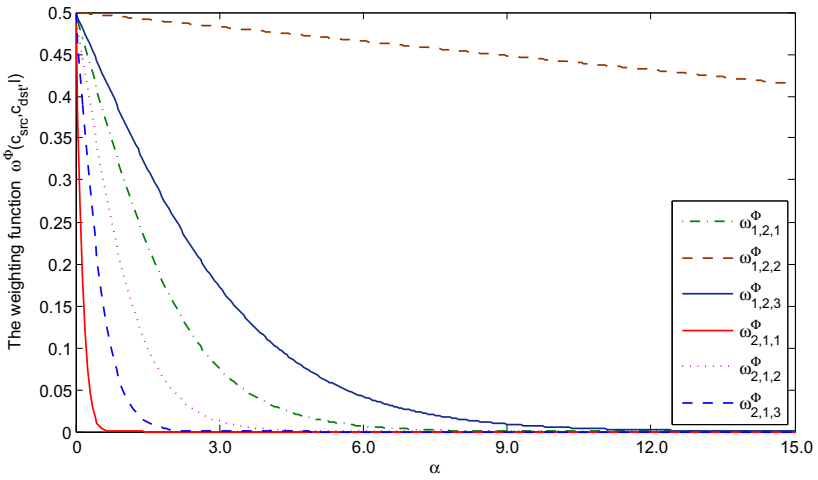


Figure 4.7: The effect of α on the kernel-based weighting function ω^{Φ} for six training samples in "thyroid" dataset.

k such as $k = N_j$, which in fact represents the maximum value for k, and substitute the weighting function with one, $m_j^{\Phi}(\phi(x_l^i))$ becomes μ_j^{Φ}, which represents the mean vector of the j^{th} class in the feature space $\mathcal{F}$. This means that the KNPDA would perform parametrically and is essentially a generalization of the KFDA. Hence, we should avoid choosing high values for k, otherwise, the KNPDA may lose its non-parametric advantage. In contrast, selecting $k = 1$ means taking into account a very small amount of training data, which may cause performance weakness. In our experiments, we chose k between 1 and 10% of the training class with the minimum number of samples. The best classification accuracies in nine of the thirteen datasets reported in tables 4.2, 4.3, and 4.4 were obtained using $k = 3$. Each of the other four datasets: "breast-cancer", "heart", "eth-80", and "pima" performs well using $k = 6$, 5, 1, and 2, respectively. This represents a practical prove that by using $k = 3$, the method performs better. This might be the reason behind the valuable improvement of our method over the one in [104] which uses only one farthest neighbor.

As mentioned above, the parameter α controls how rapidly the kernel-based weighting

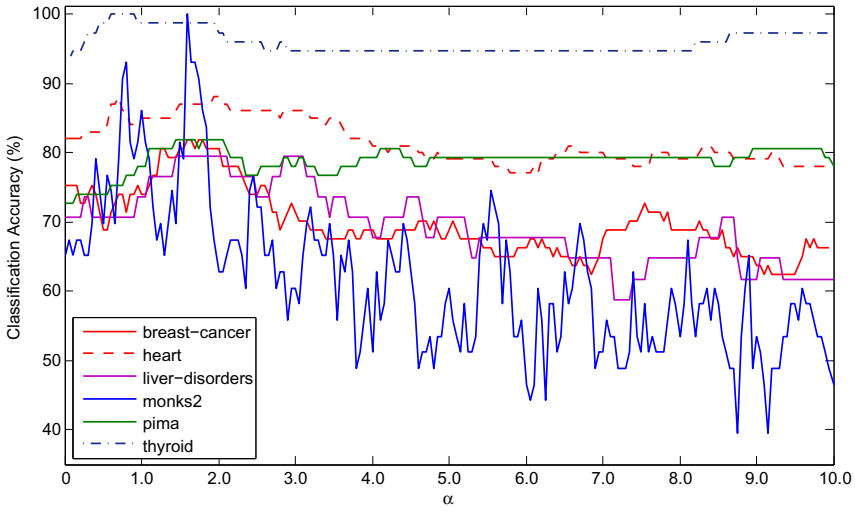


Figure 4.8: The effect of α on the classification accuracies of some datasets using the KNPDA.

function ω falls to zero with respect to the distance ratio. Figure 4.7 illustrates the influence of α on ω^{Φ}, defined in Eq. (4.41), for the first three training observations of each of the two classes in the case of *thyroid* dataset. Each sample responses differently to the change of α compared to the others. It is clear that, amongst these six samples, the nearest one to the decision boundary is sample 2 from class 1 (x_2^1) with a maximum contribution ($\omega^{\Phi}_{1,2,2}$), whereas the most distant one is sample 1 from class 2 (x_1^2) with a minimum contribution ($\omega^{\Phi}_{2,1,1}$). The change in the value of α influences the final discrimination power of this algorithm through ω^{Φ}. Figure 4.8 shows the effect of α on the classification accuracies of certain examples from some datasets using the KNPDA.

In the context of human motion classification, the method of 10-fold cross-validation is employed for model evaluation in the case of KTH, but for Weizmann, we used 9-fold cross-validation because 7 motion types out of 10 have 9 video clips each. Thus, using 10 folds in this case leads to have inaccurate model evaluation as some classes or motion types are missing their representatives in certain folds.

input→ output↓	***box***	***clap***	***wave***	***walk***	***jog***	***run***
box	100	0	3.57	1.78	0	3.57
clap	0	100	5.36	0	0	0
wave	0	0	91.07	0	0	0
walk	0	0	0	92.87	1.78	0
jog	0	0	0	1.78	96.44	3.57
run	0	0	0	3.57	1.78	92.86

Table 4.5: The confusion matrix of applying the KNPDA on the KTH motions dataset. The mean accuracy rate is 95.54%.

input→ output↓	***bend***	***wave1***	***wave2***	***jack***	***jump***	***pjump***	***sjump***	***walk***	***skip***	***run***
bend	100	0	0	0	0	0	0	0	0	0
wave1	0	100	0	0	0	0	0	0	0	0
wave2	0	0	100	0	0	0	0	0	0	0
jack	0	0	0	100	0	0	0	0	0	0
jump	0	0	0	0	100	0	0	0	0	0
pjump	0	0	0	0	0	100	0	0	0	0
sjump	0	0	0	0	0	0	100	0	0	0
walk	0	0	0	0	0	0	0	100	0	0
skip	0	0	0	0	0	0	0	0	100	0
run	0	0	0	0	0	0	0	0	0	100

Table 4.6: The confusion matrix of applying the KNPDA on the Weizmann motions dataset. The mean accuracy rate is 100%.

Tables 4.5 and 4.6 show the confusion matrices of the output recognition using each of KTH and Weizmann datasets respectively. The obtained results in Table 4.5 show that this algorithm accurately recognizes 95.54% motions with about 1.34% enhancement over the FDA and about 8.04% over the PCA that are reported in Table 3.1. In spite of little confusions between some motions, this confusion matrix reflects more stability with less variance or standard deviation compared to the PCA's and FDA's. The KNPDA works perfectly on the Weizmann dataset giving a mean recognition accuracy of 100% with no confusion at all. The reported results are obtained by huge reduction in data dimensionality using the KNPDA algorithm. The dimensionality of the KTH samples were reduced about 98.28% and 99.83% w.r.t the MII and video frame sizes respectively. The corresponding reduction rates w.r.t Weizmann samples are 95.5% and 99.67%.

Conclusions

This chapter summarizes and draws conclusions from the results of the research reported in this thesis.

We have proposed a novel and efficient appearance-based framework based on eigenspace for human motion recognition. The appearance-based MII model is simple, expressive, and effective. We align the human silhouette of each binary image to a reference point and then form a single intensity image of these silhouettes by taking into account the difference between each subsequent silhouettes. It has been shown that, by effective classification of such forms, using PCA or FDA for example, reliable human motion recognition is achieved. The experiments show that the MII, with the help of silhouette centering, encapsulates motion information in a smallest possible area and gives useful pieces of information about where motion is concentrated and what limbs are involved. Hence, the recognition process is very fast and efficient. The robustness of this human motion template to noise, temporal occlusion, and imperfect silhouettes is shown through the high recognition rates obtained. Two simple-structured eigenspaces are generated and the data dimensionality is drastically reduced using each of PCA and FDA. When compared the two peers, the FDA-based model outperforms the PCA-based one with about 6.7%, while preserving maximum class separability. The effectiveness of our approach has been demonstrated over one of the state-of-the-art datasets in motion recognition literature, which is the KTH dataset. The recognition rates of this method are directly comparable and even competitive to the results presented over this dataset. This chapter proposed also an effi-

cient appearance based method using a compound eigenspace for directed human motion recognition. Thanks to the combination of relative speed s and displacement vector θ, the motion discrimination was significantly improved. This compound model has shown clear improvement over using a simple eigenspace. It has been also shown that this model is more robust against underfitting compared to the aforementioned one. Moreover, the experimental results have demonstrated that the compound eigenspace is efficient in discriminating between single-location and multi-location human motions. The compound eigenspace is generated and the data dimensionality is reduced using the PCA technique, while preserving maximum data variance and getting rid of similar features that have no discrimination value.

An important characteristic of the proposed eigenvector-based recognition system is the fact that it can be deployed in real time applications. The only shortcoming of the MII approach is its dependence over silhouette centroid accuracy, which is computed based on the distribution of silhouette pixels. It was observed that most of the confusion occurs mainly because of two reasons, the similarity between motions such as *fast-jogging* and *slow-running* and the imperfect extracted silhouettes.

A kernel-based approach has been described for feature selection in multi dimensional classification for human motion recognition. The algorithm estimates the decision axes matrix in the non-linear feature space $\mathcal{F}$, based on the best kernel features of the covariance matrix preserving as much as possible of the variation in the original data distribution. We introduced a methodology based on the kernel technique that can deal with non-linear separable data by mapping it into a higher dimensional space using a non-linear function. It has been shown how critical choosing the most appropriate kernel function is. Although, most of the kernel-based approaches in the literature use RBF kernels, we have shown that in the application of human motion recognition based on the combined KTH-Weizman dataset, Sigmoid and Log kernels have shown superiority over the others including the RBFs represented by Gaussian, Exponential, and Laplacian.

The conventional FDA is a common used method for class discrimination and dimen-

sionality reduction. However, it encounters difficulties in dealing with the non-Gaussian aspects and with the non-linear separation of data spreads due to its normality assumption and its linear behavior. We have proposed a novel kernel-based non-parametric approach, called KNPDA, for feature selection in multi dimensional classification problems. Each feature has its corresponding scalar (eigenvalue) that can be used as a relative indicator of separability. The non-parametric version of FDA, called NDA by Fukunaga, is adopted to relax the normality assumption. To overcome the restriction of linearity, a kernel-based nonlinear formulation of the NDA's main criterion J in general, and of its between scatter matrix in particular, has been derived. The kernel-based non-parametric forms of the scatter matrices provide advantages over both linear and kernel-based parametric versions used in discriminant analysis. The KNPDA maps the original data spreads into a Hilbert space, implicitly using the *kernel trick*, where they can be separated with a hyperplane. The resulting kernel-based mapping provides many advantages. It gives the ability to separate data distributions nonlinearly, to control the degree of dimensionality reduction freely, and to preserve classification structure. The experimental results obtained when testing 12 datasets have clearly shown the advantages of our proposed approach and its superiority over other state-of-the-art methods. Furthermore, human motion recognition has been clearly enhanced by employing the KNPDA when testing two well known benchmark motions datasets, KTH and Weizmann.

Future work will be dedicated to improving the discriminatory power and robustness of this non-parametric kernel-based human motion recognition framework to target more complicated motions and multi-actor motions.

References

[1] M. Abbasian, H. S. Yazdi, and A. V. Mazloom. Kernel machine based fourier series. *International Journal of Advanced Science and Technology*, 33:13–22, 2011.

[2] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):44–58, jan. 2006.

[3] A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[4] Shotaro Akaho. A kernel method for canonical correlation analysis. *CoRR*, abs/cs/0609071, 2006.

[5] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(2):288–303, feb. 2010.

[6] M. Alizadeh and M. Ebadzadeh. Kernel evolution for support vector classification. In *Evolving and Adaptive Intelligent Systems (EAIS), 2011 IEEE Workshop on*, pages 93–99, 2011.

[7] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014 –1021, june 2009.

[8] K. Bashir, T. Xiang, and S. Gong. Gait representation using flow fields. In *British Machine Vision Conference*, 2009.

[9] J. Blackburn and E. Ribeiro. Human motion recognition using isomap and dynamic time warping. In *Human Motion Understanding, Modeling, Capture and Animation*, volume 4814, pages 285–298, Rio de Janeiro, Brazil, October 2007. Springer Berlin / Heidelberg.

[10] A. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, 352:1257–1265, 1997.

[11] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, March 2001.

[12] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[13] M. Bressan and J. Vitria. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749, 2003.

[14] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1932 –1939, 2009.

[15] F. Chelali, A. Djeradi, and R. Djeradi. Linear discriminant analysis for face recognition. In *Multimedia Computing and Systems, 2009. ICMCS '09. International Conference on*, pages 1 –10, april 2009.

[16] P. Chen, C. Lin, and B. Schölkopf. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, 2005.

[17] Y. Chen, J. Bi, and J. Wang. Miles: Multiple-instance learning via embedded instance selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):1931–1947, dec. 2006.

[18] S-C. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In S. Panchanathan and B. Vasudev, editors, *Visual Communications and Image Processing*, volume 5308, pages 881–892. SPIE, jan 2004.

[19] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, September 1995.

[20] R. Cucchiara, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on PAMI*, 25:1337–1342, 2003.

[21] H. R. David. *Semantic Models for Machine Learning*. PhD thesis, University of Southampton, feb 2006.

[22] J. Davis and A. Bobick. The representation and recognition of human movement using temporal templates. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, CVPR '97, pages 928–934, Washington, DC, USA, jun 1997. IEEE Computer Society.

[23] A. Diaf and R. Benlamri. An effective view-based motion representation for human motion recognition. In *Modeling and Implementing Complex Systems, International Symposium on*, pages 57–64, May 2010.

[24] A. Diaf, R. Benlamri, and B. Boufama. Nonlinear-based human activity recognition using the kernel technique. In Rachid Benlamri, editor, *Networked Digital Technologies*, volume 294 of *Communications in Computer and Information Science*, pages 342–355. Springer Berlin Heidelberg, 2012.

[25] A. Diaf, B. Boufama, and R. Benlamri. A compound eigenspace for recognizing directed human activities. In Aurlio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, volume 7325 of *Lecture Notes in Computer Science*, pages 122–129. Springer Berlin / Heidelberg, 2012.

[26] A. Diaf, B. Boufama, and R. Benlamri. Non-parametric fisher's discriminant analysis with kernels for data classification. *Pattern Recognition Letters*, (In press):–, 2012.

[27] A. Diaf, R. Ksantini, B. Boufama, and R. Benlamri. A novel human motion recognition method based on eigenspace. In *Image Analysis and Recognition, ICIAR-2010*, volume 6111, pages 167–175. Springer, 2010.

[28] F. Dornaika and F. Davoine. On appearance based face and facial action tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(9):1107–1124, September 2006.

[29] S. Eftakhar, J. Tan, H. Kim, and S. Ishikawa. An efficient approach to human motion recognition employing large motion-database structure. *SICE Annual Conference, 2008*, pages 2239–2243, August 2008.

[30] S. Eftakhar, J. Tan, H. Kim, and S. Ishikawa. Robust human motion recognition employing adaptive database structure. *ICROS-SICE International Joint Conference, 2009*, August 2009.

[31] S.M.A. Eftakhar, Joo Kooi Tan, Hyongseop Kim, and S. Ishikawa. Viewpoint-oriented human activity recognition in a cluttered outdoor environment. In *SICE Annual Conference 2010, Proceedings of*, pages 1506 –1511, 2010.

[32] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *the 6^{th} ECCV - Part II*, pages 751–767. Springer-Verlag, 2000.

[33] M. Elmezain, A. Al-Hamadi, O. Rashid, and B. Michaelis. Posture and gesture recognition for human-computer interaction. pages 415–440, 2009.

[34] A. Farhadi and M. Tabrizi. Learning to recognize activities from the wrong view point. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Proceedings of the 10th European Conference on Computer Vision: Part I*, volume 5302, pages 154–166. Springer-Verlag, 2008.

[35] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, 2005.

[36] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugenics*, 7:179–188, 1936.

[37] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[38] K. Fukunaga and J. Mantock. Nonparametric discriminant analysis. *PAMI*, 5(6):671–678, November 1983.

[39] Keinosuke Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[40] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2247–2253, dec. 2007.

[41] D. Hilbert. *Grundzüge einer allgemeinen Theorie der linearen Integralgleichungen.* Teubner, 1912.

[42] T. Howley and M. Madden. The genetic kernel support vector machine: Description and evaluation. *Artificial Intelligence Review*, 24(3-4):379–395, 2005.

[43] Z. Htike, S. Egerton, and Y. Kuang. Model-based viewpoint invariant human activity recognition from uncalibrated monocular video sequence. In J. Li, editor, *AI 2010: Advances in Artificial Intelligence*, volume 6464 of *Lecture Notes in Computer Science*, pages 142–152. Springer Berlin Heidelberg, 2011.

[44] H. Jiang, M. Drew, and Z. Li. Successive convex matching for action detection. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1646–1653, 2006.

[45] P.-M. Jodoin, V. Saligrama, and J. Konrad. Behavior subtraction. *Image Processing, IEEE Transactions on*, 21(9):4244–4255, sept. 2012.

[46] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[47] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[48] R. Ksantini and B. Boufama. Combining partially global and local characteristics for improved classification. *International Journal of Machine Learning and Cybernetics*, 3:119–131, 2012.

[49] R. Ksantini, B. Boufama, and I. Ahmad. A new ksvm+kfd model for improved classification and face recognition. *Journal of Multimedia*, 6(1):39 – 47, Feb 2011.

[50] B. Kuo and D. Landgrebe. Nonparametric weighted feature extraction for classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 42(5):1096–1105, may 2004.

[51] B. Kuo, T. Sheu, C. Li, and C. Hung. Hyperspectral image classification using knwfe with conformal transformation for kernel selection. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, pages 3789–3793, july 2007.

[52] T.M. Lehmann, C. Gonner, and K. Spitzer. Survey: interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18(11):1049–1075, 1999.

[53] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages 409–915, 2003.

[54] Z. Li, D. Lin, and X. Tang. Nonparametric discriminant analysis for face recognition. *PAMI*, 31(4):755 – 761, April 2009.

[55] T. Lin, S. Chen, T. Truong, C. Chen, and C. Lin. Still image compression using cubic spline interpolation with bit-plane compensation. volume 6696. SPIE, sep 2007.

[56] L. Liu, W. Jia, and Y. Zhu. Gait recognition using hough transform and principal component analysis. In *Proceedings of the 5th international conference on Emerging intelligent computing technology and applications*, ICIC'09, pages 363–370, Berlin, Heidelberg, 2009. Springer-Verlag.

[57] S. liu, J. Liu, T. Zhang, and H. Lu. Human action recognition in videos using motion impression image. In *Proceedings of the First International Conference on Internet*

Multimedia Computing and Service, ICIMCS '09, pages 174–178, New York, NY, USA, 2009. ACM.

[58] Z. Liu and R. Laganiere. Registration of ir and eo video sequences based on frame difference. In *the 4th Canadian Conference on Computer and Robot Vision*, pages 459–464, 2007.

[59] M. Loog, R. Duin, and R. Haeb-Umbach. Multiclass linear dimension reduction by weighted pairwise fisher criteria. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(7):762 – 766, 2001.

[60] C. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*, chapter 15, pages 293–319. Cambridge University Press, New York, NY, USA, july 2008.

[61] A. Martinez and A. Kak. Pca versus lda. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(2):228–233, feb 2001.

[62] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *Geosci. Remote Sens., IEEE Transactions on*, 42(8):1778 – 1790, August 2004.

[63] H. Meng, M. Freeman, N. Pears, and C. Bailey. Real-time human action recognition on an embedded, reconfigurable video processing architecture. *Journal of Real-Time Image Processing*, 3(3):163–176, 2008.

[64] H. Meng and N. Pears. Descriptive temporal template features for visual motion recognition. *Pattern Recognition Letters*, 30(12):1049–1058, September 2009.

[65] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Royal Soc. (A)*, 83(559):69–70, November 1909.

[66] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müller. *Fisher Discriminant Analysis with Kernels*, volume IX, pages 41–48. IEEE, 1999.

[67] T. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, November 2006.

[68] G. Mori, Xiaofeng R., A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages 326–333, july 2004.

[69] H. Murase and R. Sakai. Moving object recognition in eigenspace representation: gait analysis and lip reading. *Pattern Recognition Letters*, 17(2):155–162, 1996.

[70] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[71] J.C. Niebles and Li Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, june 2007.

[72] N. Noorit, N. Suvonvorn, and M. Karnchanadecha. Model-based human action recognition. In K. Jusoff and Y. Xie, editors, *Proceedings of the 2nd International Conference on Digital Image Processing*, volume 7546, pages 75460P–75460P–6. SPIE, feb 2010.

[73] T. Ogata, J. Tan, and S. Ishikawa. High-speed human motion recognition based on a motion history image and an eigenspace. *IEICE - Transactions on Information and Systems*, E89-D(1):281–289, 2006.

[74] D.H. Parks and S.S. Fels. Evaluation of background subtraction algorithms with post-processing. In *Advanced Video and Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on*, pages 192–199, sept. 2008.

[75] Y. Prasad and K.K. Biswas. Fuzzy rough based regularization in generalized multiple kernel learning. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 879 –883, may 2012.

[76] A. Psarrou, S. Gong, and M. Walter. Recognition of human gestures and behavior based on motion trajectories. *Image Vision Computing*, 20:349–358, February 2002.

[77] M. Rahman and S. Ishikawa. Human motion recognition using an eigenspace. *Pattern Recognition Letters*, 26:687–697, 2005.

[78] P. Refaeilzadeh, L. Tang, and H. Liu. Cross-validation. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 532–538. Springer US, 2009.

[79] N. Robertson and I. Reid. A general method for human activity recognition in video. *Computer Vision and Image Understanding*, 104(2):232–248, November 2006.

[80] R. Sadykhov and I. Frolov. The development features of the face recognition system. In *IMCSIT*, pages 121–128, 2010.

[81] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43 – 49, 1978.

[82] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10:1299–1319, July 1998.

[83] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 3:32–36, August 2004.

[84] Martin Sewell. Kernel methods. Technical report, Department of CS, University College London, London, UK, July 2009.

[85] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[86] A. Sierra. High-order fisher's discriminant analysis. *Pattern Recognition*, 35(6):1291–1302, 2002.

[87] C. Souza. Kernel functions for machine learning applications. Web: http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html, mar 2010.

[88] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):747–757, aug 2000.

[89] L. Sudha and R. Bhavani. Gait based gender identification using statistical pattern classifiers. *International Journal of Computer Applications*, 40(8):30–35, feb. 2012.

[90] D. Sueaseenak, S. Wibirama, T. Chanwimalueang, C. Pintavirooj, and M. Sangworasil. Comparison study of muscular-contraction classification between independent component analysis and artificial neural network. In *Communications and Information Technologies, 2008. ISCIT 2008. International Symposium on*, pages 468–472, oct. 2008.

[91] C. Therrien. *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*. John Wiley & Sons, Inc., New York, NY, USA, 1989.

[92] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[93] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *the 7^{th} IEEE ICCV*, volume 1, pages 255–261, 1999.

[94] K. Tran, L. Kakadiaris, and S. Shah. Modeling motion of body parts for action recognition. In J. Hoey, S. McKenna, and E. Trucco, editors, *Proceedings of the British Machine Vision Conference*, pages 64.1–64.12. BMVA Press, 2011.

[95] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, November 2008.

[96] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[97] G. Valls and L. Bruzzone. Kernel-based methods for hyperspectral image classification. *Geosci. Remote Sens., IEEE Transactions on*, 43(6):1351 – 1362, June 2005.

[98] Y. Wang, Y. Jia, C. Hu, and M. Turk. Face recognition based on kernel radial basis function network. In *Computer Vision (ACCV2004), The sixth Asian Conference on*, pages 174–179. Asian Federation of Computer Vision Societies, 2004.

[99] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249257, November 2006.

[100] S. Wong, T. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, June 2007.

[101] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *the IEEE Computer Society Conference on CVPR*, pages 379–385, 1992.

[102] D. You, O. Hamsici, and A. Martinez. Kernel optimization in discriminant analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):631 – 638, 2011.

[103] H. Yu and J. Yang. A direct lda algorithm for high-dimensional data - with application to face recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.

[104] X. Zhan and B. Ma. Kernel nonparametric discriminant analysis. In *Electrical and Control Engineering, 2011 International Conference on*, pages 4544 – 4547, 2011.

[105] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28 – 31 Vol.2, aug. 2004.

Vita Auctoris

Abdunnaser Diaf was born in 1970 in Tripoli, the capital city of Libya. He holds a Bachelor of Science degree in electronic engineering from the High Institute of Electronic Engineering in 1993 and a Master's of Science degree in computer science from Lakehead University in 2007.