Deep Learning for Motion Recognition

by

Sara Daraei

PhD Candidate

Submitted to the Graduate Faculty of

the Department of Computing and Information Science in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

DEPARTMENT OF COMPUTING AND INFORMATION SCIENCE

This dissertation was presented

by

Sara Daraei

It was defended on

November 5th 2021

and approved by

Dr. Paul Munro, School of Computing and Information Science

Dr. Hassan Karimi, School of Computing and Information Science

Dr. Michael Lewis, School of Computing and Information Science

Dr. Mai Abdelhakim, Department of Electrical and Computer Engineering, Swanson School of Engineering

Dissertation Director: Dr. Paul Munro, School of Computing and Information Science

Deep Learning for Motion Recognition

Sara Daraei, PhD

University of Pittsburgh, 2021

Automatic analysis and interpretation of human motion from visual data has been one of the most significant computer vision challenges since 1970. In recent years, deep learning has fueled the rapid advancement of computer vision topics. In particular, human motion analysis has drawn substantial attention due to its practical importance in many applications in a variety of domain including social behavior studies, medical assistance, robotics, sport analytics, and more.

Human motion is one of the key parts of human social behavior and a rich source of information. We move our whole body involving head, shoulders, hands, trunk, legs, and limbs combined with facial expressions flavored with our individualized style to transmit social signals. A number of studies have suggested the existence of unique motion signatures of individuals by analyzing data obtained from Kinect™ devices, and Electromyography (EMG) electrodes attached to muscles. Meaning that when we move and communicate, we tend to use our characteristic style of motion. These distinct motion patterns are attributed to behavioral and anatomical differences between individuals as well as their different muscle activation strategies.

This research aims at establishing a fully-automated framework to push the envelope of understanding information hidden in human motions from visual inputs and its potential applications on a set of fundamental tasks including classification, identification, and user authentication. For this purpose, we propose a number of deep learning approaches and try to tackle the problem from a data-driven perspective and figure out to what extend we would be able to model human motion signatures and see if it is possible to authenticate or identify people based on their movement pattern. Our results demonstrate an accuracy of 94.04% for human authentication and 92.62% for human identification among 10 subjects confirming that human motion conveys information regarding their identity and can be considered as practical biometric cues. Considering particular applications and their limitations, we further propose a generative biometric model that efficiently learns task-relevant features in data and integrate them into a probabilistic authentication setting based on limited amount of data. The proposed framework is able to authenticate the correct subject 86.11% of times.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AI	Artificial Intelligence
EER	Equal Error Rate
EM	Expectation Maximization
FRR	False Rejection Rate
GAN	Generative Adversarial Networks
GMM	Gaussian Mixture Model
HAR	Human Activity Recognition
HPE	Human Pose Estimation
HTER	Half Total Error Rate
LSTM	Long short-term memory
MAP	Maximum A Posteriori
RNN	Recurrent Neural Networks
R-CNN	Region-based Convolutional Neural Networks
UBM	Universal Background Model

Preface

I want to express my gratitude to all the individuals who contributed to this research, those who kindly supported me, and the ones who inspired me and helped flicker new insights.

Most importantly, I would like to thank my PhD supervisor, Dr. Paul Munro. He guided me through challenges of this research, and always provided me with eye-opening insights and wise comments. I also appreciate other committee members, Dr. Abdelhakim, Dr. Lewis, and Dr. Karimi for insightful discussions and exchange of ideas.

I also thank my parents and my brother, Hossein, whose unconditional love and support always brightened the way, and helped me in the moments of frustration.

1.0 INTRODUCTION

Body motions are one of the key aspects of human social behavior and a rich source of information. We move with our whole body involving head, shoulders, hands, feet, trunk, and limbs combined with facial expressions flavored with our individualized style to transmit social signals [98, 76]. Due to many potential important applications, "Looking at People" is currently one of the most active challenges being explored in parallel by different research communities. In computer vision, automatic human motion analysis from visual data has been one of the most important challenges due to the advancement of video camera technologies and the significance in many domains such as psychology, medical assistance, security, social behavior, robotics, human-computer interactions, virtual reality, sport analytics, and etc. Currently, hundreds of applications are enhanced with existing such technologies while others are in the urgent need of missing pieces of this puzzle.

During the past decade, development of deep learning algorithms and progress in processor technologies, has fueled the rapid advancement of computer vision topics including human motion analysis. Human pose estimation, in particular, caught a huge amount of attention from computer vision communities resulting in growth of this domain and implementation of many algorithms to estimate human body pose from visual data [30, 157]. The temporal evolution in the field of human pose estimation naturally became an attractive option for making progress in many other human motion analysis domains such as human activity recognition and many more. While, problems such as human pose estimation and human activity recognition are examples of *classical* computer vision challenges that caught a huge amount of efforts and attention, a numerous topics in the field of human motion analysis remained partially-explored due to the limited understanding of the patterns hidden in data or immature analytical descriptor. In this research, we aims at establishing a framework to push the envelope of understanding human motion and its potential application as an authentication/identification technique.

In recent years, a number of studies have suggested the existence of distinct and identifiable motion patterns of individuals by analyzing movement data obtained from Kinect devices [115] and Electromyography (EMG) electrodes attached to muscles [67]. From these studies, unique motion patterns can be attributed to anatomical differences and the existence of individual muscle activation strategies or signatures. Behavioural contrasts between individuals is also another reason that distinguishes their movement patterns [171, 46, 111]. These studies highlight the potential of body movements as a biometric method and its usefulness in multi-modal identification systems.

Human motion biometrics has been considered as an authentication/identification technique in a number of research papers relying on the fact that human motion patterns are unique. *Headbanger* is a software device implemented on Google GLASS to authenticate users based on their free-style head movement in response to an external audio stimulus. Furthermore, gait identification, including stride length and arm swing, has attracted significant attention as a non-contact, non-obtrusive method for authentication which is resilient to cyber attacks [162, 156, 142]. In a separate effort, [121] utilize arm swing data captured from cellphone inertial sensors, accelerators and gyroscopes, and explore convolutional temporal models for human authentication. Perhaps the most relevant study to this research is [29] which proposes an identification approach using convolutional neural network based on radar micro-Doppler patterns and achieves average accuracy of 85.6% for 10 people.

In this research, we propose a framework that is able to model human motions from visual input. The proposed framework focuses on the design and implementation of a fully automated human motion analysis network as the video inputs are not annotated and acquiring human pose annotation for large data is not feasible. We approach the problem from a data-driven perspective to figure out to what extent we can model individual's motion patterns and see if it is possible to authenticate or identify people based on their motion signature.

Historically, the difficulty of data collection for biometric research has been a challenge due to practical biasis and legal problems. Therefore, the majority of existing literature are limited to lab-scale data collection which is a poor representation of the real world data due to subjects self consciousness and also limited amount of sample data.

Furthermore, two main key challenges in implementing an automatic human authentication/identification framework are the efficient learning of task-relevant representation of the data and also the incorporation of them in a biometric setting characterized by individuals. These challenges meet limitations due to low computational power as well as limited amount of training samples.

In response to the above-mentioned challenges, we develop a non-intrusive and non-cooperative technique for human verification as well as identification which is based on the in-the-wild footage of subjects. Unlike other studies established in this domain, the proposed framework is able to model human motion only by looking at in-the-wild footage of individuals and does not require any additional devices such as Kinect™, wearable or RADAR sensors. To the best of our knowledge there is no similar work established yet that could model human motion patterns based on visual inputs.

Moreover, in the further steps of this research we intend to address challenges related to low computational power or limited amount of training samples by figuring a universal background model which is cable of learning a general human motion distribution in the motion feature space that could be trained offline before being used in any application.

Apart from the potential application which we are going to briefly discuss in next section, the key contribution of this research is in proposing a temporal architecture that is able to model multiple temporal sequences. This model will be specifically developed for motion biometric recognition while it is applicable to many other problems in computer vision.

1.1 Applications

The framework we propose in this manuscript is a research tool that could be utilized in variety range of application. Bellow, we briefly describe few potential applications of our work:

- **User Authentication** One of the key potential application of the proposed framework is user authentication for consumer devices. Authentication based on movements could be used in mobile phones, smart cars, smart homes and many other applications to effortlessly unlock the device for the user without the need for their direct communication.

- **Security and Surveillance Cameras** Motion biometric offers advantages over physiological biometrics which either require close-distance cooperation of the subject like fingerprint or their accuracy is highly dependable on image quality and lighting conditions like face and iris recognition. For example, as shown in figure 1, in low light conditions, only coarse body motions are detectable in security/surveillance camera footage and thus other biometrics fail. Moreover, in situation that the subjects are covering their face with a mask or hat, it would be difficult to identify them based on their conventional biometrics.

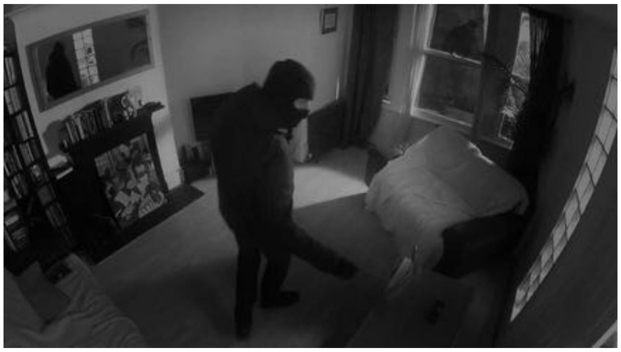


Figure 1: Example of surveillance footage in which conventional biometrics are not useful and only coarse body motions are detectable.

- **Virtual Reality** Emerging metaverses are predicted to define the future of entertainment and social interactions [145] where body motions are captured to provide an immersive experience. A key experience of these metaverses are interactive virtual avatars which are either built super-realistically from high resolution images of the real individuals or cartoonist animated 3D objects. People can walk into a casino and play a hand of poker or chat with their family/friends in remote locations as all are in the same location. While these metaverses try to emulate real life experiences and social interactions, avatars identity theft and fraud appears to be a big security concern [16, 24]. Authenticating virtual reality users by tracking their behaviour in performing goal-oriented tasks has gained attention in research community recently [89, 171].

- **Video Conferencing** On recent closures and due to Covid-19, online video conferencing has become a trend and part of everyday life, as people are confined in their places and work from home. Many of these video conferencing platforms provide the feature of avatars to communicate with others remotely. As an example, *LoomieLive* [7] is a video conferencing platform that provides customizable 3D avatars driven purely by audio. As shown in figure 2, users are able to be fully engaged in your video conference while protecting their appearance and background. This could be a situation where identity theft will become a significant security risk, specifically in confidential meeting.

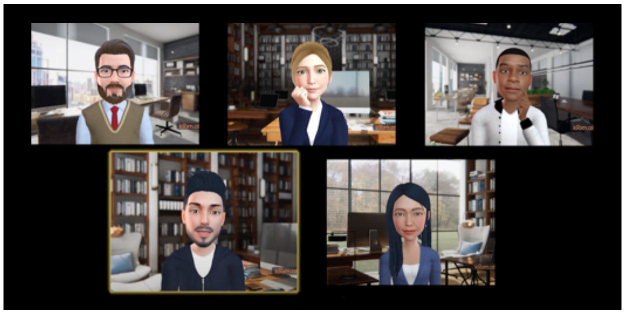


Figure 2: An example of avatar-based video conferencing: LoomieLive video conferencing allows users to fully engage in video conference meetings while protecting their appearance with a costumizable 3D avatar.

- **Content-based Video Indexing** Currently most of content-based video indexing technologies are based on based on pixels rather than perceived content [149]. Such thing can easily being manipulated by the state-of-the-art video tools and technologies. The framework proposed in this research could be considered as a step toward content-based video management.

- **Deepfake Identification** Deepfakes are synthetic media that leverage powerful techniques from machine learning, deep learning and computer vision to generate or manipulate visual and audio content with a high potential to deceive the audience [81]. In these videos one person is replaced with someone else's likeness. To mention few examples, the President Obama's 2019 video in which he was swearing during a public service announcement or Mark Zuckerberg's viral video in which he was announcing that he is deleting Facebook or Queen Elizabeth's alternative 2020 Christmas message that an image of it is shown in figure 3. Such deepfake videos are becoming more and more successful as the the quality of videos are increasing due to development of machine learning/deep learning and computer vision algorithms. Furthermore, the tools of today, and for sure those of tomorrow, provide accessibility for anyone to create such videos. In this situation, identity theft could be a big concern. Analyzing deepfake personalities motion could help with distinguishing between the real and the fake person.

Figure 3: Queen Elizabeth deepfake Christmas message vs her real Christmas message.

1.2 Key Aspects

- Firstly, as explicitly mentioned in the title of this proposal, each proposed methodology in this research is in associated with domain of **automatic analysis of human motion**

from visual data. We begin our exploration by looking at modeling human motion as a general problem.

- One significant priority of this work is the **data-driven learning** of visual data representations. Therefore, as oppose to designing hand crafted features for each specific type of input, we use **deep learning models** to achieve such purpose. In the context of our proposed research, we explore temporal strategies for learning motion features and propose a number of recurrent deep neural network architecture.
- All applications proposed in this dissertation are considered as a **machine learning problem**, regardless of their objective. Therefore, it worth to highlight the important aspects of the learning pipeline; first, the data used in this research are labeled real-data and the data representations are sequential (temporal); second, all models are trained in a fully supervised manner; and third, the problem formulation is considered as validation and classification problem.

1.3 Thesis Organization

This rest of this manuscript is organized as follows:

- Chapter 2 focuses on literature review of the existing state-of-the-art deep learning approaches implemented specifically for pose estimation, activity recognition, and biometric recognition.
- In Chapter 3, we temporarily put the proposed framework and its applications aside and focus on reviewing the existing deep learning models in detail.
- Chapter 4 is dedicated to our deep learning motion authentication framework and discussion around it.
- Chapter 5 focuses on proposing a deep learning framework for human identification among a set of 10 subjects.
- Chapter 6 evaluates and interprets the robustness of proposed methods in chapter 4 and chapter 5.

- Chapter 7 addresses the challenges of the potential human motion recognition applications and propose a lightweight framework which is applicable on devices with low computational power, storage, and memory.
- Chapter 8 concludes our work and sheds light on future potential research directions in this domain.

2.0 BACKGROUND: HUMAN MOTION ANALYSIS

In this chapter, we provide a general overview of existing approaches for human motion analysis from visual data. Even though some of the discussed context might not be directly relevant to the topic of this research, they still provide the essential background and insight for better understanding of the domain and existing challenges. For more extensive review in the area of deep learning, the reader is invited to move forward to the next chapters.

The field of human motion analysis contains a broad range of topics; e.g. human pose estimation, human activity recognition, and human authentication/recognition. Although some approaches for one problem might be applicable for another, depending on the level of abstraction, different aspects could play significant roles. In this chapter, we aim to provide a near-range subset of motion-related problems focusing on human pose estimation, human action recognition, human identification, and more general approaches when applicable.

2.1 Human Pose Estimation

The goal of Human Pose Estimation (HPE), which has been studied for decades, is to obtain posture of human body by automatically locating the position or spatial location of body keypoints from given sensors or in vision-based approaches, from a given image or video [84]. Challenges such as occlusions, clothing, lighting, small and barely visible keypoints, and strong articulation makes this task a difficult computer vision problem [23].

Traditionally, the gesture recognition process used to start with preprocessing of data to make algorithms to work on as much as useful data as possible. Then, it was followed by extracting hand-crafted feature which were designed by computer vision experts. Many of hand-crafted features were based on edge-based descriptors such as HoG [38], SIFT [104], and SURF [25]. Due to the high dimensionality of feature vector, principal component analysis is used in many of traditional approaches. However, the main difficulty with classical approaches is the necessity to claim crucial features in each task; i.g. one classifier that works for one task is not applicable to another. Also, traditional models do not scale well as the number of classes increase. As a result, classical methods showed bad generalization performance which followed by insufficiency in determining the accurate locations of body parts [112].

In 2012, a paradigm shift happened when AlexNet [85], a deep-learning based model, won the ImageNet competition by a large margin. Since then, deep neural network models has been applied in a variety range of topics including human pose recognition. Deep learning based approaches easily outperformed state-of-the-art traditional models by rapid progression as well as more significant feature extraction from metadata [40]. Figure 4 represents the differences between deep learning and traditional computer vision workflow. An interested reader may refer to [161] for detailed review of differences between deep learning and traditional computer vision approaches.

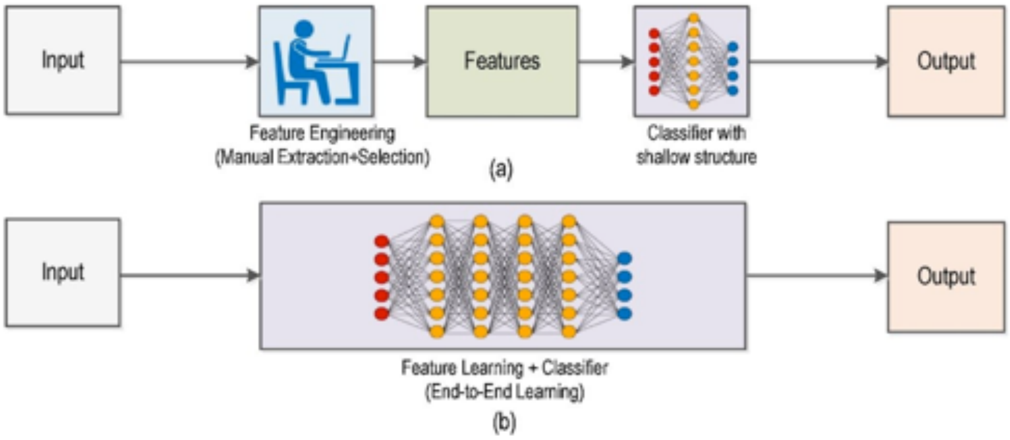


Figure 4: Traditional computer vision workflow represented in (a) vs. deep learning workflow represented in (b). Figure from [166]

While we do not study all deep learning approaches for human pose estimation in details, it is important to have an overview of existing works. For more details on current HPR deep learning models, an interested reader may refer to several notable surveys that summarized and compared the research in this area [33, 40, 176, 127]. The current literature of this area could be categorized in several different ways depending on the body type they choose as the representation or their approach to tackle the problem. As demonstrated in figure 5, there are three types of conventional human body models in human pose estimation literature; skeleton-based model, contour-based model, and volume-based model. In the next few paragraphs, We provide a general overview on each of human body models and we refer an interested reader to these two well-summarized surveys [102, 55] for more details.

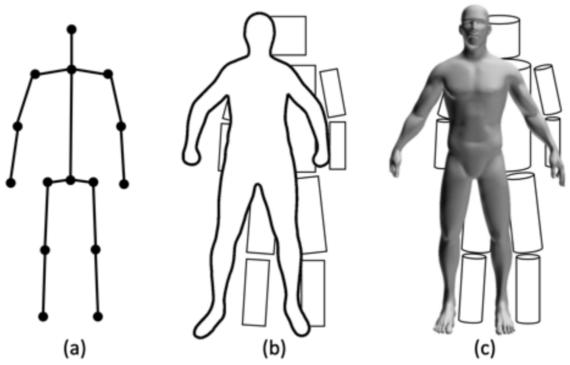


Figure 5: Conventional human body models used in HPE literature. (a) skeleton-based model, (b) contour-based models, (c) volume-based models. Figure from [33]

Skeleton-based Models Skeleton-based models could be described as a graph where nodes represent the locations of body joints and edges indicate the joints connections within the skeletal structure [49]. The simplicity and flexibility of the skeleton-based model lead to the wide utilization of this model in human pose estimation literature. However, it has some limitation in providing texture, width, and contour information.

Contour-based Models The contour-based model is used widely in earlier studies in human pose estimation. In this model, human body parts are represented by rectangular bounding boxes over the person silhouette. The bold benefit of this contour-based models is that they provide width and contour information. Cardboard models [79] and Active Shape Models (ASMs) [36] are two of widely used contour-based models.

Volume-based Model Volume-based models are 3D human body models, normally captured with 3D scans and represented in mesh form. Shape Completion and Animation of

People (SCAPE) [21] and Skinned Multi-Person Linear model (SMPL) [103] are two of the widely used volume-based models.

Moreover, current human pose estimation literature could be categorized based on their respective characteristics. Figure 6 illustrates the tree-structure taxonomy of deep learning-based monocular human pose estimation approaches.

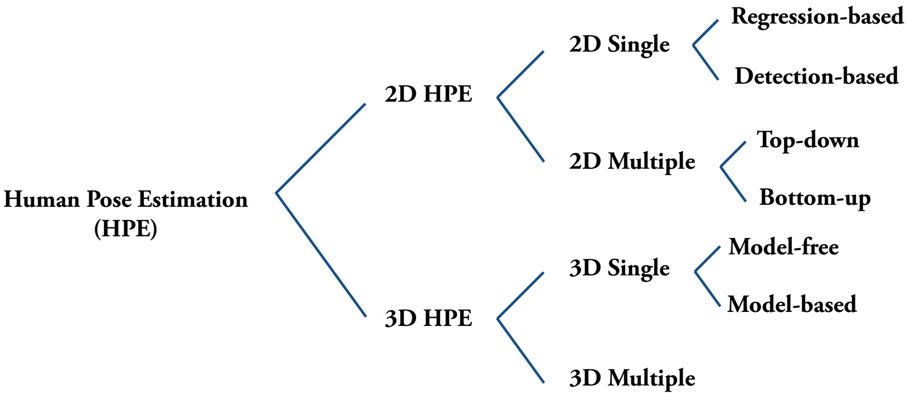


Figure 6: Taxonomy of deep learning-based monocular human pose estimation approaches

2.1.1 2D Human Pose Estimation

2D human pose estimation aims to predict the location of body keypoints/joints in 2D space from images or videos.

2.1.1.1 2D Single-Person Pose Estimation In 2D single-person pose estimation, the algorithm is able to localize one person in the input image. For images with more than one person, preprocessing is necessary to crop the image into images with single person. The single-human pose estimation pipeline could be classified into two categories depending on the different problem formulation in predicting keypoints: regression-based and detection-based methods.

Regression-based Methods In regression-based approaches, the input image is directly mapped to the body joints or parameter of human body models [157]. Due to nonlinearity of the problem, the direct mapping of the input image to body joints is a challenging task.

Detection-based Methods Detection-based methods aim to target the approximate locations of body parts or joints by rectangular bounding boxes or in recent works by heatmaps [34, 117]. The benefit of this approach is that the small region representations provide dense pixel information with stronger robustness. However, for example, heatmap representation suffer from limited accuracy due to lower resolution caused by the pooling operation in CNN. Figure 7 presents an example of heatmap-based single-person human pose estimation.

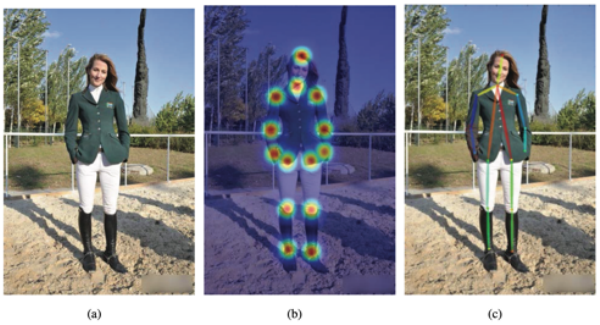


Figure 7: An example of heatmap-based single-person human pose estimation: (a) original image ; (b) generated heatmap; (c) estimated pose

2.1.1.2 2D Multi-Person Pose Estimation There is a set of challenges that makes 2D multi-person pose estimation a difficult task. For example, there is no prompt over the number of people in each input image and the interactions between individuals increase the complexity of part associations. Moreover, as the number of people increases in an image, the run-time complexity tends to grow. In order to address these challenges, 2D multi-person pose estimation approaches can be categorized into two mainstream methods: top-down and bottom-up approaches.

Top-down Methods Top-down approaches (e.g. PoseNet [123], RMPE [48]) first detect all the people in the image using a set of bounding boxes, and then try to predict individuals poses using the existing single-person pose estimation method. The efficiency of top-down approaches in multi-person pose estimation heavily depends on the performance of the pose estimator and the run time is proportional to the number of people in the image.

Bottom-up Methods Bottom-up approaches (e.g. DeepCut [128], DeeperCut [71], OpenPose [30]) directly detect the location of body joints and then assemble the into distinct human body skeletons. In a complex environment with multiple human, correct assembling of keypoints is a challenging task. The processing time of some of the existing bottom-up approaches are really fast. Even some are able to be run in real time [30, 118].

2.1.2 3D Human Pose Estimation

The goal of 3D human pose estimation is to predict the location of body joints in 3D space from visual inputs or other sources such as Kinect [6], VICON [13], or TheCaptury [11]. However, compare to monocular cameras that have been widely used for 3D pose estimation, these commercial products required special markers/devices on human body and work in a very constrained environment. Compare to 2D human pose estimation, 3D estimation is significantly more difficult problem as it needs to predict the depth information related to keypoints. Moreover, lack of 3D in the wild ground truth data is a major limitation. In this section we divide the current approaches of 3D human pose estimation into two sections of 3D single-person pose estimation and 3D multi-person pose estimation and provide a general overview on each one.

2.1.2.1 3D Single-Person Pose Estimation In most of 3D single-person pose estimation approaches, the bounding box around the person is provided; meaning that the algorithm is not necessarily enhanced with the person detection process as well. As shown in figure 6, 3D single person pose estimation could be divided into two main categories: model-free methods and model-based methods.

Model-free Methods In model free methods, human body model is not used as the predicted target or half-between cues. Instead, they either directly map an image into a 3D-pose or estimate depth (e.g. [94, 126, 95, 154, 125]) or estimate depth following intermediately predicted 2D pose from 2D pose estimation methods. The second approach

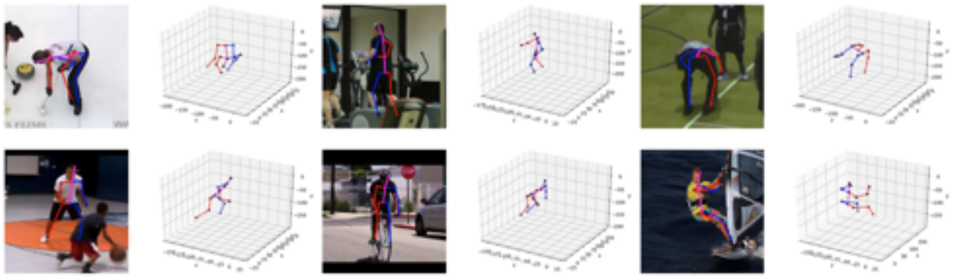


Figure 8: Predicting 3D human pose estimation using 2D approaches. Figure from [178].

gains the benefit of 2D pose estimation which involves the easy utilization of images from 2D datasets; e.g. as shown in figure 8, some of them first estimate the 2D pose and then extend that to 3D by a 2D-to-3D pose estimator which utilizes linear layers [106] or heatmaps [178, 155].

Model-based methods In model-based approaches, a parametric body model is used for human pose estimation. Recent models are majorly estimated from multiple scans of diverse people (e.g. [61, 103, 179]) or combination of different body models (e.g. [78]). The parameters in these models are updated based on separate body pose and shape components.

2.1.2.2 3D Multi-Person Pose Estimation The field of 3D multi-person pose estimation is a pretty much a new field and a few methods are propose. For example, [108] utilizes a 2D bottom up approach to estimate individuals pose in 2D and then uses an occlusion-robust pose-maps (ORPM) for multi-style occlusion information. As another example, [136] uses a three-stage neural network that first detects the location of individuals using Region-based Convolutional Neural Networks (R-CNN) is employed to detect people locations. Then using a classifier, each pose proposal is assigned with the closest anchor-pose. Finally, the poses are refined with a regressor correspondingly.

2.1.3 OpenPose

OpenPose [31], is one of the most popular single/multi-person 2D pose estimation frameworks that successfully estimates human body, hand, face, and foot keypoints (135 keypoints in total) from a single image, video, or real-time footage. OpenPose first introduced on 2016 and ever since has been widely used in many studies with different purposes to obtain human body pose [120, 113, 122, 44, 158].

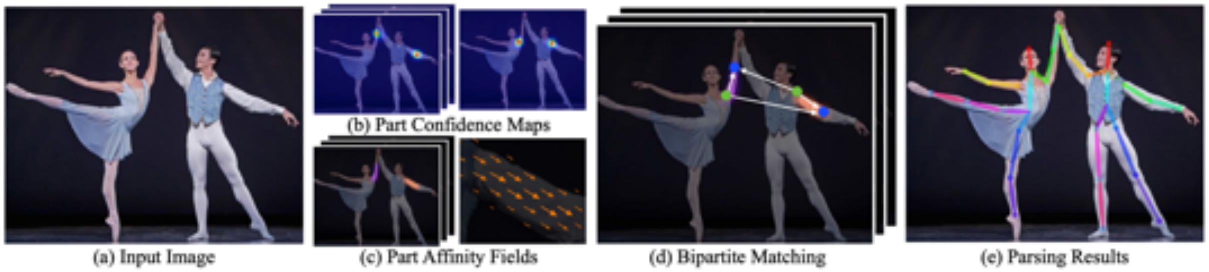


Figure 9: OpenPose overall pipeline [30]: (a) input image that is fed into the CNN network which jointly predicts: (b) part confidence map and (c) PAFs for part association. (d) the bipartite step which matches body part candidates. (e) part candidates are matched into full poses for all individuals in each image.

Taking a bottom-up approach, OpenPose uses a multi-stage Convolutional Neural Network (CNN) architecture, in which first predicts the confidence map of body keypoints using a feed-forward network and then finds the Part Affinity Fields (PAFs), a two dimensional vector that demonstrates the position and orientation of each limb over the image domain. Later, in the parsing step, a set of bipartition matching is performed to associate body part candidates and finally part candidates are assembled into full-body poses. Figure 9 represents the OpenPose general pipeline; the network takes a $w \times h$ colored image as the input (a) and estimates the 2D skeleton-based poses for each person within the image (e).

The OpenPose network architecture is presented in figure 14. The input image is first passed to a pre-trained convolutional neural network such as the first 10 layers of VGG-19 [147] for feature extraction. The feature maps F are the input to the proposed architecture.

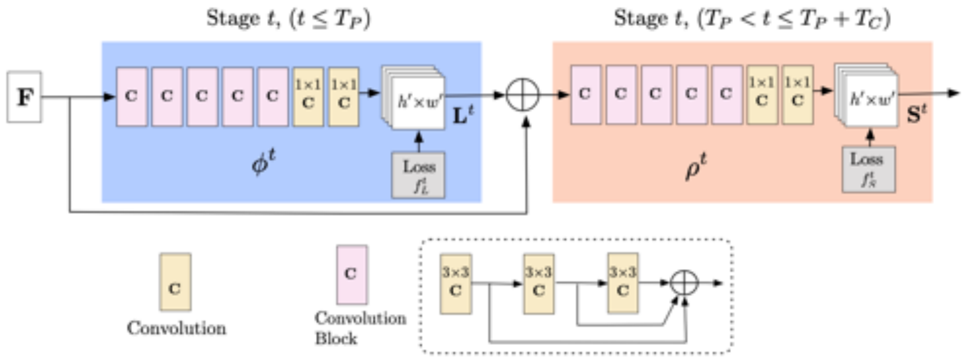


Figure 10: OpenPose multi-stage architecture: The first set of stages in blue, predicts PAFs; the last set in orange detects confidence maps. The predictions of each stage and their corresponding image features are concatenated for each subsequent stage. Each convolutional block includes 3 convolutional layer of kernel 3 [30].

Through a feed forward network, the first set of steps shown in blue, iteretively predicts the part affinity fields (L) and the steps in orange detects confidence maps (S). The symbol ϕ^t is used as function representation of the CNN with input F that outputs part affinity field L at stage t, and the symbol ρ^t is used as function representation of the CNN with input F that outputs confidence map S at stage t. With intermediate supervision at each stage, the iterative process helps to increase the precision of predictions over successive stages; i.e. original features along with the prediction from previous stage are all used to produces more refine predictions:

$$S^t = \rho^t(F, S^{t-1}, L^{t-1})), \forall t \geq 2 \tag{2.1}$$

$$L^t = \phi^t(F, S^{t-1}, L^{t-1})), \forall t \geq 2 \tag{2.2}$$

At the end of each branch one loss function is applied in order to generate the best sets of S and L. A standard L2 loss is used the estimated predictions and ground truth maps and fields:

$$f_S^t = \sum_{j=1}^{J} \sum_{P} W(p).||S_j^t(p) - S_j^*(p)||_2^2 \tag{2.3}$$

$$f_L^t = \sum_{c=1}^{C} \sum_{P} W(p).||L_c^t(p) - L_c^*(p)||_2^2 \tag{2.4}$$

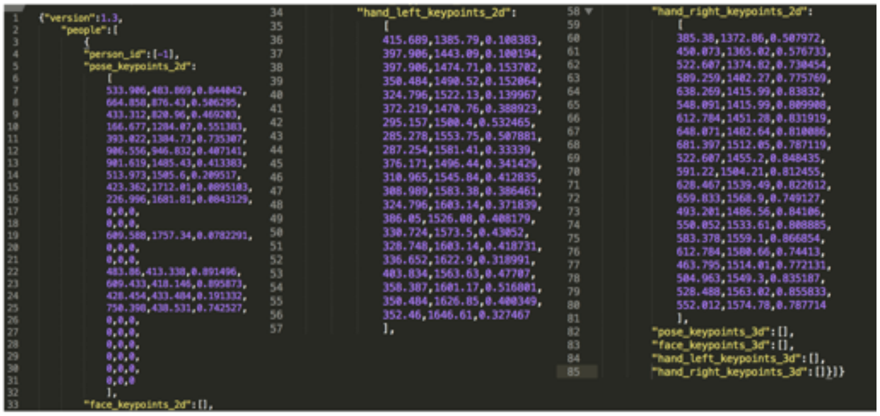
Where p represents a single pixel location in an image, * stands for ground truth, J represents the total number of body part and C represents the total number of limbs. $W(p)$ stands for the weight added to the loss functions to address a practical problem that some datasets do not completely label all individuals. The overall loss function is the combination of the two above-mentioned loss functions:

$$f = \sum_{t=1}^{T} (f_S^t + f_L^t) \tag{2.5}$$

One of the key benefits of OpenPose is that it can be run on both GPU and CPU only systems with different operating systems including Mac OSX, Windows, Ubuntu, and embedded systems. The user can select an input between image, video, webcam. The output could be in images with keypoint displays (JSON, XML, YML, ...) or could be saved as array of keypoint coordinations on disk (JSON, XML, YML, ...). The user is also able to enable/disable body, foot, face, and hand detectors, control pixel normalization, and even control the number of GPUs to use. Figure 11 represents an example of OpenPose *json* output for 2D estimation of body and hands keypoints. As can be seen, each keypoint is represented by coordinates x and y and a degree of confidence *c*. The image output of OpenPose could be seen in figure 9 part (e).

2.2 Human Activity Recognition

After successful implementation of algorithms to estimate human body pose in each image frames, understanding human movements and the type of activity they are doing



```
{"version":1.3,
    "people":[
        {
        "person_id":[-1],
        "pose_keypoints_2d":
            [
            533.906,483.869,0.844042,
            664.858,876.43,0.506295,
            433.312,820.96,0.469203,
            166.677,1284.07,0.551383,
            393.022,1384.73,0.735307,
            906.556,946.832,0.407141,
            901.619,1485.43,0.413383,
            513.973,1505.6,0.209517,
            423.362,1712.01,0.0895103,
            226.996,1681.81,0.0843129,
            0,0,0,
            0,0,0,
            609.588,1757.34,0.0782291,
            0,0,0,
            0,0,0,
            483.86,413.338,0.891496,
            609.433,418.146,0.895873,
            428.454,433.484,0.191332,
            750.398,438.531,0.742527,
            0,0,0,
            0,0,0,
            0,0,0,
            0,0,0,
            0,0,0,
            0,0,0
            ],
        "face_keypoints_2d":[],
        "hand_left_keypoints_2d":
            [
            415.689,1385.79,0.108383,
            397.906,1443.09,0.100194,
            397.906,1474.71,0.153702,
            350.484,1490.52,0.152064,
            324.796,1522.13,0.139967,
            372.219,1470.76,0.388923,
            295.157,1500.4,0.532465,
            285.278,1553.75,0.507881,
            287.254,1581.41,0.33339,
            376.171,1496.44,0.341429,
            310.965,1545.84,0.412835,
            308.989,1583.38,0.386461,
            324.796,1603.14,0.371839,
            386.05,1526.08,0.408179,
            330.724,1573.5,0.43052,
            328.748,1603.14,0.418731,
            336.652,1622.9,0.318991,
            403.834,1563.63,0.47707,
            358.387,1601.17,0.516801,
            350.484,1626.85,0.400349,
            352.46,1646.61,0.327467
            ],
        "hand_right_keypoints_2d":
            [
            385.38,1372.86,0.507972,
            450.073,1365.02,0.576733,
            522.607,1374.82,0.730454,
            589.259,1402.27,0.775769,
            638.269,1415.99,0.83832,
            548.091,1415.99,0.809908,
            612.784,1451.28,0.831919,
            648.071,1482.64,0.810086,
            681.397,1512.05,0.787119,
            522.607,1455.2,0.848435,
            591.22,1504.21,0.812455,
            628.467,1539.49,0.822612,
            659.833,1568.9,0.749127,
            493.201,1486.56,0.84106,
            550.052,1533.61,0.808885,
            583.378,1559.1,0.866854,
            612.784,1580.66,0.74413,
            463.795,1514.01,0.772131,
            504.963,1549.3,0.835187,
            528.488,1563.02,0.855833,
            552.012,1574.78,0.787714
            ],
        "pose_keypoints_3d":[],
        "face_keypoints_3d":[],
        "hand_left_keypoints_3d":[],
        "hand_right_keypoints_3d":[]}]}
```

Figure 11: OpenPose json file output for 2D estimation of hand and body keypoints.

naturally comes next. The goal of human activity recognition (HAR) is to automatically distinguish and analyze human activities from data captured through sensors or cameras. Although, identifying the category of these activities is an easy task for human being, it is a very difficult problem for intelligent computer systems as human motions space is very high dimensional and the interactions complicate the searching process into this space. Moreover, the instantiation of same task by different subjects with different style of movements, creates substantial variations [172].

However, due to the wide range of applications, a huge amount of research has been actively conducted on identifying human activities since 1990s [50]. The task could be generally described as: given a sequence of movement data, identify the class of activity that the subject is performing. Depending on the activity, task's complexity might vary. For example, walking or dancing is less complex than the scenarios when subjects are interacting with others.

The research conducted in the field of human activity recognition are divided into two primary categories depending on the method that the input data is captured: vision-based and sensor-based approaches. In sensor-based approaches, movement data are collected from wearable sensors (e.g. [45, 131]), smartphones (e.g. [90]), or environmental sensors (e.g. [170]). Signal data captured from sensors could be one-dimensional (as in wearable devices) or could be 2-dimensional or 3-dimensional (as in optical sensors). However, one problem with sensors is that when the subject is beyond their range, they are not able to receive the signals and work properly. Vision-based approaches, on the other hand, are those that the data are collected through cameras. Thus, they are more affordable and easy to collect as cameras are much more on hand than sensor devices. However, the precision of outputs relies heavily on the image quality as well as brightness changes [39]. Although human activity recognition using vision-based approaches is much more new of a field, but both of these approaches are still very popular and are actively being used.

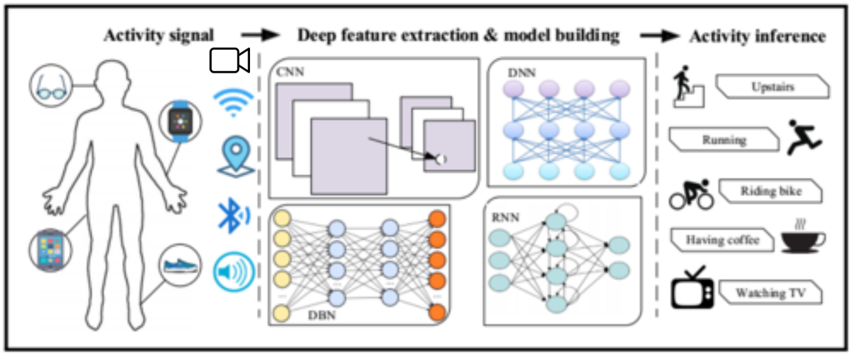


Figure 12: Human activity recognition process with deep learning approaches. Figure from [3].

For a long period of time, traditional machine learning algorithms such as support vector machine (SVM) [15], random forest [65], Bayesian networks [169], and Markov models [138] have been used to solve the activity recognition problem. In addition to the time-consuming steps for hand-crafting features, machine learning algorithms were performing great only under restricted and controlled environments and also limited input data [130]. In recent years, deep learning approaches received a great amount of attention from the computer vision community as they outperformed conventional machine learning algorithms in many tasks including human activity recognition. While the input of human activity recognition models could be captured through different devices, after they pre-processed the deep learning is a robust method that could be used in both categories.

As an example, [69] presents a user-independent deep learning-based approach using CNN for local feature extraction together with simple statistical features that preserve information about the global form of time series. They further compare the impact of time series length on the performance of their neural network. [163] also designs and implements a smartphone inertial accelerometer-based architecture for human activity recognition which basically collects the sensor data sequence and extracts high-efficient features and obtains user behavior by a three-axis accelerometers. After pre-processing data and extracting feature vectors, they use a real-time classification deep learning method using CNN, LSTM, and SVM and compared the results.

On the other hand, [120] predicts human activities from monocular videos in MHAD database [2] which contains 10 different classes of human activities such as waiving, jumping, clapping, and etc. They first extract the human pose in each video frame using a skeleton-based human pose estimation method and then extract human motion feature vectors between consecutive frames. Finally they apply an LSTM recurrent neural network on feature vectors and reach an overall accuracy of 92.4%.

The majority of studies in HAR demonstrated promising results by classifying the activities that have been already seen during training. [18] introduces a method that is able to discover and infer new unseen activities that integrates low-level sensor data with the semantic similarity of world vectors as it would be more efficient to re-utilize information obtained from existing activity recognition models instead of collecting more data with the goal of training a new model from scratch. Figure 13 represents the main idea of such work.

As, human activity recognition has been significantly studied in the literature and the state-of-the-art methods are studied and surveyed in different papers, we may refer an interested reader to following papers for more details in this area: [26, 20, 28, 151, 133, 99, 77, 148, 160, 146, 91].

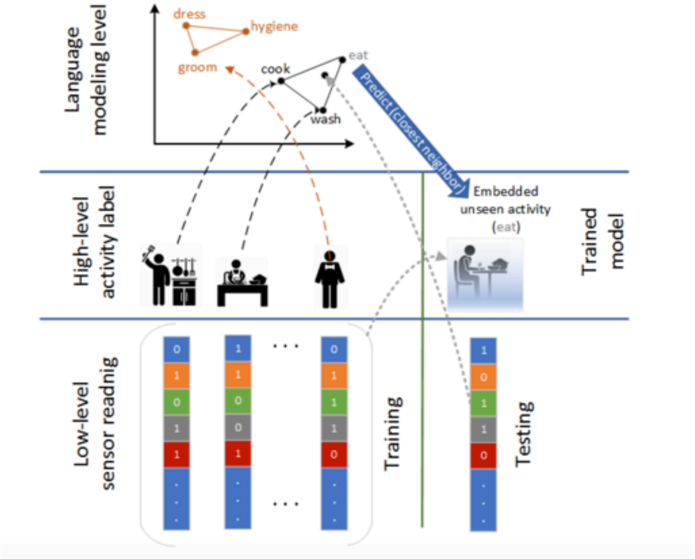


Figure 13: The main idea of Zero-shot human activity recognition using non-visual sensors [18].

2.3 Human Biometric Recognition

For personalization, authentication or security purposes, it is important to be able to distinguish a person from others and make an application less accessible. Biometrics, unlike token-based features such as ID cards or passwords, cannot get lost or easily be emulated [109]. Even though, there have been several attempts to duplicate them [52, 47], many methods have been established to distinguish the real biometrics from the fake ones [4, 110, 100]. Biometric features are generally divided into two primary group: physiological features [73] such as facial features, fingerprints, palms, iris, retinas, ears, and dental biometrics, or behavioral features [175] such as signatures, keystroke, and gait rhythm while voice and speech features could be considered as both as it has a touch of both physiological and behavioral biometric features.

Among all biometrics considered in human recognition literature, fingerprints and face are arguably the most commonly used biometrics [109]. Face, because of having several discriminative features, can be a good fit for recognition task [75]. However it is prone to change due to aging, surgeries, facial expressions, pose, angle, and image resolution [124, 59]. Fingerprints minutiae features, on the other hands, are robust from a unique pattern for each person. They were first used for murder identification dating back to 1893 [62]. Palmprints are another biometrics considered for authentication purposes by the computer vision community. Geometry of hands, principal lines, delta points, and minutiae patterns are all features that used to to distinguish individuals based on their palmprints [174].

Texture of iris or pattern of blood vessels in retinas are also two popular biometrics within eye that are used for identification task. Since there are many factors are involved in formation of retina patterns and iris textures, the probability of the false matches is extremely low [41]. It has been shown that even the two eyes of the same person have different retina texture [42]. However, [137] brings up the concern of changing in the iris texture of patients involved in a modern cataract surgery, in way that the iris recognition is no longer feasible which is a valid concern in the application that rely on this authentication technique.

Ears are another biometrics used for identification; shape of lobes, means, centroids are all considered relatively statistic for each person and all can be measured remotely and does not request for the individual's direct interaction [27, 14], however ears size does change over time which brings up some challenges for this authentication technique.

Dental identification based on radiography images is another technique mostly used in cases that other assets of identification (e.g., fingerprint, face, etc) are not available. Tooth can change appearance within time due to dental work, decays, missing, or etc; that's why in court of law, dental based identification is considered less reliable than other biometrics, but in some cases such as fire, this could be the only available biometrics [74, 32].

Additionally, our ability to recognize a friend based on their manner of walking [37], suggests that human identification could go well beyond physical features. Therefore a huge amount of studies focused on analyzing behavioral biometrics to find patterns that are robust enough to identify individuals with them. Among behavioral biometrics, signature is one biometrics that has been widely used for identification. Speed, pressure of the pen

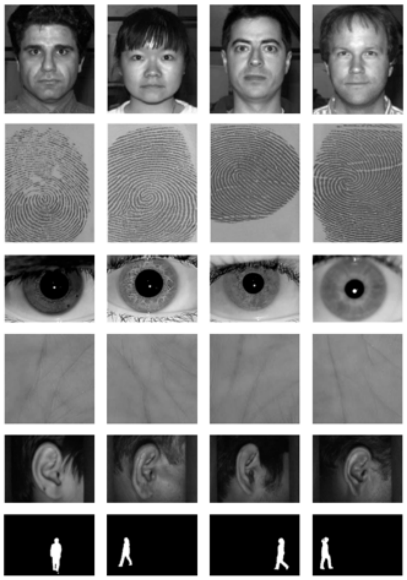


Figure 14: Sample images of several biometrics gathered for recognition task. From top to bottom: sample images for face [5], fingerprint [8], iris [86], plamprint [9], ear [87], and gait [167] recognition. Courtesy of [109].

during creation, thickness of strokes, all could be measured and compared for authentication purposes [150].

Linguistics, is another biometric domain that has been very practical during history to identify writers based on the language and writing patterns [51]. Moreover, Voice examining both physiological and behavioral biometric, applies analyzes of a person's voice to verify their identity. While twins might have a similar voice print, different algorithm has been conducted to distinguish them based on their speech pattern combined with some other biometrics such as ears shape [17].

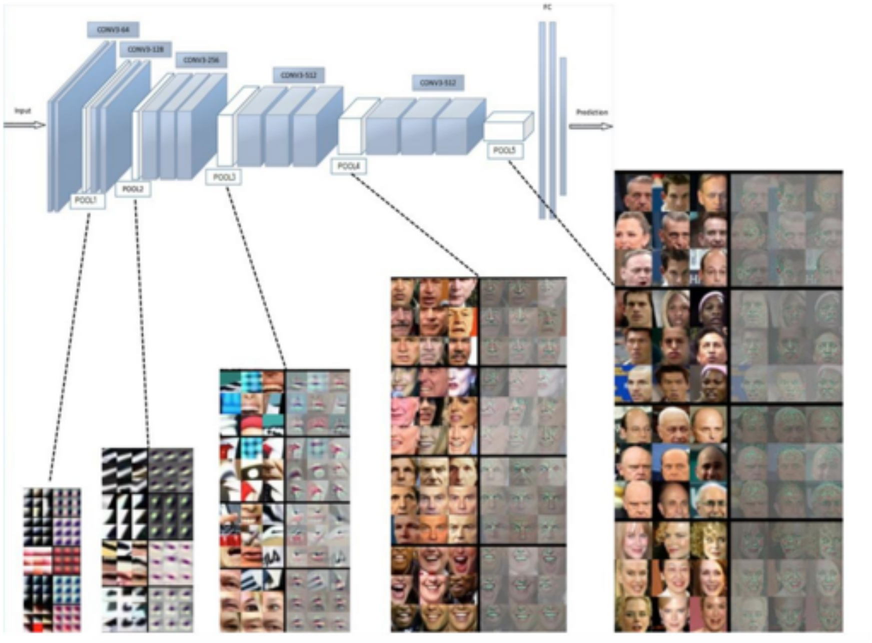


Figure 15: Illustration of learning feature representation by deep neural network [107].

Unlike classical biometric recognition which was based on hand-crafted features, deep learning based models provide an end to end learning framework which can jointly learn feature representation and the recognition task. This is achieved by a multi-layer network which is able to learn feature representation in multi level as shown in figure 15. The progress in processor technologies as well as development of new methodologies for training neural networks not only enabled scientists to train existing biometric recognition deep neural networks much faster, but also led them to explore other human biometrics for the human identification task. As an example, gait analysis, including stride length and arm swing, has attracted significant attention as a non-contact, non-obtrusive method for authentication which is resilient to cyber attacks [162]. Moreover, it is less likely to be obscured as it appears to be difficult to camouflage, especially in cases of serious crime [119]. [35] provides a detailed review of existing state-of-the-art biometric recognition approaches. [121], for example, utilizes arm swing data captured from smartphone inertial sensors, accelerators and gyroscopes, and explore convolutional temporal models for human authentication.

Human body movements, one of the key aspects of human social behavior, have enjoyed the attention in a variety of research communities [33]. Since 1970, human body motion has been studied to address a broad range of challenges including gesture [114, 101], action [129, 168], and activity recognition [133, 165].

A number of studies have suggested the existence of *distinct* and *identifiable* motion patterns of individuals by analyzing movement data obtained from Kinect devices [115] and Electromyography (EMG) electrodes attached to muscles [67]. From these studies, unique motion patterns can be attributed to anatomical differences and the existence of individual muscle activation strategies or signatures. Behavioural contrasts between individuals can be considered as another reason for individual movements and their relations to each other [171, 46, 111]. These studies highlight the potential of body movements as a biometric method and its usefulness in multi-modal identification systems. Motion biometric offers advantages over physiological biometrics which either require close-distance cooperation of the subject like fingerprint or their accuracy is highly dependable on image quality and lighting conditions like face and iris recognition. For example, in low light conditions, only coarse body motions are detectable in security/surveillance camera footage and thus other biometrics fail.

Recently, a number of researches have used human movements as an authentication method relying on the fact that individuals movements are distinct and identifiable. For example, *Headbanger* [97], is a software device implemented on Google GLASS to authenticate users through their free-style head movement in response to an external audio stimulus. The movement data are collected from the built-in accelerometer when the user performs head movements. As illustrated in figure 16, the reliability and robustness of the system of the system is evaluated and shown in two authenticating mode; authenticating the owner for login, and preventing an attacker from login. The authentication process in headbanger is consisted of four principal steps; sensor data collection, filtering data to smoothen them for subsequent processing, parameter generation, and classification. For the parameter generation the distances between two accelerometer samples are calculated and considered as parameters. For classification, a threshold–based classifier with a threshold $\mu_s + n\sigma_s$ is chosen to find the top-K samples with the smallest average distance as template. μ_s denotes the average value of the distance, σ_s is the standard deviation, and n is a tunable parameter of the classifier. In authenticating process, if the distance between the testing sample and the template is bellow the threshold, the test sample is labeled as success and the user is accepted by the system. Otherwise, the user is not considered as the owner, and is rejected.

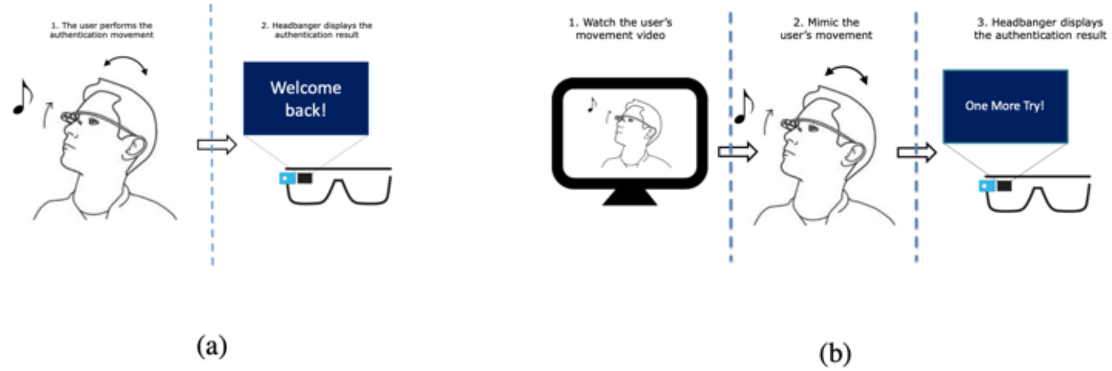

1. The user performs the authentication movement
2. Headbanger displays the authentication result
Welcome back!
(a)
1. Watch the user's movement video
2. Mimic the user's movement
3. Headbanger displays the authentication result
One More Try!
(b)

Figure 16: Illustration of the evaluation of headbanger in (a) authenticating the owner, (b) preventing attackers. Figure from [97].

In another effort, [70] proposes a method by integrating laser range finders (LRFs) in the environment and wearable accelerometers with reliable ID information to identity of pedestrians. The users are identified when the results of walking motions from feet of a pedestrian matches his/her body oscillation. Figure 17 demonstrate the proposed concept and flow of the proposed network.

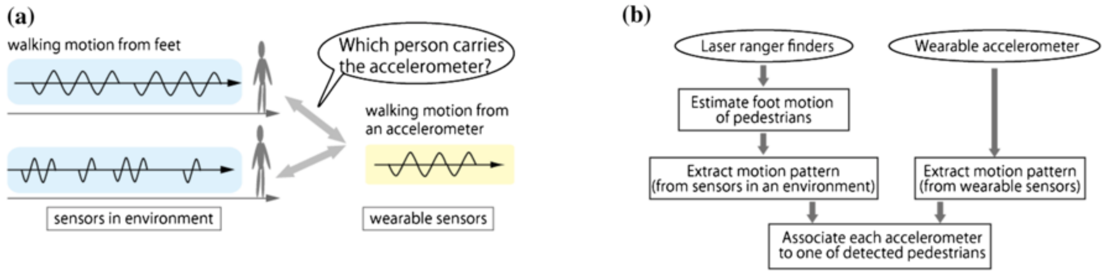

(a)
walking motion from feet
Which person carries the accelerometer?
walking motion from an accelerometer
sensors in environment
wearable sensors
(b)
Laser ranger finders
Wearable accelerometer
Estimate foot motion of pedestrians
Extract motion pattern (from sensors in an environment)
Extract motion pattern (from wearable sensors)
Associate each accelerometer to one of detected pedestrians

Figure 17: Concept (a) and flow (b) of the proposed framework [70] for pedestrian identification.

Another line of research, focused on human identification approaches based on radar micro-Doppler signatures. Doppler radar are known as a suitable tool to collect data as it is not affected by low light light and bad weather conditions. Using a three-layer deep convolutional autoencoder, [143] successfully distinguishes seven gaits based on the micro-Doppler signatures collected by a 4 GHz continuous wave radar. [135] propose a feature extraction algorithm which automatically generate a set of shape spectrum features extracted from the cadence velocity diagram of the human micro-Doppler signature collected by an X-band radar. Further, a Naïve Bayesian classifier and a shape-similarity-spectrum classifier is used to recognize individuals. Perhaps the most relevant study to this research is [29] which utilizes deep learning algorithms to address the recognition task based on individuals radar micro-Doppler signatures. This work propose a deep onvolutional neural network which is able to learn the necessary features and classification conditions from raw micro-Doppler spectrograms. The same method has been previously used in [82] for activity recognition which could be considered as a simpler task as the spectrograms of different human activities are significantly different most of the time while spectrograms of different people performing the same activity is very similar. Figure 18 demonstrates two examples of people spectrograms. Once the spectroms are collected a deep convolutional neural network with a logistic regression is employed to solve the multi-class classification problems. The achieve the average accuracy of 85.6% for 10 people identification.

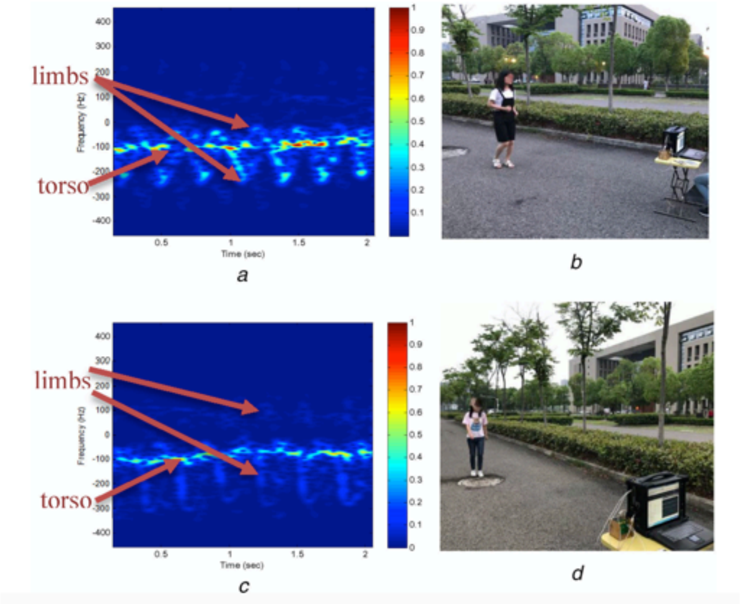


Figure 18: Sample spectrograms of two different people walking on the left, samples of visual data on the right.

In this work, we analyze human upper-body motion features captured from in-the-wild footage of individuals to to learn their movement patterns. Later, given a human movement sequence data, we predict the identity of the person based on the probabilities obtained by our proposed model.

3.0 BACKGROUND: DEEP LEARNING

Deep Learning, which is a class of machine learning, has been experiencing waves of excitement followed by a period of oblivion. However, it was not used until recent years that the progress in graphics processing units (GPUs) and the appearance of large, high-quality labeled datasets and the invention of advanced algorithms, made it possible for computers to learn in an entirely data-driven fashion with minimal feature engineering.

In 1940s-1960s early works in deep learning, or better say *cybernetics*, as it use to be called back then, was described as a simple neural network trained in supervised and unsupervised manner [252,122]. Later, in 1980s-1990s, the second wave came under the name of *connectionism* with the invention of backpropagation [254]. The modern era of deep learning as defined by [109] started in 2006 [126, 26, 245] and since around 2011 it has been actively used and made a tremendous impact in a variety of domains such as image processing, computer vision, natural language processing, machine translation, medical information processing and image analysis, art, and so many others. Experimental results show that deep learning successfully outperformed traditional machine learning approach in majority of domains. It worth mentioning that some of the connectionism works are still reconsidered and reformulated in new contexts. We may encourage an interested reader to go through this comprehensive survey of the field [259]] for more in-depth details about the history of the matter.

In this chapter, we aim to build a ground for our work by reviewing some of the most relevant classes of deep learning models. We start by reviewing the categories of deep learning approaches followed by discussing state-of the art feed-forward and temporal neural models. we further provide a brief review on milestones of unsupervised learning.

3.1 Types of Deep Leaning Approaches

Deep learning approaches are categorized into few main categories: supervised, semi-supervised (or partially supervised), and unsupervised. In **supervised learning**, labeled data are used to train the model and calculate loss function. **Semi-supervised learning**, use data that are partially labeled as it is in Generative Adversarial Networks (GAN). In **unsupervised learning**, data are not labeled at all and the model aims to figure out the internal representation or important features to discover unknown relation of data. Clustering, generative techniques, and dimensionality reduction are examples of unsupervised learning. In adition to these categories, there is also another learning category named **Reinforcement Learning (RL)** that deals with learning via interaction and feedback, or in other words learning to solve a task by trial and error. RL is sometimes discussed under semi-supervised and sometimes are discussed under unsupervised learning approaches [19]. The pictorial diagram of deep learning categories is represented in figure 19.

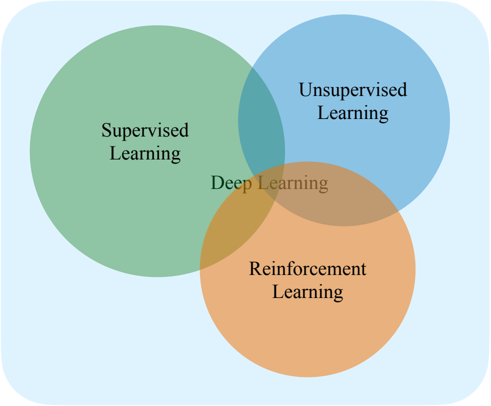


Figure 19: The categories of deep learning approaches; semi-supervised learning is considered as the shared area between supervised and unsupervised learning.

3.2 Deep Feedforward Network

Deep feedforward neural network, also called **multi layer perceptrons (MLPs)** are the quintessential deep learning models [56] which basically is a connections of neurons/units. The general purpose of feedforward networks is to map an input x to an output y by approximating function f which is defined as $y = f(x;\theta)$. The values of parameter θ is expected to be learned in a way that results into the best approximation. The term **feedforward** relates to the fact that the information flows through the function from the input to the output and there is no feedback fed again to the model. The term **network** also is used as the feedforward neural networks can be typically defined as a chain of functions $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$ where each $f^{(n)}$ represents the n-th layer of the network build from hidden units that act in parallel. Each hidden units resembles a neuron that received information (inputs) from many other units and computes its activation value. In **fully connected** networks, each neuron in a layer receives information from all neurons in the previous layer. Moreover, the term **neural** in these networks refers to hidden units (neurons) which is inspired by neuroscience.

We assume that the reader is familiar with the fundamentals of artificial intelligence models. However, we will still go over feedforward neural network basic equations but try our best to keep it as brief as possible.

In theory, *universal approximation theorem* states proves that for any arbitrary smooth function, there exists a feedforward network large enough with any squashing activation function that is able to achieve any degree of accuracy we desire. However, the theorem does not specify the sufficient amount of hidden layers for such approximation. In practice, it is enough to have a continuous function on a bounded subset of $\mathbb{R}^n$. Furthermore, it worth mentioning that feedforward networks ability to estimate any smooth functions does not mean that the approximation could be learned. Overfitting could be one example of such situation.

Design of hidden units is not as easy task. Although, there are many research actively focused on this problem, the process still is consisted of trial and errors and it is impossible to say which functions works best in advance. To choose the best hidden units, the network is trained based on a specific kind and the performance of the network is evaluated and compared.

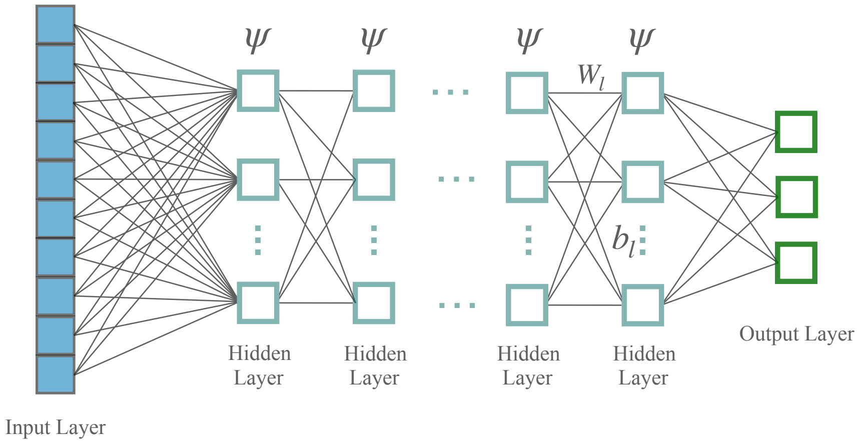


Figure 20: The architecture of fully connected deep feedforward neural network.

Finally, let's dive into the notations of fully connected feedforward neural network equation. As demonstrated in figure 20, at each layer l+1 and for each input, the values taken by hidden units x_{l+1} are recursively calculated by taking from activations of the previous layer x_l in the following vector form:

$$x_{l+1}, j = \psi(W_l x_l + b_l) \tag{3.1}$$

where ψ represents a non-linear activation function such as sigmoid, hyperbolic tangent, and rectified linear unit (ReLU), W_l is a weight matrix defining the weight coefficient between the i^{th} neuron from previous layer and the j^{th} neuron in the l^{th} layer, and b_l is the vector of biasis in each layer.

Generally neural networks are trained by *gradient decent* using chain rule and **back-propagation** [139] of the error from output to input in order to minimize the error defined in cost function. In other word, back propagation allows the information from cost flows back in the network for computing gradient decent. Depending on the type of data, the loss function varies. In supervised regression, loss function is obtained by calculating the mean square error (MSE) similar to machine learning approaches:

$$L_R(\mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{N} ||y_j^{(i)} - \tilde{y}_j^{(i)}||_2^2 \tag{3.2}$$

where $\mathcal{D}$ represents a training samples, $y^{(i)}$ is a ground truth label of *i-th* sample while the $\tilde{y}^(i)$ is the networks predicted label of sample i, and N is the number of outputs. While loss function calculates the penalty in a single training set, the cost function almost refers to the same meaning but calculates the penalty in a number of training sets.

In classification task, there is a need to represent a probability distribution over a discrete variable with n possible values that represent classes. The output function is a softmax which is a generalization of sigmoid function that is used in binary classification problems. The winner class is the one with the highest posterior probability. The posterior probability for observation x_o for class c_n is calculated as:

$$\tilde{y}_n = P(c_n|x_o) = \frac{\exp(W_j^n x_{l-1})}{\sum_j \exp(W_j^T x_{l-1})} \tag{3.3}$$

where W denotes the weight matrix and W_n is a set of weights connecting the previous layer with an output element n. The loss function in classification is calculated by taking the negative log-likelihood of the softmax which is the cross-cross-entropy between the ground truth labels and the network prediction:

$$L_c(\theta, \mathcal{D}) = -\sum_{i=1}^{|\mathcal{D}|} \log P(\mathcal{Y} = y^{(i)}|x_o^{(i)}, \theta) \tag{3.4}$$

where θ denotes all of network parameters including biases and weights for all layers.

3.3 Convolutional Neural Networks

A **convolutional neural network (CNN)** [92] is a deep learning algorithm which have been tremendously successful in capturing the spatial and temporal dependencies in data with grid-like topology such as image data (2D-grid) or time-series (1-D grid). The high dimensional of input data could be very high in these cases. For example, imagine a 7680×4320 image which is consisted of three channel; red, green, and blue. the role of a CNN is to reduce the dimentionality of the data in a way that the essential features that are critical for a good prediction are not lost. Therefore, the primary benefit of CNN architecture over fully-connected networks is the significant reduction in the number of parameters as the input is processed locally by sliding a set of convolutional filters over it [56]. For example in case of very basic binary images, the feedforward network might show an average precision, however, in complex images, the accuracy will not be reliable. CNNs, on the other hand, could be trained to understand sophisticated images with complex pixel dependencies.

Figure 21 illustrate an example of convolutional neural network used on image dataset. As can be seen, CNN architecture is consisted of two main block; feature learning and classification blocks. The feature learning block, which is the particular characteristic of CNNs, functions as a feature extractor by applying convolution filtering operations for template matching. The goal of convolutional layers which are the key component of CNNS, is to detect the presence of a set of features in the input. Each convolutional layer is also followed by a pooling layer which output for example a maximum (in max pooling) within a local neighborhood. That's how the dimentionality reduction happens without missing the bold features. In the first layer, several convolution kernels (filters) could be used to extract feature maps. These feature maps will be then resized or normalized by an activation function. This process can be repeated for several time to obtain new feature maps using a different kernel. finally, the output of this block will be a vector which is built from the last feature map.

The second block is not particular to only CNNs and can bee seen in other types of network for classification purposes. The vector values from the first block are transformed by several activation function to obtain a new vector that corresponds to the output classes.

The $i-th$ element of the output vector denoted the probability that the input image belongs to class i. The probabilities are calculated in the last layer using a logistic activation function for binary classification and softmax for multi-class classification.

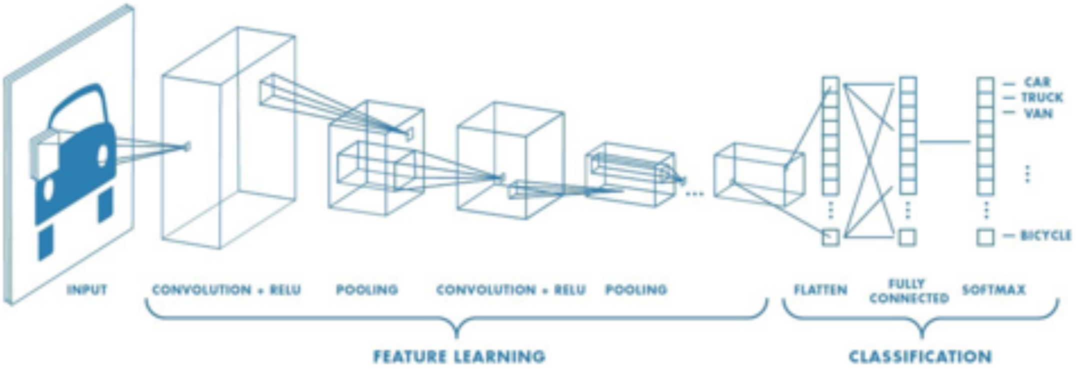


Figure 21: An example of neural network architecture [1]; the CNN architecture is generally consisted of two main block of feature learning and classification. In feature learning block, critical features are extracted using convolutional layers. In the classification block, the feature vector is transformed into a probability vector corresponding to the classes of the output.

As other neural networks, network training happens by minimizing the cross-entropy function between predicted classes and labeled classes and the parameters are calculated using gradient decent and backpropagation.

AlexNet [85] was the first convolutional neural network that achieved a significant success in computer vision problems, which was introduced in 2012. The proposed architecture of AlexNet is 8-layer deep where the last three layers are fully-connected. Since then, a number of complex and advanced convolutional neural networks has been proposed in this area. We may refer the reader into some of these state-of-the-art networks: LeNet [93], VGGNet [147], GoogLeNet [152], ResNet [63], and ZFNet [173].

3.4 Sequence Modeling

The problem of sequence modeling that aims to model, interpret, make predictions of sequential data, is one of the most important challenges in computer vision. Since the temporal modeling is one of the significant aspects of motion analysis (the main goal of this thesis), we try to provide a more comprehensive review of this type of deep neural networks.

3.4.1 Recurrent Neural networks

Recurrent Neural networks (RNNs) [140] are a family of deep neural algorithms that are designed for modeling sequential data or time-series. Like convolutional neural networks that are specialized for processing data with grid-like topology, recurrent neural networks is specially designed for processing the sequence of $x^{(1)}, x^{(2)}, ...x^{(\tau)}$. The output of an RNN network for the input $x^(t)$ is not only obtained through the feedforward process but also is affected by the information from previous time steps. While CNNs could be used in problems with time-series data and be computationally more efficient compare to RNNs, but they are not able to remember contexts or attention to local patterns as RNNs are. Therefore, depending on the problem and priorities, each of these two networks could be applied.

Figure 22 illustrates a simple one layer recurrent neural network. The rolled structure is demonstrated on the left and the unrolled structure over time is on the right. As can be seen, in this type of network the output from previous steps are considered as inputs. From bottom to top, x is the input state, h stands for hidden state, and y denotes the output state. W_xh, W_hh, and W_ho are the weights between input layer and hidden layer, the hidden layer h connection wights, and connection weights between the hidden layer and the output layer. To formulate this, For an RNN with the input sequence of $x = \{x_0, x_1, ..., x_T\}$, a state $h = \{h_0, h_1, ..., h_{T-1}\}$ will be calculated for each neural unit in hidden layers:

$$h_t = f_H(W_{hh}h_t + W_{xh}x_t + b_h) \tag{3.5}$$

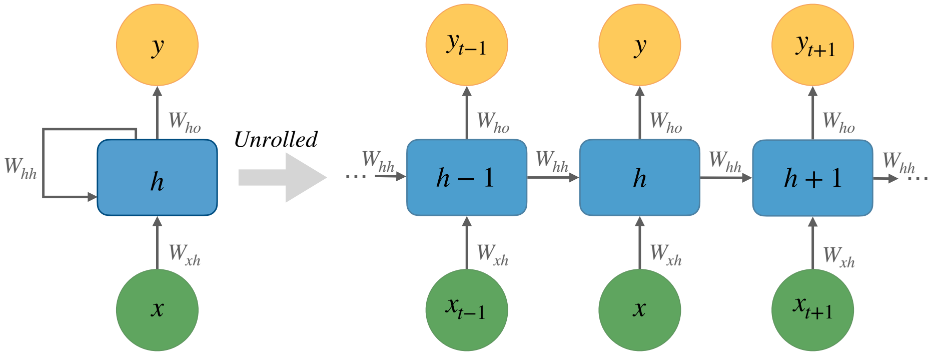


Figure 22: The structure of a simple one layer recurrent neural network; compressed structure is illustrated on the left and unrolled structure on the right.

where b_h is the bias vector, and $f_H(.)$ is the activation function in the hidden layer. Output for each hidden layer $y = \{y_0, y_1, ..., y_{T-1}\}$ will be generated as:

$$y_t = f_O(W_{ho}h_t + b_o) \tag{3.6}$$

where b_o denotes the output layer bias vector, and f_O is the output layer activation function. In deeper recurrent neural networks, many of these recurrent layers could stack up to each others.

Recurrent models, because of the dynamic nature of their representations, are capable of modeling rich temporal evolutions by generating low-latency feature vectors based on previously observed data. Due to RNN's ability to retain memory over time, they have previously shown efficiency in modeling temporal data in different context [57, 58]. However, RNNs suffer from challenges such as vanishing or exploding gradients [64] during back-propagation through time as the sequences of input data get longer. The reason for gradient vanishing could be apparent if we unfold the recurrent neural network in time, as shown in figure 22. Now that the network is unrolled, it seems very similar to a very deep feedforward network. However, in the recurrent network, the gradient of errors which are used to update weights in the back-propagation process, are affected by the recursive multiplication of recurrent weights matrix. In this situation in case of eigen-values less than one for this matrix, the gradients exponentially converge to zero. This process only could be prevented if a saturable function such as sigmoid is used as activation. However, in this case, the network would be able to catch only few steps as the history but not long-term dependencies [56].

3.4.2 Long Short-Term Memory Networks

Long Short-Term Memory Networks (LSTMs) is an extended version of RNNs that are explicitly designed to learn the long-term dependencies in the data. They have been tremendously used to address a large variety of problems and appeared to be performing well on learning long dependencies in the data. The architecture of all recurrent networks including LSTM networks is consisted of one or several repeating neurons. In vanilla RNNs, neurons have a very simple structure (e.g *tanh*) as shown in figure 22.

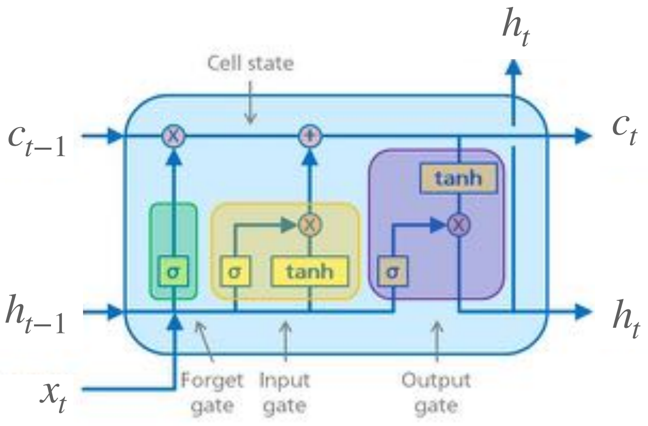


Figure 23: The structure of a single LSTM memory block. c_{t-1} is the cell state from the previous time step, h_{t-1} is the output the previous state, and x is the input. c_t and h_t are the current cell state and output. Figure from [105] with auther's modifications.

LSTM networks, as an advanced RNN architecture, replaces these simple nonlinear units with gated **memory blocks**. The impact of gates is to regulate the units to maintain and access information over long periods of time. As illustrated in figure 23, each memory blocks contains a cell state c_i with three inputs; the input gate i_t, the output gate o_t, and the forget gate f_t. The cell state could be modified by the forget, input, and output gate all placed below it.

The first step in this process starts with the forget gate which basically is a sigmoid function and decides that what part of information is going to be thrown away. The next step is to decide what information we want to store or update in cell state. This process contains two parts; the first part is a sigmoid function which decide what values needs to be updated and the next step is a *tanh* function that create a vector of values to be added to the cell state. Lastly, the memory block should decide what would be the output. This output will be based on our a filtered version of the cell state. This process has two steps either; first, the sigmoid function decides what parts of the cell state will be outputted. Then the cell state go trough the *tanh* function to be compressed between -1 and 1 and the will be multiplied by the output of sigmoid function to only outputs the selected parts [12]. To formulate this:

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{3.7}$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3.8}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + b_c) \tag{3.9}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{3.10}$$

$$h_t = o_t \tanh(c_t) \tag{3.11}$$

where all the W are the connection weights between two neighbor layers, $\sigma(.)$ is the sigmoid function, and all the b are biases.

LSTM networks have been tremendously used to address sequence modeling problems and shown shown state-of-the-art performance. What was describe, so far, is a normal LSTM. However, there are other proposed LSTM networks which are a little bit different. In one of the most popular LSTM network proposed by [53], gates are able to look at the cell state. Therefore, the output of LSTM can be calculated as:

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{3.12}$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3.13}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}c_{t-1} + b_c) \tag{3.14}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{3.15}$$

$$h_t = o_t \tanh(c_t) \tag{3.16}$$

where all the W are the connection weights between two neighbor layers, $\sigma(.)$ is the sigmoid function, and all the b are biases.

4.0 HUMAN AUTHENTICATION

4.1 Introduction

In this chapter we present an automatic deep neural network framework which is able to model human motions from visual input. We further investigates the potential capability of motion sequences as an active biometric verification technique. We approach the problem from a data-driven perspective to figure out to what extent we can authenticate individuals based on their movement patterns. We propose a deep learning approach using OpenPose and a three-layer LSTM recurrent neural networks (RNN) to verify the subject against others based on their motion signature. We use the person-specific video dataset introduced by [54] which contains 144 hours of data from 10 speakers with diverse set of backgrounds including television show hosts, university lecturers and televangelists. They cover a large range of topics such as chemistry, history of Rock music, current news, reading bibles and Qur'an.

The proposed method focuses on the design and implementation of a fully automated human authentication network as the video dataset is not annotated and acquiring human annotation for large amounts of video is not feasible. OpenPose library [30] is used to extract 33 2D skeletal body keypoints and motion features are calculated as of changes in angle and magnitude between body joints in consecutive frames. We use recurrent neural network (RNN) with LSTM to learn long-term dependencies within our data. Since multi-layers RNNs can extract more rich semantics features, we design three-layer LSTM to realize this task. Lastly, we use Dropout and L2 regularization to avoid overfitting.

4.2 Dataset

In this work, we use *speaker-specific gesture dataset* [10], which was first specifically tailored by [54] to find the connection between conversational gesture and speech for further prediction of human gesture from audio. The dataset originally contains 144 hours of in-the-wild footage of 10 gesturing single-speaker from different backgrounds including 5 television show hosts, 3 university lecturers, and 2 televangelists as shown in figure 24.

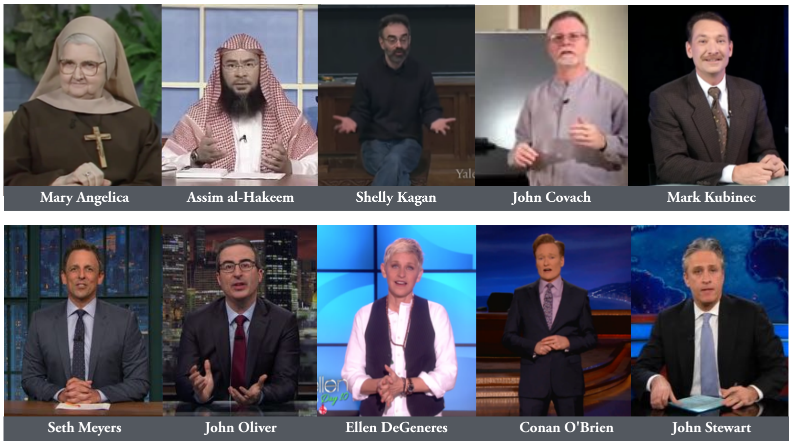


Figure 24: The speaker-specific gesture dataset containing video data of 2 televangelists (Mary Angelica and Assim Al-Hakeem), 3 university lecturers (Shelly Kagan, John Covach, Mark Kubinec), and 5 television show hosts (Seth Meyers, John Oliver, Ellen DeGeneres, Conan O'Brien, and John Stewart).

The speakers cover a large range of topics such as chemistry, history of Rock music, current news, reading bibles, and Qur'an. The set of speakers are deliberately chosen by the authors as they could find hours of clean single-speaker footage for each person available on YouTube.

The authors further used out-of-the-box face recognition and pose detection systems to split each video into intervals that only contains the speaker in the frame where all detected keypoints are visible. The final dataset contains 60,000 such intervals with an average length of 8.7 seconds and a standard deviation of 11.3 seconds which sums up to 144 hours of video.

In this research, we deliberately selected the speaker-specific gesture dataset due to three main reason; first, it only contains one speaker in all intervals; second, it is collected from in-the-wild footage of subjects; and third, all the subjects are performing the same task which is talking to audience. We further manage to use 50 hours subset of the speaker-specific gesture dataset that contains intervals with more than 10 seconds of length.

4.3 Movement Data

To estimate individuals' pose over time we use OpenPose for visual sequence of data at a rate 15 fps. OpenPose enables us to train over a large amount of data. Among 135 keypoints detected by OpenPose, we use the 33 corresponding to nose, neck, shoulders, elbows, wrists, fingers (10 keypoints), eyes, and ears as shown in figure 27. Key positions of eyes, ears, and nose are used inside the set of body keypoints as they have information about head orientation. However, facial signals like lips movement, eye-gaze, eyebrows, jaw and cheek motions are ruled out to avoid any possible impact influenced by facial expression. After obtaining the skeletal keypoints, each keypoint is normalized by subtracting the per-individual mean, divided by the standard deviation. Motion features are calculated between consecutive frames for each keypoint defined by the change x and y coordinations.

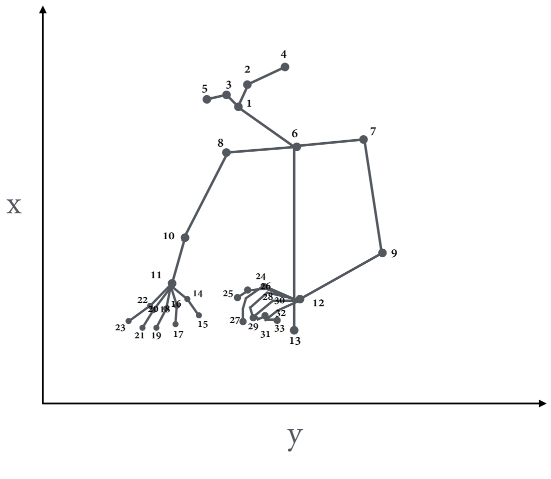


Figure 25: Skeletal keypoints used for pose detection in each frame.

It is worth noting that as shown in supplementary material of [54], these keypoints are not ground truth annotations obtained from human observers but rather pseudo ground truth generated from pose estimation machine. The distance between OpenPose estimation and the mean of human annotations is small enough to be used in our authentication framework.

4.3.1 Architecture and Implementation Details

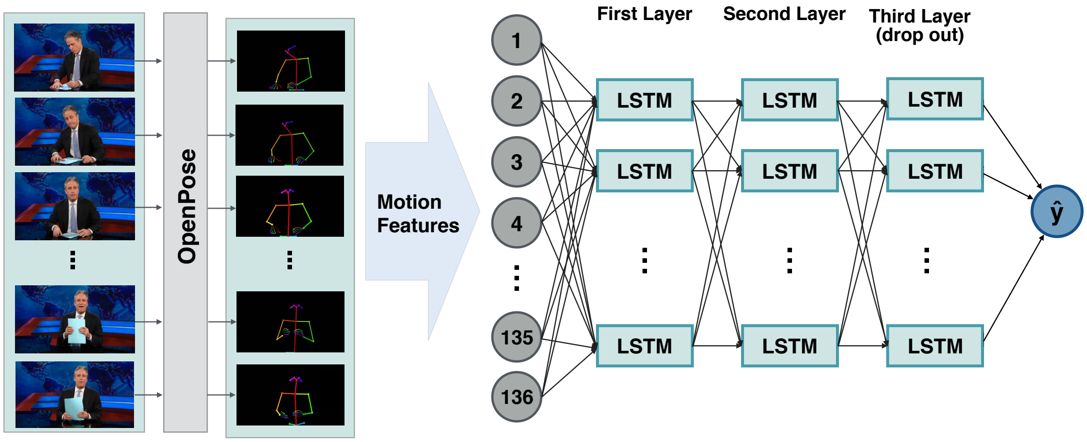


Figure 26: **Architectural overview of proposed authentication network.** *OpenPose* is used to detect individuals' pose over time. Motion features are calculated by the change in keypoints coordination between consecutive frames. Given an input sequence of motions, the 3-layer LSTM recurrent neural network outputs whether the input belongs to the subject of study or not.

In order to authenticate individuals based on their motion signatures, we implement a recurrent neural network with LSTM cells to find the long term dependencies in our data. Details about LSTM recurrent neural network has been discussed in chapter 3. The framework of the proposed model is presented in Figure 26. Our model is consisted of one input layer, three hidden layers and a sigmoid classifier considering that multi-layers RNNs can extract more rich motion features. For the input layer, we used joints coordination combined with the Δx and Δy as a feature vector of 132 dimension:

$$F_n = [x_1, y_1, \Delta x_1, \Delta y_1, ..., x_{132}, y_{132}, \Delta x_{132}, \Delta y_{132}]$$

where n demonstrate the index of frame in the video. Δx and Δy also corresponds to the motions related to keypoints between consecutive frames. The hidden layers include two regular layers with 256 LSTM cells with *tanh* activation function and one dropout layer with 0.5 rate to boost the performance of network. Furthermore, L2 regularization is applied to our network to avoid overfitting. The output of the model is obtained from a sigmoid classifier (a dense layer with a sigmoid activation function), projecting whether the input motion sequence belongs to the subject or not.

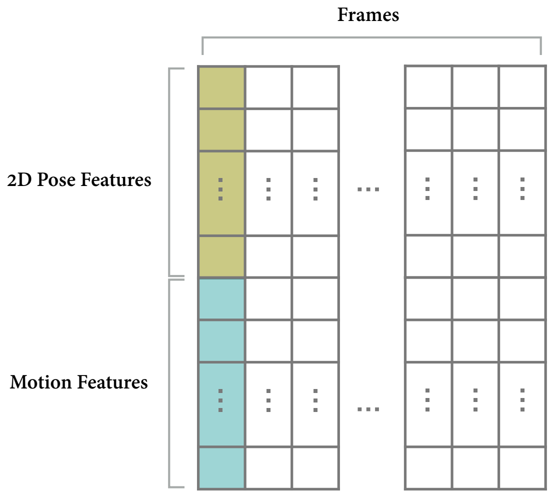


Figure 27: The two kinds of features from a sequence are concatenated to form input feature vectors.

To formulate the problem precisely, suppose F is the input motion sequence to our network, and we want to see if the input F belongs to subject s or not. The output of proposed network would be '0' meaning that the input sequence does not belong to subject s or would be '1' meaning that the user is authenticated. Since the problem could be considered as a binary classification, we use sigmoid function as the activation function in the output layer:

$$\sigma(s) = \frac{1}{1 + e^{-s}} \tag{4.1}$$

Finally, in order to measure the error between true values and the predictions, we use binary cross-entropy loss function. We also use Adam optimization algorithm with the default learning rate of 0.001 to enhance the fitting ability. We train and implement our networks using Lasagne on an NVIDIA GPU TESLA K80.

4.4 Experiments

We show that our method successfully authenticates the individuals in our dataset based on their movement patterns and quantitatively outperforms baselines.

4.4.1 Baselines

We compare our methods to several other conventional approaches. All the classifiers are trained in a fully supervised way using 80% of data for training and 20% of data for validating and testing.

Support Vector Machine (SVM) Due to high classification performance of SVM, it is used in a wide range of classification problems including human authentication/identification [72, 68, 164]. SVM attempts to reach the increase the classification accuracy by creating hyper-planes that maximize the margins between classes [120]. By minimizing the cost function, SVM reaches the optimal feasible accuracy. In this work, we use a non-linear binary SVM classifier with *sigmoid* kernel to authenticate speakers based on their motions.

Decision Trees Another method exploited in recognition studies is decision tree [144, 88]. A decision tree classifies inputs by sorting them down a tree-like graph from the root to leaves, with the leaf node representing the label of classification. In a recursive process, each node in the graph acts as a test case for some attributes and each edge descending corresponds to one of the possible answers.

Random Forests We further compare our authentication model against a Random Forest model that generally is used in several studies for human authentication/identification [83, 141, 43]. Random Forest is an ensemble learning method used in both classification and regression. The key difference between random forest and decision tree is that a decision tree is built on the entire dataset and uses all the variable of interest while a random forest generates multiple decision trees based on random selection of observations and specific variables and then average the resits.

4.4.2 Quantitative Evaluation

As the performance of models might differ depending on the input sequence, we randomly chose 360 test sequence with 10 second duration and compare our model to all other baselines. Table 10 represents the average accuracy obtained by each model. To evaluate the average accuracy of each classifier, we ran 10 repetitions of the particular experiments on a random selection of training and test sets. Table 3 demonstrates the precision and recall obtained from the prediction of a single experiment with the same training and test set for each classifier.

As shown in Table 10, among all conventional approaches considered in this study, SVM performance was generally the lowest while Random Forest and Decision tree showed better performance. However, they still were not able to differentiate all speakers properly. The proposed LSTM neural network outperformed other classifiers obtaining the highest average accuracy of 94.07%.

To validate that the subjects are statistically discriminant, student's t-test is applied.

The p-value calculated for our models is less than 0.05 representing the statistically significance of this method. Figure 45 displays the results distribution obtained by different classifiers. As shown, the LSTM achieved the superior results with a higher median and accuracy over other classifiers.

4.5 Conclusion

We presented a deep learning approach for human authentication based on body motions. Our network is consisted of a pose estimation machine (*OpenPose*) and a three layer LSTM RNN. OpenPose detects body keypoints in visual sequences and LSTM network acts as the classifier for the authentication task. In the path of architecture design, we considered different classification methods and selected LSTM as the most capable option due to its lower classification error and its capability to detect long dependencies. We ran our model over *Speaker-specific gesture* dataset and measured accuracy of authentication for 10 speakers. The results demonstrate robust high accuracy of predictions, confirming the strong link between human body motions and their identity. This approach works with monochrome or RGB video stream and does not require custom hardware like Lidar/Kinects, motion or EMG sensors on the body. The test results of our model shows average authentication accuracy of 94.07%.

Despite the strong performance of the proposed model, the current network presented in this paper is only capable of authenticating single-person in each test sequence and adding multi-person authentication requires more work. However, we see this work as a preliminary step toward active biometric authentication that does not require the direct communication and cooperation of subjects.

Table 1: Precision and Recall of four classification methods on a test set of 360 sequence.

Name	SVM		Decision Tree		Random Forest		LSTM	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Almaram	0.79	0.38	0.67	0.76	0.89	0.82	0.92	0.93
Angelica	0.44	0.63	0.72	0.78	0.82	0.87	0.96	0.92
Conan	0.74	0.91	0.74	0.89	0.87	0.87	0.95	0.93
Covach	0.63	0.82	0.68	0.62	0.86	0.82	0.96	0.94
Ellen	0.39	0.48	0.76	0.63	0.81	0.77	0.95	0.93
Kagan	0.92	0.91	0.75	0.64	0.89	0.83	0.96	0.98
Kubinec	0.79	0.74	0.76	0.74	0.82	0.86	0.97	0.98
Oliver	0.49	0.50	0.68	0.73	0.79	0.64	0.95	0.95
Meyers	0.44	0.41	0.58	0.61	0.62	0.71	0.97	0.93
Stewart	0.49	0.58	0.65	0.69	0.71	0.76	0.97	0.94
Avg	0.61	0.63	0.69	0.70	0.80	0.79	**0.95**	**0.94**

Table 2: Average accuracy of each classifier on 10 repetition.

Classifier	Accuracy (%)
SVM	64.91
Decision Tree	69.29
Random Forest	79.13
LSTM	**94.07**

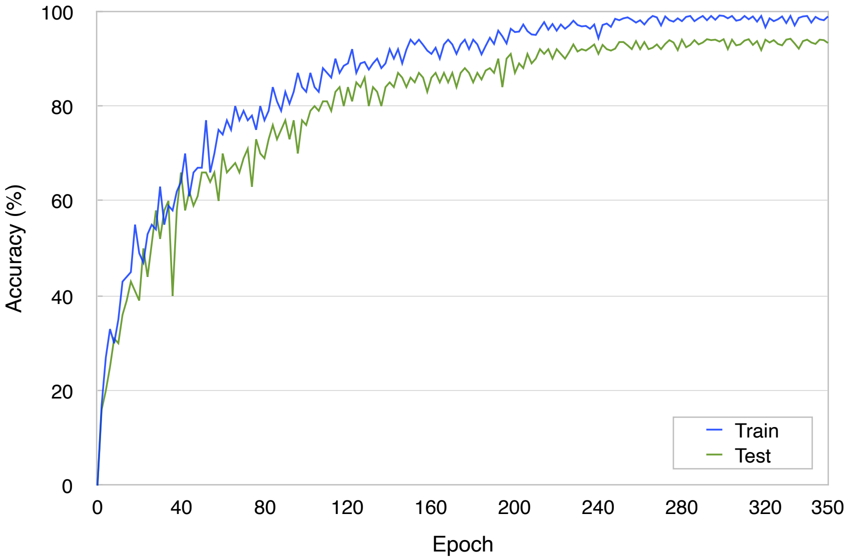


Figure 28: Training and development accuracies per epoch. The model is trained for 350 epochs with a batch size of 256 samples from one run for one subject - Almaram.

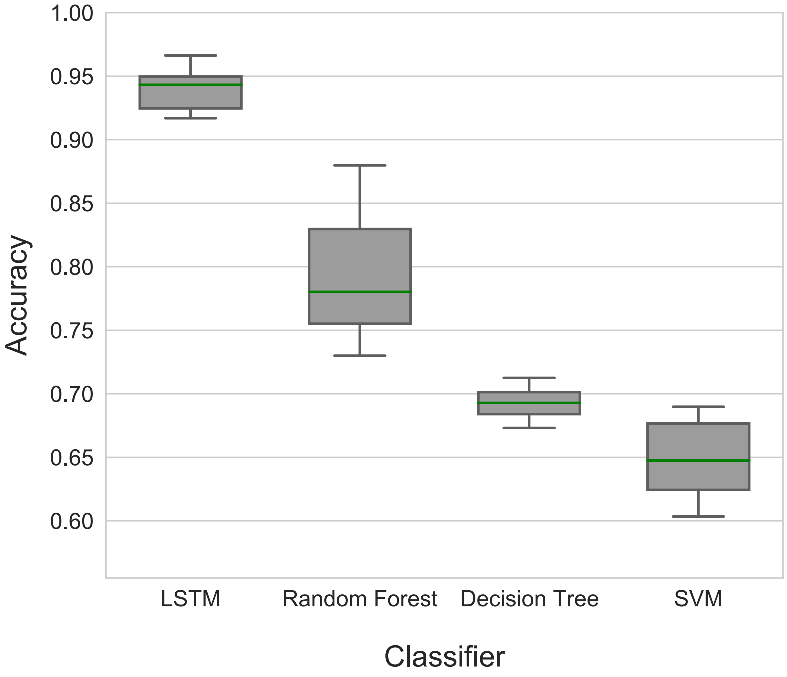


Figure 29: The comparison of accuracy between different classifiers for 10 repetition.

5.0 HUMAN IDENTIFICATION

5.1 Introduction

In the early chapters of this work, we discussed the objectives of this research, potential applications, previous works done in the domain, and existing deep learning models for approaching human motion analysis problems. In chapter 4, as the preliminary step of our research, we focused on design and implementation of a deep learning model which is able to efficiently learn human motion representations from noisy motion sequences and further authenticate individuals based on their motion signatures. The test results of the proposed authentication model over a set of 10 speakers showed an average accuracy of 94.07%. While the motion authentication model proposed in chapter 4 is very practical in many applications to distinguish the user from others, but it still is not applicable to applications that requires subject identification within a set of people.

For this purpose, in this chapter, as another step toward understanding human motion, we intend to develop a deep learning framework to identify individuals based on their motion patterns in videos. We use OpenPose to detect human body keypoints in visual sequences collected from a *Speaker-specific gesture* dataset explained in section 4.2 and extract temporal motion features between consecutive frames (see section 4.3 for more details). Motion features are fed into a 3-layer LSTM network which outputs the probability distribution vector corresponding to identities. We perform quantitative evaluations including comparing with different conventional machine learning algorithms used for biometric identification in similar works. Finally, we demonstrate that proposed framework outperforms other approaches by achieving the highest average accuracy of 92.62%.

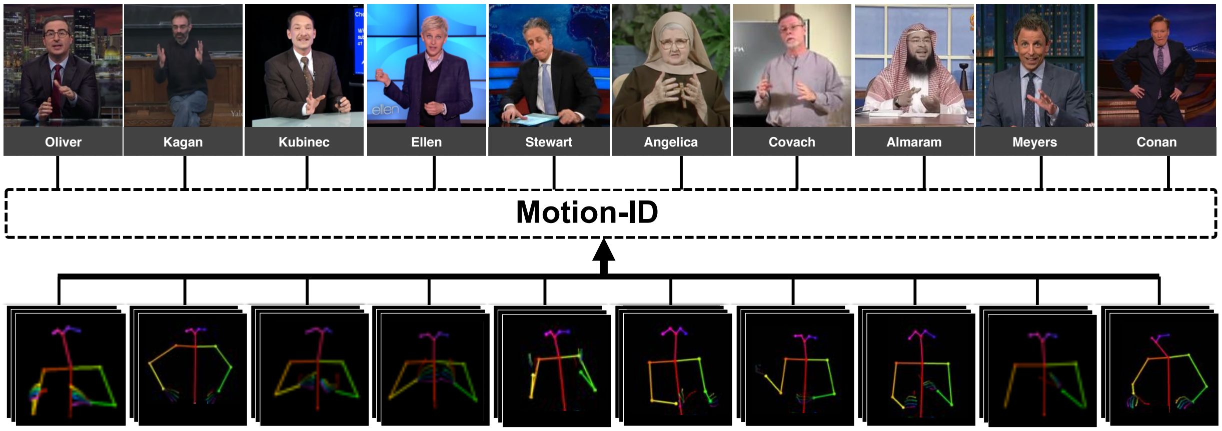


Figure 30: **General overview of Motion-ID.** In this chapter, we present a deep learning approach to identify 10 speakers from their motions in Youtube videos collected by [54]. Speakers poses are estimated utilizing [30] over a period of time in combination with an LSTM recurrent neural network to predict speakers identities.

5.2 Architecture and Implementation Details

To address this problem, a recurrent neural network with LSTM cells is implemented to find the long term dependencies in our data. The framework of the proposed model is presented in Figure 31. Our model consists one input layer, three hidden layers and one softmax considering that multi-layers RNNs can extract more rich motion features. For the input layer, we used joints coordination combined with the Δx and Δy as a feature vector of 132 (4 features for each 33 keypoints) dimension:

$$F_n = [x_1, y_1, \Delta x_1, \Delta y_1, ..., x_{132}, y_{132}, \Delta x_{132}, \Delta y_{132}]$$

where n demonstrates the index of frame in the video. Δx and Δy also correspond to the motions related to keypoints between consecutive frames. The hidden layers include two regular layers with 256 LSTM cells with tanh activation function and one dropout layer with 0.5 rate to boost the performance of network. Furthermore, L2 regularization is applied to our network to avoid overfitting. The output of the model is obtained from a softmax layer (a dense layer with a softmax activation function), yielding a class probability distribution corresponding to subjects identities.

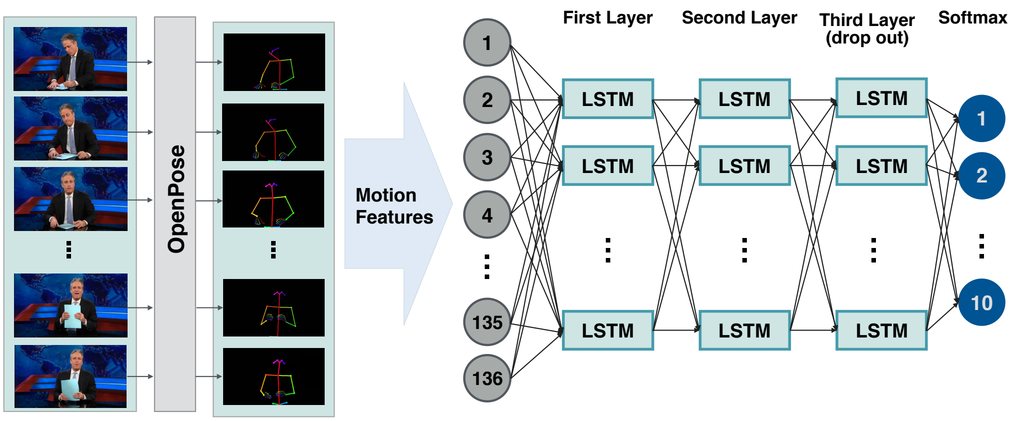


Figure 31: **Architectural overview of Motion-ID network.** *OpenPose* is used to detect individuals' pose over time. Motion features are calculated by the change in keypoints coordination between consecutive frames. Given an input sequence of motions, the 3-layer recurrent neural network with LSTM cells outputs the probability distribution vector corresponding to identities.

To formulate the problem precisely, suppose F is the input motion sequence to our network, and $s = \{s_1, s_2, ..., s_n\}$ represents n speakers considered in this study. The output of proposed network would be an n-dimensional vector $o = \{o_1, o_2, ..., o_n\}$ where $o_i = p(s_i|F)$ - the possibility that the input sequence F belongs to speaker s_i - and they can be calculated as:

$$p(s_i|F) = \frac{e^{o_i}}{\sum_{k=1}^{n} e^{o_k}} \tag{5.1}$$

where n represents the number of classes. Then the maximum possibility will be associated to input sequence F as the final identity:

$$s = argmax\{o_i | 1 \leq i \leq n\} \tag{5.2}$$

The model is trained for 350 epochs with a batch size of 256 samples from one run. The hyperparameters were chosen empirically according to which values yielded the best results for the task. Finally, we use cross-entropy loss function to measure the error between predictions and the true values and Adam optimization algorithm with the learning rate of 0.001 to enhance the fitting ability. We train and implement our networks using Lasagne on an NVIDIA GPU TESLA K80.

5.3 Experiments

We compare our methods to several other conventional approaches described in section 4.4.1 including SVM, decision trees,and random forests. All the classifiers are trained in a fully supervised way using 80% of time series data for training and 20% of them for validating and testing. The percentages refer to hours. We show that our method successfully identifies individuals based on their movement patterns and quantitatively outperforms baselines.

5.3.1 Quantitative Evaluation

As the performance of models might differ depending on the input sequence, we randomly chose 360 test sequences with 10 second duration and compare our model to all other

Table 3: Precision and Recall of four classification methods on a test set of 360 sequence.

Name	SVM		Decision Tree		Random Forest		LSTM	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Almaram	0.77	0.73	0.74	0.72	0.81	0.85	0.91	0.91
Angelica	0.74	0.90	0.71	0.87	0.87	0.86	0.92	0.90
Conan	0.91	0.91	0.72	0.62	0.87	0.82	0.93	0.98
Coach	0.47	0.49	0.65	0.71	0.78	0.63	0.92	0.93
Ellen	0.40	0.62	0.71	0.76	0.80	0.86	0.91	0.90
Kagan	0.36	0.47	0.75	0.61	0.81	0.76	0.92	0.91
Kubinec	0.77	0.37	0.65	0.74	0.89	0.81	0.91	0.94
Oliver	0.42	0.40	0.57	0.58	0.61	0.70	0.94	0.90
Meyers	0.47	0.57	0.63	0.68	0.70	0.75	0.96	0.92
Stewart	0.61	0.81	0.66	0.60	0.85	0.81	0.94	0.94
Avg	0.59	0.62	0.67	0.68	0.79	0.78	**0.92**	**0.92**

baselines. Table 4 represents the average accuracy obtained by each model. To evaluate the average accuracy of each classifier, we ran 10 repetitions of the particular experiments on a random selection of training and test sets. Table 3 demonstrates the precision and recall obtained from the prediction of a single experiment with the same training and test set for each classifier.

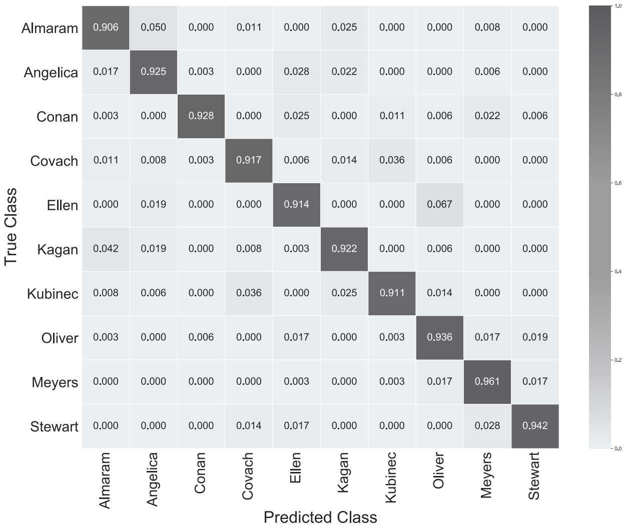
Table 4: Average accuracy of each classifier on 10 repetition.

Classifier	Accuracy (%)
SVM	56.58
Decision Tree	64.13
Random Forest	77.38
LSTM	**92.62**

As shown in Table 4, among all conventional approaches considered in this study, SVM performance was generally the lowest while Random Forest and Decision tree showed better performance. However, they still were not able to differentiate all speakers properly. The proposed LSTM neural network outperformed other classifiers obtaining the highest average accuracy of 92.62%.

Figure 32 displays the confusion matrix related to the proposed model. For every speaker in the *speaker-specific gesture* dataset our model achieves accuracy over 92% with the least accuracy of 90.8% for Almaram and the highest accuracy of 96.0% for Meyers. However, differentiating Ellen and Oliver resulted in the most confusion for our network.

To validate that the subjects are statistically discriminant, student's t-test is applied. The p-value calculated for our models is less than 0.05 representing the statistically significance of this method. Figure 45 displays the distribution of the results achieved by different classifiers. As shown, the LSTM achieved the superior results with a higher median and accuracy over other classifiers.



	Almaram	Angelica	Conan	Covach	Ellen	Kagan	Kubinec	Oliver	Meyers	Stewart
Almaram	0.906	0.050	0.000	0.011	0.000	0.025	0.000	0.000	0.008	0.000
Angelica	0.017	0.925	0.003	0.000	0.028	0.022	0.000	0.000	0.006	0.000
Conan	0.003	0.000	0.928	0.000	0.025	0.000	0.011	0.006	0.022	0.006
Covach	0.011	0.008	0.003	0.917	0.006	0.014	0.036	0.006	0.000	0.000
Ellen	0.000	0.019	0.000	0.000	0.914	0.000	0.000	0.067	0.000	0.000
Kagan	0.042	0.019	0.000	0.008	0.003	0.922	0.000	0.006	0.000	0.000
Kubinec	0.008	0.006	0.000	0.036	0.000	0.025	0.911	0.014	0.000	0.000
Oliver	0.003	0.000	0.006	0.000	0.017	0.000	0.003	0.936	0.017	0.019
Meyers	0.000	0.000	0.000	0.000	0.003	0.000	0.003	0.017	0.961	0.017
Stewart	0.000	0.000	0.000	0.014	0.017	0.000	0.000	0.000	0.028	0.942

Figure 32: Speaker-specific gesture dataset confusion matrix calculated based on a test set of 360 sequences. The proposed LSTM network achieves over 92% accuracy for each person identification.

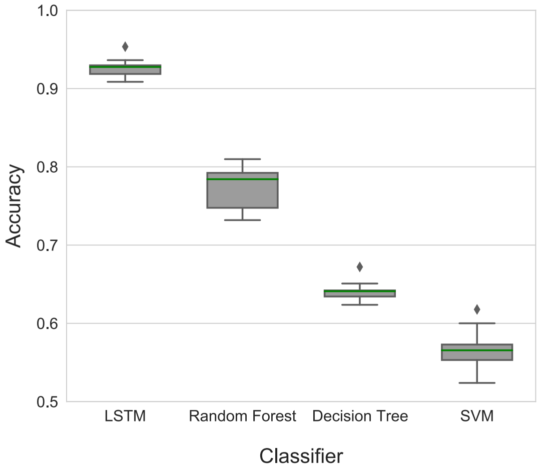


Figure 33: The comparison of accuracy between different classifiers for 10 trials each. LSTM classifier is 3-layer with dropout and L2-regularization.

We further test our model over different number of consecutive frames to evaluate accuracy variation. The results are shown in figure 34. After a certain number of frames, computational costs overweights the incremental accuracy as it gets saturated. In Speaker-specific dataset, this number is around 150 frames where the model is capable to predict the identity of the subjects with more than 90% accuracy.

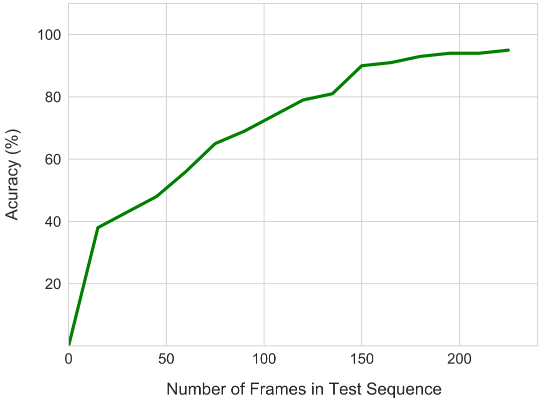


Figure 34: The average accuracy vs numbers of test frames per speaker.

5.4 Conclusion

In this chapter, we focused on human identification based on body movements. Meaning that given a sequence of body movements, we intend to relate the motion sequence to the individual's identity within a 10-person dataset (*Speaker-specific gesture* dataset). To this purpose, our proposed method focuses on the design and implementation of a fully automated human identification network as the video dataset is not annotated and acquiring human annotation for large amounts of video is not feasible. *OpenPose* is used to extract 33 2D skeletal body keypoints and motion features are calculated as of changes in x and y coordinations of body joints in consecutive frames. We use recurrent neural network (RNN) with LSTM to learn long-term dependencies within our data. Since multi-layers RNNs can extract more rich semantics features, we design three-layer LSTM to realize this task. In the path of architecture design, we considered different classification methods and selected LSTM as the most capable option due to its lower classification error and its capability to detect long dependencies. The results demonstrate robust high accuracy of predictions, confirming the strong link between human body motions and their identity. This approach works with monochrome or RGB video stream and does not require custom hardware like Lidar/Kinects, motion or EMG sensors on the body. The test results of our model over a set of 10 speakers shows average identification accuracy of 92.62%.

Despite the strong performance of the proposed model, there are still some limitations. This model is trained to predict the individual identity using 10 speakers in-the-wild footage from *Speaker-specific gesture*. Although these 10 speakers are chosen from different backgrounds to reduce any possible bias, the accuracy might vary with subjects, number of subjects, or speech topics changing. Also, the current network presented in this chapter requires high computational power and time to be trained which might cause some limitations when it comes to potential applications. In future chapters, we will focus on addressing these challenges, and introduce a generative probabilistic framework that is able to efficiently identify individuals based on their motion sequence in a low-computational power setting.

6.0 ANALYZING THE FRAMEWORK ROBUSTNESS

In chapter 4 and 5, we proposed two deep learning framework that were successfully able to authenticate and identify human based on their body movements. Although deep neural networks showed superior performance in many applications of artificial intelligence as well as biometric recognition, however, one of the discussions around vision-based deep learning approaches is that the precision of outputs relies heavily on the image quality as well as brightness change. Therefore, in real world setting, it is important to make sure that small changes in input testing data, does not yield significant loss to the performance of the framework.

The goal of this chapter is to discuss the robustness of our proposed motion recognition framework in previous chapters to a set of disturbance that might impact the samples in practice, such as random noises and brightness manipulations. We further compare our results to one of state of the art deep learning facial recognition systems, *DeepFace* [153], as a baseline to our work.

Since this work is one of the earliest steps in human authentication based on body movements from visual data, and to the best of our knowledge, there is no baseline for quantitative comparison of our robustness results, we compare our results to one of state of the art deep learning facial recognition systems, *DeepFace* [153], as a baseline to our work. DeepFace is lightweight face recognition framework implemented in Python that already reached and passed the human level accuracy in face recognition and verification. The accuracy reported for DeepFace on LFW dataset [66] for human verification equals to 97.35%. On our dataset, the obtained accuracy for DeepFace is equal to 96.92%.

We hope that this part of our study will contribute on shedding light on open research challenges in the domain of deep learning models robustness.

6.1 Random Noise

During the training process of a neural network, the goal is obtaining the best accuracy. However, most of the time generalization ability of the neural network does matter, meaning that how does the model performs on the unseen real-world dataset.

Majority of time, the model performance is satisfying. However, there are some situations that the network is trained on a huge dataset and achieves state of the art training accuracy, but once it is tested on real-world testing dataset, it doesn't generalize well on new unseen datasets. One of the main reasons in these cases is that the real-world testing data are not clean as the training dataset. In this section we want to evaluate our proposed model accuracy testing them with noisy dataset. The reason for such experiment to figure that how the framework accuracy would be affected in case of noisy data recorded by different cameras.

For this purpose, we add *Gaussian noise* to our test dataset. Gaussian noise is a statistical noise with a Gaussian distribution probability function. The noise magnitude of Gaussian noise is depending on the standard deviation (σ) value. Figure 35 demonstrates the image results after applying Gaussian noise with different σ ranging from noise-free to to σ of 5.5.

Table 5 demonstrates the accuracy obtained by our framework versus Deep face once the Gaussian noise is added to image inputs. The experiment is repeated with different Gaussian standard deviation values as the amount of noise added is a configurable hyperparameter. As can be seen, little noise has less effect, whereas too much noise makes accuracy drop in both networks. However, The overall accuracy drop in DeepFace is more compare to our network when noises magnitude gets bigger. This could be explained as a result of missing some facial information due to adding noise while pose information are still more detectable.

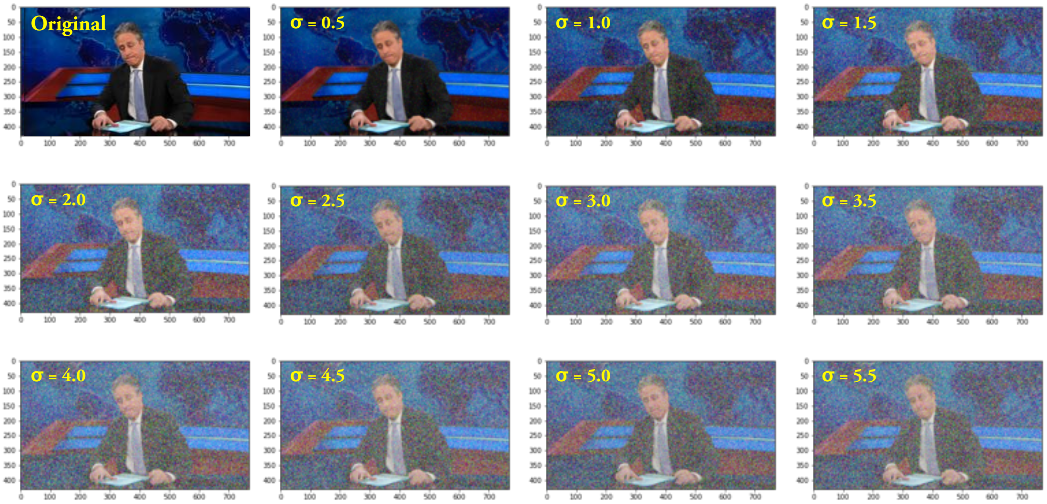


Figure 35: Test images corrupted by Gaussian noise with different magnitude of σ from 0 to 5.5.

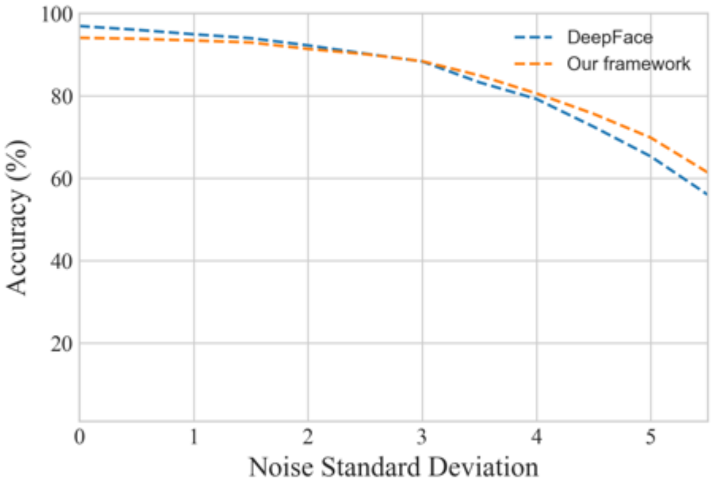


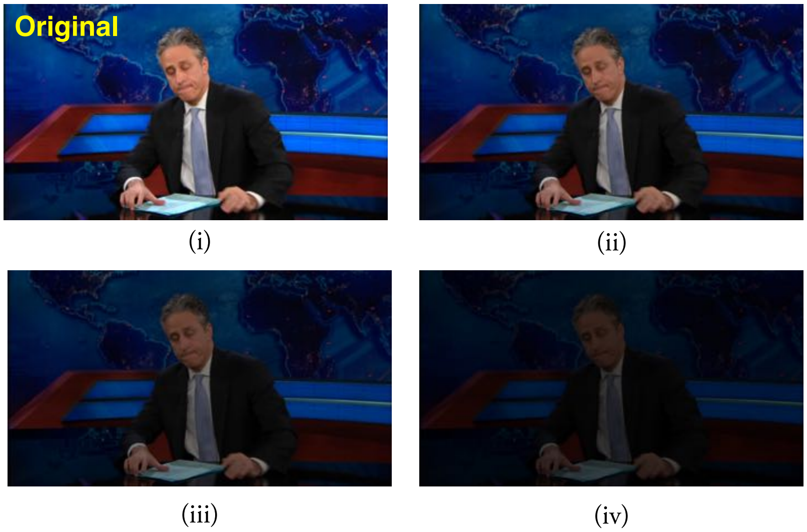
Figure 36: Plot of accuracy changes with different noise level.

Table 5: Quantitative comparison of different Gaussian noise level to input images, and motion signals.

Noise Level	Proposed Framework	DeepFace
$\sigma = 0$ - Original	0.9407	0.9692
$\sigma = 0.5$	0.9384	0.9602
$\sigma = 1.0$	0.9342	0.9493
$\sigma = 1.5$	0.9294	0.9398
$\sigma = 2.0$	0.9137	0.9223
$\sigma = 2.5$	0.9011	0.9023
$\sigma = 3.0$	0.8839	0.8834
$\sigma = 3.5$	0.8327	0.8327
$\sigma = 4.0$	0.8057	0.7923
$\sigma = 4.5$	0.7562	0.7249
$\sigma = 5.0$	0.6984	0.6529
$\sigma = 5.5$	0.6137	0.5597

6.2 Brightness

In this section we intend to understand how lighting might impact the precision of our framework. For this purpose, using Pillow (PIL) library in python, we adjust the brightness of our testing dataset to generate augmented samples with different brightness levels. Adjusting the brightness, increases or decreases the pixel value evenly across all channels for the entire image to increase or decrease the brightness. The brightness can be adjusted using *enhance*() method developed in Pillow library by changing the enhancer factor. While a factor of 1 gives original image, making the factor towards 0 makes the image black, and factors greater than 1 brightens the image. Figure 37 demonstrates the darkened results with enhancer factors of 0.25, 0.5, 0.75, and 1.0. Figure **??** shows the brightened samples with enhancer factors of 1.5, 2.0, 2.5, 3.0, and 3.5.



(i) (ii)

(iii) (iv)

Figure 37: Adjusting the brightness value by decreasing the pixel value evenly across all channels for the entire image. The brightness level are from enhancer factors of (i) 1.0 which is the original image, (ii) 0.75, (iii) 0.5, and (iv) 0.25. An enhancer factor 1 is the original image while moving toward 0 darkens the image and an enhancer factor equal to 0 makes the image black.

Figure 38 shows extracted pose in different lighting via OpenPose network. As can be seen, the network is not affected in samples with 0.75 and 0.5 lighting factor at all, while in 0.25 lighting, the subjects left fingers are not correctly detected. In 0.1 lighting factor, however, the OpenPose fails to detect more details (figure 40).

Figure 39 demonstrates the DeepFace performance on darkened test images with enhancer factor equal to 0.75, 0.5, and 0.25. In these lighting, DeepFace was able to successfully verify the subjects. We further move forward and compare our network performance to DeepFace in lower lighting with enhancer factor equal to 0.1. Figure 40 shows the DeepFace performance (left) vs OpenPose performance (right) on the darkened images. As shown in

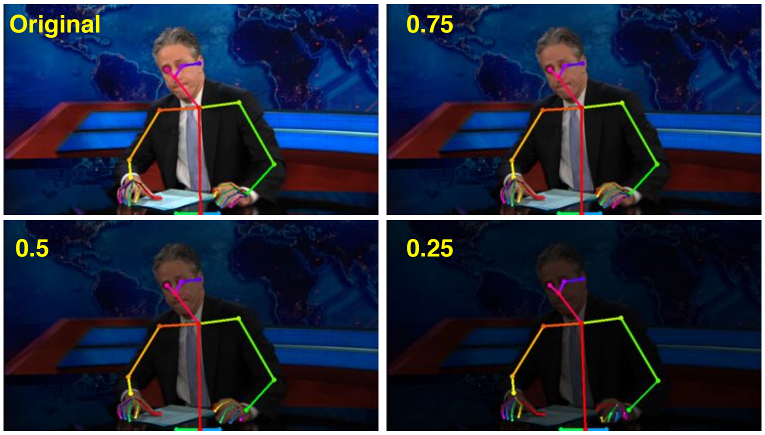


Figure 38: OpenPose performance on darkened augmented image samples with enhancer factors of 0.0 (original image), 0.75, 0.5, and 0.25.

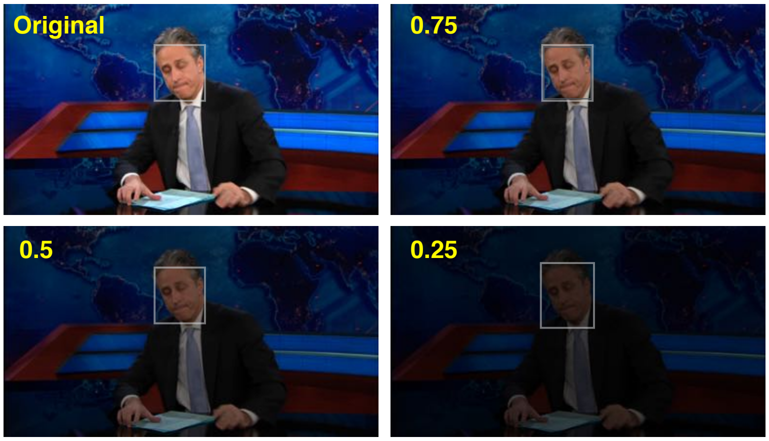


Figure 39: DeepFace performance on darkened augmented images with enhancer factor of 0.0 (original image), 0.75, 0.5, and 0.25.

this figure, DeepFace fails to detect the subject while OpenPose was able to not only detect the subject but also detect some part of body pose.

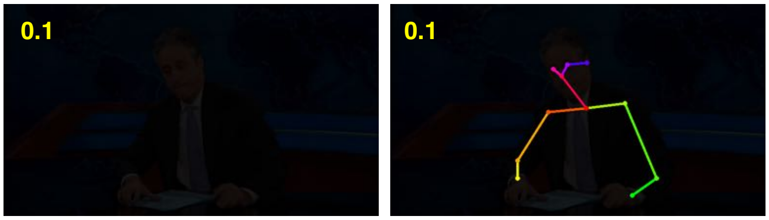


Figure 40: DeepFace (on the left) vs OpenPose (on the right) performance in brightness with enhancer factor equal to 0.1.

Table 6: Average accuracies obtained in different brightness levels for the proposed framework and DeepFace.

Brightness	Proposed Framework	DeepFace
1.0 - Original	0.9407	0.9692
0.75	0.9311	0.9604
0.5	0.8993	0.9398
0.25	0.8536	0.8813
0.10	0.7948	0.5829

Table 6 represent the accuracy obtained by our framework compared to DeepFace for human verification in different lighting. As results plotted in figure 41 demonstrate, the accuracy drops in both frameworks as brightness decrease. In brightness greater than 0.25 DeepFace slightly outperforms our framework accuracy. In lighting with enhancer factor of 0.25, our framework accuracy drops by 8.71% while the DeepFace accuracy decreases by 8.79%. In lower brightness, our proposed framework is significantly performing better.

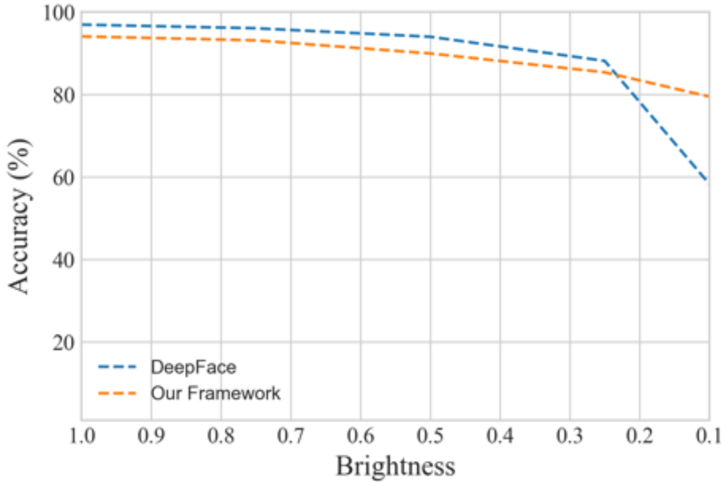


Figure 41: Plot of accuracy changes with lighting for DeepFace verification and the proposed motion verification framework.

We further evaluate our framework with brightened augmented samples to see how brightening images would impact the network performance to authenticate subjects. Figure 42 demonstrates brightened test samples of different enhancer factors from 1.0 which is the original image to 3.5. The accuracy obtained by each network on brightened test sets of different enhancer factor are reported in table 7. As shown in figure 43, overall, the accuracy in both networks drops as brightness level increases in test images. This might be due to loosing some visual information and details when the pixel values get high.

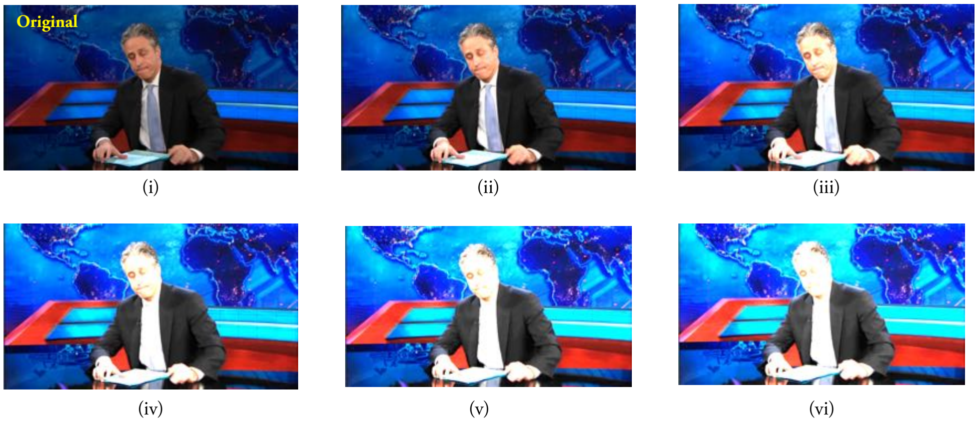


Figure 42: Increasing the brightness value by increasing the pixel value evenly across all channels for the entire image. The brightness levels are from enhancer factors of (i) 1.0 which is the original image, (ii) 1.5, (iii) 2.0, (iv) 2.5, (v) 3.0, and (vi) 3.5.

Table 7: Average accuracy of each deep learning framework in different brightness levels.

Brightness	Proposed Framework	DeepFace
1.0 - Original	0.9407	0.9692
1.5	0.9482	0.9798
2.0	0.9273	0.9537
2.5	0.8449	0.8174
3.0	0.7623	0.7593
3.5	0.7137	0.6783

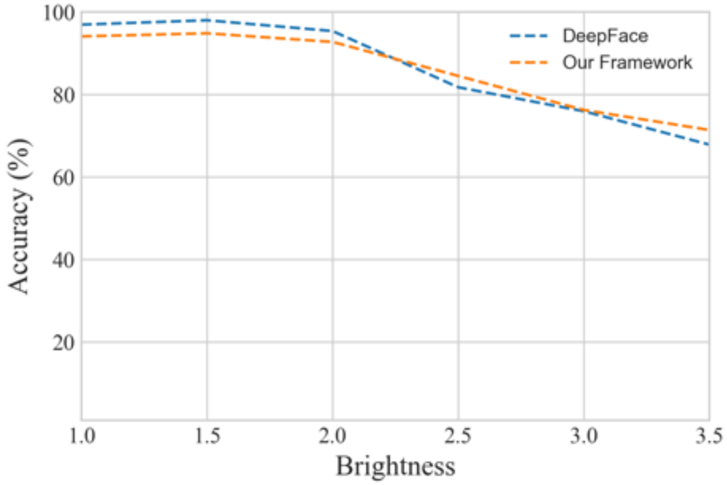


Figure 43: Plot of accuracy changes with brightened images for DeepFace verification and the proposed motion verification framework.

6.3 Conclusion

In chapter 4 we proposed a human authentication framework which is able to verify individuals based on their body movements. In chapter 5, we implemented another framework for human motion identification within a set of 10 gesturing speakers. Considering the applications of such frameworks, in verification steps, the input videos might be noisy as a result of using different cameras or might even be recorded in different lighting. To address these concerns, in this chapter, we focused on evaluating the robustness of our framework to inputs with different noise or brightness levels. The test images are added with different levels of Gaussian noise, from noise-free, to σ of $\{0.5, 1.0, 1.5, ..., 5.0, 5.5\}$. We compared our results to the state of the art face recognition deep learning frame work, DeepFace, evaluated with the same testing set for the verification task. As the results show, the verification accuracy of our framework drops by 1.5% when Gaussian noise with $\sigma = 1.0$ while DeepFace accuracy drops by 1.99% in the same experiment. With greater noise levels, both network accuracy drops as the standard deviation of the Gaussian noise gets bigger. However, as results demonstrate our network accuracy is less affected by noise compare to DeepFace.

We further evaluate our network performance with input samples from different brightness levels. Our results report that as the light level decrease or increase in images, the verification accuracy decreases in both frameworks. However, as results show, our network is more robust to extreme light changes compared to DeepFace.

Analyzing the robustness of deep neural network is a vast domain focused by lots of researcher. Since the concern of this manuscript is to provide a better understanding of human motion from visual data and investigating its potential applications as a biometric technique, we keep this chapter as simple as possible. In future chapters, we move forward with the main goal of this research and introduce a generative probabilistic framework that is able to efficiently identify subjects based on their motions in low-computational power setting.

7.0 LEARNING EFFECTIVE AND EFFICIENT

In this chapter we aim to investigate the potential capability of human temporal motion data as a non-cooperative method for on-device biometric authentication used in large scale. We introduce a novel dataset of 267 in-the-wild footage of gesturing single-speakers all collected from *YouTube*. We investigate different feature learning architectures for sequential data and incorporate them in a lightweight probabilistic generative framework. Our results shows that human body movements conveys valuable information about their identity that could be used for authentication in different devices and applications.

7.1 Introduction

One of the obstacles in biometric studies is the difficulty of data collection due to practical and legal limitations. Most of existing biometric studies are based on data collected in lab environments that due to the self-consciousness of participants is not a good representation of real world. The other challenge in existing biometric research is that they are mostly built on deep learning approaches that require high computational power as well as a huge amount of training data. Therefore, such biometric authentication/identification model is not applicable on different setting.

In previous chapters, we used LSTM neural network for human authentication and identification based on their motion signatures. The accuracy obtained by proposed framework was more that 94% for human authentication and more than 92% for human identification. Although the architectures proposed in previous steps might be applicable on many services, there are many limitations enforced by many other applications. For example, such framework requires approximately five hours of individuals data on the cloud which could create a privacy threat from the legal perspective. Thus, subject verification should be performed on devices and this creates a challenge due to the limited processing power, storage, and memory. Moreover, adaption of new individuals must be quick and based on a small sample of motion data which is not possible in a pure deep learning setting due to overfitting and low generalization power of the trained model. Therefore, learning a discriminative model per individual would not be easily feasible.

To address these challenges, we intend to propose a generative probabilistic framework that is able to efficiently learn the task-relevant features within the data in a low-computational power setting and incorporate them to a probabilistic biometric model trained using a limited amount of enrollment data. For this purpose, we start by exploring several popular deep learning feature extraction architectures including convolutional and temporal architectures. As the final step, using extracted features, we train a lightweight generative biometric model based on Gaussian Mixture Model (GMM) which is able to authenticate subjects based on their motion patterns.

7.2 Method Overview

The goal of this part of our research is to distinguish between the specific subject and the imposters based on the limited amount of enrollment motion data extracted from video inputs. For this purpose, the proposed framework is consisted of two main components:

- a **feature extraction** component that learns a collection of discriminative characteristics for each individual.
- a probabilistic **biometric model** that takes the feature vector as the input and performs the authentication task. For the purpose of this research, it is important to train a model that is lightweight enough to be used for on-device authentication task.

In the following sections, we begin to introduce the *human motion dataset.* We further discuss these two components to provide more details.

7.2.1 Human Motion Dataset

Training a UBM, requires a large sample of training data. In this work, we introduce a large dataset consisted of 267 hours of data from 267 gesturing speaker with a diverse set of backgrounds. They cover a wide range of topics including science, entertainment, personal lifestyle, religions, sports, and music. For the simplicity, all the videos in our dataset are deliberately handpicked from YouTube as they involve one gesturing speaker in a single camera setup. However, for multi-camera setups, out-of-the-box face recognition and splitting techniques could be used to extract the intervals that contains only the speakers in all frames, so the detected keypoints are visible (As the process performed in [10]). Our final dataset contains approximately 96,000 intervals with an average length of 10.42 and a standard deviation of 3.7 representing in-the-wild footage of the subjects.

Figure 44: Sample frames from our introduced dataset of 267 single speakers in single-camera setups covering a wide range of topics.

In this experiment, we used *OpenPose* at a rate of 15 fps. 33 upper body keypoints corresponding to nose, neck, shoulders, elbows, wrists, fingers, eyes and ears are extracted as explained in chapter 4. Data from all 267 speakers are used for deep feature extraction as well as training the universal background model. 10% of these sample speakers are randomly selected as the validation set to tune the hyperparameters while 80% are used for training and another 10% for testing.

7.3 Feature Extraction

Feature learning is an important stage in our proposed framework since the performance of such framework in real-time depends on the representational power of the extracted features and the speed of feature extracting. These two typically contradict each other as the speed of feature extracting decrease when the performance increase [116].

In recent years, a large number of studies proposed deep learning architecture for extracting temporal features in sequential data. Two paradigms specifically have been dominating this field with the goal of finding the balance between *speed* and *representational power* of the feature learning process. 1-dimentional convolutional neural network that aggregates the temporal statistics by temporal pooling and recurrent neural networks that explicitly models the temporal dependencies. Another model which was popular in speech recognition was consisted of *short-term* (ST) and *long-term* (LT) convolutional neural networks that are capable of learning the temporal dependencies from short and long stream of data [60]. The main challenge with each of these component is that the short term network fails to model the context while it produce output at a high rate. The long-term architecture, on the other hand, does not generalize to sequences of arbitrary length and suffers from a high degree of temporal inertia while it is capable of learning the representation at different scale.

As many deep learning architecture has been proposed for feature extraction, [80] compare these networks and propose a model that enhances the sequential data feature extraction process. In their model, they combine both convolutional and recurrent layer in a way that first they use few convolutional layers to extract features from data patches and then they feed these features to a recurrent layer to model the temporal relations in the data. One benefit of this method is that by using a pooling mechanism after convolutional layers, the input vector to the recurrent layers will be shorten, therefore learning the temporal dependencies will be performed on a smaller numbers of frames. The other benefit of their architecture is that by applying more layers of computation they allow for more details in the feature extraction process. They finally demonstrate that their proposed method outperforms both traditional CNN and RNN as well as ST and LT architectures in the area of feature extraction from sequential data.

In this research, we evaluate the performance of LSTM architecture as well as a vanilla RNN for the feature extraction task. We found it strongly beneficial to make the first few layers convolutional to reduce the size of the input data resulting to a reduction in computational load and complexity. Although the LSTM network discussed in chapter 4 obtained a satisfying accuracy to authenticate subjects, it is not efficient in terms of speed and model complexity. Feature extraction models are evaluated as a multi-class classification problem. Later, the output layer is removed and the activations of the penultimate layer are treated as the input to the probabilistic model discussed in section 7.4.1.

7.4 Biometric Model

It is important to emphasize that the methodological choices in this part our our research are affected by the specific constraints of its applications. For example, streaming the subject data into cloud potentially brings up some privacy concerns that makes the authentication task an impermissible threat. Therefore, the verification task must be doable on the device which is challenging due to the limited processing power, storage, and memory. Moreover, adaption of new individuals must be quick and based on a small sample of subject data while this sample might not be the best representation of the subject's motion signatures. Therefore, the architecture discussed in previous chapters might not be applicable in different settings as learning a separate model for each individual and fine-tuning them is hardly feasible.

Instead, in this study, using Gaussian Mixture Models (GMMs), we create a *universal background model* (UBM) to estimate a general data distribution in the dynamic motion feature space. The UBM is trained offline and prior to its use in devices using a large sample of training data. Later, the client model could be adapted from the UBM (online) by only a small sample of enrollment data from each subject. Both of these model then could be used for a continuous real-time authentication based on a trust score achieved by the likelihood test between them.

In the following sections, we will dive into more details regarding the process of the training a UBM, adaptation of the client models, and finding the proper trust score.

7.4.1 Universal Background Model

A Universal Background Model is a model mostly used in biometric authentication that represents the general subject-independent characteristics. The UBM will be then used against a subject-specific model with the specific characteristic to make a decision regarding to verification process. In this study, the UBM is a person-independent Gaussian Mixture Model trained using a large sample of pre-collected dataset. Having a person-specific GMM model trained with motion samples from a specific person, a likelihood test is performed to obtain the likelihood score between these two models.

To provide a better understanding of how the whole process works, we delay the discussion around training the UBM and client-model and start with the likelihood-ration test.

7.4.1.1 Likelihood Ratio Test Given a motion sequence of Y, and the subject S, the goal is to determine whether Y belongs to S or not. This could be written as a hypothesis testing between:

$$H_0 : \text{Y is form subject S}$$

$$H_1 : \text{Y is not form subject S}$$

The likelihood ratio test is a statistical test that decides between two hypothesis based on a calculated threshold θ and is only optimal when the likelihood functions are exactly known. θ is the decision threshold for accepting or rejecting the H_0. In practice, it could be written as:

$$\frac{P(Y|H_0)}{P(Y|H_1)} \begin{cases} \geq \theta \text{ Accept } H_0 \\ < \theta \text{ Reject } H_0 \end{cases} \tag{7.1}$$

where $P(Y|H_i), i = 0, 1$ is the probability density function for hypothesis i, which evaluates measurement M. It worth pointing out that when H_i is considered as the independent variable, $P(Y|H_i)$ is referred as the likelihood. In a verification system, the goal is to employ techniques to calculate these two likelihood functions. In the context of human biometric authentication, this could be achieved by modeling the two likelihood of $P(Y|H_0)$ and $P(Y|H_1)$. To calculate the likelihood of H_0 and H_1, let $y = f(x^{(t)}) \in R^N$ be a vector of features extracted from a raw sequence of motion vector which is performed using deep neural network. This feature vector is used to compute the likelihood of H_0 and H_1. We represent H_0 by a Gaussian Mixture Model denoted by Θ_S that characterizes the distribution of features extracted from enrolment samples from subject S by a set $\Theta = \{\mu_i, \Sigma_i, \pi_i\}$, where μ_i denotes the mean, Σ_i is a covariance matrix, and π_i is a mixture weight and coefficient. The alternative hypothesis H_1 would be the same, denoted by Θ_{UBM} that is a large GMM representing the person-independent characteristics and is referred as UBM. The likelihood ratio is then calculated as:

$$\text{LR(y)} = \frac{P(y|\Theta_S)}{P(y|\Theta_{UBM})} \tag{7.2}$$

7.4.1.2 UBM and the Subject Model While estimating a separate GMM for each individual can be learned through the training dataset, it is suboptimal regarding the goal of this part of our research which is minimizing the computations and shortening the training process. Instead, in this research, we collect a large dataset of human body motion to learn an unsupervised universal background model (UBM), which is then used to estimate the client models through an online adaptation process.

Let y be the feature vector extracted from an unprocessed sample of data. We define the UBM over these feature vectors as a weighted sum of M multi-dimensional Gaussian distributions with the mean μ_i, covariance matrix Σ_i, and mixture coefficient of π_i:

$$P(y|\Theta) = \sum_{i=1}^{M} \pi_i \mathcal{N}(y; \mu_i, \Sigma_i) \tag{7.3}$$

where the density function is defined as:

$$\mathcal{N}_i(y) = \frac{1}{\sqrt{(2\pi^D)|\Sigma_i|}} exp\left(-\frac{(y-\mu_i)'\Sigma^{-1}(y-\mu_i)}{2}\right) \tag{7.4}$$

The UBM is estimated by using the Expectation Maximization (EM) algorithm. The EM algorithm is specifically used for calculation of the probability distribution parameters in a way that maximizes the likelihood of the feature vector over the large pre-collected dataset. EM algorithm starts by placing Gaussians randomly in space by a random mean and covariance, then for each data point, it figures out the probability of this point belonging to each cluster (soft assignment). Later, the EM algorithm, use these probabilities to re-estimate the mean and covariance and it iterates until it converges.

In order to learn the subject model $P(y|\Theta_s)$, the probability distribution parameters are adapted from the UBM by using Maximum A Posteriori(MAP) estimation. In fact the UBM acts as a prior distribution over a large sample of pre-collected data, therefore the posterior distribution of the enrolment sample could be calculated using Bayes' theorem. To avoid overfitting, weights and the co variance matrices are the same while MAP is used to estimate the mean vector for a new subject. Therefore, learning subject model process is shrunk to updating a subset of parameters related to the new subject. In other words, by having a set of M enrollment samples by subject $\{y_M\}$, the updates to a specific subject model is applied by calculating the mean of each mixture component i:

$$E(\{y_M\}) = \frac{1}{n_i} \sum_{m=1}^{M} P(i|y_m) y_m \tag{7.5}$$

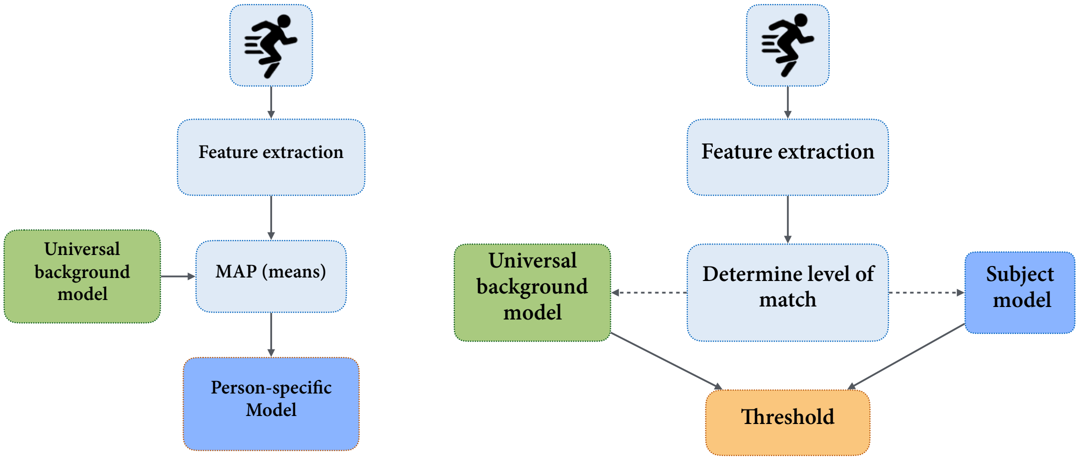


Figure 45: The overview of proposed efficient verification model using UBM; on the left is the training person specific model and MAP adaptation; on the right is the threshold estimation and testing pipeline.

where n_i is defined as:

$$n_i = \sum_{m=1}^{M} P(i|y_m) \tag{7.6}$$

and $P(i|y_m)$ is:

$$P(i|y_m) = \frac{\pi_i p_i(y_m)}{\sum_{j=1}^{C} \pi_j p_j(y_m)} \tag{7.7}$$

Using following formula, the mean of all components are updated:

$$\hat{\mu_i} = \alpha_i E_i(\{y_m\}) + (1 - \alpha_i)\mu_i \tag{7.8}$$

where:

$$\alpha_i = \frac{n_i}{n_i + r} \tag{7.9}$$

r is a relevance factor that balances the subject model and the UBM and is obtained empirically in our research (r = 3).

7.4.1.3 Subject Authentication via Scoring As mentioned in section 7.4.1.1, subject authentication is performed by scoring the feature vector extracted from a given sample sequence of data ($Y = \{y_m\}$) against the UBM and the subject model. The threshold could be defined as the log-likelihood ratio as:

$$\Lambda(Y) = \log p(Y|\Theta_S) - \log p(Y|\Theta_{UBM}) \tag{7.10}$$

Lastly, in order to avoid outlier issues the distribution of the authentic client and impostors and draw a sharper client boundaries, *zt-score normalization* is performed [134]. Generally, score normalization aims to transform the distribution of the authentication scores to increase the decision threshold robustness [177]. *Zero-normalization* is used to normalize the subject model by testing the it against a set of imposters samples resulting into an imposter similarity score distribution. The imposter similarity score distribution is then used to calculate the normalization parameters (mean and variance). These mean and standard deviation are later used to normalize future incoming scores $\Theta(Y)$ via the normalization function bellow. The benefit of Z-norm is that the normalization parameters can be estimated offline during the training step [22]:

$$\Lambda_z(Y|\Theta_S) = \frac{\Theta(Y) - \mu(Z|\Theta_S)}{\sigma(Z|\Theta_S)} \tag{7.11}$$

where Z is a set of imposter samples while Y is a given test sample. $\sigma(Z|\Theta_S)$ and $\mu(Z|\Theta_S)$ are the normalization parameters.

The *test normalization* (τ-norm) [22] is another normalization method in which each test sample is scored against a set of imposter distributions at the test time. τ-norm normalizes the score of test sample against the claimed model using bellow function:

$$\Lambda_{zt}(Y) = \frac{\Theta_z(Y|\Theta_S) - \mu_z(Z|\Theta_\tau)}{\sigma_z(Z|\Theta_\tau)} \tag{7.12}$$

The τ-distributions (Θ_τ) are adapted from UBM via MAP-adaptation the same as subject models, but using a different subset. The Z samples come from a part of train dataset that has not been used for τ-distributions [96].

7.5 Experimental Results

To train the UBM, 128 mixture components were initialized with kmeans and trained for 80 iterations. For learning subject models, MAP-adaptation is performed in 8 iterations with the relevance factor being equal to 3. For score normalization, we randomly created non-overlapping samples extracted from training set for the τ-models as well as z-samples. The τ-models are adapted from UBM using MAP-adaptation. The hyper-parameters are optimized based on the validation set.

We use the same training and testing data for the zt-score normalization process. From non-overlapping subset of data, we create 200 t-models as well as 200 z-sequence, Each of these t-models are trained using the UBM through a map-adaptation process. We then optimize all of hyperparameters on the validation set.

In following sections, we evaluate our proposed authentication framework on our single-speaker dataset introduced in section 7.2.1. *Two rounds* of evaluations are performed specifically; the first round evaluates the is performance of the different feature learning architectures while the second round more focuses on the performance of the extracted features as a part of a generative biometric model with the goal of authentication. Table 8 and 9 represents details regarding architectures used for the purpose of feature learning. All deep networks implemented in this study are trained over an NVIDIA GPU TESLA K80.

7.5.1 Feature Learning Evaluation

We start by evaluating the feature extraction networks as a multi-class classification problem where each class corresponds to one of 267 subjects in our dataset. In this step of our evaluation the generative biometric model is not considered. To estimate the accuracy, a softmax layer is used as the output layer that yields to a class probability distribution corresponding to subjects identities. We selected the class with the highest probability as the recognized identity.

Table 8 demonstrates the hyper parameters related to the long-term and short-term convnet architectures. The long-term convnet is trained over 150 samples (one data stream)

Table 8: Feedforward long-term (LT) convolutional architecture on the left, and feedforward short-term (ST) convolutional architecture on the right.

	LT Convnet		ST Convnet	
Layer	Filter size/Units	Pooling	Filter size/Units	Pooling
Input	10 × 15 × 134	-	3 × 15 × 134	-
Conv1	25 × 9 × 1	4 × 1	25 × 9 × 1	2 × 1
Conv2	25 × 9 × 1	2 × 1	25 × 9 × 1	1 × 1
Conv3	25 × 9 × 1	1 × 1	25 × 9 × 1	1 × 1
FCL1	1024	-	1024	-
FCL2	512	-	512	-
Output	267	-	267	-

Table 9: Conv-RNN sequential feature learning architecture on the left and Conv-LSTM sequential feature learning architecture on the right.

	Conv-RNN		Conv-LSTM	
Layer	Filter size/Units	Pooling	Filter size/Units	Pooling
Input	10 × 15 × 134	-	10 × 15 × 134	-
Conv1	25 × 7 × 1	2 × 1	25 × 7 × 1	2 × 1
Conv2	25 × 7 × 1	2 × 1	25 × 7 × 1	2 × 1
Conv3	25 × 7 × 1	1 × 1	25 × 7 × 1	1 × 1
Recurrent	512 RNN	-	256 LSTM	-
Output	267	-	267	-

and the short-term architecture is trained over 45 samples. We distinguished convolutional layer and fully connected layer by denoting them as Conv and FCL. Table 9 demonstrate feature learning architectures via vanilla recurrent neural network and LSTM network both combined with convolutional layers. We train all of these network using stochastic gradient decent. We further use dropout for regularization on fully connected layers. For the loss calculation, we use negative log likelihood function.

Table 10 demonstrates the classification accuracies obtained with architectures presented in table 8 and 9. As the results show, among temporal models, the LSTM recurrent network achieves the highest accuracy as the most effective architecture. The vanilla RNN accuracy, on the other hand, is lower compared to LSTM network but still higher that the accuracies obtained by LT and ST convectional networks.

Table 10: Performance of feature learning architectures on single-speaker dataset.

Model	Accuracy (%)
ST Convnet	57.84
LT Convnet	65.91
Conv-RNN	73.28
Conv-LSTM	**79.64**

7.5.2 Authentication UBM Evaluation

When it comes to binary authentication, classification accuracy is not capable of capturing the balance between false rejection and false acceptance rates. Equal Error Rate (EER) is a metric that describes when false rejection rate (FRR) and false acceptance rate (FAR) are balanced and equal. As sensitivity increase, the FRR rise and the FAR drops. EER is the intersection of these two lines and demonstrates the overall accuracy of a biometric system. Using the validation set, we optimize the authentication model to have the minimal

EER [132]. Once the EER threshold (Θ_{EER}) is obtained, we use it on our testing data to evaluate the performance via Half Total Error Rate (HTER) that corresponds to the average of both FAR and FRR errors on the test set:

$$HTER = \frac{FAR(\Theta_{EER}) + FRR(\Theta_{EER})}{2} \tag{7.13}$$

A lower ERR demonstrates a better performance of the model. For example an ERR equal to 10% indicates that 90% of the time, the model authenticates subject correctly. Table 11 demonstrates the performance measurements of the proposed GMM-based authentication model using different feature learning architectures on the single-speaker dataset. Results obtained in this step are aligned well with the performance of feature extraction models, showing that well-recognized features can efficiently incorporate in a lightweight generative framework.

Table 11: Performance metrics of the proposed GMM-based authentication model using different feature learning architectures on the single-speaker dataset.

Model	EER (%)	HTER (%)
Raw features	37.85	44.27
ST Convnet	29.15	31.32
LT Convnet	26.91	28.26
Conv-RNN	19.82	20.18
Conv-LSTM	16.14	16.95
Conv-LSTM (*zt-norm*)	**13.89**	**14.05**

To compare the proposed generative probabilistic framework with a previous framework that includes fine-tuning a separate model for each individual in another setting that is not enforced with applications limitations, we randomly selected 10 subjects from the validation set. The output of the LSTM feature learning model was replaced with a binary classification model (in this case a logistic regression). The average performance was already 3.2% lower than the generative model. The reason behind such difference could be explained in the shade of overfitting and poor generalization of the traditional model. In a generative setting, however, such challenge is handled when each subject's model adopts the parameters from the general probabilistic distribution. In other word, adapting some parameters from the UBM while learning a subject generative model helps the model to keep the generality of the UBM at the same time as it fits well to the training data specifically when the training dataset is limited. Adapting parameters from UBM that has been trained based on a huge amount of data provides more robustness for the subject model [159].

7.6 Conclusion

In this chapter, in response to challenges in biometric research such as low computational power and limited amount of training data in particular applications, we propose a non-cooperative framework which is able to authenticate human motions. The proposed framework framework includes three main component including a pose estimation machine, a temporal feature extraction model, and a generative probabilistic model for the authentication task. For pose estimation, we used OpenPose to extract body pose in each frame with a rate 15 fps. For efficient learning of dynamic features, we investigate several popular feature extraction architectures including long-term (LT) and short-term convnets as well as conv-RNN and conv-LSTM. Results show that among all architectures LSTM network combined with few convolutional layers obtains the highest accuracy (79.64). For human authentication task, we first used a probabilistic generative model to train a UBM offline and based on a large sample of single-speaker dataset. Each participant model is then adapted from UBM via MAP-adaptation. Finally, the human authentication is performed by scoring the extracted feature vector against the UBM. As the final step, we used zt-score normalization to reduce the overlap between different classes and compensate for inter-session and inter-suject variations.

We use EER as a performance metric to evaluate our authentication framework. Our results show an EER equal to 13.89 for our proposed framework, meaning that 86.11 of the time the subject is verified correctly. Given the fact that the proposed framework does not require high computational power and a huge amount of data to perform the on-device learning of subjects model, the results obtained in this study look particularly promising and we believe that this work can provide additional view for communities working on similar problem. Moreover, the proposed framework in this chapter can be easily be extended to multi-person authentication framework for specific applications.

8.0 SUMMARY AND FUTURE WORK

The current manuscript sums up our years of contribution in academic research and we are thrilled to walk the reader step-by-step through our findings and experiences. Now, we would like to provide some closing remarks and once again highlight the contribution of our work to shed light on some potential future research direction. For this final review, let us walk through what we done and emphasise on some details.

– **Part 1. Motion Authentication/Identification From Visual Input.**
  In the beginning stage of our study, we tackled the problem of human authentication/identification based on their motion pattern from visual inputs. In this context, we demonstrate that motion features extracted from videos convey information that can be considered as practical biometric cues for identification. We designed, implemented, and optimized a fully automated deep learning framework to model human body movements from visual inputs and authenticate/identify them based on their motion series. Since the video inputs are not annotated and acquiring human pose annotation for large data is not feasible, in this work, we take a data-driven approach to learn visual data representation instead of hand-crafting features for each subject. The proposed framework is consisted of a pose estimation machine, (*OpenPose*) and a three layer LSTM neural network. OpenPose detects body keypoints in visual sequences from the Speaker-specific dataset. Having the pose extracted in each frame, motion features are calculated between consecutive frames. Later, the LSTM recurrent neural network acts as the classifier to authenticate/identify individuals.
  The key contribution of this part of our research is in design and implementation of a temporal architecture that is able to model multiple temporal sequences. The test results of our model over a set of 10 gesturing speakers obtained an average accuracy of 92.62% for the human identification and 94.07% for human authentication.
  Our model performance surpasses other conventional approaches including random forest, decision tree, and SVM in both identification and authentication tasks. Our

framework is specifically designed and developed for motion biometric recognition while it could be applied to many other problems in computer vision.

- **Part 2. Robustness Analysis.**

  The second stage of this study is dedicated to analyzing the robustness of proposed authentication/identification framework. One of the discussions around vision-based approaches is that the precision of outputs relies heavily on the image quality as well as brightness changes. Therefore, it is important to make sure that the small changes in the testing data, does not yield to a significant loss to the performance of the network. For this purpose, we manipulated our test dataset to see how adding noise and changes in brightness would impact the network accuracy. Since the framework implemented in this work is the first automated motion recognition system capable of authenticating individuals from visual inputs, we used *DeepFace*, the state of the art vision-based face recognition framework as a baseline to our work to compare our results.

  As the first step to robustness evaluation, we added Gaussian noise with different magnitude to our test dataset to see how our framework reacts. We started by applying random Gaussian noise with the standard deviation of 0.5 to each RGB channel of video frames and increased it by 0.5 step by stem till it gets to 5.5. Our results demonstrate that our framework is more robust to noisy inputs as its total changes in the accuracy are less in each experiment even though that both networks' accuracies decrease when noise magnitudes gets bigger. In noise level with a σ of 5.5, our network surpasses DeepFace by 5.4%.

  In the second step of our robustness analysis, we discussed how light changes would impact the network performance by manipulating the brightness level in our test dataset. We generated augmented test samples with enhancer factors ranging from 0 to 3.5 and calculated the accuracies for both our network and DeepFace to see how different brightness levels would impact the performance. While an enhancer factor of 1.0 gives back the original image, moving toward 0.0 darkens the image and a factor greater than 1.0 makes the image brighter. Such task is done by increasing/decreasing the pixel values evenly across all channel for the entire image. A factor of exactly 0.0 is a black image.

Our results show that both framework accuracies drop equally moving from enhancer factor 1.0 to 0.25. However, in lower lighting with enhancer factor of 0.1 , our framework performs significantly better and surpasses DeepFace accuracy by 21.19%. In brightness levels with enhancer factor greater than 1.0, which are brightened images, both network accuracies drops as a result of loosing some visual information due to changes in pixel values. However, the overall drop in our network is 6% less compared to DeepFace confirming the robustness of our network to brightness changes.

- **Part 3. Learning Effective and Efficient.**

  In the final stage of this thesis, we address the challenges related to low computational power and limited amount of training samples that are enforced by potential applications of our proposed framework. In devices with limited processing power, storage and memory, it is important to make sure that the verification task is successfully completed in a reasonable time. Moreover, the adaption of new individuals must be based on a small sample of the subject data and needs to be quick.

  For this purpose, we proposed an authentication framework consisting of a feature learning network incorporated in a lightweight generative framework that allows learning the subject model directly on the device with a limited amount of training data. For efficient learning of dynamic features, we investigated several popular feature extraction architectures and showed that among all those architectures, the LSTM network combined with few convolutional layers obtains the highest accuracy. Using a large sample of pre-collected dataset, we then trained a universal background model offline containing the user-independent characteristic. Each subject model was then adapted from the UBM via MAP-adaptation process. Finally, the human authentication was performed by scoring the extracted feature vector against the UBM.

During this research, we brought up five main challenges unaddressed in previous studies. We briefly go over these challenges and explain how we addressed them in the context of our research. The first challenge is that the most of previous studies are limited to lab-scale data collection which is a poor representation of the real world due to subjects consciousness. In our study, we use in-the-wild footage of subjects in which users are not aware of the experiments at all. Moreover, all previous studies required an additional device such as Kinect, wearable or radar sensors which are not practical for most real-world situations. We conducted our framework based on video data from subjects, therefore it does not require any additional hardware. To address challenges regarding the privacy threat, and limitations such as processing power, storage, and memory, we introduced our lightweight generative framework that is trained offline based on a pool of subjects. In this framework, learning the subject model is quick since their parameters are adapted from the universal background model. Therefore, the training time is decrease by 90% which is crucial in industrial applications.

What's done in this research is a promising step toward active biometric recognition that does not require the direct communication and cooperation of subjects. For future work, there are few different directions that can be explored. First of all, further analysis of temporal feature learning for understanding different aspect of human motion characteristics could be done particularly from a biometric point of view. Secondly, the motion pattern could be considered in the context of specific applications such as games. Lastly, the proposed approach in this research could be combined with other biometric recognition approach to achieved the optimal accuracy. This area of research is fertile ground for further explorations and finding new applications.

This concludes our study about human authentication based on their motion patterns. There is an enormous amount of work could be done in this area, and we expect that it will take many years for these system to be widely used in daily life systems.

BIBLIOGRAPHY

[1] A Comprehensive Guide to Convolutional Neural Networks. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[2] Berkeley multimodal human action database. http://tele-immersion.citris-uc.org/berkeleymhad.

[3] Deep learning for sensor-based human activity recognition. https://becominghuman.ai/deep-learning-for-sensor-based-human-activity-recognition-970ff47c6b6b/.

[4] Deepfake Detection Challenge. https://deepfakedetectionchallenge.ai/.

[5] Extended yale face database. http://vision.ucsd.edu/content/extended-yale-face-database-b-b/.

[6] Kinect. https://developer.microsoft.com/en-us/windows/kinect.

[7] LoomieLive. https://www.loomielive.com/.

[8] Polyu fingerprint dataset. http://www4.comp.polyu.edu.hk/~biometrics/HRF/HRF_old.htm/.

[9] Polyu palmprint dataset. https://www4.comp.polyu.edu.hk/~biometrics/MultispectralPalmprint/MSP.htm/.

[10] Speaker Specific Gesture Dataset. https://github.com/amirbar/speech2gesture/blob/master/data/dataset.md.

[11] TheCaptury. https://thecaptury.com.

[12] Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[13] Vicon. https://www.vicon.com.

[14] Ayman Abaza and Mary Ann F Harrison. Ear recognition: a complete system. In *Biometric and Surveillance Technology for Human and Activity Identification X*, volume 8712, page 87120N. International Society for Optics and Photonics, 2013.

[15] Bilal M'hamed Abidine, Lamya Fergani, Belkacem Fergani, and Mourad Oussalah. The joint use of sequence features combination and modified weighted svm for improving daily activity recognition. *Pattern Analysis and Applications*, 21(1):119–138, 2018.

[16] Daehwan Ahn, Seongmin Jeon, and Byungjoon Yoo. What's your real age? an empirical analysis of identity fraud in online game. *Information Systems and e-Business Management*, 16(4):775–789, 2018.

[17] Cihan Akin, Umit Kacar, and Murvet Kirci. A multi-biometrics for twins identification based speech and ear. *arXiv preprint arXiv:1801.09056*, 2018.

[18] Fadi Al Machot, Mohammed R Elkobaisi, and Kyandoghere Kyamakya. Zero-shot human activity recognition using non-visual sensors. *Sensors*, 20(3):825, 2020.

[19] Md Zahangir Alom, Tarek M Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C Van Essen, Abdul AS Awwal, and Vijayan K Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3):292, 2019.

[20] Eisa Jafari Amirbandi and Ghazal Shamsipour. Exploring methods and systems for vision based human activity recognition. In *2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pages 160–164. IEEE, 2016.

[21] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005.

[22] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, 2000.

[23] Babu. A 2019 guide to human pose estimation with deep learning. https://nanonets.com/blog/human-pose-estimation-2d-guide/.

[24] Woodrow Barfield and Marc Jonathan Blitz. *Research Handbook on the Law of Virtual and Augmented Reality*. Edward Elgar Publishing Cheltenham, 2018.

[25] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[26] Djamila Romaissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79(41):30509–30555, 2020.

[27] John D Bustard and Mark S Nixon. Toward unconstrained ear recognition from two-dimensional images. *IEEE transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 40(3):486–494, 2010.

[28] Allah Bux, Plamen Angelov, and Zulfiqar Habib. Vision based human activity recognition: a review. In *Advances in Computational Intelligence Systems*, pages 341–371. Springer, 2017.

[29] Peibei Cao, Weijie Xia, Ming Ye, Jutong Zhang, and Jianjiang Zhou. Radar-id: human identification based on radar micro-doppler signatures using deep convolutional neural networks. *IET Radar, Sonar & Navigation*, 12(7):729–734, 2018.

[30] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[31] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[32] Hong Chen and Anil K Jain. Dental biometrics: Alignment and matching of dental radiographs. *IEEE transactions on pattern analysis and machine intelligence*, 27(8):1319–1326, 2005.

[33] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: overview, challenges and opportunities. *arXiv preprint arXiv:2001.07416*, 2020.

[34] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in neural information processing systems*, pages 1736–1744, 2014.

[35] Patrick Connor and Arun Ross. Biometric recognition by gait: A survey of modalities and features. *Computer Vision and Image Understanding*, 167:1–27, 2018.

[36] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[37] James E Cutting and Lynn T Kozlowski. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of the psychonomic society*, 9(5):353–356, 1977.

[38] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[39] L Minh Dang, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, 108:107561, 2020.

[40] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24(6):663–676, 2019.

[41] John Daugman. Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94(11):1927–1935, 2006.

[42] Maria De Marsico, Alfredo Petrosino, and Stefano Ricciardi. Iris recognition through machine learning techniques: A survey. *Pattern Recognition Letters*, 82:106–115, 2016.

[43] Xunfei Deng, Zhi Liu, Yu Zhan, Kang Ni, Yongzhi Zhang, Wanzhu Ma, Shengzhi Shao, Xiaonan Lv, Yuwei Yuan, and Karyne M Rogers. Predictive geographical authentication of green tea with protected designation of origin using a random forest model. *Food Control*, 107:106807, 2020.

[44] Erika D'Antonio, Juri Taborri, Eduardo Palermo, Stefano Rossi, and Fabrizio Patanè. A markerless system for gait analysis based on openpose library. In *2020 IEEE Inter-*

national Instrumentation and Measurement Technology Conference (I2MTC), pages 1–6. IEEE, 2020.

[45] Muhammad Ehatisham-Ul-Haq, Ali Javed, Muhammad Awais Azam, Hafiz MA Malik, Aun Irtaza, Ik Hyun Lee, and Muhammad Tariq Mahmood. Robust human activity recognition using multimodal feature-level fusion. *IEEE Access*, 7:60736–60751, 2019.

[46] Mohamed Elhoseny, Amir Nabil, Aboul Ella Hassanien, and Diego Oliva. Hybrid rough neural network model for signature recognition. In *Advances in Soft Computing and Machine Learning in Image Processing*, pages 295–318. Springer, 2018.

[47] Sefik Emre Eskimez, Ross K Maddox, Chenliang Xu, and Zhiyao Duan. Generating talking face landmarks from speech. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 372–381. Springer, 2018.

[48] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017.

[49] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[50] F Foerster and M Smeja. Joint amplitude and frequency analysis of tremor activity. *Electromyography and clinical neurophysiology*, 39(1):11, 1999.

[51] KS Fu. Syntactic (linguistic) pattern recognition. In *Digital pattern recognition*, pages 95–134. Springer, 1980.

[52] Javier Galbally, Raffaele Cappelli, Alessandra Lumini, Davide Maltoni, and Julian Fierrez. Fake fingertip generation from a minutiae template. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[53] Felix A Gers and Jürgen Schmidhuber. Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 3, pages 189–194. IEEE, 2000.

[54] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning individual styles of conversational gesture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3497–3506, 2019.

[55] Wenjuan Gong, Xuena Zhang, Jordi Gonzàlez, Andrews Sobral, Thierry Bouwmans, Changhe Tu, and El-hadi Zahzah. Human pose estimation from monocular images: A comprehensive survey. *Sensors*, 16(12):1966, 2016.

[56] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[57] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772, 2014.

[58] Alex Graves, Marcus Liwicki, Horst Bunke, Jürgen Schmidhuber, and Santiago Fernández. Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in neural information processing systems*, pages 577–584, 2008.

[59] Yulan Guo, Yinjie Lei, Li Liu, Yan Wang, Mohammed Bennamoun, and Ferdous Sohel. Ei3d: Expression-invariant 3d face recognition based on feature and shape matching. *Pattern Recognition Letters*, 83:403–412, 2016.

[60] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[61] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and H-P Seidel. A statistical model of human pose and body shape. In *Computer graphics forum*, volume 28, pages 337–346. Wiley Online Library, 2009.

[62] Mark Hawthorne. *Fingerprints: analysis and understanding.* CRC Press, 2017.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[64] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[65] Chunyu Hu, Yiqiang Chen, Lisha Hu, and Xiaohui Peng. A novel random forests based class incremental learning method for activity recognition. *Pattern Recognition*, 78:277–290, 2018.

[66] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[67] François Hug, Clément Vogel, Kylie Tucker, Sylvain Dorel, Thibault Deschamps, Éric Le Carpentier, and Lilian Lacourpaille. Individuals have unique muscle activation signatures as revealed during gait and pedaling. *Journal of Applied Physiology*, 127(4):1165–1174, 2019.

[68] Salam Allawi Hussein, Alyaa Abduljawad Mahmood, and Mohammed Iqbal Dohan. A hybrid global local adaptive particle swarm optimization-based support vector machine model for human facial authentication. *Journal of Southwest Jiaotong University*, 55(1), 2020.

[69] Andrey Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915–922, 2018.

[70] Tetsushi Ikeda, Hiroshi Ishiguro, Takahiro Miyashita, and Norihiro Hagita. Pedestrian identification by associating wearable and environmental sensors based on phase dependent correlation of human walking. *Journal of Ambient Intelligence and Humanized Computing*, 5(5):645–654, 2014.

[71] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[72] Shekh MM Islam, Ashikur Rahman, Narayana Prasad, Olga Boric-Lubecke, and Victor M Lubecke. Identity authentication system using a support vector machine (svm) on radar respiration measurements. In *2019 93rd ARFTG Microwave Measurement Conference (ARFTG)*, pages 1–5. IEEE, 2019.

[73] Anil Jain, Lin Hong, and Sharath Pankanti. Biometric identification. *Communications of the ACM*, 43(2):90–98, 2000.

[74] Anil K Jain, Hong Chen, and Silviu Minut. Dental biometrics: human identification using dental radiographs. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 429–437. Springer, 2003.

[75] Anil K Jain and Stan Z Li. *Handbook of face recognition*, volume 1. Springer, 2011.

[76] Alexander Refsum Jensenius. Action-sound: Developing methods and tools to study music-related body movement. 2007.

[77] Charmi Jobanputra, Jatna Bavishi, and Nishant Doshi. Human activity recognition: a survey. *Procedia Computer Science*, 155:698–703, 2019.

[78] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8320–8329, 2018.

[79] Shanon X Ju, Michael J Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 38–44. IEEE, 1996.

[80] Gil Keren and Björn Schuller. Convolutional rnn: an enhanced model for extracting features from sequential data. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3412–3419. IEEE, 2016.

[81] Jan Kietzmann, Linda W Lee, Ian P McCarthy, and Tim C Kietzmann. Deepfakes: Trick or treat? *Business Horizons*, 63(2):135–146, 2020.

[82] Youngwook Kim and Taesup Moon. Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks. *IEEE geoscience and remote sensing letters*, 13(1):8–12, 2015.

[83] Oleg V Komogortsev and Corey D Holland. Biometric authentication via complex oculomotor behavior. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2013.

[84] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11977–11986, 2019.

[85] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[86] Ajay Kumar and Arun Passi. Comparison and combination of iris matchers for reliable personal authentication. *Pattern recognition*, 43(3):1016–1026, 2010.

[87] Ajay Kumar and Chenye Wu. Automated human identification using ear imaging. *Pattern Recognition*, 45(3):956–968, 2012.

[88] Amioy Kumar, Madasu Hanmandlu, and H_M Gupta. Fuzzy binary decision tree for biometric based personal authentication. *Neurocomputing*, 99:87–97, 2013.

[89] Alexander Kupin, Benjamin Moeller, Yijun Jiang, Natasha Kholgade Banerjee, and Sean Banerjee. Task-driven biometric authentication of users in virtual reality (vr) environments. In *International conference on multimedia modeling*, pages 55–67. Springer, 2019.

[90] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[91] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209, 2012.

[92] Yann Le Cun, Lionel D Jackel, Brian Boser, John S Denker, Henry P Graf, Isabelle Guyon, Don Henderson, Richard E Howard, and William Hubbard. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11):41–46, 1989.

[93] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14, 2015.

[94] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

[95] Sijin Li, Weichen Zhang, and Antoni B Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2848–2856, 2015.

[96] Stan Z Li and Anil Jain. *Encyclopedia of biometrics*. Springer Publishing Company, Incorporated, 2015.

[97] Sugang Li, Ashwin Ashok, Yanyong Zhang, Chenren Xu, Janne Lindqvist, and Macro Gruteser. Demo of headbanger: Authenticating smart wearable devices using unique head movement patterns. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 1–3. IEEE, 2016.

[98] Sugang Li, Ashwin Ashok, Yanyong Zhang, Chenren Xu, Janne Lindqvist, and Macro Gruteser. Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9. IEEE, 2016.

[99] Xinyu Li, Yuan He, and Xiaojun Jing. A survey of deep learning-based human activity recognition in radar. *Remote Sensing*, 11(9):1068, 2019.

[100] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2018.

[101] Hongyi Liu and Lihui Wang. Gesture recognition for human-robot collaboration: A review. *International Journal of Industrial Ergonomics*, 68:355–367, 2018.

[102] Zhao Liu, Jianke Zhu, Jiajun Bu, and Chun Chen. A survey of human pose estimation: the body parts parsing based methods. *Journal of Visual Communication and Image Representation*, 32:10–19, 2015.

[103] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[104] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[105] Nailya Maitanova, Jan-Simon Telle, Benedikt Hanke, Matthias Grottke, Thomas Schmidt, Karsten von Maydell, and Carsten Agert. A machine learning approach to low-cost photovoltaic power prediction based on publicly available weather reports. *Energies*, 13(3):735, 2020.

[106] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.

[107] Iacopo Masi, Yue Wu, Tal Hassner, and Prem Natarajan. Deep face recognition: A survey. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 471–478. IEEE, 2018.

[108] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. *arXiv preprint arXiv:1712.03453*, 2017.

[109] Shervin Minaee, Amirali Abdolrashidi, Hang Su, Mohammed Bennamoun, and David Zhang. Biometric recognition using deep learning: A survey. *arXiv preprint arXiv:1912.00271*, 2019.

[110] Huaxiao Mo, Bolin Chen, and Weiqi Luo. Fake faces identification via convolutional neural network. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pages 43–47, 2018.

[111] Fabian Monrose and Aviel Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56, 1997.

[112] Tewodros Legesse Munea, Yalew Zelalem Jembre, Halefom Tekle Weldegebriel, Longbiao Chen, Chenxi Huang, and Chenhui Yang. The progress of human pose estimation: A survey and taxonomy of models applied in 2d human pose estimation. *IEEE Access*, 8:133330–133348, 2020.

[113] Masato Nakai, Yoshihiko Tsunoda, Hisashi Hayashi, and Hideki Murakoshi. Prediction of basketball free throw shooting by openpose. In *JSAI International Symposium on Artificial Intelligence*, pages 435–446. Springer, 2018.

[114] Pradyumna Narayana, Ross Beveridge, and Bruce A Draper. Gesture recognition: Focus on the hands. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5235–5244, 2018.

[115] Natalia Neverova, Christian Wolf, Griffin Lacey, Lex Fridman, Deepak Chandra, Brandon Barbello, and Graham Taylor. Learning human identity from motion patterns. *IEEE Access*, 4:1810–1820, 2016.

[116] Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706, 2015.

[117] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[118] Xuecheng Nie, Jiashi Feng, Junliang Xing, and Shuicheng Yan. Pose partition networks for multi-person pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 684–699, 2018.

[119] Mark S Nixon, John N Carter, Jason M Nash, Ping S Huang, David Cunado, and Sarah V Stevenage. Automatic gait recognition. 1999.

[120] Farzan Majeed Noori, Benedikte Wallace, Md Zia Uddin, and Jim Torresen. A robust human activity recognition approach using openpose, motion features, and deep recurrent neural network. In *Scandinavian Conference on Image Analysis*, pages 299–310. Springer, 2019.

[121] Fuminori Okumura, Akira Kubota, Yoshinori Hatori, Kenji Matsuo, Masayuki Hashimoto, and Atsushi Koike. A study on biometric authentication based on arm sweep action with acceleration sensor. In *2006 International Symposium on Intelligent Signal Processing and Communications*, pages 219–222. IEEE, 2006.

[122] Tsukasa Okumura, Shuichi Urabe, Katsufumi Inoue, and Michifumi Yoshioka. Cooking activities recognition in egocentric videos using hand shape feature with openpose. In *Proceedings of the Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management*, pages 42–45, 2018.

[123] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017.

[124] Unsang Park, Yiying Tong, and Anil K Jain. Age-invariant face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):947–954, 2010.

[125] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7307–7316, 2018.

[126] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.

[127] Xavier Perez-Sala, Sergio Escalera, Cecilio Angulo, and Jordi Gonzalez. A survey on model based approaches for 2d and 3d visual human pose recovery. *Sensors*, 14(3):4189–4210, 2014.

[128] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016.

[129] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[130] Ivens Portugal, Paulo Alencar, and Donald Cowan. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97:205–227, 2018.

[131] Jun Qi, Po Yang, Martin Hanneghan, Stephen Tang, and Bo Zhou. A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors. *IEEE Internet of Things Journal*, 6(2):1384–1393, 2018.

[132] Vijay V Raghavan, Venkat N Gudivada, Venu Govindaraju, and Calyampudi Radhakrishna Rao. *Cognitive computing: Theory and applications*. Elsevier, 2016.

[133] Sreenivasan Ramasamy Ramamurthy and Nirmalya Roy. Recent trends in machine learning for human activity recognition—a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1254, 2018.

[134] Douglas A Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *Fifth European Conference on Speech Communication and Technology*, 1997.

[135] Roberto Ricci and Alessio Balleri. Recognition of humans based on radar micro-doppler shape spectrum features. *IET Radar, Sonar & Navigation*, 9(9):1216–1223, 2015.

[136] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3433–3441, 2017.

[137] Roberto Roizenblatt, Paulo Schor, Fabio Dante, Jaime Roizenblatt, and Rubens Belfort. Iris recognition as a biometric method after cataract surgery. *Biomedical engineering online*, 3(1):1–7, 2004.

[138] Charissa Ann Ronao and Sung-Bae Cho. Recognizing human activities from smartphone sensors using hierarchical continuous hidden markov models. *International Journal of Distributed Sensor Networks*, 13(1):1550147716683687, 2017.

[139] David E Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. Backpropagation: The basic theory. *Backpropagation: Theory, architectures and applications*, pages 1–34, 1995.

[140] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[141] Ayan Seal, Debotosh Bhattacharjee, and Mita Nasipuri. Human face recognition using random forest based fusion of à-trous wavelet transform coefficients from thermal and visible images. *AEU-International Journal of Electronics and Communications*, 70(8):1041–1049, 2016.

[142] Vijay Bhaskar Semwal, Manish Raj, and Gora Chand Nandi. Biometric gait identification based on a multilayer perceptron. *Robotics and Autonomous Systems*, 65:65–75, 2015.

[143] Mehmet Saygın Seyfioğlu, Ahmet Murat Özbayoğlu, and Sevgi Zubeyde Gürbüz. Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities. *IEEE Transactions on Aerospace and Electronic Systems*, 54(4):1709–1723, 2018.

[144] Yong Sheng, Vir V Phoha, and Steven M Rovnyak. A parallel decision tree-based method for user authentication based on keystroke patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(4):826–833, 2005.

[145] Jonathan Shieber. Investors say emerging multiverses are the future of entertainment, May 2020.

[146] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. A survey of online activity recognition using mobile phones. *Sensors*, 15(1):2059–2085, 2015.

[147] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[148] Roshan Singh, Ankur Sonawane, and Rajeev Srivastava. Recent evolution of modern datasets for human activity recognition: a deep survey. *Multimedia Systems*, pages 1–24, 2019.

[149] Stephen W Smoliar and HongJiang Zhang. Content based video indexing and retrieval. *IEEE multimedia*, 1(2):62–72, 1994.

[150] D Srinivasan, WS Ng, and AC Liew. Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, 21(1):398–405, 2005.

[151] T Subetha and S Chitrakala. A survey on human activity recognition from videos. In *2016 International Conference on Information Communication and Embedded Systems (ICICES)*, pages 1–7. IEEE, 2016.

[152] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[153] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[154] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[155] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3941–3950, 2017.

[156] Hoang Minh Thang, Vo Quang Viet, Nguyen Dinh Thuc, and Deokjai Choi. Gait identification using accelerometer on mobile phone. In *2012 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 344–348. IEEE, 2012.

[157] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[158] Yu-Shiuan Tsai, Li-Heng Hsu, Yi-Zeng Hsieh, and Shih-Syun Lin. The real-time depth estimation for an occluded person based on a single image and openpose method. *Mathematics*, 8(8):1333, 2020.

[159] Florian Verdet. Exploring variabilities through factor analysis in automatic acoustic language recognition. Avignon, 2011.

[160] Michalis Vrigkas, Christophoros Nikou, and Ioannis A Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.

[161] J Walsh, N O'Mahony, S Campbell, A Carvalho, L Krpalkova, G Velasco-Hernandez, S Harapanahalli, and D Riordan. Deep learning vs. traditional computer vision. In *Computer Vision Conference (CVC)*, 2019.

[162] Changsheng Wan, Li Wang, and Vir V Phoha. A survey on gait recognition. *ACM Computing Surveys (CSUR)*, 51(5):1–35, 2018.

[163] Shaohua Wan, Lianyong Qi, Xiaolong Xu, Chao Tong, and Zonghua Gu. Deep learning models for real-time human activity recognition with smartphones. *Mobile Networks and Applications*, 25(2):743–755, 2020.

[164] F Wang and J Han. Multimodal biometric authentication based on score level fusion using support vector machine. *Opto-electronics review*, 17(1):59–64, 2009.

[165] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11, 2019.

[166] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48:144–156, 2018.

[167] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *IEEE transactions on pattern analysis and machine intelligence*, 25(12):1505–1518, 2003.

[168] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer vision and image understanding*, 115(2):224–241, 2011.

[169] Qinkun Xiao and Ren Song. Action recognition based on hierarchical dynamic bayesian network. *Multimedia Tools and Applications*, 77(6):6955–6968, 2018.

[170] Chi Xu, Lakshmi Narasimhan Govindarajan, and Li Cheng. Hand action detection from ego-centric depth sequences with error-correcting hough transform. *Pattern Recognition*, 72:494–503, 2017.

[171] Roman V Yampolskiy and Venu Govindaraju. Behavioural biometrics: a survey and classification. *International Journal of Biometrics*, 1(1):81–113, 2008.

[172] Mao Ye, Qing Zhang, Liang Wang, Jiejie Zhu, Ruigang Yang, and Juergen Gall. A survey on human motion analysis from depth data. In *Time-of-flight and depth imaging. sensors, algorithms, and applications*, pages 149–187. Springer, 2013.

[173] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[174] David Zhang, Zhenhua Guo, and Yazhuo Gong. *Multispectral biometrics: systems and applications*. Springer, 2015.

[175] David D Zhang. *Automated biometrics: Technologies and systems*, volume 7. Springer Science & Business Media, 2013.

[176] Hong-Bo Zhang, Qing Lei, Bi-Neng Zhong, Ji-Xiang Du, and JiaLin Peng. A survey on human pose estimation. *Intelligent Automation & Soft Computing*, 22(3):483–489, 2016.

[177] Rong Zheng, Shuwu Zhang, and Bo Xu. A comparative study of feature and score normalization for speaker verification. In *International conference on biometrics*, pages 531–538. Springer, 2006.

[178] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 398–407, 2017.

[179] Silvia Zuffi and Michael J Black. The stitched puppet: A graphical model of 3d human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3537–3546, 2015.