FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Open-World Face Recognition

Arthur Johas Matta

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Jaime Cardoso

Co-supervisor: João Ribeiro Pinto

October 30, 2020

Open-World Face Recognition

Arthur Johas Matta

Mestrado Integrado em Engenharia Informática e Computação

October 30, 2020

Resumo

A autenticação biométrica é um campo de estudo em segurança da informação que visa proteger as informações usando características físicas ou biológicas únicas de cada indivíduo. Dentre essas características, o Reconhecimento Facial (RF) tem a vantagem de não requerer interação humana e ser um método natural de identificação empregado por humanos. No entanto, os sistemas de RF tradicionais usam uma abordagem closed-set, em que os dados de treinamento e teste pertencem ao mesmo conjunto de features e labels. Portanto, esses sistemas apresentam um mal desempenho em cenários do mundo real, onde situações imprevistas podem acontecer a qualquer momento, como a introdução de um indivíduo desconhecido. O Open-Set Face Recognition (OSFR) é uma abordagem introduzida para resolver essa deficiência, aplicando um threshold no score de confiança e atribuindo todas os dados desconhecidas a uma única classe. OSFR, no entanto, não aprende ou de outra forma tira proveito dos novos dados. O Open-World Face Recognition (OWFR) resolve essa limitação estendendo o OSFR com Class Incremental Learning (CIL), aprendendo novas identidades a partir de individuos anteriormente desconhecidas. Embora ambas as abordagens lidem com identidades desconhecidas, sua precisão geral e tempo de execução sofreram significativamente em comparação com os sistemas tradicionais como consequência. Portanto, este trabalho propõe três algoritmos OWFR rápidos e diretos que lidam com a questão do tempo de execução enquanto equilibra a precisão. O primeiro algoritmo calcula uma distância característica por característica entre os dados. O segundo algoritmo usa Gaussian Mixture Models (GMMs) para calcular um teste de razão de verossimilhança. O terceiro algoritmo se concentra em cenários de vídeo e estende o anterior aplicando um mecanismo de análise temporal. Os experimentos realizados empregam imagens estáticas e de vídeo e mostram que a eficiência dos métodos propostos é compatível com o estado da arte e apresentam tempos de execução mais rápidos.

Abstract

Biometric authentication is a field of study in information security that safeguards information by using physical or biological traits unique to each individual. Among these traits, Facial Recognition (FR) has the advantage of not requiring human interaction and being a natural identification method employed by humans. However, traditional FR systems use a closed-set approach where both training and test samples belong to the same label and feature space. Therefore, these systems perform poorly in real-world settings where unpredicted situations may happen at any time, such as the introduction of an unfamiliar individual. Open-Set Face Recognition (OSFR) is an approach introduced to address this deficiency by applying a threshold on the confidence score and assigning all those unknown samples to a single class. OSFR, however, does not learn or otherwise takes advantage of newly available data. Open-World Face Recognition (OWFR) tackle this limitation by extending OSFR with Class Incremental Learning (CIL), learning new identities from the previously unknown subjects. Although both approaches handle unfamiliar identities, their overall accuracy and execution time suffered significantly compared to the traditional systems as a consequence. Hence, this work proposes three fast and straightforward OWFR algorithms that tackle the execution time issue while balancing accuracy. The first algorithm calculates a feature-by-feature distance between the samples. The second algorithm uses Gaussian Mixture Models (GMMs) to calculate a likelihood ratio test. The third algorithm focuses on video scenarios and extends the previous by applying a temporal analysis mechanism. The experiments conducted employ both static- and video-images and show that the proposed methods' efficiency is consistent with the state-of-the-art while having faster execution times.

Acknowledgements

First of all, I would like to thank my parents for all the care, love, and support they have provided me. Without them, I wouldn't be the person I am today; neither would have reached where I am now. Second, thanks to my friends, whose continuous support was fundamental for my life, and whose companionship helped me get through hard moments. Last but not least, thank you, professors, for the support, patience, effort, and dedicated work. You have taught me almost all the knowledge I keep nowadays and will carry for the rest of my life.

Arthur Johas Matta

"Power comes not from knowledge kept but from knowledge shared"

Bill Gates

Contents

List of Figures

List of Tables

Abbreviations and Symbols

CIL	Class Incremental Learning
EM	Estimation-Maximization
EVM	Extreme Value Machine
FCN	Fully Connected Networks
FR	Face Recognition
GMM	Gaussian Mixture Model
KKC	Known Known Classes
KUC	Known Unknown Classes
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LSH	Locality-Sensitive Hashing
MAP	Maximum A Posteriori
OSFR	Open-Set Face Recognition
OSR	Open-Set Recognition
OWFR	Open-World Face Recogntion
OWR	Open-World Recognition
PLS	Partial Least Squares
PSEI	Pattern Specific Error Inhomogeneities
UBM	Universal Background Model
UUC	Unknown Unknown Classes

Chapter 1

Introduction

Information security is the practice of safeguarding information from unauthorized modification, disruption, destruction, and inspection. It typically entails the use of something you possess (e.g., security token and ID cards), something you know (e.g., passwords and PINs), or something you are (e.g., fingerprint and gait). The latter approach, commonly referred to as biometric authentication, has replaced the former methods in most settings as those are more prone to be stolen or replicated by an unapproved individual, thus unfulfilling their purpose.

Biometric authentication measures and matches the unique physical (e.g., gait, keystroke rhythm) and biological traits (e.g., fingerprints, iris pattern) of a person to verify their identity. It has the advantage of being reasonably accurate, fast, natural, and non-intrusive; hence corporations and governmental entities begun to implement it widely. Biometric systems are generally composed of sensors, quality assessment and feature extraction, matching, and database modules (Fig. 1.1). The former captures the individual's characteristics, which are evaluated and processed by the subsequent module, transforming the raw data into numerical, comparable data. During enrollment, the database module stores and allocate these features into known identities. During verification, the matching module compares the new data against the database and returns a similarity score (e.g., probability) for one or more entries in the database.

Amidst the several biometric traits, face recognition (FR) has the benefit of not requiring human interaction, consequently enabling its identification from a distance. This benefit allowed law enforcement agencies to recognize individuals among crowds without them even being aware of it, thus finding missing children and uncovering criminals. Face recognition systems (Fig. 1.2) detect prominent facial features (also known as facial landmarks or fiducial points) such as the eye corners, the nose tip, and the mouth corners, and use them to crop the face and adjust its orientation. From the face image, the system measures distinguishable facial landmarks, such as the distance between the eyes, the width of the nose, and the shape of the cheekbones, transforming them into a feature vector, called a faceprint, representing the face in the database. During verification, the system matches the unidentified faceprint against the database and returns a similarity score between the new sample and the known identities.

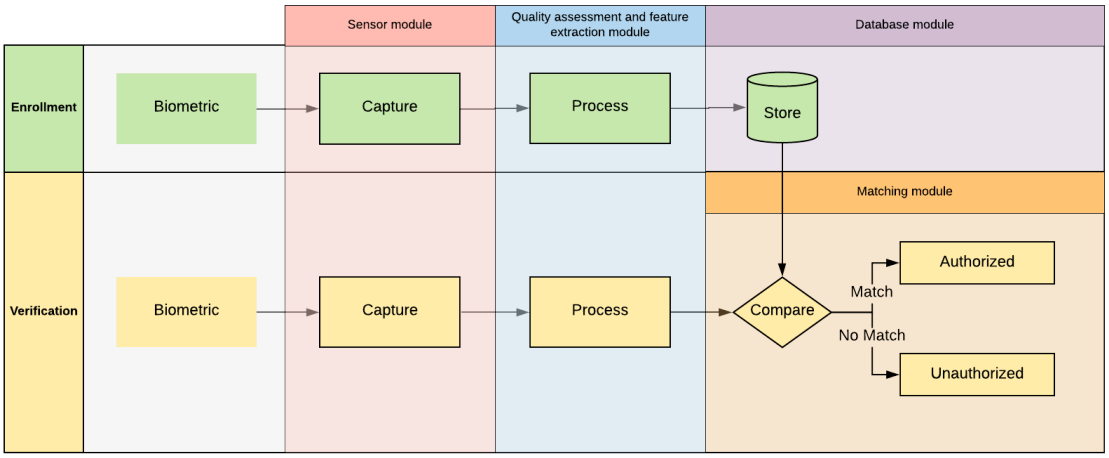


Figure 1.1: Biometric authentication system architecture

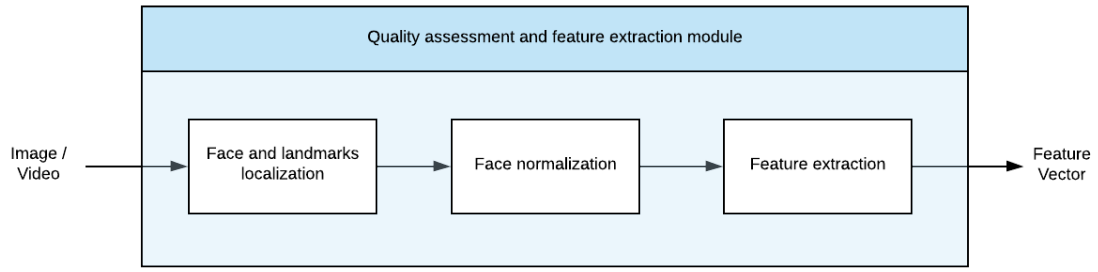


Figure 1.2: Face recognition system architecture

FR became popular in the early 1990s by introducing the eigenfaces method [40] witnessing accelerated growth until the beginning of the 2000s when the facial recognition community realized that current approaches fail to address uncontrolled facial changes. In a controlled environment, details such as pose, lighting, expressions, and occlusion are not present and, therefore, identifying and recognizing the face becomes a more straightforward task. However, in real-world settings, the facial images are acquired in-the-wild, and no control over the face exposure exists. For example, video surveillance is subject to climate conditions, lighting variations, partial occlusion, and many other factors. Straightforward methods were not suitable for such scenarios as they typically represent the face using one or two layers, e.g., filtering responses and histogram of feature codes. This panorama changed with Viola and Jones, whose boosting based face detection method [42], still used in some modern approaches, made FR nearly feasible in those settings. Nonetheless, the FR models' accuracy increased slowly, and most of these models addressed only one aspect of the unconstrained facial changes; there was no technique to address them all. Finally, with the development of Deep Neural Networks and improved hardware capability, that scenario evolved into an entire new patamar.

Deep Learning for face recognition first became popular in 2012 with the AlexNet [13] algorithm's debut in the ImageNet competition, winning by a large margin. Previous techniques

usually recognized human face by one- or two-layer representations. Deep Learning, on the other hand, employs a cascade of multiple layers, learning various levels of abstraction. This architecture increased the algorithm's robustness to face pose, lighting, and expression changes. Deep-learning-based techniques reshaped the FR landscape by changing algorithms' designs, databases, application scenarios, and even the evaluation protocol. In just a few years, these methods were able to surpass human performance. Figure 1.3 shows the evolution of face recognition, some famous approaches, and the respective range of accuracy they could reach.

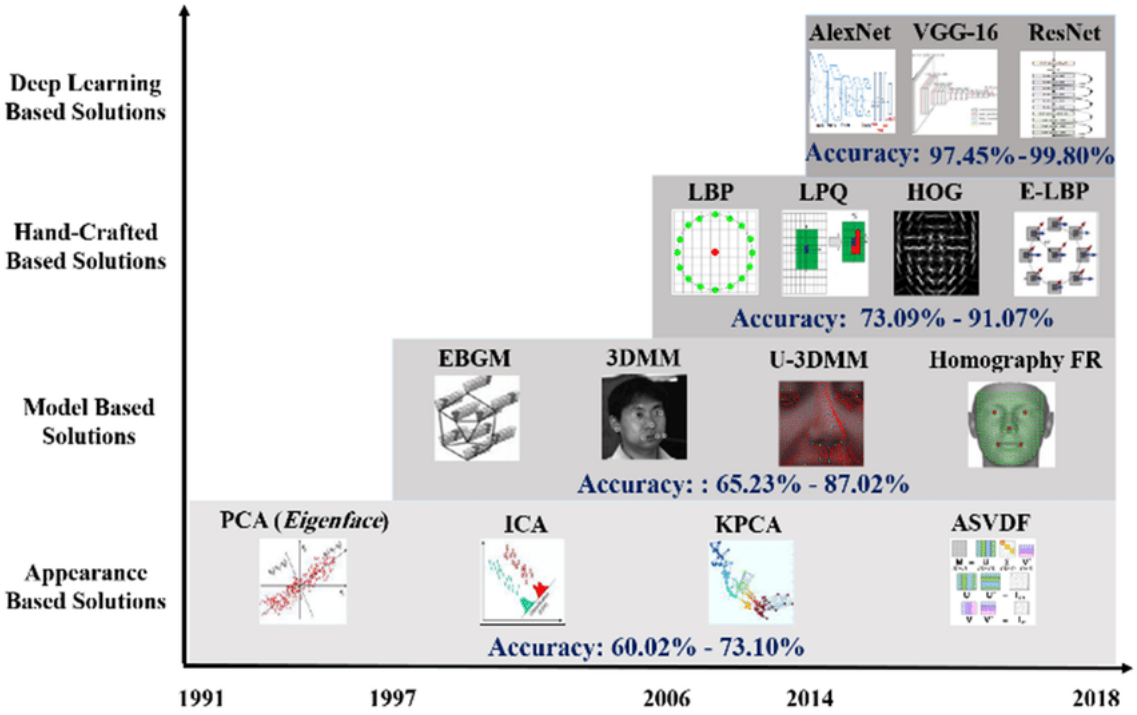


Figure 1.3: Evolution of Face recognition [32]

Nevertheless, a challenge arises when dealing with unfamiliar faces (i.e., faces never seem by the system). Conventional algorithms operate on a closed-set setting and associate familiar and unfamiliar faces to the known class with the highest confidence score (Fig. 1.4a). The presence of never-seen-before instances can confuse the system and thus decrease its efficiency.

One technique to solve this challenge arose with the introduction of Open-Set Face Recognition (OSFR), formalized by Günther et al. [9]. In OSFR, the confidence score is thresholded, and an instance is assigned to a known class only if the correspondent score is higher than the threshold. Otherwise, the system attributes the sample to an "unknown" group together with all unfamiliar individuals (Fig. 1.4b). However, what if instead of allocating all these never-seen-before instances to a single class, the algorithm could learn and create new identities for them? That is the key-concept behind Open-World Face Recognition (OWFR), introduced by Bendale and Boult [2]. Unlike the OSFR, in OWFR, if the confidence score is lower than the threshold, instead of attributing the instances to an "unknown" group, it arranges them into new known classes (Fig. 1.4c). Thus, as the OWFR systems are continually learning new identities, if a previously

unfamiliar face appears to the system again, it'll be now classified as familiar.

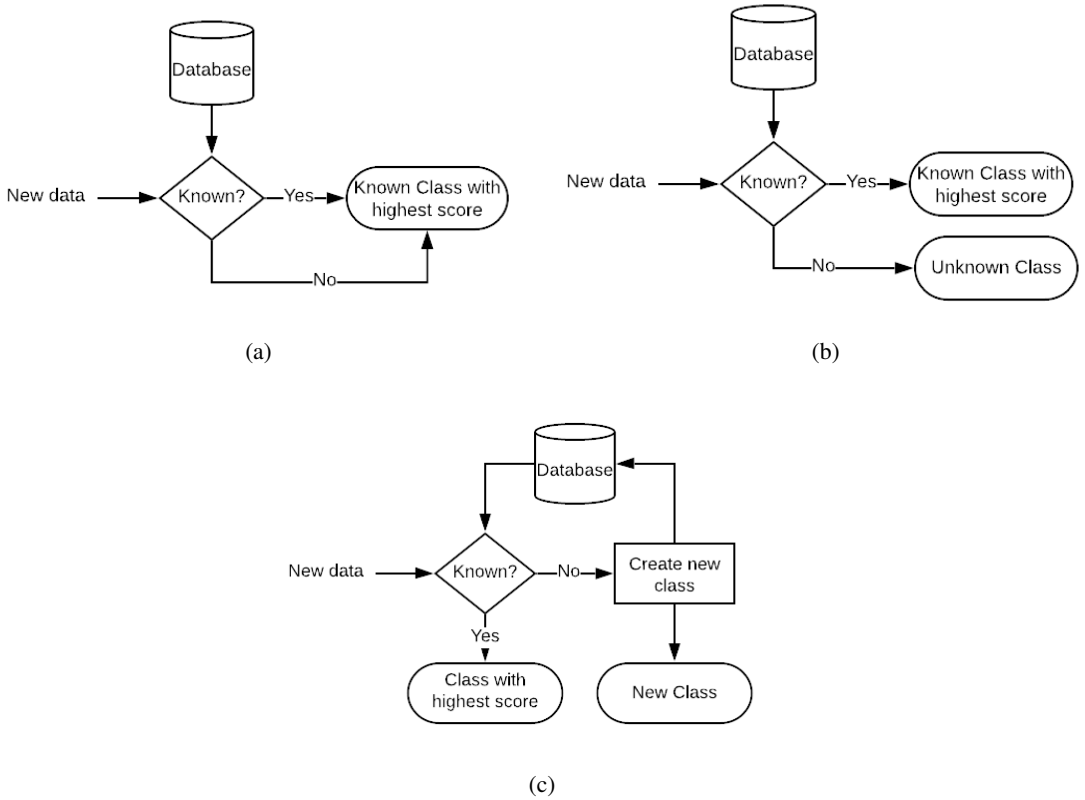


Figure 1.4: How (a) Closed-Set, (b) Open-set, and (c) Open-world Face Recognition algorithms handle unknowns

In this context, this work proposes three open-world approaches. The former represents every sample as a curve and assigns it to a class based on the number of features whose distance to the class' center is smaller than a threshold. The second approach employs a Gaussian Mixture Model (GMM) to represent each group and a Universal Background Model (UBM) as a normalization factor. The latter adds a sliding window mechanism to reduce misclassification during execution time and a Maximum A Posteriori (MAP) estimation to update existing models incrementally. Experimentally these approaches achieved results comparable to conventional face recognition methods while also managing unfamiliar individuals.

Besides the introduction, three more chapters compose the remainder of this paper. Chapter 2 introduces some state-of-the-art approaches to Open-World Face Recognition; Chapter 3 details the three algorithms developed; Chapter 4 describes the experimental settings, the results obtained, and a comparison to a state-of-the-art method.

Chapter 2

Related work

Biometrics is the science of establishing one's identity based on physiological and behavioral traits intrinsic to an individual, such as the face, iris pattern, fingerprint, gait, and keystroke dynamics. Biometric systems employ these traits to verify or attribute an identity to an individual, with the primary goal of preventing unauthorized access to classified information. Traditional identification methods include knowledge-based (e.g., passwords) and token-based (e.g., ID cards) mechanism. However, these mechanisms are flawed since they are easily forgotten or stolen, thus compromising the intended security [12]. By using intrinsic characteristics of a person, however, biometric systems provide a natural and reliable identification method. Among these characteristics, facial recognition has been one of the most popular (Fig. 2.1), mainly because it does not require human cooperation; therefore, its images can be acquired discreetly from a distance [17].

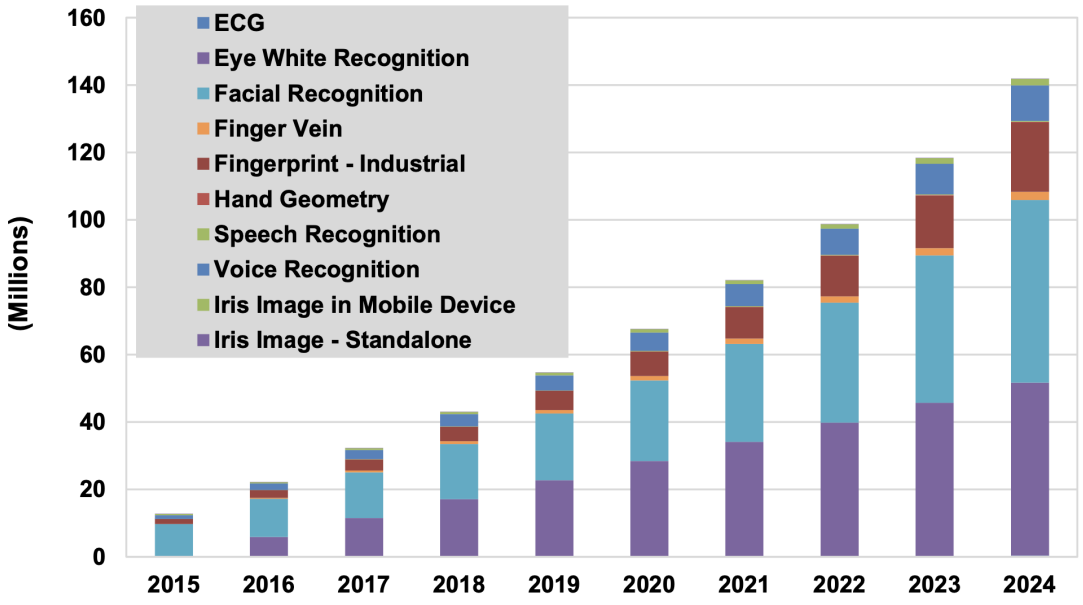


Figure 2.1: Enterprise Biometrics Devices and Licenses by Modality, World Markets: 2015-2024 [39]

In the 1960s, Bledsoe et al. [3] introduced the first face recognition approach by manually marking facial landmarks, such as the eye centers and the nose and mouth positions, and feeding to a computer that compared images by measuring the distance between those points. As an unnamed intelligence agency was funding this project, little to no information was available to the research community. Together with the computational power limitations of that time, the face recognition field evolved slowly. It was only in the late 1980s and early 1990s that it began drawing attention again with the introduction of the Eigenface method. First presented by Sirovich et al. [34], this method produced a low-dimensional representation of the face by employing a principal component analysis to extract a set of essential features. Turk et al. [40] later expanded this method by automating its process and allowing computers to perform eigen-decomposition on several images by calculating a covariance matrix's eigenvectors. However, this and similar low-dimensional representation methods fail to address changes associated with unconstrained scenarios, such as lighting and expression variations. This issue gave rise to local-feature-based face recognition. Scale-Invariant Feature Transform (SIFT) [24], Gabor filters [18], and Local Binary Patterns (LBP) [1], among others, could deliver robust performance by extracting invariant properties of local filtering. Yet, handcrafted features experienced a lack of distinctiveness and compactness, being replaced by learned-based local descriptors [4, 15] in the early 2010s, solving these two problems. But the issue with facial appearance variations from unconstrained scenarios persisted, as most methods only addressed one aspect of the problem.

In 2012, however, the FR landscape experienced a substantial impact due to the introduction of AlexNet and its deep learning technique [13]. This technique divides the computations related to feature extraction and transformation into multiple layers of processing units that amplify or dampen the received input based on a set of coefficients, presenting robustness to changes related to face pose, lighting, and expression, as can be seen in Figure 2.2. Two years later, DeepFace [38], a deep learning algorithm developed by Facebook, almost surpassed the human-level performance in face detection for the first time, reaching an accuracy of 97.35% versus the 97.53% achieved by humans. Its method consisted in comparing face descriptors, obtained by applying the same convolutional neural network to pairs of faces, using the Euclidean distance. Since then, deep learning became a trend, with several approaches being developed in the following years using this technology. Hence, in just three years, the accuracy experienced a tremendous boost to above 99%. For instance, Google proposed a method called FaceNet [31] that learns a Euclidean embedding per image, with the embedding space directly corresponds to face similarity, achieving an accuracy of 99.63% on the widely used Labeled Faces in the Wild (LFW) database [11]. Table 2.1 presents several state-of-the-art deep-learning-based approaches and their respective accuracies.

A known limitation regarding any traditional recognition/classification algorithm lies in employing a closed identity set where both training and testing data belong to the same label and feature spaces. However, the unpredictability of realistic scenarios allows for unseen situations to emerge at any given time. In the face of such circumstances, conventional algorithms typically perform inefficiently.

The challenging nature of dealing with the unknown instigated the research community. Among

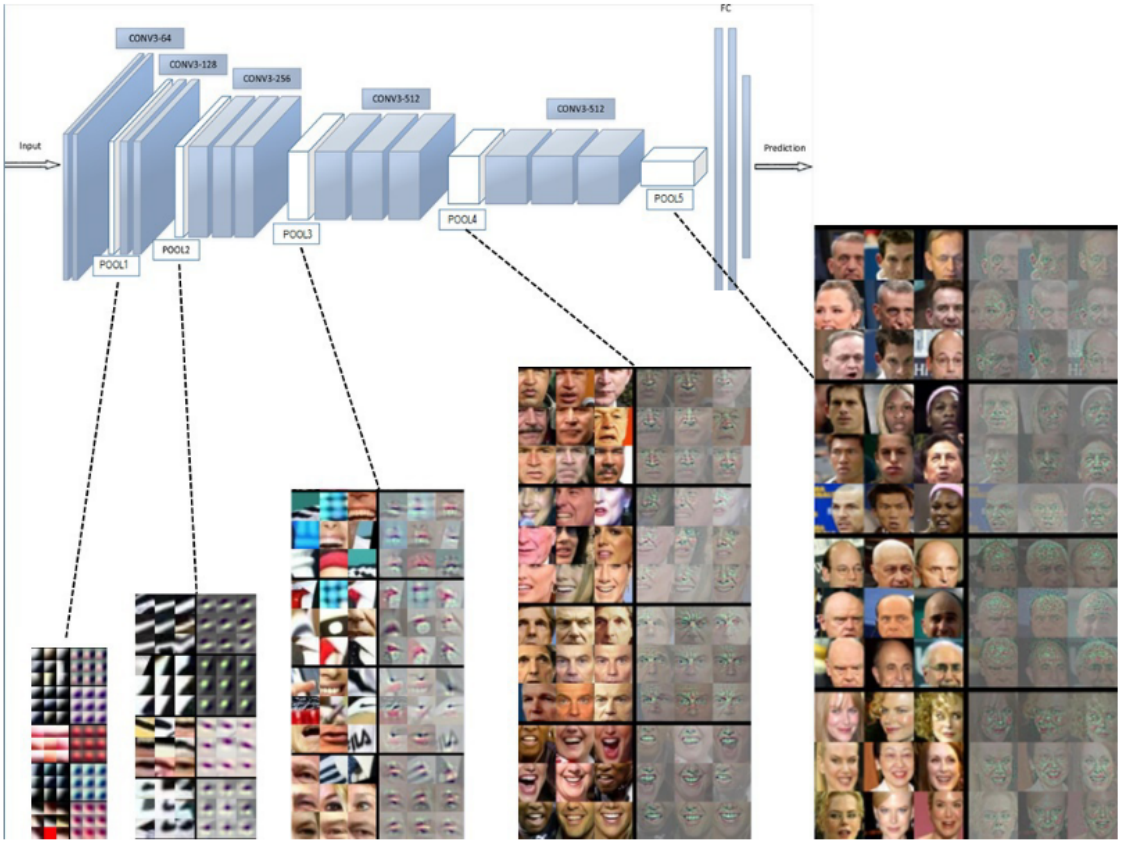


Figure 2.2: Example of a deep neural network with its several layers, each layer learning a deeper intermediate representation of the face. Note how this architecture shows robustness to face pose, lighting, and expression changes [46]

many proposed solutions, Open-Set Recognition (OSR) was one with the highest notoriety, mostly because of its need for strong generalization. Its concept was formalized by Scheirer et al. [30], who proposed a 1-vs-set machine solution and introduced the open space risk to account for what lies beyond the known closed space.

The OSR introduced three categories of classes: KKCs, KUCs, and UUCs. The former represents identities known and of high interest to the system, while the second comprises known but uninteresting individuals, and the latter includes subjects never seen before. Establishments like airports are a great example of using these categories because they need to classify travelers with increasing interest levels. For example, a list of wanted criminals has the highest interest level, therefore, comprising the KKCs. On the other hand, although known, the airport personnel don't have a high-interest level and should not confuse the system, thus, belonging to the KUCs. Lastly, the passengers who do not stay at the airport are unknown and uninteresting and should be discarded, comprising the UUCs.

Open-set recognition also instigated the field of face recognition. Günther et al. [9] formulated an Open-Set Face Recognition (OSFR) protocol and evaluated three approaches using threshold verification-like scores, Linear Discriminant Analysis (LDA), and an Extreme Value Machine

Method	Publication Year	Architecture	Accuracy (%)
DeepFace [38]	2014	AlexNet	97.35
DeepID [37]	2014	–	97.45
DeepID2 [36]	2014	AlexNet	99.15
DeepID3 [35]	2015	VGGNet-10	99.53
FaceNet [31]	2015	GoogleNet-24	99.63
VGGFace [25]	2015	VGGNet-16	98.95
Center Loss [47]	2016	LeNet+-7	99.28
L-Softmax [21]	2016	VGGNet-18	98.71
Range Loss [50]	2016	VGGNet-16	99.52
L2-Softmax [28]	2017	ResNet-101	99.78
Normface [44]	2017	ResNet-28	99.19
CoCo Loss [22]	2017	–	99.86
vMF Loss [10]	2017	ResNet-27	99.68
Marginal Loss [6]	2017	ResNet-27	99.48
SphereFace [20]	2017	ResNet-64	99.42
CCL [27]	2018	ResNet-27	99.12
AMS Loss [43]	2018	ResNet-20	99.12
CosFace [45]	2018	ResNet-64	99.33
ArcFace [5]	2018	ResNet-100	99.83
Ring Loss [53]	2018	ResNet-64	99.50
AdaCos [51]	2019	ResNet-50	99.71
AdaptiveFace [19]	2019	LResNet50A-IR	99.62
RegularFace [52]	2019	ResNet-20	99.61
UniformFace [8]	2019	ResNet	99.80

Table 2.1: Several state-of-the-art methods, their architecture, and their accuracy in the LFW data set

(EVM) probabilities. They concluded that the three methods performed well in closed-set scenarios, but only the LDA and EVM could deal with KUCs, and only the latter satisfactorily handled the UUCs. Li and Wechsler [16] developed the TCM-KNN (Transduction Confidence Machine - K Nearest Neighbors) method by employing a relation between transduction and Kolmogorov complexity. Vareto et al. [41] combined Locality-Sensitive Hashing (LSH) with Partial Least Squares (PLS) and Fully Connected Networks (FCN) to create the HPLS and HFCN algorithms, respectively, accomplishing better discrimination between positive and negative samples.

Although open-set face recognition handled the problem with unknown identities, it does not provide an incremental scenario nor scale gracefully with the number of classes. With that in mind, Bendale and Boult [2] extended the open-set concept by combining it with Class Incremental Learning (CIL), introducing and formally defining the Open-World Recognition (OWR) problem. In this scenario, a system would need not only to handle UUCs but also to decide which new samples to label to update the classifier. Furthermore, it should tackle "open space risk" as the newly labeled data tends to move away from the known data into open space.

Besides introducing the OWR problem, Bendale and Boult [2] proposed a metric learning algorithm to solve it. Nearest Non-Outlier (NNO) extends the traditional Nearest Class Mean

(NCM) approach to tackle open space risk while balancing accuracy. However, Rosa et al. [29] stated that several metric learning algorithms, like NNO and NCM, estimate its parameters on an initial closed set and keep them unchanged as the problem evolves, contradicting the very own definition of OWR. Therefore it was necessary to learn the parameters in an online fashion. Hence, they extended three algorithms, the Nearest Class Mean, the Nearest Non-Outlier, and the Nearest Ball Classifier, to update their metric and novelty threshold online. Following this line of thought, Doan and Kalita [7] developed a similar approach with the difference that, aside from employing their solution for the incremental addition of new classes, they optimized the nearest neighbor search for determining the most proximate local balls. Lonij et al. [23] approached OWR differently, using knowledge graph embedding to add semantic meaning to images by employing smoothing constraints in the graph embedding loss function and an attention-based scheme to improve predictions of novel graphs. For the action recognition task, Shu et al. [33] proposed the Open Deep Network (ODN), which applies a multi-class triplet thresholding technique to detect new classes and then dynamically reconstructs the classification layers of the network by appending predictors for new categories continually. Most existing OWR systems need some form re-training to incorporate new classes in the overall model. As an alternative, Xu et al. [49] proposed a meta-learning algorithm that only requires the training of a meta-classifier. It can then continually include new classes when sufficient labeled data is available and detect/reject later unseen subjects.

Chapter 3

Proposed methods

This section introduces the baseline as well as the three approaches proposed by this paper. The former is an adaptation from Rosa et al.'s oNNO algorithm [29]. This paper's first approach mainly focuses on allowing the visual picturing of the N-dimensional data into a bidimensional space. The other two methods extend the base approach by employing Gaussian Mixture Models and a sliding window technique, respectively. These algorithms will be applied to still- and video-images scenarios to predict the label to which each face belongs.

3.1 Baseline

The Online Nearest Non-Outliner algorithm (oNNO) by Rosa et al. [29] was selected and adapted to the face recognition environment as the baseline. It extends the Nearest Non-Outliner (NNO) from Bendale et al. [2] by learning the metric and the confidence thresholds incrementally.

Given $y \in Y = \{y_1, y_2, ..., y_k\}$, where Y is the set of all possible classes, with μ_y representing the respective class' center, oNNO defines the probability for class y as the soft-max function:

$$p(y|x) = \frac{exp(-\frac{1}{2}\, d_w(x, \mu_y))}{\sum_{y' \in Y} exp(-\frac{1}{2}\, d_w(x, \mu_{y'}))} \tag{3.1}$$

where $d_w(x, \mu)$ is the squared low-rank Mahalanobis distance, parameterized by W, given by:

$$d_w(x, \mu) = (x - \mu)^T W^T W (x - \mu) \tag{3.2}$$

where x and μ are N-dimensional vectors, and the metric $W \in R^{MxN}$, with $M \leq N$, acts as a regularizer to project the feature vector into a low-dimensional space, allowing faster computations and more compact representations. M is the "rank" of the metric, i.e. the number of dimensions of the low-dimensional space that the metric projects the data into.

In the open-world scenario, the number of classes is unknown upfront, and therefore the model's parameters need to be learned in an online fashion. Thus, for each pair $(x_t, y_t) \in R^{NxY}$ inputted, the model updates its parameters using the formulations:

$$\mu_{y_t}^{t+1} = \left(1 - \frac{1}{n(y_t)}\right) \mu_{y_t}^{t} + \frac{1}{n(y_t)} x_t \tag{3.3}$$

$$W^{t+1} = (1-\gamma) W^t + \gamma \nabla_{W^t} \; logp(y_t|x_t) \tag{3.4}$$

where $n(y_t)$ is the number of instances to class y_t at time t (including the current sample), and γ is a fixed learning rate in the range [0.0, 1.0]. Note that the initial value for μ^1 is equal to the first sample x_1, while the initial metric W^1 is the truncated identity matrix. The gradient of W w.r.t. the model is a single step of stochastic gradient descent:

$$\nabla_{W^t} \; logp(y_t|x_t) = \sum_{y \in Y} (p(y_t|x_t) - [[y_t = y]]) \; W^T \; (\mu_y^{t+1} - x_t)(\mu_y^{t+1} - x_t)^T \tag{3.5}$$

With the Iverson brackets [[]] denoting the indicator function, returning 1 if the condition inside it is True or 0 otherwise. The prediction confidence formulation employed by oNNO is similar to an RBF-kernel, thus having the advantage of being strictly bounded:

$$C_y(x_t, \theta^t) = exp\left(-\frac{1}{2\theta^t} d_{W^t}(x_t, \mu_y^t)\right) \tag{3.6}$$

This equation attributes a confidence value between [0,1] to a sample x_t at time t for class y. The bandwidth parameter θ is learned incrementally and represents the expected value of distances to all class means:

$$\theta^{t+1} = \left(1 - \frac{1}{t}\right) \theta^t + \frac{1}{t} \sum_{y \in Y} d_{W^t}(x_t, \mu_y^t) \tag{3.7}$$

The algorithm accepts the instance x_t as belonging to class y if its confidence is higher than a threshold parameter τ; otherwise, it assigns the instance to the unknown class. This parameter τ can be seen as the expected value of confidence since it considers the mean between the prediction scores since the last added novel class:

$$\tau^{t+1]} = \begin{cases} 0 & \text{if } y_t \text{ is from a novel class} \\ \left(1 - \frac{1}{t^*}\right) \tau^t + \frac{1}{t^*} C_{y_t}(x_t, \theta^t) & \text{otherwise} \end{cases} \tag{3.8}$$

where x_t is the current instance, and t^* is the number of samples since the last novel class. The assignment of x_t to class y triggers the update of the correspondent class mean, the metric, the bandwidth, and the threshold.

Note that it is necessary to create new identities for never-seen-before individuals to adapt the algorithm to the Open-World settings. Therefore, instead of assigning the unknown samples to a disposal class, the adapted algorithm allocates them into temporary classes that become new known classes after reaching a minimum number of instances; otherwise, they get discarded after a period. The process of assigning a sample to a temporary class is the same as described above, using a confidence score for each class and thresholding it.

3.2 First Approach

The first approach employs a data simplification by converting any N-dimensional sample into a curve on a bidimensional space. By doing so, it portrays data in a way that humans can picture. It then assigns an instance to a class by calculating a unidimensional distance feature-by-feature between an instance's curve and a centroid's curve. Figure 3.1 illustrates the representation scheme for this approach. The experiments conducted showed that this approach is practical and obtained results similar to state-of-the-art methods while also reducing the execution time considerably compared to the baseline.

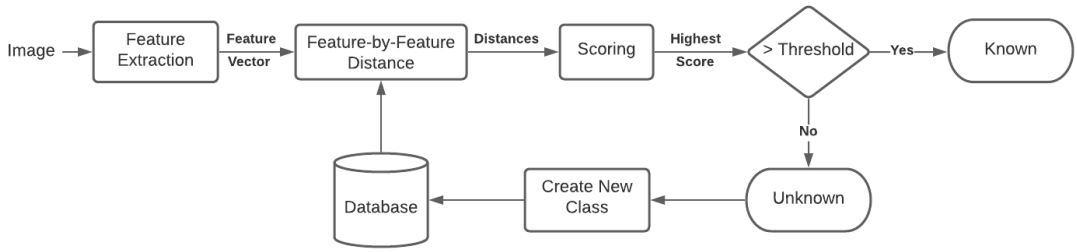


Figure 3.1: Representation scheme of the first variant.

Given an instance $x \in X = \{x_1, x_2, ..., x_i\}$, with $X \in \mathbf{R}^N$, its visual representation is given by a graph with the x-axis representing the N features and the y-axis the nth-feature value. Figure 3.2 shows an illustration of five samples with 100 elements.

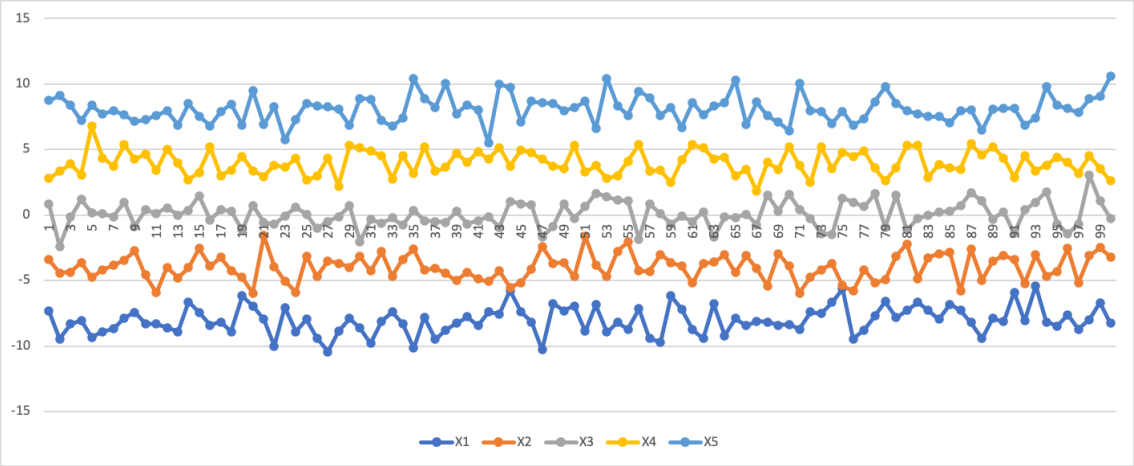


Figure 3.2: Curve representation of five samples with 100 features

This approach allows the same class to have multiple clusters, each with its centroid, using a label encoder that maps each class to an index. Therefore, given $Y = \{y_1, ..., y_j\}$ the set of all classes, the index representation is given by $I = \{i_1^1, i_1^2, i_2^1, ..., i_k^g\}$, where i_k^g is the index pointing to cluster g of class k, with u_k^g representing the cluster's centroid. For simplification, consider $I = \{i_1, i_2, ..., i_c\}$ the set of all indexes with u_c representing the correspondent center. The algorithm

first step calculates a threshold for each feature indicating the interval to which an element can be considered related to that cluster:

$$T^n = F * \sigma^n \tag{3.9}$$

where F is a constant scale factor used to adjust the interval's width, and σ^n is the standard deviation of the nth-feature considering all μ_c. Figure 3.3 illustrates an example of applying the threshold applied to three class centroids.

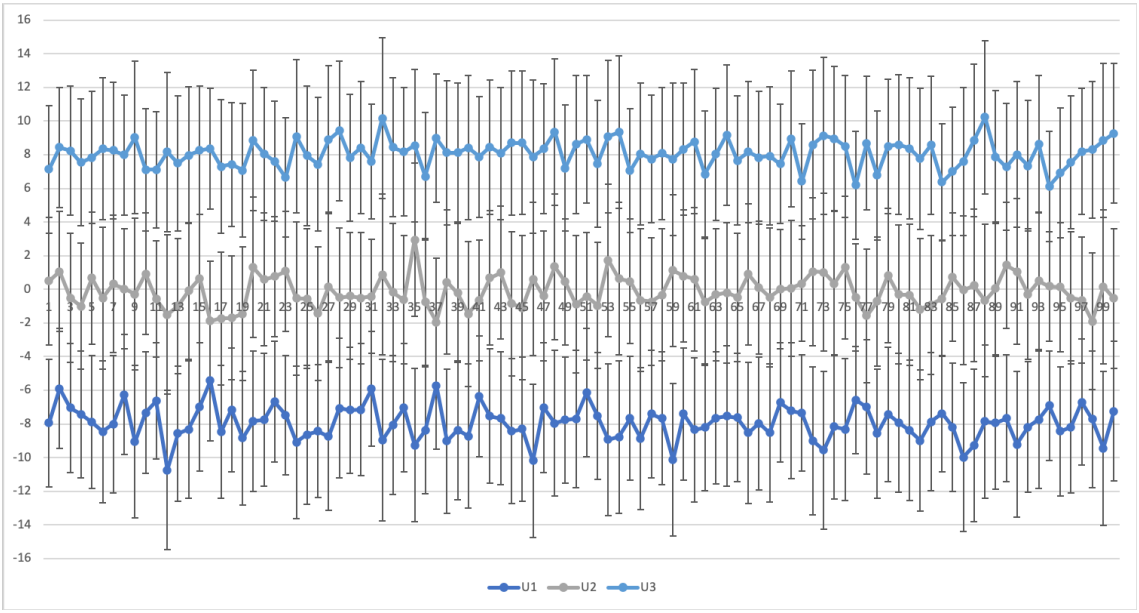


Figure 3.3: Thresholds of three class centroids

The algorithm's second step is calculating a unidimensional distance feature-by-feature between the instance x and each class' center μ_c, as follows:

$$D^n_{xc} = |\mu^n_c - x^n| \tag{3.10}$$

where μ^n_c and x^n are the nth-feature of μ_c and x, respectively. This distance is employed to calculate a score value between [0, N] to an instance x for cluster c using the formulation:

$$S_{xc} = \sum_{n=1}^{N} [[D^n_{xc} \leq T^n]] \tag{3.11}$$

where [[]] symbolizes the Iverson Brackets that returns 1 if the condition inside is True and 0 otherwise. The algorithm accepts the instance x as belonging to cluster c if $S_{xc} \geq H$, where H is a threshold between [1, N], updating the respective centroid using equation 3.3. Otherwise, it assigns x to one of the unknown classes $U = \{u_1, u_2, ..., u_i\}$ by repeating the same procedure described above but replacing the set I for U. If $S_{xu} < H$, create a new label u_{i+1} and attribute x to it.

The final step is, after some period, converting u_i into a known class when it reaches a minimum number of samples or discarding it otherwise. When a new cluster is created, an index pointing to it is also generated.

3.3 Second Approach

The second approach employs Gaussian Mixture Models (GMMs) instead of the curve representation aforementioned. The GMMs increase the algorithm's robustness as they don't rely on distance-from-cluster-center, which may lead to poor performance in real-world scenarios, and allow the clusters' shape to be flexible, therefore better fitting the data. Moreover, this method employs a likelihood ratio test using a Universal Background Model to discriminate faces more clearly. For reasons of simplification, this algorithm used the GMM library provided by Sklearn [26]. Figure 3.4 illustrates the representation scheme for this variant.

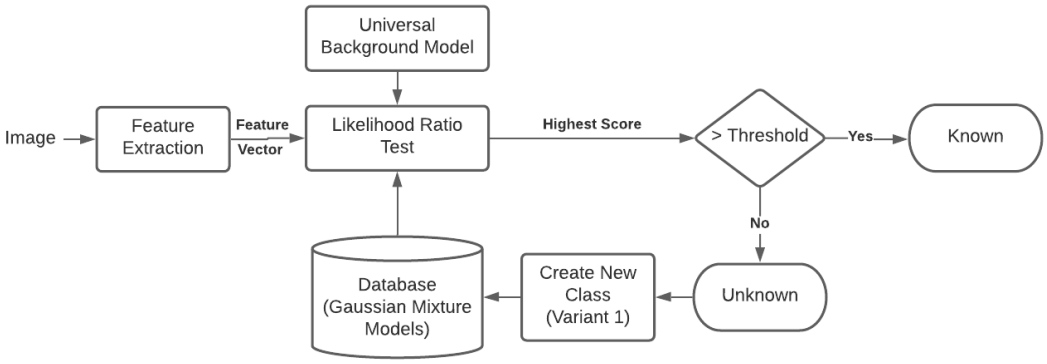


Figure 3.4: Representation scheme of the second variant.

Gaussian Mixture Models (GMMs) are probabilistic models that employ a finite number of Gaussian distributions to model any arbitrarily-shaped cluster more accurately. They apply an Expectation-Maximization (EM) algorithm to calculate a weight encoding the probability of membership to each arrangement for each point and then use it to update each cluster's parameters, guarantying a convergence to a local maximum. Universal Background Models (UBMs) are typically target-independent models constructed by fitting a GMM with samples from several clusters to create a global model. This model can increase the algorithm performance and suitability to real-world scenarios as it functions as a normalization factor for the rejection/acceptance decision and can also be used to derive a new target-dependent model.

The general notation of a GMM is $\lambda = \{\omega_i, \mu_i, \Sigma_i\}$, with $i = 1, ..., M$, where M is the number of Gaussian components used, and ω_i, μ_i, and Σ_i are the component's weight, mean vector, and covariance matrix, respectively. The mean vector defines the gaussian distribution's location in space, while the covariance matrix determines its density contours' direction and length. Typically, the covariance matrix can be full, tied, diagonal, or spherical (Figure 3.5). Full and tied covariance matrices allow the distributions to adopt any shape; however, the former enables each arrangement

to assume a different configuration, which is not permitted by the latter. The diagonal matrix binds the contour axes' orientation to the coordinate axes (although the eccentricities may vary between components), while the spherical, as the name suggests, restrict the distribution to a spherical shape.

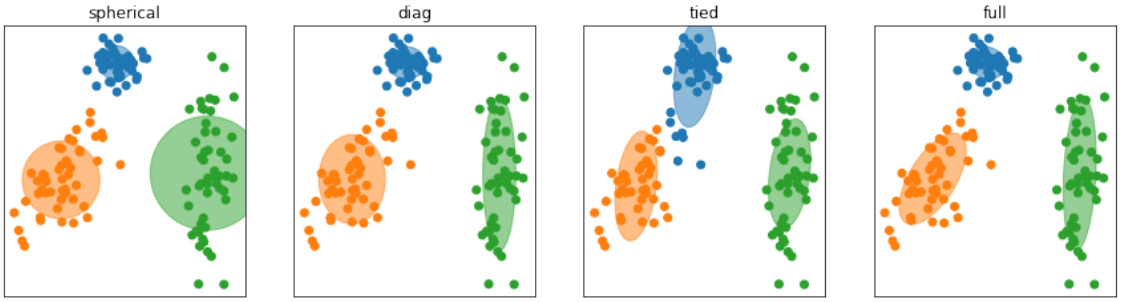


Figure 3.5: Covariance types for Gaussian Mixture Models

Unlike the previous approach, the algorithm presented here implements a model based on GMMs. It creates only one GMM for each class and one to act as UBM. Both employ a diagonal covariance matrix since it provides some degree of freedom to the clusters' shape while utilizing the same number of parameters as the feature vectors, suiting a scenario with limited data like this work. During training, the algorithm feeds each GMM with the corresponding data and the UBM with all the available data. However, instead of calculating a distance to the cluster's center during enrollment, it applies a likelihood ratio test.

The likelihood ratio (LR) test assesses the fitness between two models by employing a hypothesis test: given an observation, O, and a person, P, define the hypothesis $H_0 = O$ *is from* P, and $H_1 = O$ *is not from* P; then calculating the ratio between the probability density function (or likelihood) for both hypothesis:

$$LR(O) = \frac{p(O|H_0)}{p(O|H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{reject } H_0 \end{cases} \tag{3.12}$$

In this work, the GMM describes a feature's distribution derived from the corresponding person, hence characterizing a hypothesis, and each feature vector represents an observation. Therefore, the LR test becomes:

$$LR(x,P) = \frac{p(x|\lambda_P)}{p(x|\lambda_{\overline{P}})} \tag{3.13}$$

where λ_P and $\lambda_{\overline{P}}$ are models denoting the weights, means, and covariance matrices of the corresponding GMM. The $p(x|\lambda_P)$ is a probability density function given by the weighted sum of the GMM's M components:

$$p(x|\lambda) = \sum_{i=1}^{M} \omega_i * g(x|\mu_i, \Sigma_i) \tag{3.14}$$

with each component $g(x|\mu_i,\Sigma_i)$ being a function of the form:

$$g(x|\mu_i,\Sigma_i) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma_i|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}\,(x-\mu_i)'\,\Sigma_i^{-1}\,(x-\mu_i)\right\} \tag{3.15}$$

The issue, however, is how to define the likelihood of the alternative hypothesis. λ_P can be acquired using the training data, but $\lambda_{\overline{P}}$ must encompass the entire space of possible alternatives to person P. One could use a set of alternative models $\{\lambda_1,...,\lambda_N\}$ such that:

$$\lambda_{\overline{P}} = F(p(x|\lambda_1),...,p(x|\lambda_N)) \tag{3.16}$$

where F is a function, such as average or maximum, of the likelihood values. However, this is not suitable for applications with many alternatives, and thus a second approach must be employed. The use of UBM allows creating a model pooling samples from many different groups, hence having one single model to represent all alternative hypotheses:

$$p(X|\lambda_{\overline{P}}) = p(x|\lambda_{ubm}) \tag{3.17}$$

Thus, given an instance $x \in X$, this work calculates the LR for each class using both the corresponding GMM and the UBM:

$$LR(x,y) = \frac{p(x|\lambda_y)}{p(x|\lambda_{ubm})} \tag{3.18}$$

accepting x as belonging to the class y' with the highest LR if its $LR(x,y') > \theta$, where θ is a provided threshold ranging between $]0.0,+\infty[$. In case all classes reject the instance, the algorithm assigns it to one of the unknown classes, updating the respective class' center using equation 3.3. This assignment occurs using the method described in the first approach, except that, after accumulating the minimum number of samples, it instead generates a new GMM fitted to the corresponding data.

3.4 Third Approach

The third and last approach extends the previous by adding a sliding window step to correct some misclassification and a Maximum A Posteriori estimation to adapt existing models using the incoming data. This method is better suited for video scenarios because it assumes that the same person is present in a sequence of images during an interval. Figure 3.6 illustrates the representation scheme for this variant.

The algorithm works the same way as the second approach. However, before dealing with those whose likelihood ratio is lower than a given threshold, it submits the predictions to a sliding window analysis step. During this step, the algorithm analyses every prediction P_n until it finds a P_{n+1} different from P_n. Then, given a window W, the algorithm analyses the interval $[I_{n+1} - W, I_{n+1} + W]$, where I_{n+1} is the index of P_{n+1}. If the mode M of that interval represents more than

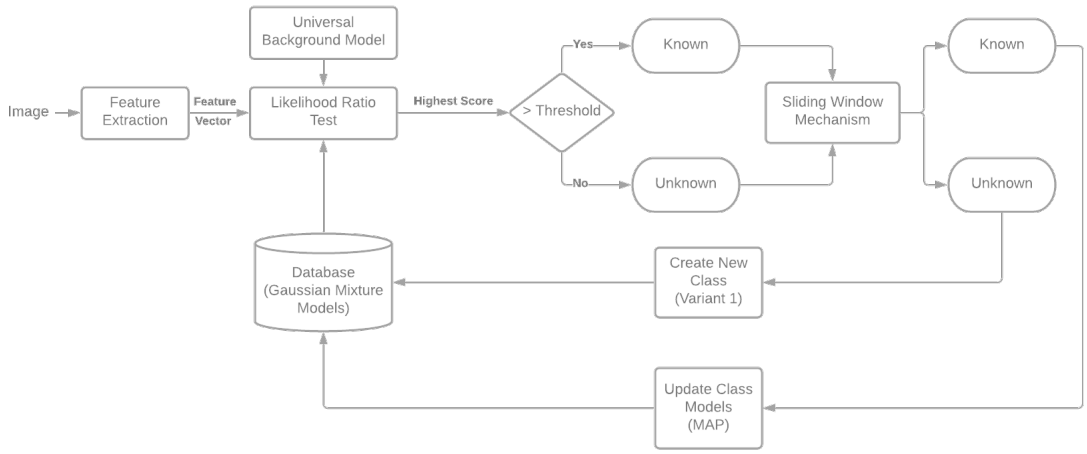


Figure 3.6: Representation scheme of the third variant.

a minimum specified percentage (preferably above 50%) of that range, then P_{n+1} is replaced by M.

Besides the mechanism mentioned above, unlike the previous method, this approach also applies a continuous update of its models using Maximum A Posteriori (MAP) estimation. The MAP estimation is an alternative to the EM method where, instead of creating a new model from the data, it uses the data to adapt an existing model. It is also a 2-step process, with the first step being equivalent to the Expectation step from the EM method where it estimates the parameters for each model:

$$n_i = \sum_{t=1}^{T} Pr(i|x_t, \lambda_{prior}) \quad \text{weight} \tag{3.19}$$

$$E_i(x) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_t, \lambda_{prior}) x_t \quad \text{mean} \tag{3.20}$$

$$E_i(x^2) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_t, \lambda_{prior}) x_t^2 \quad \text{variance} \tag{3.21}$$

with the posterior probability $Pr(i|x_i, \lambda_{prior})$ being given by:

$$Pr(i|x_t, \lambda) = \frac{\omega_i \; g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^{M} \omega_k \; g(x_t|\mu_k, \Sigma_k)} \tag{3.22}$$

However, on the second step, these new estimates are combined with the originals using a data-dependent mixing coefficient. This coefficient defines the model's reliance on the data according to its amount: the more data it receives, the more will be its reliance on them. The equations bellow define the new parameters for the adapted model:

$$\hat{\omega}_i = \left[\frac{\alpha_i^w \; n_i}{T} + (1 - \alpha_i^w) \; \omega_i \right] \gamma \quad \text{adapted mixture weight} \tag{3.23}$$

$$\hat{\mu}_i = \alpha_i^m \, E_i(x) + (1 - \alpha_i^m)\mu_i \quad \text{adapted mixture mean} \tag{3.24}$$

$$\hat{\sigma}_i^2 = \alpha_i^v E_i(x^2) + (1 - \alpha_i^v)(\sigma_i^2 + \mu_i^2) - \hat{\mu}_i^2 \quad \text{adapted mixture variance} \tag{3.25}$$

where α_i^w, α_i^m, and α_i^v are the adaptation coefficients for the weights, means, and covariances, respectively, and y is a scale factor used to ensure the sum of the weights is unity. Each mixture and each parameter has its adaptation coefficient, which is given by:

$$\alpha_i^\rho = \frac{n_i}{n_i + r^\rho} \tag{3.26}$$

where n_i is the probabilistic count of new data, and r^ρ is a fixed relevance factor for parameter ρ that controls the amount of data required before the new parameters begin replacing the old ones. If n_i is low, then $\alpha_i^\rho \to 0$ causes the emphasis of the old (better trained) parameters over the new (potentially under-trained) parameters. If, on the other hand, n_i is high, then $\alpha_i^\rho \to 1$ causes the emphasis of the new parameters.

This third approach stores the incoming data for a brief period and, at the end of it, applies the MAP estimation with the collected data to update the existing models.

Chapter 4

Experiments

This section presents the experiments conducted to validate the algorithms mentioned in the previous chapter and a comparison between the results obtained by those.

4.1 Databases

For the static-images scenario, this work utilized a subset of the VGGFace 1 database [25], an extensive database composed of 2.6M images from 2.6K individuals. However, as the algorithms described here are geared towards indoor environments with few identities, only a subset of the database was used containing 20k images from 200 identities (~100 images per identity). Figure 4.1 presents some examples of pictures from the VGGFace 1 database.

Figure 4.1: Four example photos taken from the VGGFace 1 database

For video-images, this work employed a subset of the ChokePoint database [48] to simulate the pretended scenario closely: people walking through a closed space (e.g., a corridor) and then

going off-scene, as illustrated by figure 4.2. The subset is composed of the video sequences from 25 subjects (19 male and 6 female) with a frame rate of 30 fps and an image resolution of 800x800 pixels.

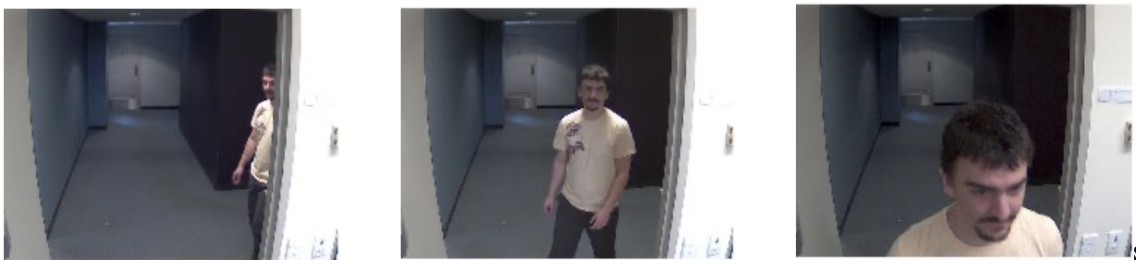


Figure 4.2: Example of three frames from the ChokePoint dataset employed in the video-based experiment

All images were processed using Facenet [31], a state-of-the-art method that extracts and converts each face into a 512-dimensional embedding (feature vector).

4.2 Closed-Set scenario

The first scenario employs static-images to validate the algorithms in closed-set settings by inputting only familiar samples. In this scenario, the number of subjects increases from 50 to 100, 150, and 200. The system trains on an initial set of 70 images per individual, while the enrollment set contains the remaining 30 images per individual. The images are shuffled before being inputted to the algorithms, and no other preprocessing technique is employed since Facenet already returns a whitened feature vector. The performance is measured using Top-1 accuracy over a single execution.

Figure 4.3 presents the results obtained. Note that when using 50 individuals, the proposed methods have a significantly higher performance compared to the baseline. As the number of individuals increase, the proposed approaches remain almost unchanged. Simultaneously, the baseline improves considerably, probably because it updates its parameters incrementally with every inputted sample, fitting the data more adequately. Nevertheless, the first method proposed still presents the best results since it considers all features and the standard deviation of all samples when calculating the confidence. Approach 2 only updates its parameters when creating a new class, while method 3 updates its parameters after gathering a minimum number of samples.

4.3 Open-World scenario

The second scenario employs static-images to validate the algorithms in open-world settings by feeding both familiar and unfamiliar samples. The number of known individuals is set to 50 and 100 while the unknown subjects increase from 50 to 100, 150, and 200. The systems train on an initial set containing 70 images per known individual, while the enrollment set includes 30

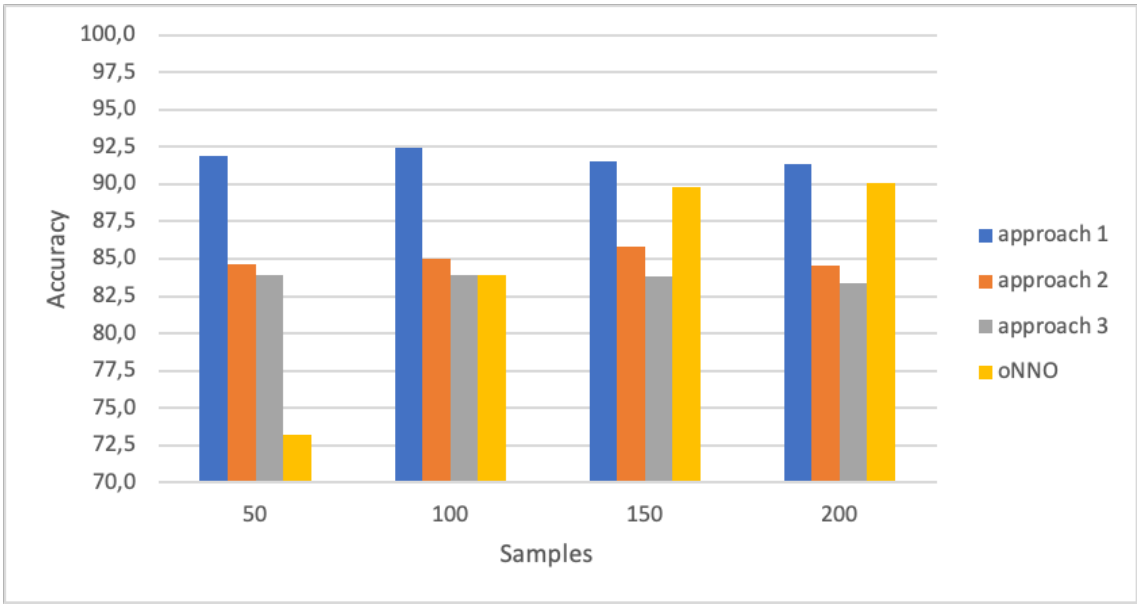
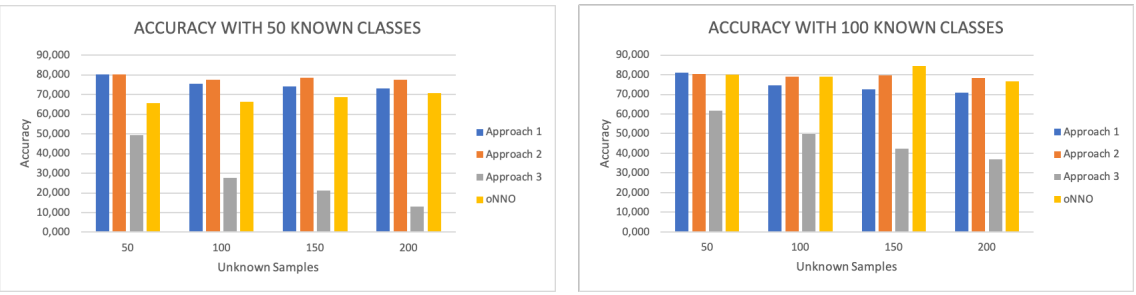


Figure 4.3: Comparison of results on the closed-set scenario obtained using 50, 100, 150, and 200 subjects

photos for each known and unknown individual. The images are shuffled before being fed to the algorithms, and, for the same reasons as the previous scenario, no additional preprocessing was performed. The algorithms' performance in the open-world can be viewed as an assignment problem of maximization. Therefore, this paper implements the Hungarian method [14] to match the ground truth and the predictions to find the correspondence that maximizes accuracy.



(a) (b)

Figure 4.4: Comparison of results on the open-world scenario obtained using (a) 50 and (b) 100 known individuals and 50, 100, 150, and 200 unknowns

Figure 4.4 illustrates the results obtained by the proposed approaches in comparison to the online NNO. The latter is designed for open-set recognition, assigning never-seen-before instances to an unknown class. In this work, however, it was adapted to learn new identities instead, hence fitting the open-world scenario. All algorithms train on an initial set and update some parameters incrementally during enrollment.

Except for approach 3, while the other two proposed methods remain almost unchanged or

with a small decrease, the baseline performance increases with the number of subjects. Thus, in a larger scenario, the baseline might outperform the proposed methods.

The behavior of the third proposed method is expected since it was designed for video-images rather than static-images. Because the data is being shuffled, the sliding window and model update steps of that approach become meaningless. The sliding window will analyze every prediction but won't change any since they are randomly ordered and won't correspond to a minimum percentage of the interval. Also, by the time the models will be updated, probably few or none instances were collected to represent a meaningful change in the corresponding models.

4.4 Video-images scenario

The third scenario employs video-images to validate the algorithms in tracking individuals while moving through a closed space, e.g., a corridor. For this purpose, a subset of the ChokePoint dataset was used, consisting of 25 subjects (19 male and 6 female) who walk through a corridor towards the camera. Since there is no previous knowledge of those individuals, the algorithms were trained using the images from 50 individuals from the initial experiments' dataset (70 images per individual). The frames were organized into batches of size 32. For each group, the faces present were detected, cropped, and fed to the algorithms. Note that some frames have more than one person, and therefore a batch can contain more than 32 faces. It was expected that the algorithms could create new identities for those 25 subjects and adequately recognize them. However, all four algorithms failed to do so.

The first approach was the algorithm that most behaved as expected, creating a small set of new identities for the subjects, but couldn't correctly differentiate them. The second and third approaches assigned most individuals to one of the 50 known identities learned during training, creating only very few new identities. The sliding window functionality of the third algorithm made the predictions more stable, i.e., most subjects were assigned to a single identity instead of multiple identities. Finally, the baseline suffered from the same problem, attributing the individuals to a small subset of identities.

Since many executions with different parameters' values were made, and the results were almost unchanged, there might be an issue related to the database, how the faces are being extracted, or the algorithms are just not suitable for video-based scenarios. No experiments with other video databases were performed, and thus the first option can't be discarded. The face extraction follows the same procedure of the previous experiments and, therefore, should not be the problem. The faces cropped present variations in lighting and pose. Consequently, they might be confusing the algorithms, which imply these are not suited for this type of scenario.

4.5 Algorithms' parameters

Table 4.1shows the parameters' values used by the algorithms on the scenarios described previously. These values are the result of several tests carried out to obtain the algorithms' best

performances.

In the first and second experimental scenarios, the images are shuffled before being inputted to the algorithms. Consequently, the *Timeout* parameter becomes less effective unless it is provided a higher value that increases according to the number of samples during enrollment. To avoid unnecessary extra work, that parameter was ignored during those scenarios, hence the None value.

Approach 1	Approach 2	Approach 3	Online NNO
$N = 512$	$N = 512$	$N = 512$	$N = 512$
$F = 1.0$	$F = 1.0$	$F = 1.0$	$M = 100$
$H = 360$	$H = 360$	$H = 360$	$\gamma = 10^{-4}$
$S_{min} = 3$	$S_{min} = 3$	$S_{min} = 3$	$S_{min} = 5$
$Timeout = None$	$Timeout = None$	$Timeout = None$	$Timeout = None$
	$\theta = 1.0$	$\theta = 1.0$	
	$UBM_{comp} = 5$	$UBM_{comp} = 5$	
	$GMM_{comp} = 3$	$GMM_{comp} = 3$	
	$r^w = 30$	$r^w = 30$	
	$r^m = 30$	$r^m = 30$	
	$r^v = 30$	$r^v = 30$	
		$W_{size} = 5$	
		$W_{threshold} = 50\%$	

Table 4.1: Algorithms' parameters used during experiments.

Where N is the number of features of the feature vectors; F is the constant scale factor employed by eq. 3.9; M is the number of features used by the metric in eq. 3.2; H is the first variant's threshold; γ is the learning rate used in eq. 3.4; S_{min} is the minimum number of samples required to create another class; *Timeout* is the maximum time to which a temporary class stores data before being deleted or converted into a new class; θ is the likelihood ratio test threshold employed by the second and third variants; UBM_{comp} and GMM_{comp} are the number of components used by the UBM and GMMs, respectively; r^w, r^m, and r^v are the relevance factors for the weight, mean, and variance, respectively (Eq. 3.26); and W_{size} and $W_{threshold}$ are the size and threshold of the temporal analysis mechanism.

Chapter 5

Conclusion

This paper addressed the open-world recognition problem by proposing three algorithms. The first algorithm represents any N-dimensional vector as a curve in a bidimensional space and calculates a feature-by-feature distance between the curves. The second algorithm substitutes the curve representation by using Gaussian Mixture Models to represent each cluster. The third algorithm extends the previous one by applying a Maximum A Posteriori estimation to update the models continuously and a Window mechanism to correct misclassifications during execution time. These approaches were compared with the online NNO and evaluated over three experimental settings: close-set, open-world, and video-images. The latter closely approaches the intended scenario for the proposed methods. For the first scenario, the proposed approaches outperformed the baseline when using a few individuals but performed worst when increased. For the second scenario, the proposed methods were as performant as the baseline. Unfortunately, however, in the third experiment all algorithms failed to correctly identify the subjects.

Future work will focus on three main topics: performance, scalability, and differentiation. Aside from the first approach, the other two didn't perform as expected in the closed-set scenario, leaving room for improvement in their performance. In both closed and open-set settings, the algorithms had a satisfactory performance when dealing with few identities, but as the number of identities increases, their performance changed slightly. In the third experimental setting, however, all algorithms failed to identify the subjects correctly, and therefore a more in-depth study of the algorithms' capabilities and differentiation mechanism is necessary.

References

[1] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[2] Abhijit Bendale and Terrance E. Boult. Towards open world recognition. *CoRR*, abs/1412.5687, 2014.

[3] Woodrow Wilson Bledsoe. The model method in facial recognition. Technical Report PRI 15, Panoramic Research, Inc., Palo Alto, CA, 1964.

[4] Zhimin Cao, Qi Yin, Xiaoou Tang, and Jian Sun. Face recognition with learning- based descriptor. In *2010 IEEE Computer society conference on computer vision and pattern recognition*, pages 2707–2714. IEEE, 2010.

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. January 2018.

[6] Jiankang Deng, Yuxiang Zhou, and Stefanos Zafeiriou. Marginal Loss for Deep Face Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2006–2014. IEEE, July 2017.

[7] Tri Doan and Jugal Kalita. Overcoming the challenge for text classification in the open world. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–7, 2017.

[8] Yueqi Duan, Jiwen Lu, and Jie Zhou. UniformFace: Learning Deep Equidistributed Representation for Face Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3410–3419. IEEE, June 2019.

[9] Manuel Günther, Steve Cruz, Ethan M. Rudd, and Terrance E. Boult. Toward open-set face recognition. *CoRR*, abs/1705.01567, 2017.

[10] Md. Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. von Mises-Fisher Mixture Model-based Deep learning: Application to Face Verification. pages 1–16, June 2017.

[11] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[12] Anil K. Jain and Arun Ross. *Introduction to Biometrics*, pages 1–22. Springer US, Boston, MA, 2008.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25(2):1097–1105, January 2012.

[14] Harold William Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[15] Zhen Lei and Stan Z. Li and. Learning discriminant face descriptor for face recognition. In *Computer Vision – ACCV 2012*, pages 748–759, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[16] Fayin Li and Harry Wechsler. Open set face recognition using transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1686–1697, 2005.

[17] Stan Z. Li and Anil K. Jain, editors. *Handbook of Face Recognition*. Springer, London, 2011.

[18] Chengjun Liu and Harry Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476, 2002.

[19] Hao Liu, Xiangyu Zhu, Zhen Lei, and Stan Z. Li. Adaptiveface: Adaptive margin and sampling for face recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 11939–11948, 2019.

[20] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SphereFace: Deep Hypersphere Embedding for Face Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6746, April 2017.

[21] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-Margin Softmax Loss for Convolutional Neural Networks. *Proceedings of the 33rd International Conference on Machine Learning*, 48, December 2016.

[22] Yu Liu, Hongyang Li, and Xiaogang Wang. Rethinking Feature Discrimination and Polymerization for Large-scale Recognition. *31st Conference on Neural Information Processing Systems*, October 2017.

[23] Vincent P. A. Lonij, Ambrish Rawat, and Maria-Irina Nicolae. Open-world visual recognition using knowledge graphs. *Computing Research Repository*, abs/1708.08310, 2017.

[24] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[25] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep Face Recognition. *British Machine Vision Association and Society for Pattern Recognition*, pages 41.1–41.12, 2015.

[26] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[27] Xianbiao Qi and Lei Zhang. Face Recognition via Centralized Coordinate Learning. January 2018.

[28] Rajeev Ranjan, Carlos D. Castillo, and Rama Chellappa. L2-constrained Softmax Loss for Discriminative Face Verification. March 2017.

[29] Rocco De Rosa, Thomas Mensink, and Barbara Caputo. Online open world recognition. *CoRR*, abs/1604.02275, 2016.

[30] Walter J. Scheirer, Anderson Rezende de Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013.

[31] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, June 2015.

[32] Alireza Sepas-Moghaddam, Fernando Pereira, and Paulo Correia. Face recognition: A novel multi-level taxonomy based survey. 01 2019.

[33] Yu Shu, Yemin Shi, Yaowei Wang, Yixiong Zou, Qingsheng Yuan, and Yonghong Tian. Odn: Opening the deep network for open-set action recognition. *Computing Research Repository*, abs/1901.07757, 2019.

[34] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, Mar 1987.

[35] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. DeepID3: Face Recognition with Very Deep Neural Networks. February 2015.

[36] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Representation by Joint Identification-Verification. *Advances in Neural Information Processing Systems*, 3:1988–1996, June 2014.

[37] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[38] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. DeepFace: Closing the gap to human-level performance in face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014.

[39] Tractica. Biometrics for enterprise applications. https://tractica.omdia.com/wp-content/uploads/2015/09/BIOE-15-Brochure.pdf, 2015. Accessed: 2020-09-01.

[40] Matthew Turk and Alex Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, January 1991.

[41] Rafael Vareto, Samira Silva, Filipe Costa, and William Robson Schwartz. Towards open-set face recognition using hashing functions. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 634–641, 2017.

[42] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, December 2001.

[43] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive Margin Softmax for Face Verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[44] Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. NormFace: L2 hypersphere embedding for face verification. *Proceedings of the 2017 ACM Multimedia Conference*, pages 1041–1049, October 2017.

[45] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large Margin Cosine Loss for Deep Face Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[46] Mei Wang and Weihong Deng. Deep Face Recognition: A Survey. April 2018.

[47] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A Discriminative Feature Learning Approach for Deep Face Recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 499–515, Cham, 2016. Springer International Publishing.

[48] Yongkang Wong, Shaokang Chen, Sandra Mau, Conrad Sanderson, and Brian C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 81–88. IEEE, June 2011.

[49] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. Learning to accept new classes without training. *Computing Research Repository*, abs/1809.06004, 2018.

[50] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range Loss for Deep Face Recognition with Long-tail. November 2016.

[51] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. AdaCos: Adaptively Scaling Cosine Logits for Effectively Learning Deep Face Representations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10815–10824. IEEE, June 2020.

[52] Kai Zhao, Jingyi Xu, and Ming-Ming Cheng. RegularFace: Deep Face Recognition via Exclusive Regularization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1136–1144. IEEE, June 2019.

[53] Yutong Zheng, Dipan K. Pal, and Marios Savvides. Ring loss: Convex Feature Normalization for Face Recognition. February 2018.