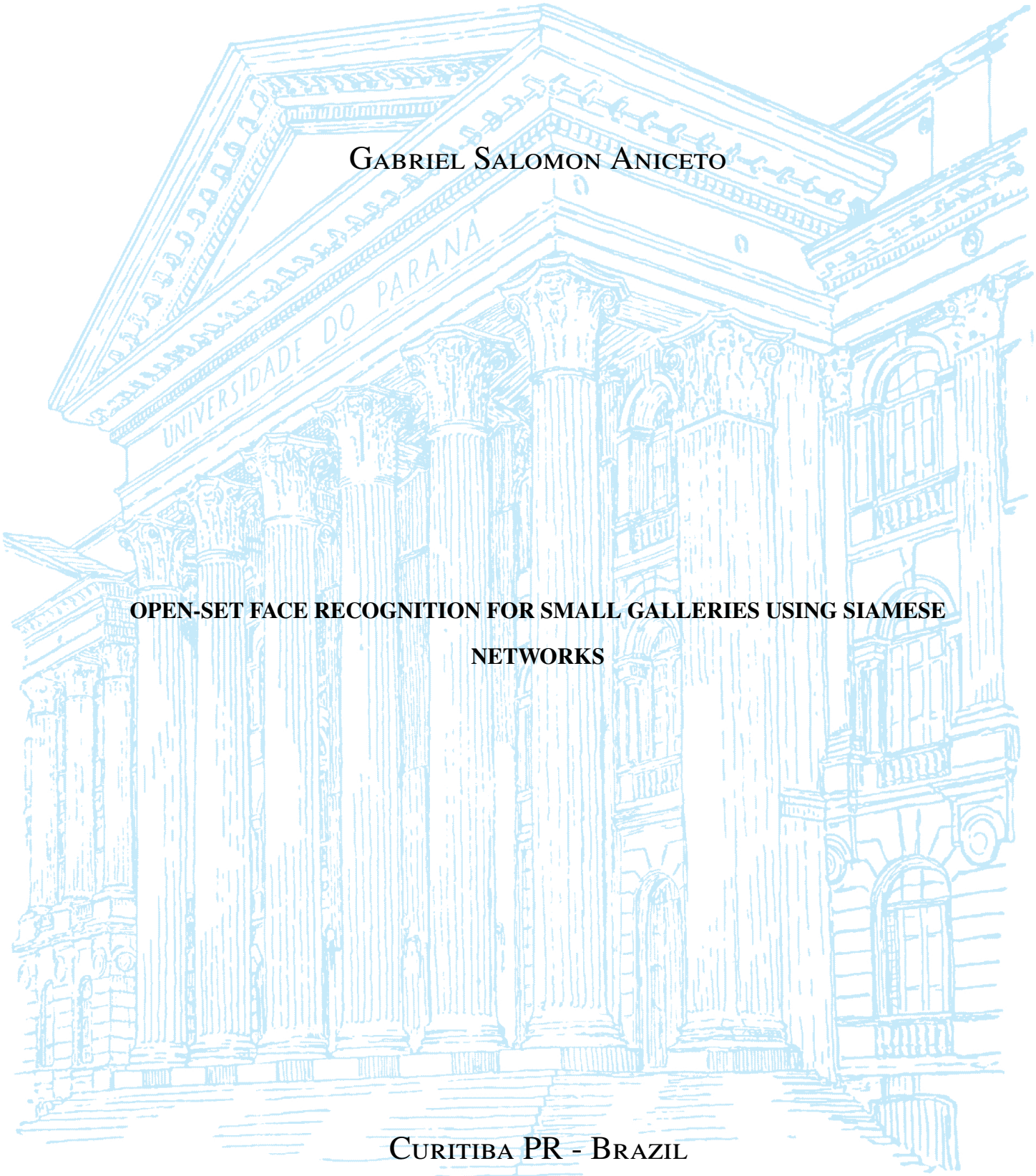
Federal University of Paraná

Gabriel Salomon Aniceto



OPEN-SET FACE RECOGNITION FOR SMALL GALLERIES USING SIAMESE NETWORKS

Curitiba PR - Brazil
2018

Gabriel Salomon Aniceto

OPEN-SET FACE RECOGNITION FOR SMALL GALLERIES USING SIAMESE NETWORKS

Bachelor thesis presented as a partial requirement for the degree of Bachelor in Computer Science in the Undergraduate Program in Informatics, Exact Sciences Sector, of the Federal University of Paraná, Brazil.

Field: *Computer Science*.

Advisor: David Menotti.

Curitiba PR - Brazil
2018

Ficha catalográfica

Substituir o arquivo 0-iniciais/catalografica.pdf (PDF em formato A4) pela ficha catalográfica fornecida pela Biblioteca da UFPR a pedido da secretaria do PPGInf/ UFPR.

O conteúdo exato da ficha catalográfica é preparado pela Biblioteca Central da UFPR, a pedido da secretaria do PPGINF. Portanto, não "invente" um conteúdo para ela.

ATENÇÃO: por exigência da Biblioteca da UFPR, esta ficha deve ficar no verso da folha de rosto (que contém o nome do orientador e área de concentração). Cuide desse detalhe quando imprimir as cópias finais.

Ficha de aprovação

Substituir o arquivo 0-iniciais/aprovacao.pdf pela ficha de aprovação fornecida pela secretaria do programa, em formato PDF A4.

To my family, for always being there for me...

Acknowledgements

I am very grateful for all the support my family provided my whole life, both in personal and academical life.

I would like to thank my advisor and professor David Menotti Gomes for the orientation provided and belief in my work, crucial element for this research to be completed.

I would also like to thank all my colleagues from the undergraduation, and from the Laboratory of Vision, Robotics and Imaging (VRI) for all the help and support provided.

A special thanks to Rafael Henrique Vareto and professor William Robson Schwartz for the cooperation.

Last, but not least, I must thank all the professors that inspired me and teach me the Computer Science's paths.

Resumo

Reconhecimento facial é uma tarefa de bastante relevância no mundo digital em que vivemos. Há várias pesquisas nesta área, mas poucas em cenários abertos. Estes cenários são importantes para aplicações reais onde há muito mais indivíduos desconhecidos em comparação com os conhecidos. Neste trabalho, Redes Siamesas foram combinadas com Redes Neurais Convolucionais para abordar este problema em pequenas galerias. Foram realizados testes nas bases de imagens: FRGCv1 e PubFig83. Os resultados foram bastante satisfatórios em comparação com o estado da arte atual.

Palavras-chave: cenários abertos, redes siamesas, reconhecimento facial.

Abstract

Face recogniton is a really relevant task in the digital world that we live. There are a lot of research in this field, but only a few broaches open-set scenarios. This kind of scenarios are important for real-world applications in which there are more unknown individuals than known ones. In this work, Siamese Networks were combined with Convolutional Neural Networks to approach this matter in small galleries. Tests were conducted on two datasets: FRGCv1 and PubFig83. The results were promising in comparison to state-of-the-art methods.

Keywords: open-set scenarios, Siamese Networks, face recognition.

List of Figures

List of Tables

List of Acronyms

CNN	Convolutional Neural Network
SVM	Support Vector Machine
COTS	Currently Off-the-shelf
GOTS	Government Off-the-shelf

Contents

1 Introduction

Face recognition is a subject inside Computer Vision of increasing interest in the last years. But, before getting into this matter, it is important to get an overview of Computer Vision. Huang (1996) claims that Computer Vision's objective is "to build autonomous systems which could perform some of the tasks which the human visual system can perform". The main tasks inside Computer Vision revolves around recognition and classification. The Cambridge Dictionary [1] defines classification as "the act or process of dividing things into groups according to their type". Recognition, on the other hand, can be defined as "the fact of knowing someone or something because you have seen or [...] experienced it before".

In Computer Vision, classification is the task of, given a sample, to identify to which class it belongs. Recognition consists in taking into account that there are some classes that we are able to recognize in a space that contains classes that we do not recognize.

Stan e Anil (2011) define face recognition as a problem where "the face, represented as a three-dimensional object [...] needs to be identified based on acquired images". In other words, the problem of matching a representation (image) of a face to the person it belongs (if it belongs to a known individual). The face recognition pipeline is illustrated in Figure 1.1. There are four steps:

- **Face and Landmark Localization:** in this initial stage, the face and landmarks are located in the input image. This consists in face detection (check where the face is located, if there is a face). Landmarks are points of interest, e.g. corners of the eyes, nose, eyebrows. Finally, a face is output (normally cropped) with corresponding landmarks.
- **Face Normalization:** in this step, background is removed (neck and hair), the face can be warped to become frontal and geometric normalized. Contrast, illumination and occlusion corrections can also be performed in this stage.
- **Feature Extraction:** an aligned, normalized face is the input. A feature extractor like HOG, LBP, SIFT, Convolutional Neural Network or other method is used to extract relevant compact information from the face image. Some type of face representation is the result of the feature extraction.
- **Feature Matching:** the last stage. The training data, or a model of it, is stored containing useful and, hopefully, discriminative information. The recognition algorithm (e.g. Neural Network, SVM, kNN) will compare a test image to the gallery to check if the sample represents someone in the gallery, and in positive case to define to whom it belongs.

According to Chellappa et al. (2010) there are three possible scenarios for face recognition: verification, identification and watch list, as Figure 1.2 depicts.

[1] https://dictionary.cambridge.org

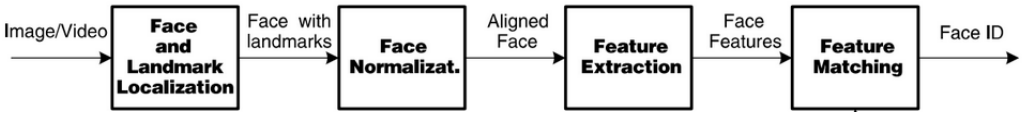


Figure 1.1: Illustration of the face recognition pipeline. Source: Stan e Anil (2011).

In a face verification scenario, the goal of the Recognition System is to verify if the given face representation matches a desired model. In other words, if a representation (image) of a face matches a predefined identity. This is a 1-against-1 problem.

In the face identification, the Recognition System needs to identify who is the person, given a representation of the person's face and a model of several people. This is a 1-against-all problem.

In the watch list scenario, the problem consists in verify if the facial representation of the person belongs to the predefined watch list (gallery of known people). If it belongs to the watch list, then a face identification can be performed to determine the identity.

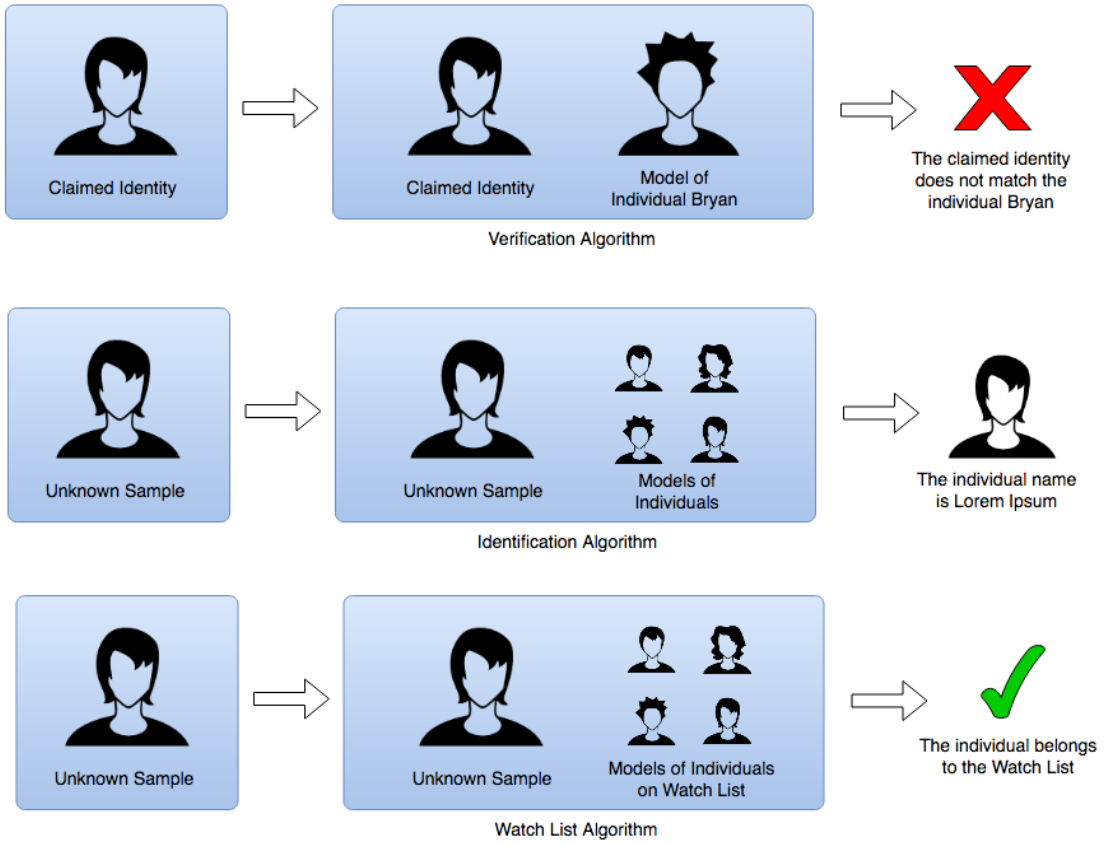


Figure 1.2: Illustration of the three algorithms: verification, identification and watch list. Source: the author. Based on illustration from Chellappa et al. (2010).

This last type corresponds to an open-set scenario problem, as mentioned by Scheirer et al. (2013). The main difficulty of this type of recognition lies in the fact that it is possible to obtain examples of the known people and create a model, although this is not necessarily true for the unknown people. Zhou e Huang (2001) quoting Leo Tolstoy, stated that: "All positive examples are alike; each negative example is negative in its own way".

Another challenge relies on the problem being unbalanced, as there are more negatives classes of individuals (infinite individuals are unknown), than positive ones (finite individuals

are known). Even if the negative classes were known in advance, it would still be difficult as the problem would remain unbalanced.

1.1 Motivation

As technology increases, the world getting more connected and the automation of a series of tasks promoted by Machine Learning, recognizing faces using only human eyes is not efficient anymore. For this reason the number of computer-based solutions to this issue is growing at a fast rate. Only this year, there has been more than ten thousand publications related to face recognition indexed by Google Scholar [2].

According to Stan e Anil (2011), face recognition has a lot of advantages over other biometrics as it is non intrusive and can be captured at a distance, even without the person's knowledge (which might raise privacy concerns). They also state that it has become more popular than iris or fingerprints due to the availability of face photos on the Internet (more data to train).

The business news website *Business Insider* [3] published that more than 1.8 billion photos were being posted in social media everyday, in 2014. With this increase in photo's availability, the possible applications are no far behind. Some retail stores are using this kind of technology to identify possible shoplifters. [4]

As authentication remains one of the biggest issues in this digital world, face recognition can also help prevent fraud and avoid unauthorized access to data and services available online.

One of the more addressed issues, inside face recognition field of research, in the last years, is face identification on a closed-set scenario. Nowadays, it is not really a big challenge. For instance, recent work by Sun et al. (2015) achieved 96% identification rate on tests using the LFW (Labeled Faces in the Wild), proposed by Huang et al. (2008) dataset with their proposed DeepID3 architecture.

A bigger and still unsolved issue is face identification in an open-set scenario.

1.2 Hypothesis

The main hypothesis is: Siamese Networks can be used to learn differentiating subjects on a small gallery and are able to allow recognition for an open-set scenario.

1.3 Objectives

The present work proposes a new open-set face recognition approach based on Siamese Networks for small galleries. To the best of our knowledge, this is the first time that: i) such approach usually used for verification task is employed for open-set face recognition; ii) small galleries are the focus of an open-set face recognition.

The method will be meticulously explained in Chapter 4, but briefly explaining, the method can be divided in two stages. First, the Siamese Networks will be trained on generated pairs on verification task, learning to differentiate the members of the gallery. After that, they will be used to calculate the minimum euclidean distance between a test image and the members

[2] https://scholar.google.com.br

[3] https://www.businessinsider.com/were-now-posting-a-staggering-18-billion-photos-to-social-media-every-day-2014-5

[4] https://www.racked.com/2018/5/22/17380410/facial-recognition-technology-retail

of the gallery. It is expected that there would be a threshold that separates the distance obtained for test images that belong to the gallery (smaller distance) from the ones that do not belong (higher distance).

1.4 Contributions

- A new approach of open-set recognition was proposed for small galleries
- From the experiments, new state-of-the-art results for FRGCv1 dataset, on experiment 4 were achieved.

1.5 Roadmap

This work contains the following structure for the next chapters:

Chapter 2 presents concepts for the reader to understand the methods and techniques used. Chapter 3 reviews several relevant publications of progress made so far, in a condensed way, in the fields of face feature extraction, face verification, identification, open-set and Siamese Networks, with introduction of relevant concepts and methods. Chapter 4 contains the proposed method detailed. In Chapter 5, the experimental protocol and datasets are discussed, with results presented. Finally, Chapter 6 presents conclusions, and paths to explore in future works.

2 Theoretical Background

2.1 Convolutional Neural Networks

The CNN architecture is composed by convolutional, subsampling and fully connected layers.

In the convolutional layers, a convolution operation is performed in the whole image using kernels, in other words, a filter is applied in the whole image, to extract features, but maintaining a notion of locality.

The subsampling layers (sometimes called pooling layers) are used to remove dimensionality of the feature maps, but maintaning important information. The subsampling method most used is maximum pooling, maintaining the highest values of a window with predefined size. But the pooling can also be performed calculating the average, minimum value. A common size is 2×2 windows. The windows are slided over the feature maps of the last layer to obtain the maximum value between the 4 pixels in the window and put the result in the position of the first pixel, on a new feature map.

The fully connected layers are similar to a MLP (Multilayer Perceptron). They have a nonlinear activation function. This layer can make combinations of features, enhancing the representation power. The output of this layer is a feature vector. If it is the last layer it can be normalized using a Softmax function (a function that makes the sum of all outputs be always 1, like a probability function).

For example, Lawrence et al. (1997) used a convolutional layer that outputs 20 feature maps, followed by a subsampling layer that generates 20 feature maps. After that there is another convolutional layer producing 25 feature maps, followed by a subsampling layer that also makes 25 feature maps. The last layer is a fully connected MLP like layer that outputs 40 features normalized using the Softmax function. Figure 2.1 illustrates the process.

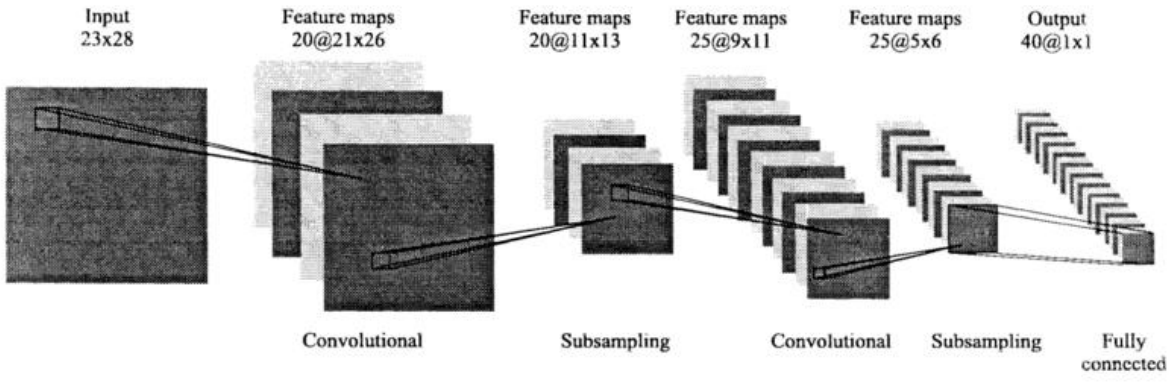


Figure 2.1: The layers of the Convolutional Neural Network. Source: Lawrence et al. (1997).

2.2 Siamese Networks

Siamese Networks are basically composed of two Neural Networks that have the exact same architecture and the same weight values. Two samples are fed to the two networks, they extract the features and a distance function calculates a scalar distance based in the distance of the output values of the two networks. It is common to use Cosine, L2, L1, Mahalanobis distance functions. The output of the network is the measured distance between the two inputs. Figure 2.2 illustrates the architecture.

Usually the distance between two inputs, if they correspond to the same individual is close to 0, otherwise it is closer to 1. If the problem is well balanced (if there are similar number of samples for each class), the recognition threshold normally is set to 0.5.

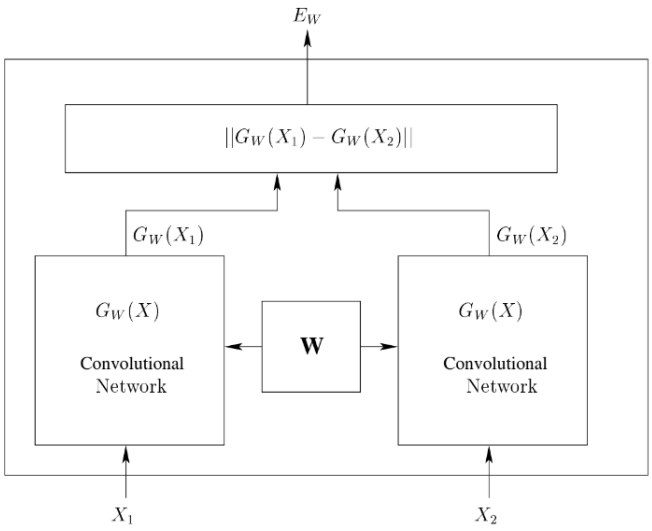


Figure 2.2: Siamese Networks, with shared weights "W" and euclidean distance welding the outputs of them. Source: Chopra et al. (2005).

2.3 Open-set Recognition

An equation to measure the unbalance of classes in an open-set recognition, or the “openness” of a particular problem, considering the number of individuals in training, testing and in the gallery was formalized by Scheirer et al. (2013). Openness is the rate of unknown samples in testing. The following formulation can range between 0 and 1 indicating the level of openness of a given problem, where 0 would be a closed problem and greater values would enter the open world:

$$\text{openness} = 1 - \sqrt{\frac{2 \times |\text{training} \quad \text{classes}|}{|\text{testing} \quad \text{classes}| + |\text{target} \quad \text{classes}|}}$$

By example, if we have 12 classes (people) that we would like to identify, 12 classes that we use on training and 50 classes (12 known and 38 unknown) in testing, we would have approximately 0.38 openness. Statistically, as the openness grows, the ability to model the problem should decay. The unknown classes are hard to be modeled, and if there are lots of them in testing, the consequence is poor predictions, hence the difficulty around open-set problems. Scheirer et al. (2013) uses open-space risk concepts, from Statistics to help improve method performance, this technique will be discussed in the next chapter.

Another relevant concept, introduced by Günther et al. (2017) was the notion of "knowns", "known unknowns" and "unknowns unknowns" subjects in the open-set problem. The known individuals are the ones seen on training, and that belong to the watch list. The task lies upon identifying them. The known unknowns are the ones we also see on train, but are not in the watch list – the algorithm should reject them. The unknown unknown, on the other hand, are the ones not seen during training, only during testing and are not in the watch list – they should be rejected as well.

3 Literature Review

The main and most difficult tasks in face recognition are the feature extraction and the recognition itself. Having that in mind, this chapter will approach several works in the last decades and their proposed algorithm to perform these tasks. Inside recognition there is an even harder task: dealing with open space (present in open-set scenarios).

3.1 Feature Extraction

Feature extraction is a process used to generate a representation, preferably compact and unique, of a given input image. There are lots of algorithms that can extract representation, dimensional reduction algorithms approaches like PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) that are even sometimes applied in raw data (pixel intensity values) without pre-processing. But, there are more discriminative and representative methods like HOG (Histogram of Oriented Gradients) Dalal e Triggs (2005), LBP (Local Binary Patterns), SIFT (Scale-invariant Feature Transform) Otero (2015), Gabor filters. The most expensive (as it requires training), but also most effective are the CNNs (Convolutional Neural Networks).

3.1.1 Convolutional Neural Networks

One of the first considerations of Convolutional Neural Networks (under this name) for recognition was made by Bengio et al. (1994) to recognize handwritten words. A few years later, Lawrence et al. (1997) introduced CNNs to face recognition. The advantages obtained in feature extraction, over most of the other methods, if trained with data augmentation (or in large datasets), were: invariance of rotation, translation, scale and deformation, by introducing locality, shared weights and spatial subsampling.

In recent works, Parkhi et al. (2015) used a Deep CNN (known as VGG Face) trained only on a large self-made dataset, achieving a maximum accuracy of 98.95% in face verification tests conducted over the LFW dataset (Huang et al. (2008)) as shown in Table 3.1 . Although the problem addressed is not open-set nor recognition, this shows the power of representation and feature extraction of the Deep CNNs, in unrestricted datasets. For comparison, we can observe in Table 3.1 that SIFT with Fisher Vector Simonyan et al. (2013) obtained only 93.10% maximum accuracy, but with the disadvantage of needing to use the same data in training and in testing (it does not generalize as the CNN does).

The greatest changes in image recognition, and consequently, face recognition in the last years were promoted mainly by the use of Deep CNNs. This was only possible due to the creation of large datasets, making this type of empirical algorithms able to train in millions of subjects and achieve remarkable results. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[1], an annual event since 2010, compares algorithms used in large scale image classification. In

[1] http://www.image-net.org/challenges/LSVRC/

2009, Deng et al. (2009) made the FaceNet dataset publicly available. Containing more than 3.2 million of images, organized over 5,247 hierarchic categories (like mammals, carnivores, dogs, poodles), this dataset was used in several of the challenges for algorithms comparison. Most of the algorithms in Table 3.1 were submitted to compete in the ILSVRC and, since Krizhevsky et al. (2012) winner algorithm in 2012, no other competing algorithms were able to beat CNNs in the classification category.

Table 3.1: Comparison between Deep CNNs over unrestricted dataset LFW on face verification task. Source: Parkhi et al. (2015).

Method	Images on training	Networks	Max. Accuracy(%)
Fisher Vector Faces	-	-	93.10
DeepFace	4M	3	97.35
Fusion	500M	5	98.37
DeepID-2,3 (Sun et al., 2015)	-	200	99.47
FaceNet	200M	1	98.87
FaceNet + Alignment	200M	1	99.63
Parkhi et al. (2015) Method	2.6M	1	98.95

One of the pioneers to reduce the abyss between human and machine performance on face verification was DeepFace (Taigman et al. (2014)). The DeepFace uses 2D alignment, followed by a 3D alignment and, at the end, face frontalization made using fiducial points and warping. After this process of frontalization, it uses a Deep Neural Network (not really a convolutional) to train and learn the face features. The operation between some layers is similar to convolution, but with a great difference: instead of uniform filters that are applied in the whole image, DeepFace uses different filters depending on the area of the face. This leads to not sharing weights in this layers (as they layers are different from each other). It is easy to see this relevance as a face has more contour details around the eyes than on the cheeks, by example. Interestingly, the higher layers are sparse, with more than 75% of the feature components being 0. This method achieves maximum face verification accuracy of 97.35% in the LFW dataset. Tests were also performed with a Siamese architecture, using L1 as the distance measure. The Siamese Network did not perform as great due to the number of parameters surpassing 1.2 million and LFW containing less than 15,000 images, this disparity led to overfitting.

The DeepID2 proposed by Sun et al. (2014) is also a deep CNN for face recognition: verification and identification. The main innovation over the others deep CNNs are the learning face identification and face verification signals as that act as supervision in the learning process. The identification signal helps improving the inter-class separation and the verification signal enhances the similarity of intra-class feature variations by regularizing features extracted.

Sun et al. (2015) released later the DeepID3, an improved DeepID2, based on VGG and GoogLeNet concepts of **very** deep network. By introducing more layers of convolution and subsampling, there was an improvement, but not so significant, in the accuracy.

The GoogLeNet (Szegedy et al. (2015)) was one of the first CNNs to go deeper: 27 layers (22 with parameters). Also, instead of using fixed convolution filters, the so-called Inception architecture learns the filters itself. To avoid bottlenecks and extra computational costs caused by the increase in depth and width were used small convolution filters, some of them of size 1×1. This type of resource also allows dimensionality reduction. The Inception architecture also approximates, using dense components, optimal local sparse structures. Using regular convolutional "blocks", the connections between layers occur in the regions with higher

correlation, this way the sparsity of the connections is increased, this may help preventing overfitting, improving speed and lowering computational costs.

Schroff et al. (2015) introduced a really ingenious approach to face recognition: FaceNet. This Deep CNN generates a 128-dimension embedding to represent every face sample. It also uses a Triplet Loss function that improves significantly the accuracy and it is a crucial part of one of the most interesting appliances: Euclidean space representation. The FaceNet generates an embedding that in a feature space has L2-distance minimized for same subject examples and maximized for different individuals – in training phase a general threshold is learned to differentiate identities.

In this deep network some structures were based in the Inception model (Szegedy et al. (2015)), including the 1×1 convolutional filters. The Triplet Loss is obtained by using two samples that belong to the same individual and one that belongs to another one and testing both distances (intra and inter-class) to determine the current loss. Mini-batches are made with training samples, and the hard positive and negatives are chosen (the examples that are more different from each other) to improve separation in euclidean space.

Another really interesting element in the FaceNet lies upon its possibility of generating clusters (using a 128-dimension Euclidean space). An example of what can be achieved by the clustering is represented in Figure 3.1. A version more compact (and with lower precision) of the network was also created, possibly to be used in mobile devices. Interestingly, although the performance decreased, it was still relatively high. Tests were conducted with this network on the LFW dataset and the maximum face verification accuracy was 99.63%±0.09 standard deviation.

Figure 3.1: The samples above belong to a cluster that represents one person's identity. The unconstrained conditions of the samples show that FaceNet is robust to lighting, pose and age variations. Source: Schroff et al. (2015).

3.2 Siamese Networks

There are not many works in literature that used Siamese Networks (at least, not using this nomenclature). One of the first approaches of verification using Siamese Networks was made by Bromley et al. (1994) for verifying signatures using Neural Networks.

Chopra et al. (2005) used Siamese Networks on face recognition. Instead of distance, an EBM (Energy Based Model) was used as a loss function instead of probability models – the latter normalize the probabilities for all training examples. The main advantage is that there is no

need to estimate probability over all inputs (architecture freedom). The main disadvantage is that if two samples belong to the same subject their loss function result is 0, but if they do not belong to the same subject there is no guarantee that the distance will be greater than 0 (as it is not necessarily the complement). Therefore, the loss function needs a term to ensure a distance between those distinct samples.

Khalil-Hani e Sung (2014) used Siamese CNNs with fused convolutional and subsampling layers to perform very fast classification. It was also used 2D cross-relation instead of convolution to improve even more the classification time, achieving impressive 0.6 milisseconds for every face verification.

3.3 Open-set Recognition

Most of the classical recognition algorithms in Computer Vision (all of the methods in the last two sections) offer solutions based on "closed set" scenarios, i.e. when you have a defined number of classes and certainty that all subjects that you will test belong to one of these classes. This type of solution is not so realistic. In the real world, there are more things, or people, that we do not know, than the ones that we know (the negative space of things that are known is bigger than the positive space).

Scheirer et al. (2013) defines this more realistic problem as “open-set recognition” and proposes a method entitled “1-vs-Set Machine” based on regular binary and 1-class SVMs (Support Vector Machines), but adding the concept of open space risk.

The regular binary SVM creates a hyperplane splitting the positives examples and negatives ones (assuming they are linear separable). The main complication of this approach on open-set lies on the probability of any point in the positive half-space being equal, no matter the distance to the hyperplane, implying in a great generalization. In other others, it does not matter if the subject is close or really far from the hyperplane, it is treated equally. This is ineffective in the open-set context, a reduction in the open half-space is necessary to avoid under-fitting.

The solution presented combines a risk model for the open space with empirical risk obtained during training, to minimize an error function that aims to reduce the positive half-space using a new hyperplane Ω. The minimization of the positive region implies overgeneralization (pulling the hyperplanes apart), and the minimization of empirical risk generates overespecialization (bringing the hyperplanes closer to each other). The Figure 3.2 illustrates the process.

Although Scheirer et al. (2013) focus in object recognition, were presented results in face verification over the LFW database (Huang et al. (2008)). The best method was the proposed 1-vs-Set Machine, significantly better than the binary SVM. The F-measure decreases as the openness of the problem increases, as expected.

Günther et al. (2017) introduces a new method using EVM (Extreme Value Machines). This method is compared to LDA (Linear Discrimination Analysis) and a common approach: thresholding verification-like scores.

The thresholding score used most common among authors: cosine similarity. The feature extraction is made using deep features (deep convolutional networks as feature extractors), and there is no further processing other than calculating the similarity measure in this method.

LDA and EVM also use cosine similarity. The LDA technique is not new in face recognition as it was previously explored by Yu e Yang (2001). However it is proved to be efficient in its capability of increasing intra-class vector similarity and reducing inter-class vector similarity over the training set by learning a sub-space projection.

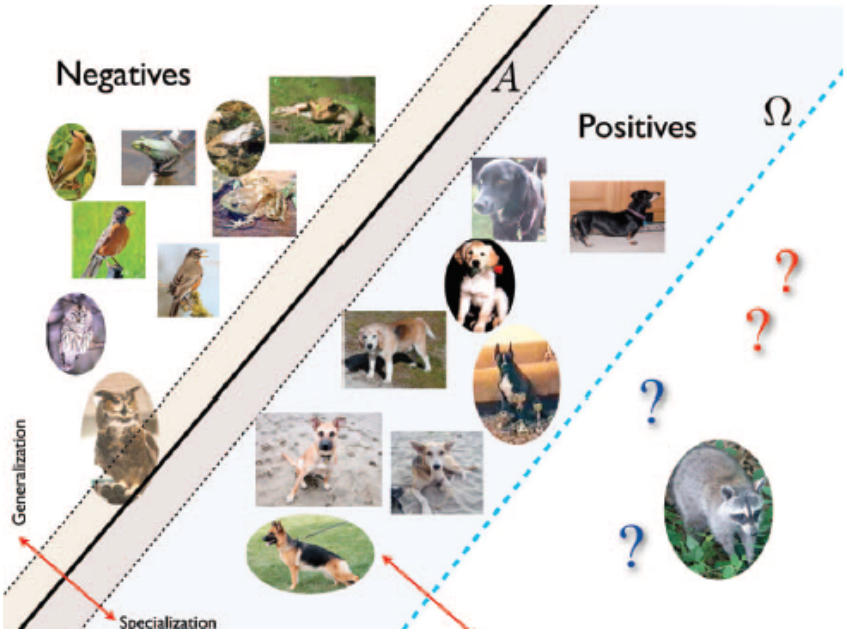


Figure 3.2: Regular linear SVM creates the hyperplane A. The error minimization process of 1-vs-Set Machine using the open space risk model and empirical risk produces the hyperplane Ω and also adjusts the hyperplane A, to generalize or specialize, according to the minimal error possible. This prevents the racoon from being identified as a dog in the example, as its far away from the other training examples for the dogs' class. Source: Scheirer et al. (2013).

EVM is based in the Extreme Value Theory. It assumes a new statistical data distribution if the distance values are greater than a pre-defined threshold. Instead of learning the identities and reducing dimensionality as LDA does, it creates a statistical model that can handle outliers.

Proedrou et al. (2002) introduced the TCM-NN (Transduction Confidence Machine Nearest Neighbors), a method based on transduction, similar to kNN (k-Nearest Neighbors) but, that has the advantage of producing probability measures for every prediction, apart from most methods, like SVM, Neural Networks, that only output a prediction. That way, it is possible to estimate the level of correctness of a classification, given a training set.

It is important to define "transduction" for better understanding the method. First of all, transduction is different from induction. Most learning methods, like SVM and Neural Networks learn by induction. Creating a generalization over the training examples, they induce a model (or function). After that, when a testing example is inserted into the model, a prediction is made, using deduction.

On the other hand, methods like kNN, do not train on the training examples. No model is inducted (or learned). Each time a testing example is given to the algorithm, all the training examples are considered and used to make the prediction. That is how transduction works, by using all available information of the training examples to infer a prediction, instead of a pre-trained model.

The advantages of using a transductive method lies upon there is no need to retrain the algorithm once a new identity or sample of a subject needs to be inserted. The disadvantages are high computational cost for every prediction and high amount of data necessary.

Li e Wechsler (2005) proposed Open Set TCM-kNN (Transduction Confidence Machine-k Nearest Neighbors). Based on TCM-NN, but enhanced with the notion of openness, using the PSR (Peak-side-ratio), an equivalent of the likelihood ratio (LR), used in Statistics for hypothesis testing. LR is the ratio between hypotheses H_0 and H_1. H_0 is the hypothesis that a subject's face belongs to the gallery and H_1 is that it does not.

The PSR requires a training phase, where every training sample is assigned to every other class (identity) an the PSR (which is low, due to the fact it is an impostor) is recorded. After that, the PSR distribution is used to compute a prior threshold for the task of determining if a sample face image correspond to one of the identities in the watch list.

Tests were made by Li e Wechsler (2005) comparing this novel approach in the FERET dataset with traditional methods such as PCA and Fisherfaces, and the results were close, depending on the number of components used by PCA and Fisherfaces. But, PCA and Fisherfaces best results were obtained by testing several thresholds, and Open Set TCM-kNN automatically detects the best threshold on training. Also giving the confidence and credibility of the classification.

dos Santos Junior e Schwartz (2014) tested five methods for open-set face recognition: background set (Bk-Set), SVM-All, SVM-Single, LS (Least Squares) and Chebyshev Inequality.

In the Bk-Set, a PLS (Partial Least Squares) model is learned to check whether a sample image belongs to the known list or if it is unknown. A PLS regression learns known subjects' common attributes and tries to differentiate from unknown samples seen in training.

The SVM-All learns one SVM (with radial kernel) model for all the known subjects and uses negative examples for better separation. To check whether a prediction is correct, the distance from the test sample to the hyperplane is used. If it is too distant, it is probably an impostor. Otherwise, if the distance is below the defined threshold, the identification is expected to be correct.

The SVM-Single is similar to the SVM-All, but the difference is that for each known subject a different model is learned, with a corresponding threshold for each model.

The Least Squares method is trained in a similar fashion as SVM-Single, but it has lower computational costs, by using regression to learn a PLS model for each individual and also determining a threshold.

The Chebyshev Inequality (CI) method focuses on modelling the distribution of the unknown samples. As there can be a lot of difference between the samples in training and testing, as the data sets are of unconstrained images, this method is expect to handle this variation for being more statistically based than empirical (like the previous methods).

The methods that accomplished better results were SVM-Single, LS and CI. The CI method outperformed the others when the number of subjects in the gallery increases, as it requires a higher number of samples to proper estimate mean and variance.

3.3.1 Scalability

One issue related to face recognition in real world situations is: scalability. A problem faced by previously mentioned DeepFace (Taigman et al. (2014)) is that it has more than 100 million parameters. Training this type of network with hundreds of millions of images is a really difficult task. Taigman et al. (2015) implemented improvements in the DeepFace CNN. One improvement proposed is applying bootstrapping in the training set selecting a subset of samples that are hard to classify (samples from different subjects that have high similarity). This method is applied by training 1-vs-all SVMs for a few classes and comparing hyperplanes distance to find similar models. The results of this technique were interesting as it helped accuracy to keep increasing as the training set enlarges (instead of stagnating).

Another contribution was the removal of the DeepFace's classification layer (last layer). It was the CNN's bottleneck, because the second to last layer had fewer neurons than the last layer. This makes the network more capable of generalization, instead of training-set specific biased. A new embedding of 256 dimensions (instead of the original 4,030 dimensions classification layer)

performed really well. Now the CNN is for feature extraction only. The scalability lies upon the compact representation achieved. The face verification results were similar to state-of-the-art methods on the LFW dataset. Identification in both closed and open-set scenarios were also executed, and the method outperformed COTS baselines.

Pinto et al. (2011) also addressed the scalability issue from a biologically-inspired point of view, using a rather simple but powerful method called V1-like-Plus (Pinto et al. (2008)). V1-like-Plus is a one-layer model that is composed by a sequence of linear and nonlinear steps mimicking cortical processing in a primate brain. A two and three layer models were also tested. Random filter kernels are generated using a Gaussian distribution. Gabor filters convolutions are made in the filtering step (linear) and thresholds and saturation are performed for each filter (nonlinear), the whole process is shown in Figure 3.3. More than 5,000 models were generated for each of the two and three layer. The best ones were evaluated in Facebook100 and PubFig83 datasets for face identification and verification. These models only generate representation for the faces, the task of identification is made by a 1-vs-all SVM trained for every identity. The 3-layer network outperformed the other approaches in identification and verification.

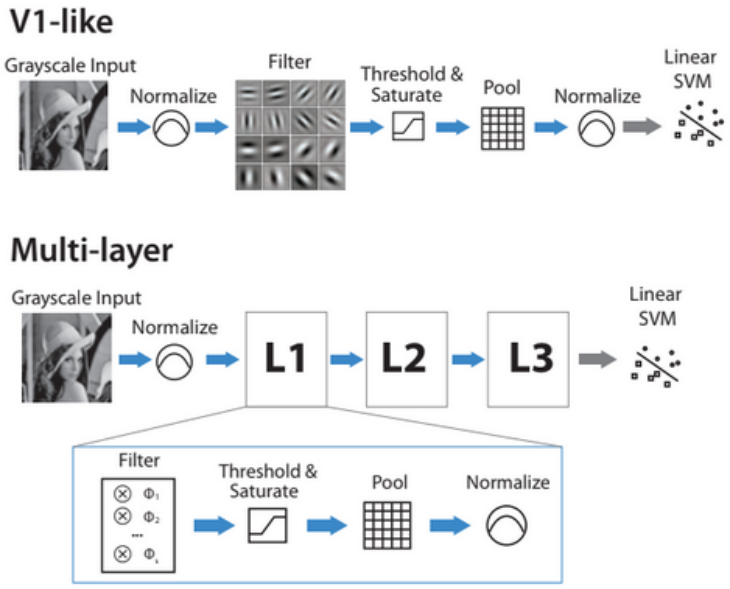


Figure 3.3: The steps of V1-like-Plus and Multi-layer V1-like. Source: Pinto et al. (2011).

AbdAlmageed et al. (2016) invested in deep multi-pose representation for face recognition. It uses pre-trained convolutional networks such as VGG19 and AlexNet. First, a transfer learning process takes place (as the networks are pre-trained) and frontal faces models are generated (one for each pre-trained network). These models are the result of refining VGG19 and AlexNet with aligned and averaged frontal faces of each subject (this corrections are made in pre-processing) from the CASIA-WebFace (Yi et al. (2014)). Other pose variations are generated from these frontal faces models, like profile, 45º and 75º yaw (head rotation on vertical axis). Also, discarding the last layer (classification layer) of these nets, it was able to extract features instead of using their default classifier. For classification, cosine similarity scores are calculated over the feature vectors generated by the different pose models. This scores are fused using softmax for every pair of features extracted. Only face verification was evaluated in the tests. The results surpassed the test database IJB-A COTS and GOTS baselines (Klare et al. (2015)) significantly. Evaluation was also performed on LFW.

Partial Least Squares models were introduced for face recognition by de Paulo Carlos et al. (2015). The main task intended was closed-set face identification.The feature extraction was composed by combined HOG, LBP, Gabor filters, average pixel colors features. The PLS models were used for identification. They were trained in one-against-one, one-against-some and one-against-all schemes for comparison. The one-against-none models focused on learning the features of a subject's face, one model were created for each distinct subject (n identities result in models). The one-against-some models were trained by choosing a value k (less than n), then, for every subject, randomly selecting k-1 remaining subjects and using them as negative examples, generating $k \times n$ models. The one-against-all scheme consisted in for every subject a model was created using the other $n - 1$ subjects as negative examples. Notably, the last is the most computationally expensive approach, but also the most effective, as per tests conducted.

The work of Vareto et al. (2017) differently from most of the other approaches, focused mainly in open-set recognition, not in the feature extraction. The VGG Face descriptor was used to extract features that were used to train Partial Least Squares, Neural Networks and SVM models for the recognition task. Similar to de Paulo Carlos et al. (2015), the models are trained in a one-against-all, so that there is a model for each distinct individual in the gallery. The models predictions are inserted into a vote array (hashing) and normalized between 0 and 1. If one vote stands out from the others, it is assumed that a match has occurred, otherwise, the sample is considered unknown. A threshold is used to determine if the vote stands out from the others. This vote process is illustrated in Figure 3.4.

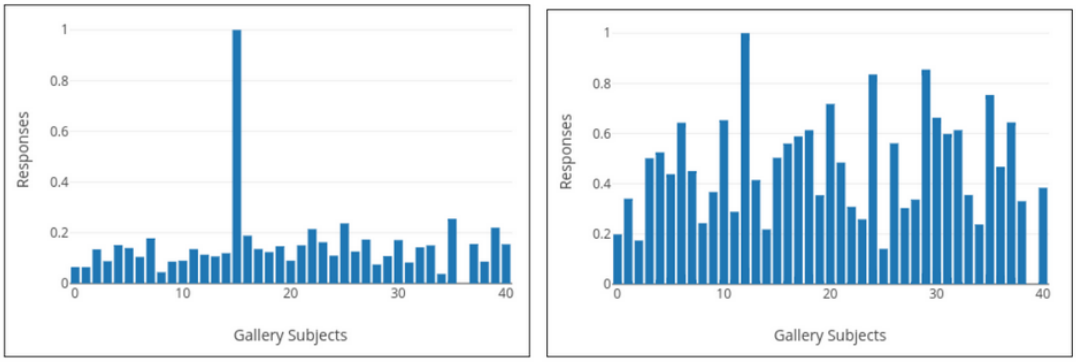


Figure 3.4: On the left, a representation of the vote-list normalized array, indicating that a subject was recognized. On the right, no prediction really stands out from the others, so the sample is considered as unknown. Source: Vareto et al. (2017).

3.4 Baseline and State-of-the-art Methods

Most of the relevant methods introduced in the last section are presented here for comparison. FV indicates Face Verification task, FID indicates Face Identification and Open indicates if there were tests performed in open-set scenarios.

Table 3.2: Comparison between methods.

Article	Feature Extraction	Classifier	Subjects	Testing Dataset	FV	FID	Open
Pinto et al. (2008)	V1-like-Plus	1-vs-all SVM	-	Facebook100, PubFig83, LFW	•	•	•
Taigman et al. (2015)	DeepFace⋆	NN	500K	LFW	•	•	•
AbdAlmageed et al. (2016)	VGG19 ⋆, AlexNet⋆	Cosine Similarity	400K	IJB-A	•	•	•
Schroff et al. (2015)	FaceNet⋆	L2-distance (thresholded)	100-200M	LFW, Youtube Faces	•	-	-
Taigman et al. (2014)	DeepFace⋆	Siamese network	4.4M	LFW, Youtube Faces	•	-	-
dos Santos Junior e Schwartz (2014)	HOG, LBP	SVM-Single, LS, CI	1K-10K	FRGC, PubFig83	•	•	•
Sun et al. (2015)	DeepID3⋆	Joint Bayesian	300K	LFW	•	•	•
Lawrence et al. (1997)	CNN+SOM	MLP, kNN	200	Own dataset	-	•	-
Parkhi et al. (2015)	DeepConv ⋆		2.6M	LFW, Youtube Faces	•	-	-
Vareto et al. (2017)	VGG Face ⋆	PLS	1K-10K	FRGCv1, Pubfig, FG-Net Aging.	•	•	•
Li e Wechsler (2005)	PCA and Fisherfaces	TCM-kNN	750	FERET	•	•	•
Günther et al. (2017)	VGG Face ⋆	EVM, LDA	13K	LFW	•	•	•
Szegedy et al. (2015)	GoogLeNet⋆		1.2M	ImageNet	•	-	-
de Paulo Carlos et al. (2015)	HOG, Gabor, LBP	PLS	1K-10K	FRGC, PubFig83, Youtube Faces	•	•	-

⋆ Methods based on Deep CNNs.

3.5 Datasets Benchmark

The most used datasets were LFW and YFW, they became the reference for face verification, face identification and open set face identification. ImageNet was used mainly by Deep CNNs to train, as there are lots of classes and images for it to generalize well and enhance the feature extraction abilities. The other datasets are also used, but not as frequently as these previous.

Table 3.3: Comparison between datasets.

Article	Data base	# samples	# subjects	Images/subject	Faces	Unconstrained
Huang et al. (2008)	LFW	13,233	5,749	2.3	•	•
Wolf et al. (2011)	YFW	3,425	1,595	2.1	•	•
Deng et al. (2009)	ImageNet	3M	5,247	-	-	•
Pinto et al. (2011)	PubFig83	8,3K	83	100.0	•	•
Phillips et al. (2005)	FRGC	50K	200	250.0	•	•
Klare et al. (2015)	IJB-A	5,712	500	11.4	•	•

3.6 Final remarks

There are few considerations after analyzing the work in the past decades in face recognition: the representation step is not a really big issue nowadays, with the use of Deep CNNs. The recognition task is still filled with issues, but there a few interesting solutions. The main challenge is still the open-set recognition, as it is really hard to predict how unknown individuals will be like: in one hand methods like PLS are prone to overfitting as they try to model the known examples. On the other hand, methods like SVM do not take into account the unbalance of the scenario. Promising approaches for open set are: using statistical risk models, reducing the open space, trying to improve representation (3D models) to differentiate better distinct classes or even ensemble of methods. Metric learning (methods like Siamese Networks) is a promising field as they transform the feature space trying to bring the gallery together and separating it from the unknown subjects, but the rigorousness of this separation is still an issue to be addressed.

4 Proposed Method

The method trains on face gallery samples, use a test dataset to determine the known/unknown best separation threshold, and is ready to recognize probe images as known or unknown. The figure 4.1 shows the whole system. In this work, the testing is evaluated. In a real world scenario, in testing, a recognition threshold would be defined. This part was not implemented as it is beyond the scope of the present work. Each step of the method is explained in the chronological sequence it would occur:

- **Crop/Align:** each image is aligned and cropped to enhance the accuracy of the method in the feature extraction phase, to make the faces comparing more robust to variations
- **Feature Extraction:** a CNN was used for extracting discriminant features of the gallery images
- **Training on pairs:** negative and positive pairs (same and not the same person) are generated in this phase and the training is performed by the Siamese Networks to learn to differentiate between the gallery members
- **Threshold Definition:** validation samples are used to determine a threshold that can separate well: known from unknown individuals
- **Recognition:** after the system concludes training/testing stages, the trained Siamese Network and the gallery extracted features are used to compare with the features extracted from the probe images to determine the minimum similarity between probe and the gallery members. According to a determined threshold the person is recognized (known) or rejected (unknown)

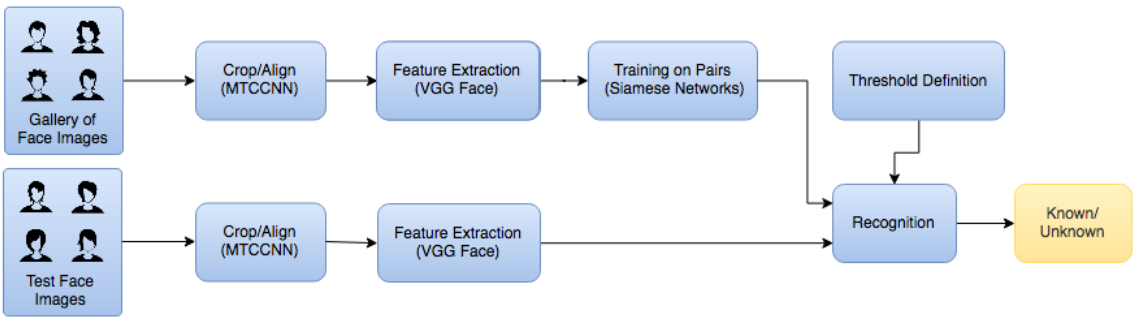


Figure 4.1: Illustration of the proposed method pipeline. Source: the author.

More details about the steps of the pipeline will be discussed in the next sections.

4.1 Pre-processing

For every face cropped image an alignment was performed and also resize if necessary as the feature extractor's input is a square image of dimensions 224×224 with 3 color channels.

The alignment was made using Joint Face Detection and Alignment using Multi-task Cascaded CNNs (Zhang et al. (2016)) as algorithms that do not use Neural Networks for face detection like the famous Viola e Jones (2004) (that also use cascade classifiers) are outdated as they were outperformed by Deep Learning in real-world scenarios.

The Multi-task Cascaded CNN can perform both face detection and alignment at the same time. There are three stages to perform both tasks:

- A shallow CNN produces shallow windows in the image that might be a face or not
- A more complex CNN reject most of the windows, as they are not faces
- An even more complex CNN analyzes the results refining it and outputs landmarks positions

Although there are "complex" CNNs in this method, they are lightweight as the method must detect faces in real time. This method outperformed state-of-the art methods and that is why it was used in the pre-processing stage, as the face alignment may improve feature extraction accuracy.

4.2 Feature Extraction

The extraction of features from the images for training and testing was made using a Deep Convolutional Network: VGG Face Descriptor Parkhi et al. (2015). This network has 21 layers (18 for convolution and pooling and 3 for classification) plus a Softmax function in the end for the purpose of prediction, as shown in Figure 4.2. VGG was pre-trained on a celebrity database that contained 2,622 individual identities with 375 images per subject resulting in almost 1 million images for training.

For only feature extraction task, the last Softmax function is removed, and the resulting size of the feature vector for every image is 2,622 floating point numbers, composing a feature vector of size 2622.

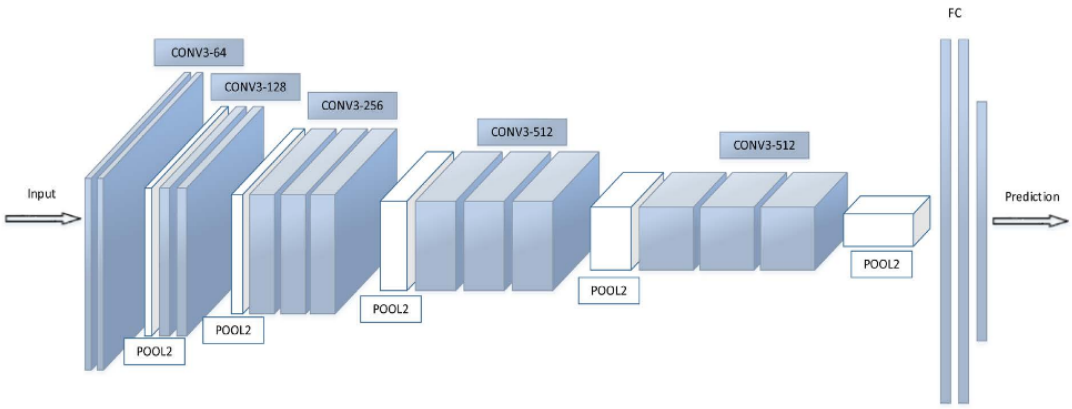


Figure 4.2: Illustration of the VGG Face Architecture. Source: El Khiyari e Wechsler (2016).

Table 4.1: Enumeration of layers, volume and parameters of the VGG Face. Source: El Khiyari e Wechsler (2016).

Layer	Volume	Parameters
INPUT	224 × 224 × 3	0
CONV3-64	224 × 224 × 64	(3 × 3 × 3) × 64 = 1,728
CONV3-64	224 × 224 × 64	(3 × 3 × 64) × 64 = 36,864
POOL2	112 × 112 × 64	0
CONV3-128	112 × 112 × 128	(3 × 3 × 64) × 128 = 73,728
CONV3-128	112 × 112 × 128	(3 × 3 × 128) × 128 = 147,456
POOL2	56 × 56 × 128	0
CONV3-256	56 × 56 × 256	(3 × 3 × 128) × 256 = 294,912
CONV3-257	56 × 56 × 256	(3 × 3 × 256) × 256 = 589,824
CONV3-258	56 × 56 × 256	(3 × 3 × 256) × 256 = 589,824
POOL2	28 × 28 × 256	0
CONV3-512	28 × 28 × 512	(3 × 3 × 256) × 512 = 1,179,648
CONV3-512	28 × 28 × 512	(3 × 3 × 512) × 512 = 2,359,296
CONV3-512	28 × 28 × 512	(3 × 3 × 512) × 512 = 2,359,296
POOL2	14 × 14 × 512	0
CONV3-512	14 × 14 × 512	(3 × 3 ×512) × 512 = 2,359,296
CONV3-512	14 × 14 × 512	(3 × 3 × 512) × 512 = 2,359,296
CONV3-512	14 × 14 × 512	(3 × 3 × 512) × 512 = 2,359,296
POOL2	7 × 7 × 512	0
FC-1	1 × 1 × 4096	7 × 7 × 512 × 4096 = 102,760,448
FC-2	1 × 1 × 4096	4096 × 4096 = 16,777,216
FC-3	1 × 1 × 2622	4096 × 2622 = 10,739,712

4.3 Siamese Network

As previously introduced in Chapter 2, Siamese Networks consist of a network that has 2 inputs and processes them independently in their own pipeline, with the exactly same weights and same architecture for each pipeline. In the end, the results of the 2 pipelines are merged into one, as the distance between the output of the networks is calculated using a distance function (L1, L2, Manhattan, Cosine Similarity etc). Each component will be discussed next.

4.3.1 Network Architecture

The input layer has size 2,622, which is the exact size of the output of the VGG Face.

Each Fully Connected Layer of the Siamese Network has 2,048 neurons, this final size was obtained empirically.

The weights of each FCN (Fully Connected Layer) are shared among the two separated internal pipelines of the network, this means that the weights are the same (not that they are connected).

The Siamese Network uses 3 fully connected layers as shown in Figure 4.3. On tests performed, fewer numbers of layers caused underfitting, and more caused overfitting. Dropout layers (Hinton et al. (2012)) were tested also. This type of layer removes some connections between the fully connected layers that may be paths that, sometimes, were only being used by a few subjects. This types of paths can cause generality loss. By removing the connections, Dropout layers force the training algorithm to use alternative paths and recalculate weights to correct the network output. Although, this is a great technique, in the proposed method, there

was no significant improvement in the results. Sometimes they can even add time to the training without really improving accuracy.

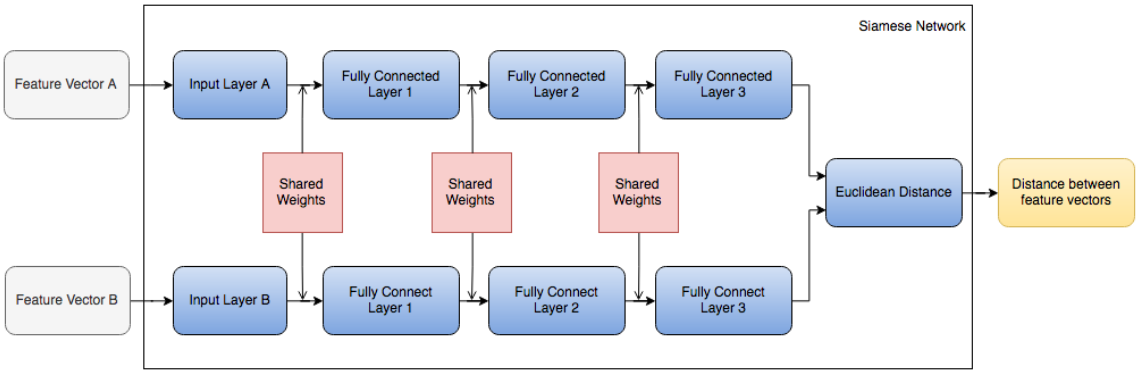


Figure 4.3: Illustration of the Siamese Architecture. Source: the author.

4.3.2 Distance Function

The Siamese Network's distance chosen was the Euclidean Distance. The other commonly used distance is cosine similarity, but in the tests performed there were convergence issues with this metric. The Euclidean Distance can be obtained between two vectors $\vec{p}$ and $\vec{q}$ of the same dimension n by the formula:

$$d(\vec{p},\vec{q}) = d(\vec{q},\vec{p}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

4.3.3 Loss Function

The loss function chosen was Contrastive Loss, introduced by Hadsell et al. (2006). It is appropriate in the Siamese parameters learning as it is models an energy based model to learn a mapping function, that is indeed metric learning. Also, it runs on pairs of samples.

The main goal is to map intra-class samples to neighbor areas and inter-class samples far away.

The weights (W) of the network are the parameters of the mapping function (called G), that reduces dimensionality by mapping two feature vectors of 2,622 dimensions into a distance measure: 0 if the feature vectors belong to the same person and 1 if they represent different identities. The distance measure is euclidean distance. The function G parameterized by the weights (parameters) W can be defined as G_W.

Given two vectors of the same size $\vec{X}_1$ and $\vec{X}_2$ as input to the system, and Y a label that equals 0 if the pair represents the same person, and 1 if it does not.

The euclidean distance D_W where W are the weights of the network (or parameters of the mapping function G_W):

$$D_W(\vec{X}_1,\vec{X}_2) = \| \; G_W(\vec{X}_1) - G_W(\vec{X}_2) \; \|$$

For simplicity $D_W(\vec{X}_1, \vec{X}_2)$ will be denoted as only D_W and the final loss function is proposed by Hadsell et al. (2006) has the following definition:

$$L(W, Y, \vec{X}_1, \vec{X}_2) = \frac{(1-Y)(D_W)^2 + Y\{\max(0, m - D_W)\}^2}{2}$$

The variable m is a predefined margin of $G_W(\vec{X}_1)$ that should be greater than 0. If the two vectors $\vec{X}_1$ and $\vec{X}_2$ belong to different people and the distance between them is within this margin, the loss function will try to separate them, otherwise, it will no take into account. The margins tested range between 0.7 and 1.3.

4.4 Training Stage

4.4.1 Pair Generation

The maximum number of unique pairings of the training samples can be calculated as the combination 2 by 2 of the n samples:

$$\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$$

If the cost of processing each pair is a constant c, the method cost of training once on all pairs will be:

$$cost = \frac{c \times (n^2 - n)}{2} = O(n^2)$$

This can be costly, taking into account that a few iterations on training over almost n^2 pairs can take very long, as neural networks are usually expensive to train. Subsampling is needed to reduce the number of pairs used in training. Two forms of pair subsampling that are much less expensive than pairing all samples were evaluated. For explanation purposes they will be called: P1 and P2.

For every training face feature sample, P1 generates pairs using the sample and for each one of the other identities (inter-class), it generates 1 or more pairs, and for balancing purposes, generates pairings with same identity samples (intra-class).

On the other hand, P2 generates fewer pairs. It generates for each training sample n pairs with the sample and a random other sample of another individual (inter-class). It also generates n pairs between the sample and other samples from the same individual (intra-class). In the end, 2 additional pairings of the sample with other individuals samples are generated.

These techniques are critical for the method to work correctly, as they determine the quality of the Siamese Networks learning and the discrimination capability.

These specific techniques were achieved by trial and error on tests performed.

4.5 Threshold Definition

In testing stage, a threshold would be selected based on system sensitivity to false positives. Precision vs recall and ROC curves (these concepts will be discussed in Chapter 4) could help in determining this threshold. This part of the method will not be implemented in this work, as it is not the main problem and there are no reference protocols to test it.

4.6 Recognition

The Recognition process occurs as illustrated in Figure 4.4. First, the probe feature vector is compared with one feature vector from each gallery individual (randomly chosen) using the Siamese Network. If the gallery has i identities, i comparisons will be executed.

After the distance from the probe vector to the individuals of the gallery is calculated, the minimum distance between them will be compared to a threshold (defined at testing). If the distance is lower than the threshold the probe is classified as known, otherwise it is classified as unknown. As mentioned before, the definition of this threshold will not be presented, as tests conducted deal with a range of thresholds (ROC curve).

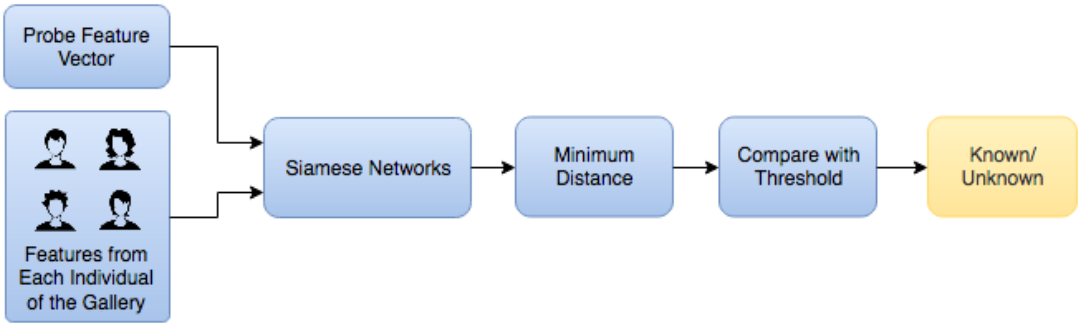


Figure 4.4: Illustration of the Recognition process. Source: the author.

5 Experiments

5.1 Experimental Protocol

The main issue on choosing the experimental protocol is that Face Verification, by example, has well-known protocols for testing (LFW, YoutubeFaces are broadly used), but in open-set recognition there is a lack of consensus among authors for which is the best protocol. This makes it really difficult to compare different papers as many of them does not determine precisely how the tests were conducted. Luckily, the protocol that will be used was well described, and a fair comparison with the original paper can be made. The protocol that will be used for literature comparison was proposed by Vareto et al. (2017).

Tests were conducted in a 16 core machine (4 processors Quad-Core AMD Opteron 8387) and a Titan X Pascal GPU (3,584 cores) with FP32 performance of 11 TFLOPS/s. The operating system was Ubuntu 16.04.

5.1.1 Protocol Description

It consists in sectioning the database in training and testing data. The training data has only known individuals (individuals that should be recognized by the method) and the testing data has known individuals (but samples of them that the method has never seen before) and unknown individuals (samples of people the method has never seen before). Bringing back the concept of known knowns and unknown unknowns, there are no known unknowns in the training stage. This is a hard issue, as the method can not know how to separate known individuals from unknown ones (what the mean distance is between them in space).

The whole dataset is first divided between knowns/unknowns individuals. The knowns are 10%, 50% and 90% of the identities and the rest is considered as unknown and used only for testing.

The knowns are then split between training/testing in a fixed 50% rate.

Each test on each base is repeated 10 times by randomly splitting the data as aforementioned, training, testing and plotting the ROC curves with AUC. The mean AUC for each these 10 tests is computed, together with the standard deviation.

5.1.2 Metrics

ROC curve and AUC

The ROC (Receiver Operation Characteristics) curve is a well-used technique for algorithm evaluation due to two reasons: i) its curve describes the algorithm performance on various thresholds; ii) it is robust to lack of balance in the classes (measures like accuracy can be deceiving in unbalanced data).

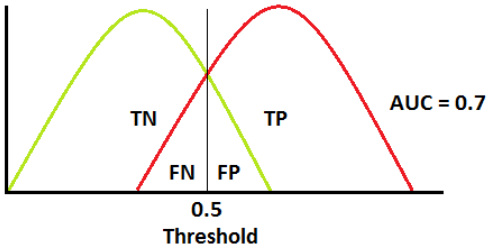
The ROC curve shows the relation between TPR (True Positive Rate) and FPR (False Positive Rate). TPR oscillates between 0 and 1 indicating how many samples that were correctly labeled as positive in relation to all that are positive. Values closer to 1 indicate high recall rates. FPR also oscillates between 0 and 1 and shows how many negative samples were labeled as positives in relation to all negative samples. Values closer to 0 indicate almost perfect false acceptation rate. As show in Figure 5.1(b), the ROC curve that indicates great performance is closer to the left upper side. Figure 5.1(a) illustrates the distribution of results in relation to true positives, true negatives, false negatives and false positives, with threshold 0.5.

A high TPR indicates that the algorithm correctly classifies lots of known people. A high FPR indicates that the algorithm classifies lots of people as known that are in fact unknown.

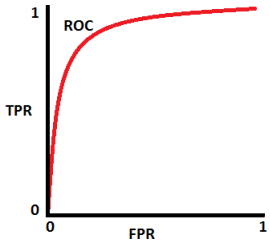
AUROC (Area Under the Receiver Operation Characteristics) or AUC for simplicity, indicates the separation degree between the True Negatives and True Positives. In the ROC curve, it indicates the area under the ROC curve (bigger area indicates better performance).

If the AUC equals 1, the method is perfect and separates the positives from the negatives samples. However, if the AUC equals 0.5 (Figure 5.2(a)) that means that the method does not separate the data, it has the same probability of measuring correctly the sample's class as "tossing an honest coin". The corresponding ROC curve for AUC=0.5 will be a straight line as depicted in Figure 5.2(b).

We can say that the method evaluated in Figure 5.1 is better than the one in Figure 5.2 as the AUC is higher, and the separation between classes is better (as shown in the left images).

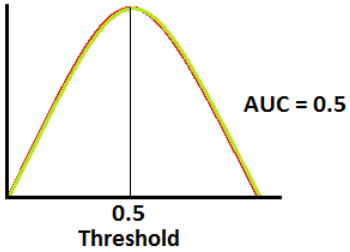


(a) Distribution of TP and FP, the algorithm perfomance is indicated by the AUC=0.7.

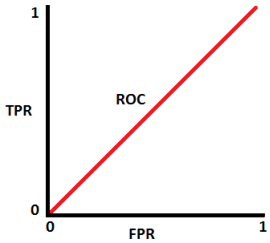


(b) ROC curve that shows the relation between TPR and FPR for various operating thresholds.

Figure 5.1: ROC curve illustration. Source: Narkhede (2018).



(a) Distribution of TP and FP, there is no separation between them.



(b) ROC curve is a straight line.

Figure 5.2: ROC curve illustration. Source: Narkhede (2018).

Precision vs. Recall

Another metric that will be used is PR (Precision vs. Recall). Precision is the number of true positives in relation of the samples considered as positive:

$$P = \frac{TruePositive}{TruePositive + FalsePositive}$$

Recall is the number of true positive in relation to the true positives and false negatives (the correctly classified as positives and the wrongly classified as negative):

$$R = \frac{TruePositive}{TruePositive + FalseNegative}$$

An increase in recall reduces the chances of a known person being considered as unknown, but also increases the probability of an unknown person being considered as known. An increase in precision does the opposite: it reduces the false positives, but also has greater chance of considering known people as unknown.

5.2 Tests Results

5.2.1 Datasets

One of the most challenging and used datasets for face recognition is certainly LFW. But, as previously stated there is no universal protocol for open-set evaluation and for comparing with the literature. It is dispendious to replicate every protocol of other authors. As the same protocol is being used, for comparison with state-of-art, two datasets that Vareto et al. (2017) used for testing were considered: Pubfig83 (Pinto et al. (2011)) and FRGCv1 (Phillips et al. (2005)).

PubFig83

It has 83 subjects and 13,838 uncontrolled images (average of 166 samples per subject), as Figure 5.3 demonstrates.

Figure 5.3: Dataset image examples for several celebrities on PubFig83. Source: Pinto et al. (2011).

FRGCv1

This is the dataset provided for the Face Recognition Grand Challenge v1.0, created by Phillips et al. (2005). Although it is discontinued (currently there is only v2.0 available for licensed download), it will be used to compare with Vareto et al. experiments.

It has a total of 50,000 images and 6 experiments, Figure 5.4 shows samples from two identities present in the dataset. Tests were only conducted on experiment 1 and 4. As the other ones (except experiment 2) are composed of 3D images. The first experiment is composed by controlled 760 images and the fourth has 152 controlled photos + uncontrolled 1064. Both experiments have 152 distinct subjects.

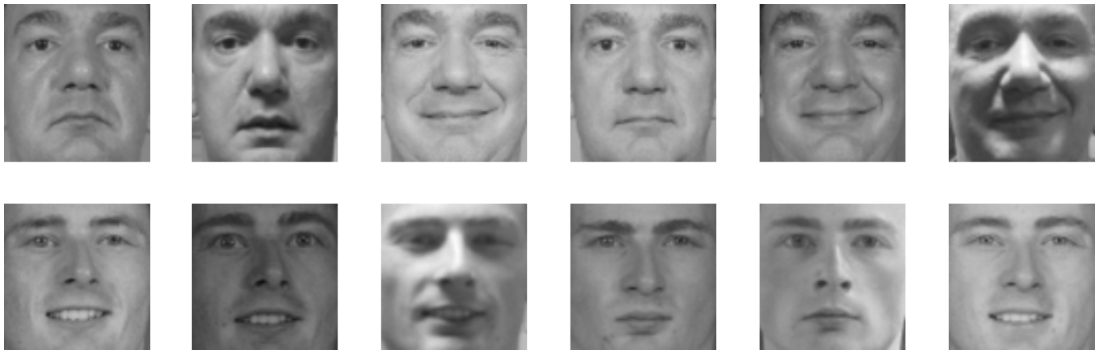


Figure 5.4: Dataset image examples for two subjects on FRGCv1 with light and pose variations. Source: Pagano et al. (2015).

5.2.2 Optimizer and Training Parameters

Several optimizers were tested like Adam, Root Mean Square Prop (RMSProp), with several adjustments in the parameters. But, the best results were achieved with the Stochastic Gradient Descent (SGD).

Several neuron activation functions were tested (with their respective optimizer), like ReLU, SELU, ELU, but in the end, the chosen one was Sigmoid. Although there is the risk of suffering from exploding gradient problem, in tests it outperformed the other options and had the most stable convergence.

There are hyperparameters to be adjusted in the Siamese training. One of them is learning rate, basically how much the weights of the network are updated towards the gradient, The second parameter is decay: how much the learning rate is diminished after every training batch is processed.

Momentum can also be set, like in Physics, momentum is the quantity of motion an object has. In the context of neural networks, momentum is used as a force trying to maintain the gradient descent towards the same direction as the previous descents (weight corrections) were.

The training metric utilized was accuracy, there was also a validation split for the network to know where to stop the learning process, according to the validation loss value stagnation. The splitting function initially was stratified but some bases had few samples per subject, demanding a high number of samples for validation (some cases were more than 35% of the training pairs).

By changing to non-stratified splitting it was possible to lower the splitting to 10-20%, increasing training performance (as there is more data to train instead of being use for validation only).

The last parameter is batch size, that determines how many inputs will be input through the network before the weights are corrected. Also how many pairs will be loaded to the GPU or memory.

The best parameters discovered in exhaustive experiments for this network are:

- **learning rate** = 0.01
- **decay** = 1×10^{-6}
- **momentum** = 0.9
- **validation split** = 10-40%
- **batch size** = 64

The protocol required that for each dataset and percentage of known individual experiments to be executed 10 times. Each experiment was executed 10 times (5 times using pairing P1, 5 times using pairing P2), and the ROC curve and PR curve were calculated. For each execution the datasets were split randomly.

5.2.3 Pubfig83

The method achieved great results in Pubfig83 for small galleries (few subjects known). Although with the increase of number of subjects, the AUC dropped. Significantly in 90% known. The PR results were good as the precision only dropped on high recall values, which indicates that if the method is more permissive with the threshold, fewer false positives are generated. Table 5.1 presents the results obtained. The graphics showing results can be seen in Appendix A.1.

Table 5.1: Results for experiments on Pubfig83

Pair	Known ratio	ROC (AUC)	PR
P1	10%	0.981 ± 0.008	0.897 ± 0.031
P2		0.981 ± 0.007	0.895 ± 0.028
P1	50%	0.922 ± 0.015	0.904 ± 0.021
P2		0.916 ± 0.018	0.896 ± 0.023
P1	90%	0.856 ± 0.017	0.962 ± 0.004
P2		0.854 ± 0.025	0.962 ± 0.006

5.2.4 FRGCv1 - Experiment 1

In the experiment 1 from FRGCv1 great results were obtained. This is due to the fact that there are only controlled images on this dataset. There is no great challenge in discriminating controlled images. There was no difference in the results concerning the type of pairing used in training. The increase in gallery size did not influence dramatically the AUC, the difference was quite small. Table 5.2 presents the results obtained. The graphics showing results can be seen in Appendix A.2.

Table 5.2: Results for experiments on FRGCv1, experiment 1

Pair	Known ratio	ROC (AUC)	PR
P1	10%	0.996 ± 0.002	0.966 ± 0.017
P2		0.996 ± 0.003	0.965 ± 0.019
P1	50%	0.986 ± 0.004	0.982 ± 0.004
P2		0.986 ± 0.003	0.982 ± 0.003
P1	90%	0.972 ± 0.018	0.994 ± 0.004
P2		0.974 ± 0.008	0.994 ± 0.002

5.2.5 FRGCv1 - Experiment 4

The small size gallery obtained better results than the bigger galleries, as in other test. But a weird phenomenon occurred in PR for the 10% known tests. With the increase of Recall, the Precision dropped almost exponentially, this indicates that if we diminish the threshold for avoiding known people to be classified as unknown, the number of false positives (unknown people being considered as known) increases almost more than linearly. Table 5.3 presents the results obtained. The graphics showing results can be seen in Appendix A.3.

Table 5.3: Results for experiments on FRGCv1, experiment 4

Pair	Known ratio	ROC (AUC)	PR
P1	10%	0.950 ± 0.026	0.828 ± 0.050
P2		0.904 ± 0.032	0.688 ± 0.082
P1	50%	0.841 ± 0.020	0.820 ± 0.020
P2		0.816 ± 0.019	0.794 ± 0.021
P1	90%	0.797 ± 0.030	0.954 ± 0.007
P2		0.770 ± 0.021	0.946 ± 0.006

5.2.6 Comparison

Table 5.4: Results comparison with the work of Vareto et al. (2017) on FRGCv1 - experiment 4:

Known Individuals		10%	50%	90%
Proposed (Siamese)	AUC	**0.935**	0.828	0.783
	STD	±0.041	±0.023	±0.029
HPLS	AUC	0.794	0.850	**0.856**
	STD	±0.078	±0.009	±0.022
HFCN	AUC	0.872	**0.866**	**0.856**
	STD	±0.015	±0.022	±0.014

As shown in Table 5.4, for small number of identities (small galleries) the proposed method with Siamese Networks can surpass a state-of-art method in experiment 4 of FRGCv1, using the same protocol, and with 10 executions for each experiment. As the gallery grows, the recognition rate drops.

6 Conclusion

A recognition system, with well defined pipelines using Siamese Networks is indeed a good option for open-set face recognition, a task that still has not been extensively researched. Going against the flux, as most work invest in scalability, the focus was on small galleries and the Siamese Networks outperformed state-of-art methods like HPLS and HFCN.

On the controlled test the method performed really well, achieving great results. The PubFig83 evaluation results were also remarkable as it is not a controlled gallery. Unfortunately, there was no data available to compare it to Vareto et al. (2017) published work, as the FRGCv1 - experiment 4. So it is not possible to determine if it outperforms the other methods.

In future works, the pairing process can be improved, that can be the most critical part of the training, along with some sort of dynamic generation of pairs (generate pairs for each epoch of the network training). Also, association with other metric learning techniques, loss function improvements and hyperparameter optimizations could improve the final results. Furthermore, implementing the complete system and testing it on real world scenarios can be an interesting path.

References

AbdAlmageed, W., Wu, Y., Rawls, S., Harel, S., Hassner, T., Masi, I., Choi, J., Lekust, J., Kim, J., Natarajan, P. et al. (2016). Face recognition using deep multi-pose representations. Em *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, páginas 1–9. IEEE.

Bengio, Y., LeCun, Y. e Henderson, D. (1994). Globally trained handwritten word recognizer using spatial representation, convolutional neural networks, and hidden markov models. Em *Advances in neural information processing systems*, páginas 937–944.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. e Shah, R. (1994). Signature verification using a" siamese" time delay neural network. Em *Advances in neural information processing systems*, páginas 737–744.

Chellappa, R., Sinha, P. e Phillips, P. J. (2010). Face recognition by computers and humans. *Computer*, 43(2).

Chopra, S., Hadsell, R. e LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, páginas 539–546. IEEE.

Dalal, N. e Triggs, B. (2005). Histograms of oriented gradients for human detection. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, páginas 886–893. IEEE.

de Paulo Carlos, G., Pedrini, H. e Schwartz, W. R. (2015). Classification schemes based on partial least squares for face identification. *Journal of Visual Communication and Image Representation*, 32:170–179.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. e Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. Em *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, páginas 248–255. IEEE.

dos Santos Junior, C. E. e Schwartz, W. R. (2014). Extending face identification to open-set face recognition. Em *Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on*, páginas 188–195. IEEE.

El Khiyari, H. e Wechsler, H. (2016). Face recognition across time lapse using convolutional neural networks. *Journal of Information Security*, 7(03):141.

Günther, M., Cruz, S., Rudd, E. M. e Boult, T. E. (2017). Toward open-set face recognition. Em *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. IEEE*.

Hadsell, R., Chopra, S. e LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. Em *null*, páginas 1735–1742. IEEE.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. e Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Huang, G. B., Mattar, M., Berg, T. e Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. Em *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*.

Huang, T. (1996). Computer vision: Evolution and promise. *Cern*.

Khalil-Hani, M. e Sung, L. S. (2014). A convolutional neural network approach for face verification. Em *High Performance Computing & Simulation (HPCS), 2014 International Conference on*, páginas 707–714. IEEE.

Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A. e Jain, A. K. (2015). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 1931–1939.

Krizhevsky, A., Sutskever, I. e Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Em *Advances in neural information processing systems*, páginas 1097–1105.

Lawrence, S., Giles, C. L., Tsoi, A. C. e Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113.

Li, F. e Wechsler, H. (2005). Open set face recognition using transduction. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1686–1697.

Narkhede, S. (2018). Understanding auc - roc curve. `https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5`. Accessed in 2018-11-30.

Otero, I. R. (2015). *Anatomy of the SIFT Method*. Tese de doutorado, École normale supérieure de Cachan-ENS Cachan.

Pagano, C., Granger, E., Sabourin, R., Tuveri, P., Marcialis, G. e Roli, F. (2015). Context-sensitive self-updating for adaptive face recognition. Em *Adaptive Biometric Systems*, páginas 9–34. Springer.

Parkhi, O. M., Vedaldi, A., Zisserman, A. et al. (2015). Deep face recognition. Em *BMVC*, página 6.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J. e Worek, W. (2005). Overview of the face recognition grand challenge. Em *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, volume 1, páginas 947–954. IEEE.

Pinto, N., Cox, D. D. e DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27.

Pinto, N., Stone, Z., Zickler, T. e Cox, D. (2011). Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook. Em *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, páginas 35–42. IEEE.

Proedrou, K., Nouretdinov, I., Vovk, V. e Gammerman, A. (2002). Transductive confidence machines for pattern recognition. Em *European Conference on Machine Learning*, páginas 381–390. Springer.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A. e Boult, T. E. (2013). Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772.

Schroff, F., Kalenichenko, D. e Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 815–823.

Simonyan, K., Parkhi, O. M., Vedaldi, A. e Zisserman, A. (2013). Fisher vector faces in the wild. Em *BMVC*, página 4.

Stan, Z. L. e Anil, K. J. (2011). *Handbook of face recognition*. Springer.

Sun, Y., Chen, Y., Wang, X. e Tang, X. (2014). Deep learning face representation by joint identification-verification. Em *Advances in neural information processing systems*, páginas 1988–1996.

Sun, Y., Liang, D., Wang, X. e Tang, X. (2015). DeepID3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. e Rabinovich, A. (2015). Going deeper with convolutions. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 1–9.

Taigman, Y., Yang, M., Ranzato, M. e Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 1701–1708.

Taigman, Y., Yang, M., Ranzato, M. e Wolf, L. (2015). Web-scale training for face identification. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 2746–2754.

Vareto, R., Silva, S., Costa, F. e Schwartz, W. R. (2017). Towards open-set face recognition using hashing functions. Em *Biometrics (IJCB), 2017 IEEE International Joint Conference on*, páginas 634–641. IEEE.

Viola, P. e Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.

Wolf, L., Hassner, T. e Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. Em *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, páginas 529–534. IEEE.

Yi, D., Lei, Z., Liao, S. e Li, S. Z. (2014). Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*.

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

Zhang, K., Zhang, Z., Li, Z. e Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.

Zhou, X. S. e Huang, T. S. (2001). Small sample learning during multimedia retrieval using biasmap. Em *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, páginas I–I. IEEE.

Appendix A: ROC and PR curves for databases on experiments

The following appendixes show the ROC curves (with AUC) and PR curves for all experiments performed in each dataset. On top of each graphic, you can see the mean AUC, with the standard deviation. On the bottom right, it shows a legend of every curve, one to every execution, with its own corresponding color, AUC and standard deviation.

A.1 ROC and PR curves for PubFig83

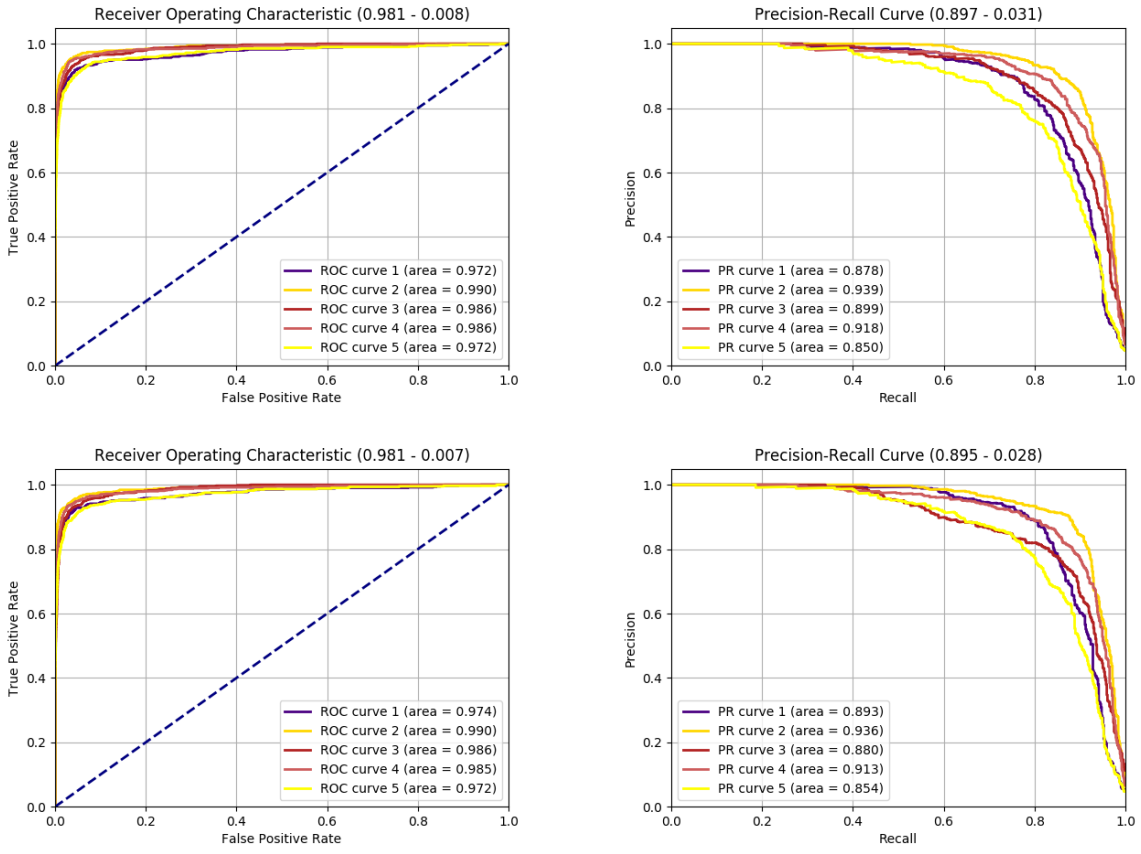


Figure A.1: ROC and PR curves for PubFig83 using P1 (upper row) and P2 (lower row), the known ratio is 10%

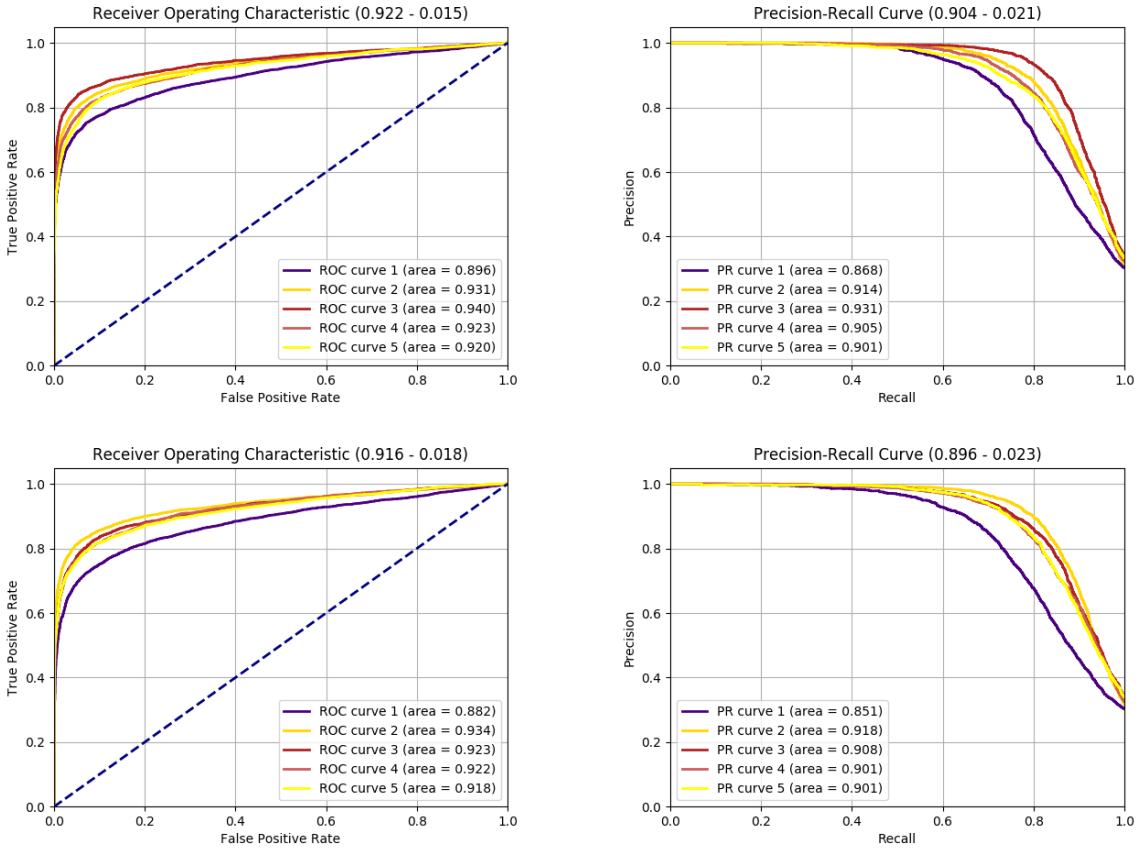


Figure A.2: ROC and PR curves for PubFig83 using P1 (upper row) and P2 (lower row), the known ratio is 50%

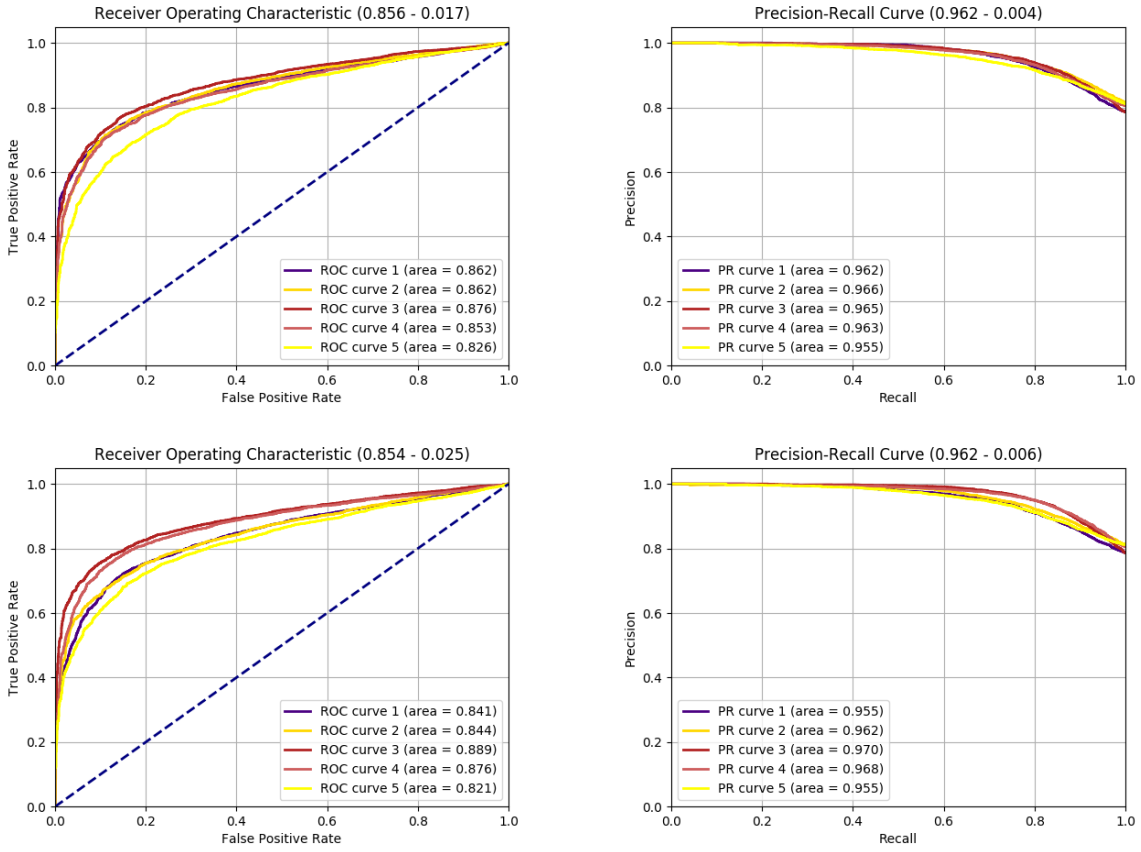


Figure A.3: ROC and PR curves for PubFig83 using P1 (upper row) and P2 (lower row), the known ratio is 90%

A.2 ROC and PR curves for FRGCv1 - Experiment 1

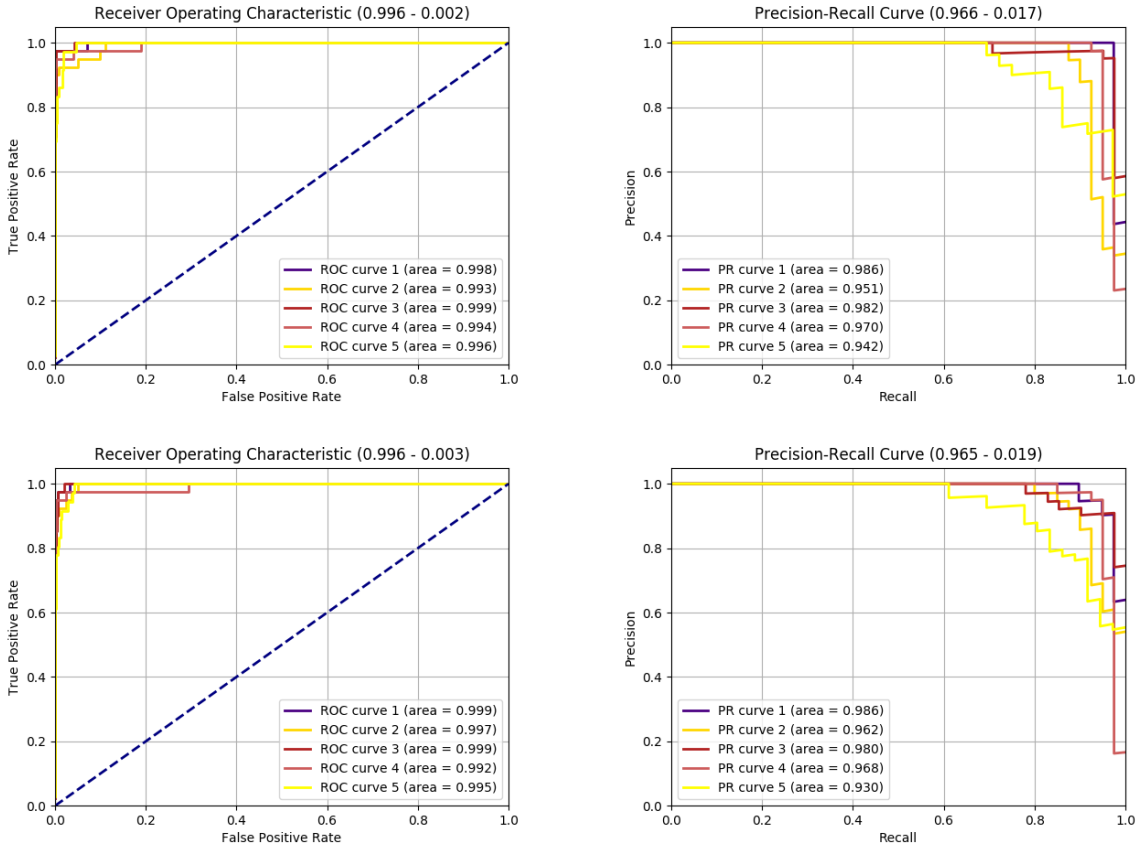


Figure A.4: ROC and PR curves for FRGCv1 experiment 1 using P1 (upper row) and P2 (lower row), the known ratio is 10%

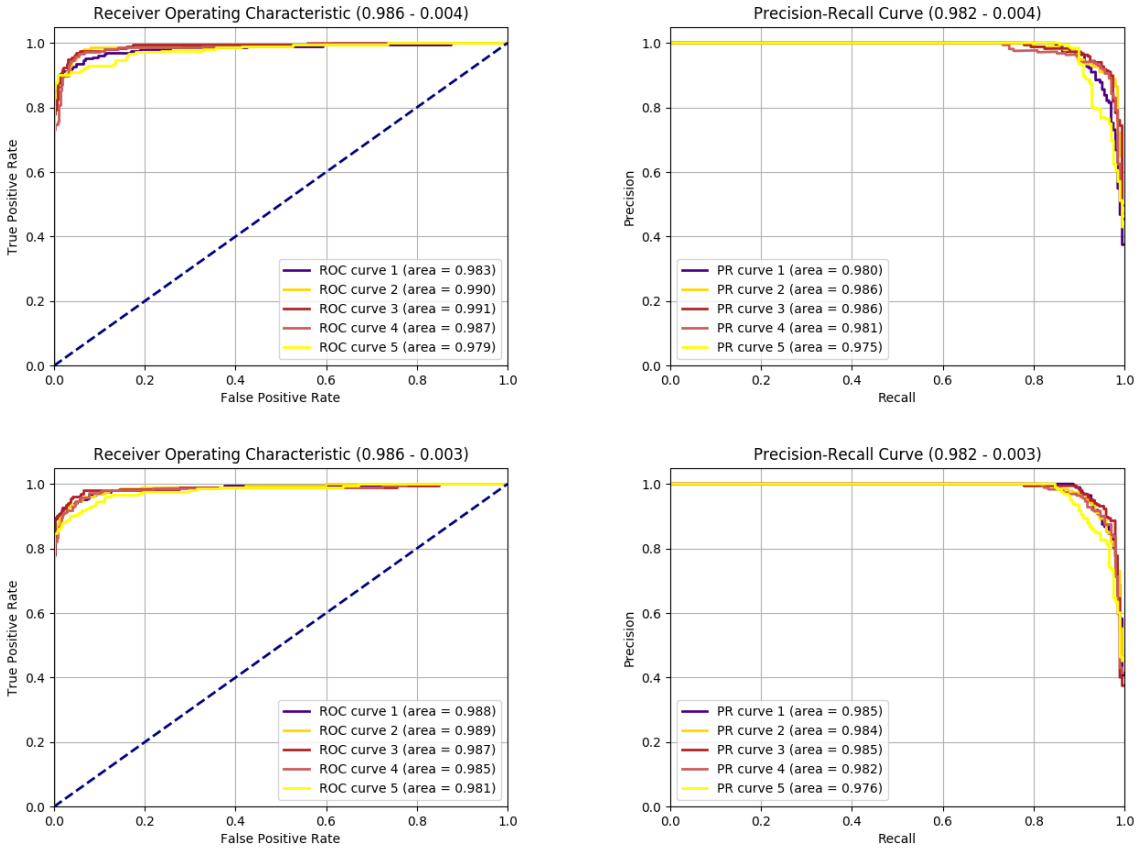


Figure A.5: ROC and PR curves for FRGCv1 experiment 1 using P1 (upper row) and P2 (lower row), the known ratio is 50%

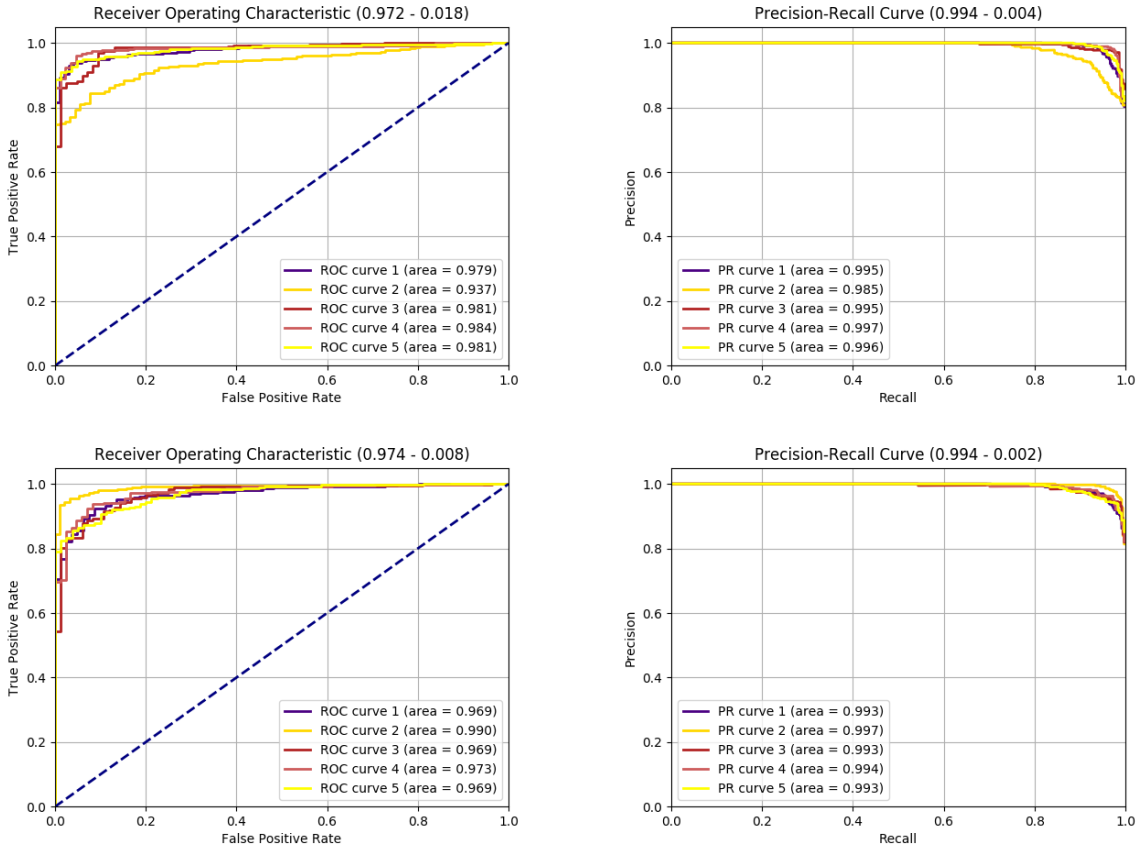


Figure A.6: ROC and PR curves for FRGCv1 experiment 1 using P1 (upper row) and P2 (lower row), the known ratio is 90%

A.3 ROC and PR curves for FRGCv1 - Experiment 4

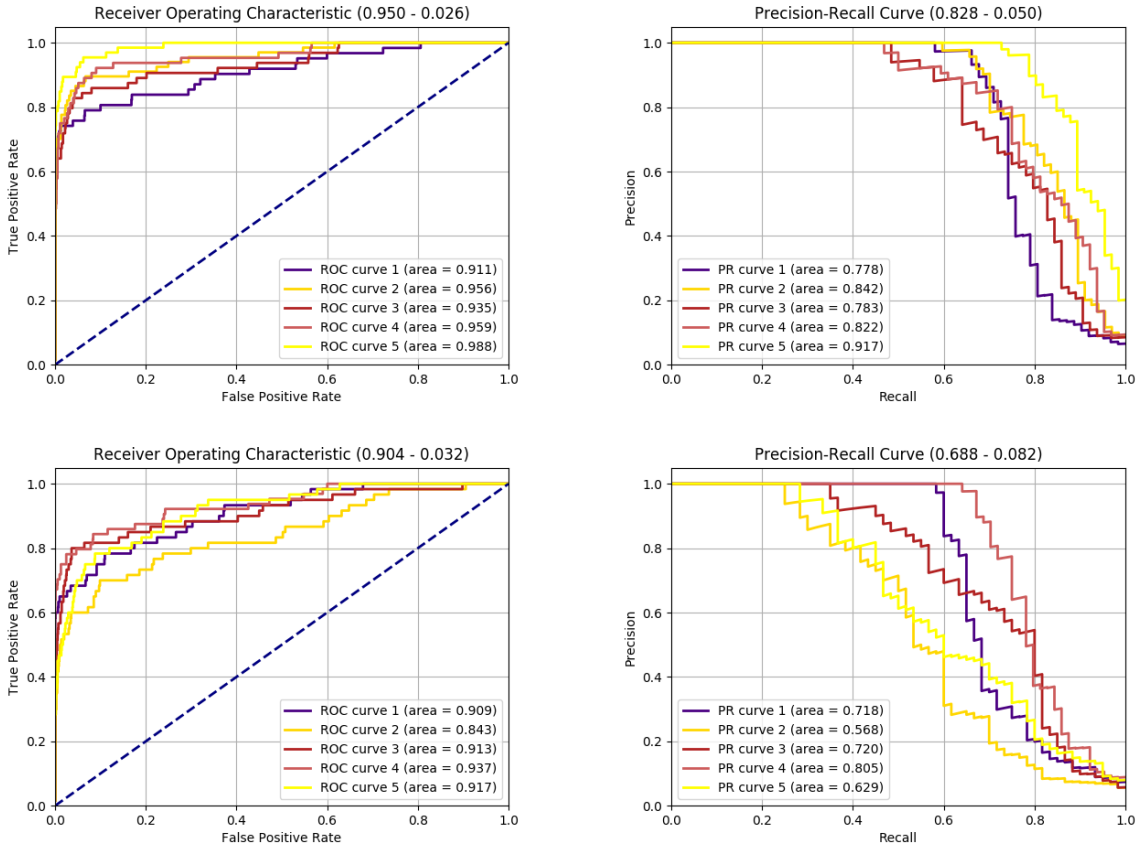


Figure A.7: ROC and PR curves for FRGCv1 experiment 4 using P1 (upper row) and P2 (lower row), the known ratio is 10%

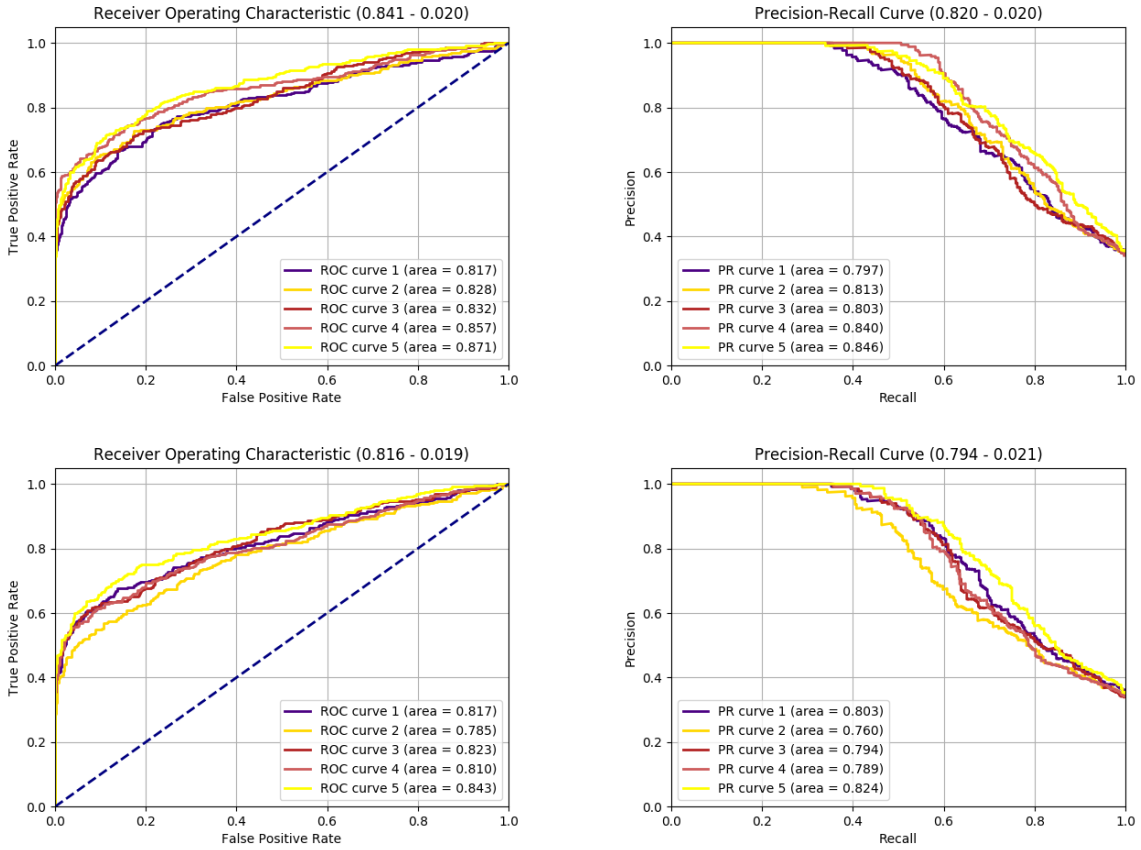


Figure A.8: ROC and PR curves for FRGCv1 experiment 4 using P1 (upper row) and P2 (lower row), the known ratio is 50%

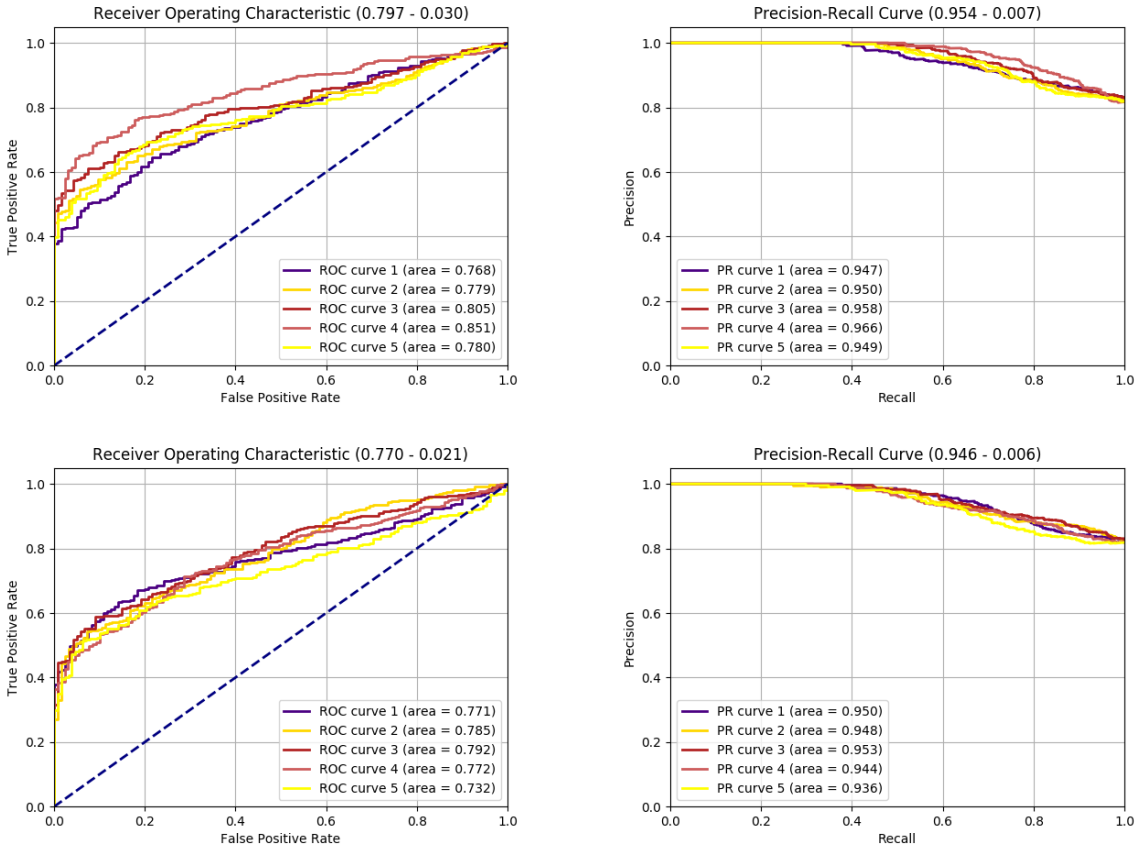


Figure A.9: ROC and PR curves for FRGCv1 experiment 4 using P1 (upper row) and P2 (lower row), the known ratio is 90%