Towards Open-Set Face Recognition using Hashing Functions

Rafael Vareto, Samira Silva, Filipe Costa, William Robson Schwartz
Smart Surveillance Interest Group, Department of Computer Science
Universidade Federal de Minas Gerais, Minas Gerais, Brazil
`{rafaelvareto, samirasilva, fcosta, william}@dcc.ufmg.br`

Abstract

Face Recognition is one of the most relevant problems in computer vision as we consider its importance to areas such as surveillance, forensics and psychology. Furthermore, open-set face recognition has a large room for improvement since only few researchers have focused on it. In fact, a real-world recognition system has to cope with several unseen individuals and determine whether a given face image is associated with a subject registered in a gallery of known individuals. In this work, we combine hashing functions and classification methods to estimate when probe samples are known (i.e., belong to the gallery set). We carry out experiments with partial least squares and neural networks and show how response value histograms tend to behave for known and unknown individuals whenever we test a probe sample. In addition, we conduct experiments on FRGCv1, PubFig83 and VGGFace to show that our method continues effective regardless of the dataset difficulty.

1. Introduction

Face Recognition (FR) systems are in great rush to help fight crime and terrorism. It has numerous applications to areas such as forensics, surveillance and law enforcement, especially public and financial security. Other applications involve user authentication for access control to physical and virtual spaces to ensure higher security.

Traditional face recognition approaches conventionally extract image features that correspond to facial components. Particularly, these methods would initially search for shape of the eyes, mouth contour, nose appearance, face silhouette to name a few and use them as discriminative features while exploring other images. Face Recognition is a general term that is used to refer to distinct problems. Chellappa et al. [1] define three main tasks. The *face verification*, in which the goal is to determine whether a pair of images corresponds to the same subject. That is, the verification task is equivalent to 1:1 matching problems; The *face Identification*, a 1:N matching task, which assumes that every queried subject was previously cataloged, ensuring that the probe face holds a corresponding identity in the gallery set. It characterizes a closed-set problem; the *watch-list*, similar to face identification but it does not guarantee all query subjects are registered in the face gallery, representing an open-set problem.

There are several works on closed-set face identification [2–7]. Real-world applications cannot assume every query image is known and, as a result, they are better represented by the open-set task since there is only a partial knowledge of the world and countless unknown inputs. A scenario like this comprehends a classification model where only few classes are known at training time, but a myriad of unknown classes appear at test time [8].

This work is inspired on an method proposed by Santos et al. [9], which provides a scalable closed-set face identification approach to galleries with hundreds and thousands of subjects. Instead of working on the face identification task, we focus on the watch-list setting, where most query images presented to the recognition procedure do not match any of the subjects previously enrolled in the gallery set.

Our main hypothesis is that vote list histograms proceed differently whether we present probe face images whose identity are enrolled in the gallery or whether we examine "unseen" individuals. Figure 1 illustrates two different queries: one corresponds to querying an enrolled individual and the other searches for an unknown subject. Notice that there is a highlighted bin when the query image has a matching identity. It probably represents the probe sample's identity. On the contrary, several bins are incremented when no subject from the gallery corresponds to the probe sample.

In the proposed approach, we combine locality-sensitive hashing (LSH), partial least squares (PLS) and fully connected networks (FCN) to get the best of the three worlds. LSH was designed to solve near neighbor search in high dimensional spaces. It hashes data in such a way that similar items tend to map to the same "bucket". PLS weights feature descriptors to best discriminate throughout different classes, handling high-dimensional data and

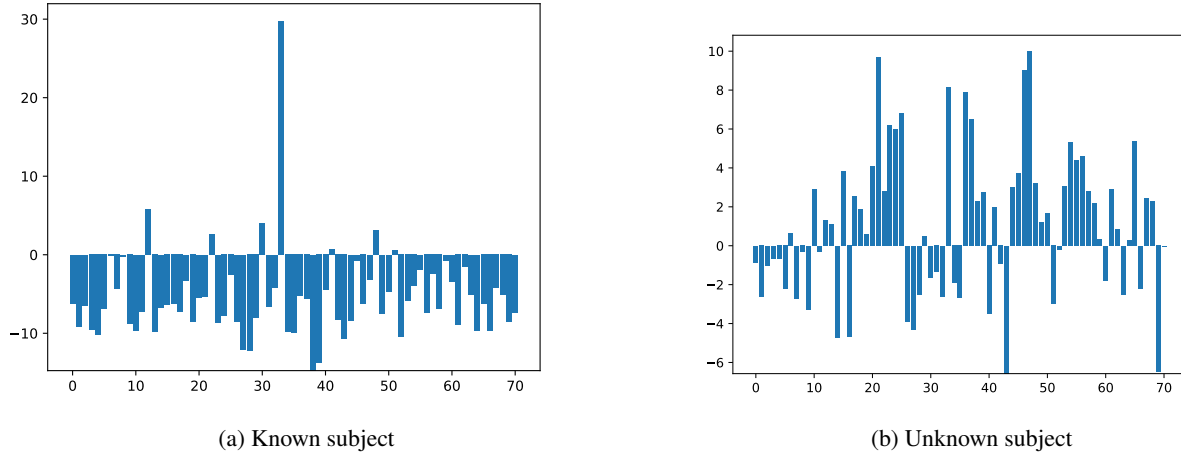| (a) Known subject | (b) Unknown subject |

Figure 1: Two queries for the same individual of the FRGCv1 dataset when he is either registered in the gallery set *(a)* or a completely unknown person *(b)*. A considerable number of subjects from the gallery set turn into candidates when there is no clue what the identity for the query image is.

successfully overcoming the problem of having just a few samples per class. FCN is a biologically-based programming paradigm that enables computer systems to learn from observational data. It is composed of numerous highly interconnected processing elements (neurons) working in harmony to solve specific problems. Therefore, we replace each LSH random projection either by PLS or FCN to obtain better discrimination between positive and negative samples.

We reimplement and adapt the partial least squares algorithm for face hashing [9] to find out whether a query sample is known, that is, it has a corresponding identity in the face gallery. Then, we carry on replacing PLS regression models with fully connected networks. We present our results on different datasets – FRGCv1 [10], PubFig83 [11] and VGGFace [12] – and show that they continue stable regardless of the chosen dataset.

The predominant contributions of this work are: *1)* an adaptation of locality-sensitive hashing combined with two different classifiers to the open-set recognition problem in a supervised learning setting; *2)* an easy-to-implement algorithm with a single trade-off parameter to be estimated.

2. Related Work

Only recently has open-set recognition been explored in the literature. Specifically for face recognition, few works focus on finding thresholds that must be satisfied so that probe face images are identified as known. There has been a lot more works intended to solve closed-set problems, either in unrestrained scenarios or in relatively "small" datasets [5, 13–15]. It is needless to say that these studies accomplished substantial progress in the last 10 years; however, FR is far from being solved since many applications have failed to deliver in watch-list scenarios.

Support Vector Machines (SVM) are broadly used in image retrieval problems. With the advent of one-class SVM [16, 17], some researchers became involved in open-set tasks [18–21] as it seems reasonable to train a classifier with only known positive data. In cases of restricted training time or too much data, one-class support vector machine faces some issues since it generates kernel models and, therefore, it is not very scalable.

The work of Kamgar-Parsi et al. [22] consists in performing open-world face recognition by learning a classifier for every single subject from a watch-list. Each subject's model is likely to accept images from corresponding targets and reject everybody else. Thus, there is no experiment on open-set face recognition.

Liao et al. [23] developed a protocol to explore all face images available in the Labeled Faces in the Wild (LFW) dataset [24] for large-scale recognition under verification and open-set identification scenarios. They concluded that open-set recognition persists unresolved for large galleries.

Santos et al. [25] proposed five different methods in a single work: one of them stands on discriminating face samples between known and background set. The four remaining methods are based on identification responses. They encompass *Chebyshev* inequality, SVM classifiers and least squares. Their approaches did not attain satisfactory accuracy in large gallery sets.

Bendale et al. [26] introduced a formal definition to the open-world problem. They came up with Nearest Non-Outlier (NNO), an algorithm that continuously update its inner model with additional unseen object categories and no need to retrain the entire model. NNO minimizes the risk of falling into open space, a space sufficiently far from any known positive training sample. The algorithm rejects

a query q as an outlier when it is too far from any training sample. As consequence, it labels q as *unknown* when all classes reject q as an outlier

Zhang et al. [27] proposed a new Sparse Representation and Classification algorithm (SRC). They introduced a training stage to the SRC algorithm so that it can be adapted to tackle open-set recognition problems by seeking the sparsest representation in terms of the training data.

The last direction of relevant work lies on Locality-Sensitive Hashing (LSH) [28, 29], a family of embedding approaches that reduces the dimensionality of high dimensional data. It is data independent, i.e., it does not explore the data distribution. With LSH, there is a high probability feature descriptors map to the same hashing location when they are located in neighboring regions in the feature space. One of the main deprivations of the LSH family is that LSH generally requires a long bit length to attain sufficient precision and recall. This drawback may lead to storage overhead and end up limiting the number of applications LSH could be utilized in.

3. Proposed Approach

The proposed method is based on two works in the literature [9, 30]. Both combine Locality Sensitive Hashing (LSH) and Partial Least Squares (PLS) to speed up image retrieval in large galleries. On the other hand, we combine and adapt LSH, PLS and FCN towards the watch-list task so that we can determine whether a subject is registered in a gallery of known individuals.

3.1. Open-set Recognition

Like most supervised learning problems, our approach is based on two canonical steps: training and testing. Figure 2 presents the pipeline of our method. Our approach analyzes feature vectors and their corresponding identities to learn an inferred function for every single hashing model, which are used to generate vote list histograms whenever a query is requested.

3.1.1 Training Stage

We start the training stage by randomly partitioning all subjects registered in the gallery set into two disjoint collections: positive and negative subset. In pursuance of a balanced division, samples are drawn from a binomial distribution (parameter $n = 1$ and parameter $p = 0.5$) in the interest of associating a gallery subject with the positive class when the distribution value gets closer to one.

Just as we split all subjects into the positive or negative collection, we guarantee that each subset contains approximately the same number of individuals. Besides, we make sure that all samples belonging to a certain individual will be in the same class. At that moment we execute

a learning algorithm so that a single model is learnt for both positive and negative subsets in a binary classification fashion.

The generation of binary classifiers for hashing is repeated several times[1] and, therefore, each classification model contains different individuals pertaining to the positive or negative collection. These classifiers are essential because they determine whether a query face sample is a member of the positive or negative class. Feature descriptors are obtained from samples in the positive set with target values equal to $+1$ in contrast to samples in the negative set, which hold target values equal to -1. These features must be extracted and combined with their corresponding target values so that classification models can be successfully learnt.

3.1.2 Testing Stage

At the time that we move to the testing stage, we engage the same feature descriptors employed during the training step to extract features from query face images. We present this feature vector to each one of the classifiers in exchange for the response value r.

Given a probe face image, we start a vote list replete with zero values. Each position in the vote list corresponds to a subject from the set of known subjects that is required for training. Each classification model has a record of which individuals were randomly categorized as pertaining to the positive and negative subset.

As we compare a probe sample to all classification models, we increment each model's response value r on the vote list only for those subjects belonging to the positive subset. In the end, we sort the vote list in decreasing order (see Figure 1) in behalf of keeping individuals with higher probability of matching the probe sample on the top of the vote list ranking. In essence, we simply want to find out how much the top scorer (leading subject of the sorted vote list) stands out from all other individuals.

Unlike Santos et al. [30], which sort the vote list in descending order of incremented scores toward generating a list of candidates for face identification, we arrange the vote list in the interest of computing the ratio of the top scorer to the remaining individuals. We fully detail the ratio estimation in Section 4.3. The proposed method also differs from the original implementation in some aspects: we store both positive and negative collections for every regression model. Moreover, we increment regression values on the vote list even when they are negative in an attempt to intensify the difference among the vote list scores.

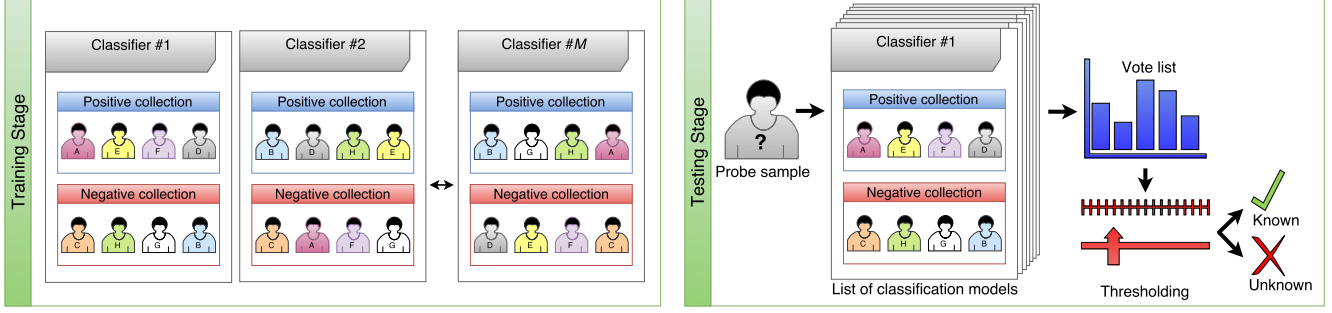[1]The number of hashing models is a parameter defined by the user.

Figure 2: An overview of binary classifiers embedded into hashing functions: *Training:* Feature descriptors are obtained for all subjects' samples before they are partitioned into positive and negative sets. Then, different classification models are learnt containing different subjects in each subset. *Testing:* Same features are extracted from the query image. It is compared to all hashing functions and their response values are used to increment a vote list. If the ratio of the top scorer to the remaining subjects satisfies a threshold, it is considered a known individual.

3.2. Classifiers

In the interest of generating classification models, we explore two approaches: Partial Least Squares (PLS) and Fully Connected Network (FCN).

3.2.1 Partial Least Squares

The purpose of PLS [31] is to create latent variables as a linear combination of the independent zero-mean variables X and Y. More precisely, X represents a matrix of feature descriptors whereas Y describes a vector of response variables. Then, PLS searches for latent vectors so they can be simultaneously decomposed into $X = TP^T + E$ and $Y = UQ^T + F$ in order to identify the maximum covariance between these variables. Matrix $T_{n \times p}$ portrays latent variables from feature vectors and matrix $U_{n \times p}$ denotes latent variables from target values. Variables $P_{p \times d}$ and $Q_{1 \times d}$ can be compared to the loading matrices from principal component analysis. Eventually, variables E and F represent residuals.

We employ the Non-linear Iterative PLS (NIPALS) [32] algorithm to compute the low-dimensional data representation. NIPALS estimates the maximal covariance between latent variables T and U and outputs a matrix of weight vectors $W_{d \times p}$. Then, it estimates the regression coefficients vector β using least squares as follows: $\beta = W(P^T W)^{-1} T^T Y$. The PLS regression response for a query image's feature vector is given by $\hat{y} = \bar{y} + \beta^T (x - \bar{x})$ where $\bar{y}$ is the sample mean of Y and $\bar{x}$ the average values of X.

3.2.2 Fully Connected Network

To boost the recognition results, we also propose a second approach by replacing all PLS models with FCN classifiers. All FCN models target at modeling the relationship of observable variables and determining whether a probe sample's identity was enrolled in the gallery set at training time. It works very much alike the PLS approach: a matrix of weight vectors is calculated considering the results of each FCN model and the regression response for a query image's feature vector is then returned.
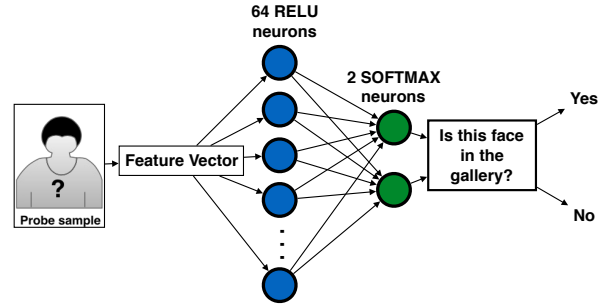

Figure 3: A diagram of the fully connected network model employed.

As depicted in Figure 3, we propose a small network architecture with two layers. Each node is a neuron with a nonlinear activation function that is connected to every neuron in the previous layer. Therefore, each node in one layer connects with a certain weight w to every node in the subsequent layer. This was the chosen architecture because it reported the best results in an exploratory experiment with several other architectures, considering different numbers of neurons and depths.

4. Experimental Results

In this section we evaluate locality-sensitive hashing (LSH) combined with partial least squares (PLS) and fully connected network (FCN)[2].

---

[2]HPLS and HFCN code and experimental data can be downloaded from https://github.com/rafaelvareto/HPLS-HFCN-openset

4.1. Face Datasets

The proposed methods are evaluated on a recent non-constrained dataset and on two well-known datasets. In favor of demonstrating its effectiveness, we select datasets with different characteristics, ranging from frontal cropped images taken under controlled scenarios to images in the wild with lighting and pose variations.

FRGCv1. Face Recognition Grand Challenge v1.0 [10] consists of 152 subjects and six different experiments. We only evaluate the methods on three of them: *experiment one, two* and *four*. *Experiment four* considers a gallery with one controlled still picture for each subject plus a probe set having multiple uncontrolled images. *Experiments one* and *two* only contain controlled images. *Experiments three*, *five*, and *six* do not correspond to 2D face recognition.

PubFig83. PubFig83 [11] is a fragment of the original Public Figures dataset [33]. It comprises several uncontrolled images with pose and expression variations. PubFig83 is composed of 83 individuals with 100 samples each.

VGGFace. VGGFace dataset [12] contains about 2.6 million samples of more than 2600 celebrities and public figures collected from the web. Its initial list of public figures was taken out of the Internet Movie Data Base (IMDB) celebrity list. Due to its massive size and high training time, we arbitrary select a portion of the original VGGFace containing a thousand subjects with 15 samples each[3].

4.2. Feature Descriptors

In this section, we depict the two feature descriptors selected in this work: HOG and VGG. The former was designed for object detection whereas the latter is based on convolutional neural networks for face detection and recognition.

HOG. Histogram of Oriented Gradients (HOG) [34] generates descriptors that comprise shape information in the form of histograms. Before feature extraction, images are rescaled to 128×144 pixels and each sample is decomposed into a set of overlapping blocks which features are extracted from. Each block is 16×16 pixels, with an 8-pixel stride and an 8×8-pixel cell size. After extracting features for all blocks, descriptors are concatenated in a feature vector and that turns into a feature descriptor.

VGG. The VGGFace CNN descriptor is computed using the implementation of Parkhi et al. [12], which is based on the VGG-Very-Deep-16 CNN architecture [35]. We consider the standard training weights utilized by the

---

[3]Subjects are chosen according to alphabetical ordering of all subjects followed by the selection of the first 1000 individuals. Samples are also sorted in ascending order and the first fifteen available images are selected.

authors, therefore, there is not any sort of fine tuning towards the selected datasets.

4.3. Evaluation Protocol

There is not a worldwide consensus when it comes to protocols for open-set face recognition. As a result, most works in the literature proposes distinct protocols for different datasets. With that in mind, we evaluate PubFig83 alone on a protocol exploited by few researchers [11,25,36]. This protocol is applied to experiments depicted in Table 1 and Table 4.

Proposed Protocol. We propose a new protocol for the experiments carried out with FRGCv1 and VGGFace. We partition the entire dataset, varying the known individuals set size in 10%, 50% and 90%. All the remaining individuals become unseen classes during training time. For each subject in the known subset, 50% of the samples are randomly selected for training and the remaining is left for testing. This protocol is applied to experiments exposed on Table 2 and Table 5.

Evaluation Metric. We consider both extensively employed Receiver Operating Characteristic (ROC) curves and its Area Under Curve (AUC) for all datasets. ROC curves usually present true positive rate on the Y axis, and false positive rate on the X axis. It indicates that the plot's top left corner is the optimal point. Good open-set recognition systems would present true positive rates for the ROC curve equal to one. Similarly, AUC ranges from zero to one, being preferable values approaching one.

Threshold Selection. An evaluation of three different thresholds is executed in the interest of finding out the one that better impacts our algorithms. Figure 4 shows the ROC curve for each threshold, which are detailed below:

$$\tau_1 = \frac{\mathcal{H}_{TS_1}}{AVG(\mathcal{H}_{TS_2} + \mathcal{H}_{TS_3})} \quad (1)$$

$$\tau_2 = \frac{\mathcal{H}_{TS_1}}{\mathcal{H}_{TS_2}} \quad (2)$$

$$\tau_3 = \frac{\mathcal{H}_{TS_1}}{AVG(\mathcal{H}_{TS_2} + ... + \mathcal{H}_{TS_p})}, p = \lceil 0.15 \times |\mathcal{H}| \rceil \quad (3)$$

Basically, they are based on the ratio of the top scorer TS_1 to the succeeding subjects. The chart indicates that the ratio among only the leading three top scores does not drastically interfere the area under the curve.

4.4. Recognition Evaluation

We evaluate the approaches described in Section 3 for VGGFace, PubFig83 and experiment number four of FRGCv1 dataset. From now on we refer to the combination of locality-sensitive hashing and partial least squares as
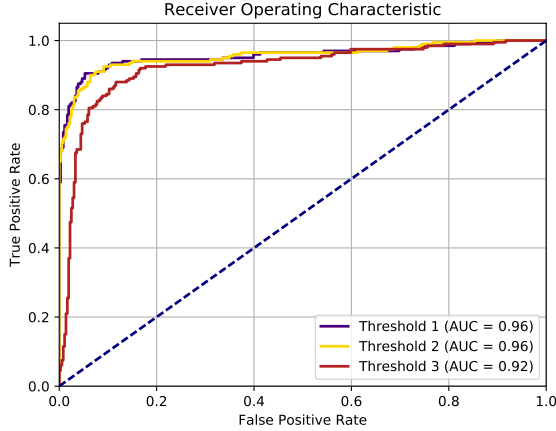
Figure 4: Average ROC curves for the FRGCv1 dataset on experiment one. We repeated this experiment five times, fixing variable p to 15% of all subjects in the gallery set.

HPLS. Equivalently, HFCN turns into the association of hashing methods with fully connected network.

4.4.1 Baseline: One-Class SVM

One-class SVM [16] generates a spherical boundary around the data in the feature space. The idea is to add most data into the hyper-sphere so that it becomes an optimization problem. There are two key parameters we should concern when using one-class SVM: $gamma$ and nu values.

In our experiments, we execute the one-class WSVM algorithm proposed by Scheirer et al. [17], which is publicly available in the form of a library called libSVM. The best results were obtained with $gamma$'s default value. With respect to nu, we perform a grid search in pursuance of the value that provides the best AUC for each experiment's protocol.

4.4.2 Descriptor Selection

There are two feature descriptors used in this work: HOG and VGGFace CNN descriptor. Table 1 presents a comparison between both descriptors on PubFig83 dataset. We have chosen this dataset since there is no room for data influence, which would not be possible with FRGCv1 or VGGFace, resulting then in a proper contrast. The considered approaches are HPLS and HFCN which stands for the proposed approaches alternating the classifier in PLS and FCN.

As we can see in Table 1, both approaches using VGGFace CNN descriptor notably outperform HOG-based algorithms. While HOG only holds shape information, VGGFace CNN descriptor comprises much more information related to faces since its network was previously trained on a face dataset.

Approach	AUC	STD	#Execs
HPLS-HOG	0.658	0.014	5
HPLS-VGG	0.954	0.008	5
HFCN-HOG	0.640	0.021	5
HFCN-VGG	0.974	0.005	5

Table 1: Comparison between HOG and VGG descriptors on PubFig83 with HPLS and HFCN algorithms.

4.4.3 Literature Comparison

Table 2 shows experiments on the FRGCv1 dataset. In their analysis, Santos et al. [25] combine four descriptors: HOG, LBP, mean color and Gabor filters. We attain very good results using either HOG or VGG. We presented the proposed approach of this paper in the form of HFCN and HPLS for fully connected network and partial least squares, respectively. WSVM symbolizes the one-class SVM of Scheirer et al. [17]. We also assess the performance of each feature descriptor independently (HOG and VGG). We fix both the number of hashing models to 100 and the quantity of individuals in the known set to 50%, in accordance to the protocol specified in Section 4.3.

Approach	AUC	STD	#Execs
Least Squares [25]	0.869	-	-
SVM-Single [25]	0.853	-	-
Chebyshev [25]	0.838	-	-
WSVM-VGG [17]	0.862	0.014	10
WSVM-HOG [17]	0.515	0.027	10
HPLS-HOG	0.910	0.022	10
HFCN-VGG	0.877	0.021	10
HPLS-VGG	0.850	0.008	10
HFCN-HOG	0.613	0.105	10

Table 2: Average AUC, standard deviation (STD) and number of executions (#Execs) for the experiment four of FRGCv1 dataset. We employ the proposed protocol with 100 hashing models, selecting 50% of the subjects to compose the gallery set.

There are blank cells in the first three rows of Table 2 because we did not reproduce those experiments. HOG is a low-level feature descriptor; however, it performed well with partial least squares. We believe it can be explained by HOG's structure and FRGCv1's predominant characteristics since it encompasses high-resolution images acquired under partial controlled conditions and no pose variation. VGGFace was learnt considering more than two thousand unique individuals with all sorts of pose variations and expression changes. Therefrom, HOG outperforming VGG seems plausible.

4.4.4 Single Classifier Evaluation

For the purpose of analyzing each classifier's behavior individually, experiments considering PLS and FCN are performed in the FRGCv1 experiment four, described in Section 4.1. Table 3 presents the hit rate for 100 executions.

	PLS (%)	FCN (%)
AVG	73.559	77.026
STD	1.943	1.648
MIN	70.614	73.245
MAX	77.631	80.592

Table 3: Evaluation of PLS and FCN classifiers individually. The presented results are: average (AVG), standard deviation (STD), minimum (MIN) and maximum (MAX). They are computed on 100-execution hit rates.

Results show that a FCN classifier alone provides better results than a PLS model. While a PLS model achieves a hit rate of approximately 73.56%, the FCN classifier attains 77.03%. Both methods hold tight standard deviation values. The variation between their minimum and maximum values remained close to the average values.

4.4.5 Additional Evaluation

To check how the method responds to some parameter adjustments, we analyze the behavior of the approaches by varying the number of hashing models for the PubFig83 dataset and alternating the size of the subset of known individuals for both VGGFace and FRGCv1. Table 4 and Table 5 expose how these parameters affect the proposed methods.

#Models		100	300	500
HPLS-VGG	AUC	0.946	0.954	0.972
	STD	0.009	0.015	0.004
HFCN-VGG	AUC	0.970	0.972	0.980
	STD	0.000	0.004	0.007

Table 4: Variable number of hashing models: AUC and STD for PubFig83, considering 75/83 randomly chosen subjects in the known subset. Eight individuals left compose the unknown subset.

Table 4 shows no clear accuracy improvement. The little increase in AUC for PubFig83 with increasingly hashing models may be justified with the fact that algorithms trained with multiple-sample gallery sets – for this experiment, PubFig83 holds 90 samples per class and only 83 classes – are inclined to remain stable regardless of the number of hashing functions. If we reduce the number of samples per subject available at training time and increase the number of subjects enrolled in the gallery set, chances are more hashing models will be required in order to keep AUC high.

Known individuals			10%	50%	90%
FRGCv1 4	HFCN	AUC	0.872	0.877	0.856
		STD	0.015	0.022	0.014
	HPLS	AUC	0.794	0.850	0.856
		STD	0.078	0.009	0.022
	WSVM [17]	AUC	0.866	0.862	0.848
		STD	0.035	0.015	0.019
VGGFace	HFCN	AUC	0.996	0.974	0.964
		STD	0.011	0.008	0.005
	HPLS	AUC	0.972	0.962	0.924
		STD	0.004	0.004	0.005
	WSVM [17]	AUC	0.841	0.839	0.835
		STD	0.013	0.007	0.007

Table 5: Variable known individuals: AUC and STD for FRGCv1 experiment four and VGGFace dataset. We secure 100 hashing models for WSVM, HPLS and HFCN with VGG descriptor as we execute each algorithm 10 times.

In general, the accuracy of a recognition system tends to reduce as we have more individuals enrolled in the gallery set. Surprisingly, in Table 5, our methods efficiency did not deteriorate with the enrollment of new subjects in the FRGCv1 dataset since having more samples increase the discriminability of classifiers when there are only few samples per subject. We believe that the stable behavior observed on the proposed approaches lies on their capability of remaining robust despite of parameter adaptation and dataset selection.

5. Conclusions

The proposed methods seemed promising in solving a task not frequently considered in the literature, namely, open-set face recognition. Two basic approaches were introduced: HFCN and HPLS. Experiments have shown that VGGFace CNN descriptor contains more valuable information than HOG in the unrestrained open-set face recognition task. In addition, a comparison with the literature shows high accuracy with HPLS-HOG and HFCN-VGG protocols on FRGCv1 experiment four. For future works, we intend to apply the designed approaches to huge galleries [37], aiming at performing a pre-filtering of the individuals to be compared with the gallery samples.

Acknowledgements

References

[1] Rama Chellappa, Pawan Sinha, and P Jonathon Phillips. Face recognition by computers and humans. *Computer*, 2010.

[2] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *PAMI*, 2006.

[3] Brendan F Klare and Anil K Jain. Heterogeneous face recognition using kernel prototype similarities. *PAMI*, 2013.

[4] Jiwen Lu, Yap-Peng Tan, and Gang Wang. Discriminative multimanifold analysis for face recognition from a single training sample per person. *PAMI*, 2013.

[5] Dong Yi, Zhen Lei, and Stan Z. Li. Towards pose robust face recognition. In *CVPR*. IEEE, 2013.

[6] Yinjie Lei, Mohammed Bennamoun, Munawar Hayat, and Yulan Guo. An efficient 3d face recognition approach using local geometrical signatures. *PR*, 2014.

[7] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[8] Fayin Li and Harry Wechsler. Open-set face recognition using transduction. *PAMI*, 2005.

[9] Cassio dos Santos, Ewa Kijak, Guillaume Gravier, and William Robson Schwartz. Partial Least Squares for Face Hashing. *Neurocomputing*, 2016.

[10] P Jonathon Phillips, Patrick J Flynn, Todd Scruggs, Kevin W Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. Overview of the face recognition grand challenge. In *CVPR*. IEEE, 2005.

[11] Nicolas Pinto, Zak Stone, Todd Zickler, and David Cox. Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook. In *CVPRW*. IEEE, 2011.

[12] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*. IEEE, 2015.

[13] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Image Processing*, 2010.

[14] Xiangyu Zhu, Junjie Yan, Dong Yi, Zhen Lei, and Stan Z Li. Discriminative 3d morphable model fitting. In *AFGR*. IEEE, 2015.

[15] Rafael Vareto, Filipe Costa, and William Robson Schwartz. Face identification in large galleries. In *SIBGRAPI*. IEEE, 2016.

[16] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *NC*, 2001.

[17] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *PAMI*, 2014.

[18] Vasileios Mygdalis, Iosifidis Alexandros, Anastasios Tefas, and Ioannis Pitas. Large-scale classification by an approximate least squares one-class support vector machine ensemble. In *Trustcom/BigDataSE/ISPA*. IEEE, 2015.

[19] Filipe de O Costa, Ewerton Silva, Michael Eckmann, Walter J Scheirer, and Anderson Rocha. Open set source camera attribution and device linking. *PRL*, 2014.

[20] Hakan Cevikalp and Bill Triggs. Efficient object detection using cascades of nearest convex model classifiers. In *CVPR*. IEEE, 2012.

[21] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *PAMI*, 2013.

[22] Behrooz Kamgar-Parsi, Wallace Lawson, and Behzad Kamgar-Parsi. Toward development of a face recognition system for watchlist surveillance. *PAMI*, 2011.

[23] Shengcai Liao, Zhen Lei, Dong Yi, and Stan Z Li. A benchmark study of large-scale unconstrained face recognition. In *IJCB*. IEEE, 2014.

[24] Erik Learned-Miller, Gary B Huang, Aruni RoyChowdhury, Haoxiang Li, and Gang Hua. Labeled faces in the wild: A survey. In *AFDFIA*. Springer, 2016.

[25] Cassio dos Santos Junior and William Robson Schwartz. Extending face identification to open-set face recognition. In *SIBGRAPI*. IEEE, 2014.

[26] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*. IEEE, 2015.

[27] He Zhang and Vishal Patel. Sparse representation-based open set recognition. *PAMI*, 2016.

[28] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *CG*. ACM, 2004.

[29] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing. *PAMI*, 2012.

[30] Cassio dos Santos, Ewa Kijak, Guillaume Gravier, and William Robson Schwartz. Learning to hash faces using large feature vectors. In *CBMI*. IEEE, 2015.

[31] Herman Wold. Partial least squares. *ESS*, 1985.

[32] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *SLSFS*. Springer, 2005.

[33] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *Computer Vision*. IEEE, 2009.

[34] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*. IEEE, 2005.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[36] Gerson Carlos, Helio Pedrini, and William Robson Schwartz. Fast and scalable enrollment for face identification based on partial least squares. In *AFGR*. IEEE, 2013.

[37] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*. IEEE, 2016.