

# ONLINE FACE RECOGNITION SYSTEM FOR VIDEOS BASED ON MODIFIED PROBABILISTIC NEURAL NETWORKS

Jun Fan, Nevenka Dimitrova and Vasanth Philomin

Philips Research USA  
Briarcliff Manor, NY 10510  
[junfan@ee.columbia.edu](mailto:junfan@ee.columbia.edu),  
[nevenka.dimitrova@philips.com](mailto:nevenka.dimitrova@philips.com)

Philips Research Aachen,  
Germany  
[vasanth.philomin@philips.com](mailto:vasanth.philomin@philips.com)

## ABSTRACT

Video retrieval in consumer applications demands high level semantic descriptors such as people's identity. The problem is that in a variety of videos such as *home videos*, *Hollywood content*, *TV broadcast content*, *mobile phone videos* faces are not easy to recognize. Even more, a closed system trained to recognize only a predetermined number of faces will become obsolete very easily. We developed an online-learning face recognition system for a variety of videos based on Modified Probabilistic Neural Networks (MPNN). This face recognition system can detect and recognize known faces, as well as automatically detect unknown faces and train the unknown faces online into new face classifiers such that this "unknown face" can be recognized if it appears again. MPNN is a variant of the PNN with thresholding on the category (output) layer of a Probabilistic Neural Network (PNN) in order to detect unknown categories of input data. The PNN training makes the online training very fast because adding new faces does not require retraining of the known categories. Our experimental results show that on-line learning gives somewhat lower hit rate, while at the same time reducing the false positive rate.

## 1. INTRODUCTION

Most face recognition systems are trained on a fixed number of faces that are known in advance [1][2][3]. These systems will only recognize the faces with known models and the face database cannot be updated during the classification procedure. For example, they will work for surveillance systems, which have to recognize all employees of a company and alert to any intruders, or an airport surveillance system that is trained to recognize known terrorists. However, in the area of home video, TV broadcast video, wearable video, in addition to the known people there is a need to recognize the new people appearing with each new video. In home videos for example, if a system is trained to recognize only family members then a visitor is labeled as "other" or "unknown". Of course there are travel videos with many new faces that are transient. A system that categorizes images and videos based on people presence has to distinguish all these categories of important and

unimportant faces. Moreover, the system has to be flexible enough to incorporate and retain important unknown faces.

Our approach can automatically detect the new faces and extend the database based on the new faces. Our online learning system can learn features of new, reoccurring faces and store corresponding models of new faces for future use. This method is similar to our human perception of known faces and incorporation of new faces. Also, our approach can generate a confidence measurement for each recognized face in the database and sort the candidate by the confidence measurement, which make post-processing easier.

## 2. SYSTEM ARCHITECTURE

Figure 1 shows our face recognition system architecture. We start with an initial database which has a limited number of faces. The system has a training phase and classification phase just like any other face recognition system (depicted with a dashed line). However, the novel aspect here is that there is a feedback arrow to the training face for unknown faces. These persistent (reoccurring) faces become new sample faces for the online training (dotted ellipse in Figure 1). We assume these persistent faces are important faces.

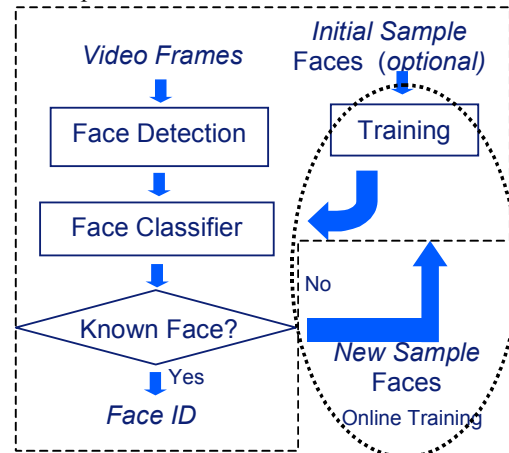


Figure 1. Face Recognition System Architecture

During the training phase, the system reads face examples for each face (actor/character) and trains the Probabilistic Neural Networks (PNN) [4][6] based on the features of these faces. We choose Vector Quantization

Histogram features as face features [1]. During the classification phase, the system will decode the MPEG video file into video frames first. For each frame, we use a variant of the face detector described in [8]. If there is a face found by the face detector, the face segment is forwarded to the PNN based Face Classifier. A confidence measurement for each face ID is generated by PNN. Based on a thresholding of the confidence values and a set of conditions, the system determines if the face is known or unknown. Persistent unknown faces are evaluated and forwarded to online learning phase. After we have the confidence measurement for each Face ID, we can easily choose the Face ID with the maximum confidence measurement as the output from Face Classifier by using a Winner takes all principle.

### 3. FACE DETECTION

This section briefly describes the face detection algorithm used in our framework. In [8], Viola and Jones applied the popular AdaBoost [9] learning technique to the problem of rapid object detection. They used “attentional cascade” - a combination of complex classifiers that includes strong classifiers consisting of a set of computationally efficient binary features (also called weak classifiers). Each round  $t$  of boosting added a single feature  $h_t$  to the current set of features by minimizing:

$$Z_t = \sum_i D_t(i) \exp(-\alpha_t y_i h_t(x_i)) \quad (1)$$

where  $D_t(i)$  is the weight on example  $x_i$  at round  $t$ ,  $y_i \in [-1, 1]$  is the target label of the example,  $\alpha_t$  is the influence of this weak hypothesis on the strong classifier and  $h_t(i)$  is the weak binary hypothesis restricted to  $[-1, 1]$ . In our variant, we use boosting stumps (decision trees that partition the domain into two pieces and yield a prediction for each partition) as the weak classifiers, which results in  $\alpha_t$  being folded into  $h_t$ , thereby allowing the weak hypotheses to have a range over all  $\mathcal{X}$  rather than the restricted range  $[-1, +1]$ . The prediction values for the left and right partitions that minimize  $Z_t$  above are:

$$c_{left} = \frac{1}{2} \ln\left(\frac{W_+^{left} + \varepsilon}{W_-^{left} + \varepsilon}\right); c_{right} = \frac{1}{2} \ln\left(\frac{W_+^{right} + \varepsilon}{W_-^{right} + \varepsilon}\right) \quad (2)$$

where the  $W$ 's denote the weight of the examples that are assigned to the left or right partition with true labels “positive” or “negative”. The predictions are also smoothed with the term  $\varepsilon$  to avoid numerical problems caused by large predictions. From these prediction values, we can greedily choose the splitting criterion for the decision tree (dropping the subscript  $t$ ) as

$$Z = 2(\sqrt{W_+^{left} W_-^{left}} + \sqrt{W_+^{right} W_-^{right}}) \quad (3)$$

rather than the Gini index or an entropic function [9].

Most variants of the algorithm described in [8] reduce the training set error during training and count on the

generalization performance of AdaBoost that is rigorously proved in [9]. It is our experience that using a validation set during training as in [8] yields the most effective cascades with fewer features. This is due to the fact that we get multiple hits around each face while scanning the validation set and we can pick the strong classifier threshold as high as possible in order to retain just one hit, thereby eliminating more false alarms in the process.

### 4. ONLINE FACE RECOGNITION

This section describes the online face recognition algorithm used in our framework. Firstly, we introduce a face classifier based on Modified Probabilistic Neural Networks (MPNN), which derive from PNN [6] by adding threshold in PNN category layer [5]. We have two reasons for choosing a MPNN as our face classifier: 1) we can measure the output confidence based on the preset threshold, and 2) Training of the PNN is accomplished by setting each input vector equal to the weight vector in one of the hidden units, and then connecting the pattern unit's output to the appropriate category unit [6]. Therefore, training of new pattern into a trained MPNN requires no retraining of the existing links in PNN. After presenting the MPNN, we introduce the conditions we used for unknown face detection and the online learning algorithm for new faces in the later sub-sections.

#### 4.1. Modified Probabilistic Neural Networks

The MPNN is based on Probabilistic Neural Networks [6][7]. The PNN is one of the implementations on Bayes Strategy, which seeks the minimum risk cost based on the Probability Density Function (PDF). The Bayes Decision rule used in PNN is shown below:

$$d(X) = \omega_i \text{ if } f_i(X) > f_j(X) \quad \forall i \neq j \quad (4)$$

where  $\omega_k \quad k=1 \dots n$  is the output category,  $n$  is the number of categories,  $f_i(X)$  is the PDF of input vector  $X$  for category  $i$ ,  $d(X)$  is the decision function for vector  $X$ .

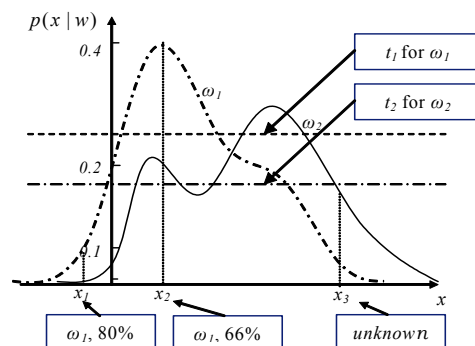


Figure 2. Threshold for PDFs in MPNN

The PNN can generate a confidence measurement by comparing the relative value from the PDF of the trained examples [5]. However, this can make the PNN generate a high confidence output for one category even though the output from PDF for this category is very low. For

example, in Figure 2, a trained two category PNN with one-dimension input vector is shown. The PDFs are generated by the trained examples in the PNN. Without adding a threshold, the input vectors  $x_1$  and  $x_2$  are classified as  $\omega_1$  with 80% and 66% confidence respectively, which is calculated by  $C_i = f_i(x_i) / \sum f(x)$  where  $C_i$  is the confidence for vector  $x_i$ . However, from Figure 2, the maximum PDF value of the category  $\omega_1$  is 0.8 and the PDF value for input vector  $x_1$  is 0.1, which makes the PDF value for the input vector  $x_1$  low enough to be identified as unknown category although the confidence of input vector  $x_1$  is 80% high.

By adding a threshold of 70% of the maximum PDF value for each category in the category layer of PNN (the  $t_1$  and  $t_2$  in Figure 2), the MPNN can deal with low PDF output vectors, which enables identification of unknown categories [5]. For example, in Figure 2, the input vector  $x_3$  has low PDF values, which is below threshold for category  $\omega_2$ . It is then classified as “Unknown” by MPNN. This modification makes the Bayes Decision rule updated as below:

$$d(x) = \omega_i \quad \text{if} \quad f_i(x) > f_j(x) \geq t \quad \forall i \neq j$$

$$d(x) = \text{unknown} \quad \text{if} \quad f_i(x) > f_j(x) < t \quad \forall i \neq j \quad (5)$$

where  $t$  is the threshold in category layer.

#### 4.2. Unknown Face Detection in Videos

The solution for detecting the unknown faces takes advantage of the use of MPNN and the temporal nature of videos: unknown faces are detected when an identified face is not classified to any of the known category by the MPNN for some period in time. We design several conditions to detect unknown faces, which utilize the advantages of MPNN and videos:

1. MPNN face classifier identifies the face as unknown
2. Mean of the PDF output is low for the face sequence
3. Variance of the input vectors is small to make sure the input vectors are for the same face
4. If all the above three conditions last for  $n$  (e.g.  $n=10$ ) seconds, we conclude that an unknown face has appeared in the video. We save this face sequence into buffer.

This is a simple use of “memory”. However if a face appears many times in a video for very short periods (high-cut rate in a conversation for instance) then we need to employ a more sophisticated, accumulative memory where a face of a stranger is learned over time (e.g. repeating faces in home video that appear at different social gatherings.)

#### 4.3. Online Face Learning

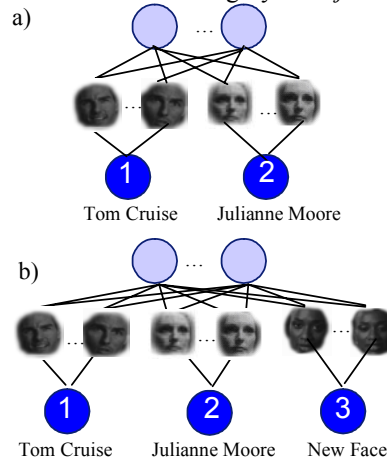
Once the algorithm detects an unknown face, the online learning of the unknown face is performed using PNN learning algorithm. The advantage of using PNNs is that we do not need to update all the existing weights during training [6]. This allows online learning without too many calculations during the updating of weights.

As we described in previous subsection, we store face input vectors in the buffer and we evaluate the variance and mean of these input vectors. In the buffer, the lower variance input vectors contain more precise information of the unknown face. We choose 10 input vectors  $X_i$  in the buffer, which contain the low variance from all the inputs (i.e. the closest to the average in the buffer). Then, the PNN learning algorithm is performed for the new input vectors in the buffer. The procedure of the online training is similar to off-line training: normalize the input vector  $X_i$  with formula (5). For every  $X_i$  add a new node into the hidden layer and initialize the weights  $W_i$  of the node to the normalized input vector  $X'_i$ . Then, add a new category  $\omega_{\text{new}}$  in the category layer and link the added hidden nodes to the new category  $\omega_{\text{new}}$ .

The algorithm for online learning is shown below:

1. for  $X_i, i = 1, 2, \dots, 10$
2. normalization:
$$x'_{ik} = x_{ik} \cdot (1 / \sqrt{\sum X_i^2}) \quad k = 1, 2, \dots, m$$
3. assign weights:  $W_i = X'_i$
4.  $\omega_{ji} = 1$
5. end

where  $X_i$  is the input vector and  $m$  is the number of input dimensions,  $W_i$  is the weight vector between input nodes and the new hidden node  $i$ , and  $\omega_{ji}$  is the link between hidden node  $i$  and the new category node  $j$ .



**Figure 3. Adding a new face using PNN Online Learning/Extension**

Figure 3a shows a trained MPNN with 2 faces in the database. In this diagram, each hidden node is represented by a face because the nodes save the information of the face during training. Figure 3b shows a PNN after the online learning for a detected unknown face. In this diagram, the nodes in hidden layer increased and the information for unknown faces is added into the hidden layer and category layer of the PNN. Therefore, when the “new face” appears again, the face classifier can recognize this face as a known face.

## 5. EXPERIMENTAL RESULTS

We tested 4 genres of videos: Movies, News Video, Video Conference and Home Video on both offline mode and online detecting/training mode. For the offline mode, we train the face classifier with all the faces detectable by our face detection in the video, and 5 training samples for each face. For the online mode, we initialize the face classifier with 4 actors/actresses and set threshold to 70% of the maximum of PDFs. The experimental results are shown in Table 1.

Video Category	Min	# of Faces	Offline		Detect -ed	Online	
			Hit %	FP %		Hit %	FP %
Movies	303	39	82 %	32 %	30	77 %	21 %
News	22	24	93 %	27 %	9	81 %	18 %
Conference	45	6	91 %	10 %	6	90 %	6 %
Home Video	28	6	74 %	29 %	2	52 %	21 %

**Table 1. Experimental results for different genre**

In Table 1, the second column labeled as “Min” refers to the length of the video in minutes; the third column labeled as “# of Faces” refers to the number of detectable faces by our face detection. “Hit” means the hit ratio and “FP” means the false positive ratio. The “Offline” and “Online” label distinguish the result of Offline mode and Online Mode. The “Detected” label means the number of faces that have been detected and online trained in PNN by our online face recognition system.

We should note here that by design we chose to detect unknown faces that are persistent for more than 10 seconds. From Table 1, we see that the algorithm can detect most unknown faces in Movies and Video Conference, however, for the News video, the algorithm is instructed to only learn online 9 faces out of 24 faces. This is because most of the new faces in News video appear for a short time, and the algorithm ignores them as not persistent and therefore not important.

Movie	Actor	Offline		Online	
		Hit	FP	Hit	FP
Minority Report	T. Cruise	91%	15%	76%	11%
	C. Farrell	87%	17%	80%	16%
	L. Smith	73%	19%	72%	21%
	S. Morton	91%	29%	82%	23%
Magnolia	T. Cruise	86%	28%	81%	11%
	J. Moore	91%	24%	72%	22%
	P. B. Hall	67%	19%	65%	16%
	J. Blackman	81%	29%	69%	23%

**Table 2. Experiment results for particular actors**

For the movies Minority Report and Magnolia, we compare the performance between offline and online learning for the same actor/actress. Table 2 shows the experimental results for particular actors/actresses. We

note that while the hit rate for offline is higher, result of online learning yields lower false positive rate.

## 6. CONCLUSIONS

Open systems for detection and recognition of high level semantic descriptors are going to be increasingly valuable in the consumer’s world where the amount of multimedia content is exploding. Once in operation the system should be able to learn and adapt just like babies learn with time to recognize the faces of their parents, close relatives, friends and keep expanding. In this paper we introduced an online face recognition system that uses a variant of the PNN. The main goal is to recognize known faces, detect unknown faces and apply automatic online learning for unknown faces in video. After the online learning, our Classifier is able to recognize the new (unknown) faces presented before. After the recognition, the Classifier will assign recognized face IDs to the faces. In our experiments on a wide variety of videos (total of 400 minutes) we found that while the hit rate for offline learning is higher, the result of online learning gives lower false positive rate. The real added benefits are: 1) we can build open systems, and 2) With PNN there is no retraining required for known faces.

In the future we would like to explore this concept further to include intermittently persistent faces. There are different forms of memory that can enrich the system to attain more human-like recognition capabilities and consequently provide tools that are more targeted to user actual needs.

## 7. REFERENCES

- [1] Koji Kotani, Chen Qiu, and Tadahiro Ohmi, “Face Recognition Using Vector Quantization Histogram Method”, *IEEE ICIP, Sept. 2002 PP: 105-108*
- [2] M. Turk and A. Pentland, “Eigenfaces for Recognition”, *J. Cognitive Neuroscience, Vol. 3, pp. 71-86, 1991*
- [3] W. Y. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips. “Face recognition: A literature survey” UMD CAR Technical Report CAR-TR-948, 2000
- [4] Patra P.K, Nayak M., Nayak S.K., Gobbak N.K., “Probabilistic neural network for pattern classification”, *IEEE 2002 IJCNN, May 2002, PP: 1200 –1205.*
- [5] Washburne, T.P.; Specht, D.F.; Drake, R.M., “Identification of unknown categories with probabilistic neural networks”, *IEEE 1993, April 1993, PP: 434 -437.*
- [6] Specht, D.F., “Probabilistic neural networks for classification, mapping, or associative memory”, *IEEE 1988 ICNN, July 1988, PP: 525-532.*
- [7] R. O. Duda, P. E. Hart, D. G. Stork, “Pattern Recognition, Second Edition”. *Wiley & Sons, NY, 2001, PP: 172-174.*
- [8] P. Viola and M. Jones, “Rapid Object detection using a boosted cascade of simple features”, *IEEE CVPR, HI, 2001*
- [9] R. Schapire and Y. Freund, “Improved boosting algorithms using confidence-rated predictions”, *Machine Learning, 37(3), pp. 297-336, 1999*