15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

An Incremental Face Recognition System Based on Deep Learning

Lufan Li, Zhang Jun, Jiawei Fei, Shuohao Li
National University of Defense Technology
Changsha, China
lilufan0315@gmail.com, {zhangjun1975, feijiawei11, lishuohao}@nudt.edu.cn

Abstract

In recent years, face recognition technologies develop rapidly especially in those systems based on deep learning. However, the models of the general face recognition system are fixed in use after trained, which is difficult to adapt to the new data collected during use. In this paper, we design a face recognition system based on OpenFace. Different from other systems, we propose an intelligent model training method S-DDL(self detection, decision and learning) using incremental SVM algorithm which makes our system able to update the classification model in real time during execution. With this method, the accuarcy of our system will increase on specific human groups and the time consumption of training new models can be limited with the incremental SVM algorithm. The results show that our face recognition system has a good real time performance and accuracy . The S-DDL method can obviously improve the accuracy within little time during the execution of system and the incremental SVM algorithm has a better performance than the traditional one.

1 Introduction

In this paper, we build a face recognition system based on a deep learning algorithm which is implemented by OpenFace [1] using FaceNet network architectures [2] and improve the classification with an incremental SVM to achieve self-learning to boost the efficiency and effectiveness of the system. And we design and implement the algorithm which can update and optimize through the growing data amount. This machanism can make our system get more intelligent with the application time getting longer. The system consists of four modules, including subsystems of image capturing, face recogniton, data management and interaction module.. The implementation of neural network is based on Torch7 [3]. The framework is shown in Figure 1.

Similarly with other methods with deep learning, our system uses a Convolutional Neural Network(CNN) to learn the feature representation directly from the pixels of the images. In this system, we apply an architecture based on the Inception model of Szegedy et al. [4], the winning approach for ImageNet 2014. The architecture use mixed layers in parallel and concatenate the responses so that it can reduce the number of parameters by many times and the number of FPS required. As to loss function, we employ triplet loss [5]to obtain the more precise features. In the classification step, we use the features obtained in the CNN to train a SVM.

There is a wide range of methods using deep learning and getting pretty good results. Zhenyao et al. [6] employ a deep network to align faces into a canonical frontal view and then learn a CNN that classifies each face as belonging to a known identity. The output of network uses the PCA and SVMs in the face verification section. Taigman et al. [7] propose a multi-stage approach with a multi-class network is trained for the face recognition task on over four thousand identities. Their best performance on LFW is 97.35% and predicts the distance combining the networks and a non-linear SVM. Sun et al. [8] propose a compact and therefore relatively cheap to compute network. Their final performance on LFW [9] is 99.47%. The networks are trained by using a combination of classification and verification loss which is similar to the triplet loss we employ in the our system. Boukharouba et al. [10] propose an online multi-category SVM. In this work, the incremental SVM can achieve an efficient update of the decision function after the incorporation of the newly added data and the old data.

In this paper, we make some contributions. First, we design and build the face recogniton system and achieve good results. Second, we present the architecture of the S-DDL algorithm. The third is we introduce the algorithm in the classification and improve the effectiveness.

The rest of this paper is organized as follows. In the Sec.2, we introduce the architecture and work procedure of our system. In the Sec.3, we describe the S-DDL algorithm. In the Sec.4, The results of our slgorithm and advantages are illustrated to show the effectiveness. In the Sec.5, we provide a conclusion.

2 Face Recognition System Architecture

In this section, we will introduce the architecture and implementation of the system. Figure 1. shows the general architecture of the system. Different from other face recognition systems, we modified the face recognition to be self-learning. Our system will automatically train new models based on new data and current models during execution of the system. The accuracy will increase in practical application.

Focusing on the process of face recognition task, it contains four main steps: face detection, face alignment, face feature extraction and face classification. First we find the largest face in every image and return the bounding box around the face. Then we use the landmark with 68 keypoints for each face to make the nose and eyes close to the mean locations to get it aligned. For each image, we obtain the largest face and align it with the size of 96*96 to make our model more efficiently in training process. Figure 2. shows the process of face alignment.

In this system, we use OpenFace based on FaceNet

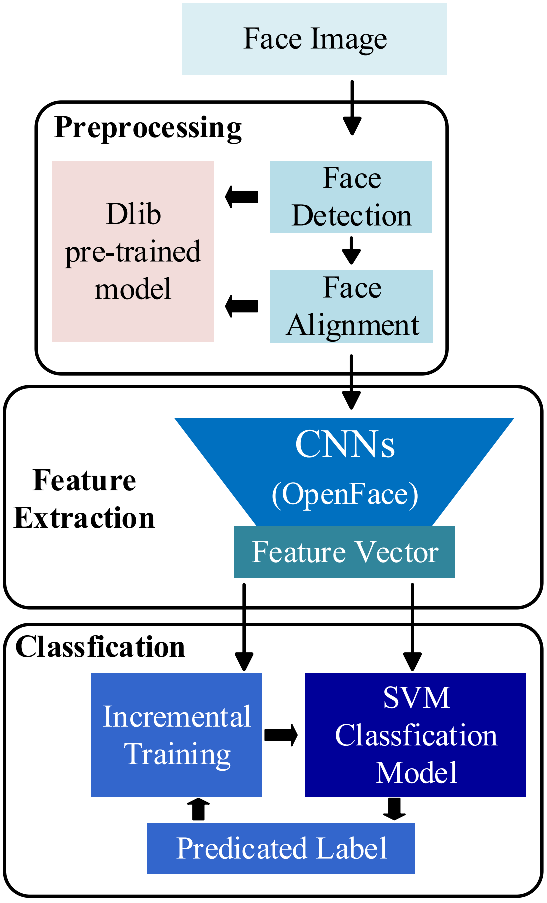

Figure 1. The architecture of our face recognition system.

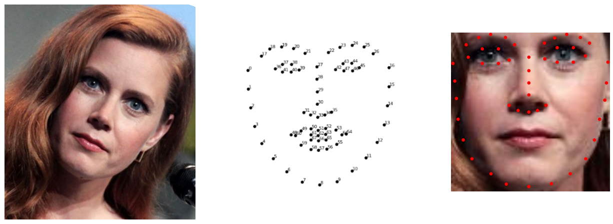

Figure 2. OpenFaces alignment transformation which is based on the large blue landmarks and the final image is cropped to the boundaries and resized to 96*96 pixels

to extract features. OpenFace is an open resource implementation of FaceNet and uses the triplet loss to provide a enbedding on the unit hypersphere and euclidean distance to represent similarity. Different with other networks, we use an incremental classifier to replace the softmax layer. In fact, we can generate a SVM classifier with several hundreds of images.

2.1 S-DDL Algorithm with Incremental SVM

In the traditional face recognition systems, the models are trained before use. However, these methods neglect the value of data collected in the system operating phase. If these images are fully utilized, the system will be more intelligent. Focusing on this demand, we modify the classification algorithm with an incremental SVM and implement this algorithm as S-DDL (self-detection, decision, and learning algorithm). We apply this incremental algorithm to modify the classification models in order to take full advantage of resources generated during the use of system.

As shown in Figure 3, for new images captured from web cameras, our system is able to recognize them using current classification model of the system. After these images are saved in train set, the newly added images will be labeled. When the system detects that the amount of newly added images reaches a certain amount, a new SVM classification model will be trained based on the former model and new data with incremental SVM training algorithm. And then the classification model will be updated immediately and the accuracy will be calculated. In this way, our algorithm can realize self-detection, self-decision and self-learning during the execution of system.

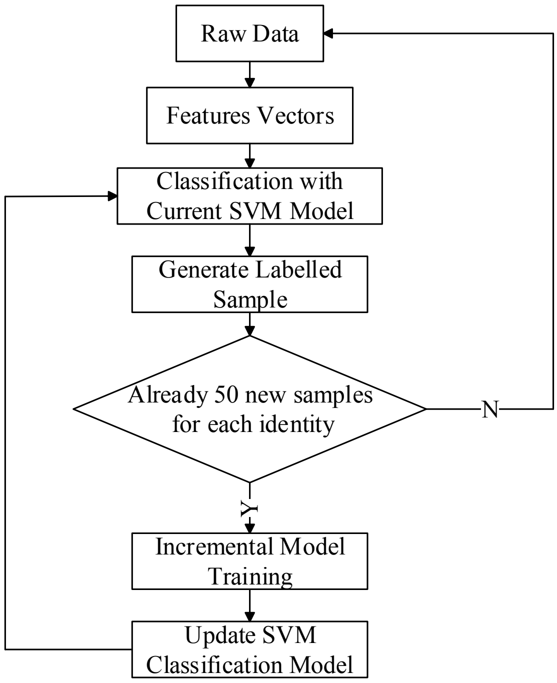

Figure 3. The framework of S-DDL algorithm based on an incremental SVM.

The main idea of incremental SVM we apply is to train a SVM with a crucial partition of the dataset, reserve the support vectors at each training step and create the training set for the next step with these vectors. Syed et al.[11] showed that the decision function of an SVM depends only on its support vectors. The key of incremental algorithm is to preserve the KKT conditions on all existing training data while adding a new vector. And our aim is to use support vector data and new added data to update the current model in the incremental training.

The algorithm is described as follows. Consider a dataset D, the KKT conditions on the point $x_m \in C_{ij}$ divide data D into three categories according to the value of g_m^{ij} for all $i = 1, ..., K, j - i + 1, \ldots, K$, and

$C \geq 0$ controls the number of outliers and corresponds to assign a higher penalty to errors when C is larger:

$$g_m^{ij} = \frac{\partial W}{\partial \alpha_m^{ij}} \begin{cases} > 0; if \alpha_m^{ij} = 0; D(dv_m^{ij}) \\ = 0; if 0 < \alpha_m^{ij} < C; S(sv_m^{ij}) \\ < 0; if \alpha_m^{ij} = C; E(ev_m^{ij}) \end{cases} \tag{1}$$

Where support vectors S are on the boundary, error vectors E exceed the margin and data vectors D are inside the boundary.

When a new data x_c is added, we initially set the coefficient $\alpha_c^{pq} = 0$, $p = 1, ..., K, q = p+1, ..., K$ change value incrementally and the parameters of the existing support vectors are adapted to satisfy to the KKT conditions. In particular, the adaptation of g_m^{ij} when new data x_c is added can be expressed differentially as:

$$\Delta g_m^{ij} = y_m^{ij}(\beta^{ij,pq}\Delta\alpha_c^{pq}\alpha_c^{pq}K_{cm} + \sum_{x_l \in C_i}(2\Delta\alpha_l^{ij} \tag{2}$$

$$+ \sum_{\substack{n=1 \\ n\neq i,j}}^{K}\Delta\alpha_l^{in})K_{lm} - \sum_{x_l \in C_j}(2\Delta\alpha_l^{ij} + \sum_{\substack{n=1 \\ n\neq i,j}}^{K}\Delta\alpha_l^{nj})K_{lm}$$

$$- \sum_{\substack{n=1 \\ n\neq i,j}}^{K}\sum_{x_l \in C_n}(\Delta\alpha_l^{in} - \Delta\alpha_l^{nj})K_{lm} + (\Delta b_i - \Delta b_j))$$

$$\gamma^{i,pq}\Delta\alpha_c^{pq} + \sum_{\substack{n=1 \\ n\neq i}}^{K}(\sum_{x_l \in C_i}\Delta\alpha_l^{in} - \sum_{x_l \in C_j}\Delta\alpha_l^{in}) = 0 \tag{3}$$

where $i = 1, ..., K, j = i+1, ..., K$, α_c^{pq} is the coefficient to be incremented and K is the kernel function of the SVM, so that $K_{lm} = K(x_l, x_m) = \Phi(x_l)^T\Phi(x_m)$. Coefficient $\beta^{ij,pq}$, $\gamma^{i,pq}$ are defined in[10].

From equation(1), for all the support vectors sv_n^{ij}, we get $g_m^{ij}(sv_m^{ij}) = 0$, then $\Delta g_m^{ij}(sv_m^{ij}) = 0$.Therefore equation(2) and (3) can be written as the following matrix equation: (as described in detail in [10])

$$\begin{bmatrix} \Delta b \\ \Delta \alpha \end{bmatrix} = -RH^{pq}\Delta\alpha_c^{pq} \tag{4}$$

Where $b = [b_1, \ldots, b_K]$, and $\alpha = [\alpha_1^{12}, \alpha_2^{12}, \ldots, \alpha_i^{ij}, \alpha_j^{ij}, \ldots, \alpha_{K-1}^{(K-1)K}, \alpha_K^{(K-1)K}]$, α_i^{ij} expresses the weights of support vectors sv_n^{ij} that belong to the class C_i.

3 Experiments

In this section we examine the face recognition system with the face images captured by our own web cameras. The dataset we create consists of 26 identities. And for each identity, we collect over 400 images in different illumination and direction. To test the real time performance and accuracy of our face recognition system, we design the different experiments. At first, we test the time spent by different steps of the face recognition process and choose a suitable input image resolution for our system to balance accuracy and efficiency. In addition, we also test the performance of the incremental training algorithm we apply by simulating the practical use of the system. We also compare our algorithm with the traditional classifier from different aspects including time consumption and accuracy.

Table 1. Comparison of time efficiency and accuracy on different resolutions and processing steps of images. Bold number represents the better resolution which obtains the better time consumption and accuracy. The unit of time is millisecond.

	1280*720	**640*360**	320*180	160*90
Image Read	8.9	2.7	0.97	0.46
Detection	2394.5	**1602.0**	1438.7	1352.2
Alignment	4.9	5.1	5.0	5.0
Representation	897.2	842.3	822.3	642.0
Recognition	0.96	0.97	0.96	0.95
Accuracy	0.970	**0.969**	0.955	0.954

3.1 Accuracy and Time Consumption of Different Image Resolution

It is obvious that images with higher resolution contain more detail information that may be helpful for face recognition. However, higher resolution also means greater computation. To balance the real time performance and the accuracy, we test the time consumption of every step in our system including face detection, face alignment and face classification and also discuss the influence of different resolutions on the accuracy. The classification models in this experiments are trained with 50 images for each identity.

As shown in Table 1, the image resolution mainly affects the time consumption of face detection step. And this step accounts for almost 75 percent of the whole recognition process. This means the image resolution is very meaningful for the real time performance of the face recognition system. As what is expected, higher resolution brings higher accuracy rate. However, when the resolution is higher than 640*360, the accuracy remains generally stable. Based on the results, we choose to set the input image resolution 640*360 which guarantees high accuracy and real-time of the system.

3.2 Experiments on Incremental Training Algorithm

In this section, we mainly test the performance of the self-learning method with incremental classification training algorithm. We divide the dataset into six partitions and each partition contains 50 images for every identity. We use these partitions to train the incremental SVM classification models with the number of images increasing from 50 to 300. We also carry on the same experiments on the traditional SVM training algorithm with the same datasets for comparison. The time consumption for training each partition of the dataset is recorded as shown in Figure 4.

As shown in Figure 4, the incremental SVM training algorithm spends almost the same even less time on training each partition. In contrast, the training step of the traditional SVM algorithm spends larger amounts of time when the dataset becomes larger. It is obvious that traditional SVM algorithm cannot ensure the system to be stable with the increase of the amounts of the dataset. Besides, the accuracy changes of the incremental algorithm and traditional algorithm are displayed in Table 2. Although it illustrates the traditional algorithm performs a better accuracy rate,

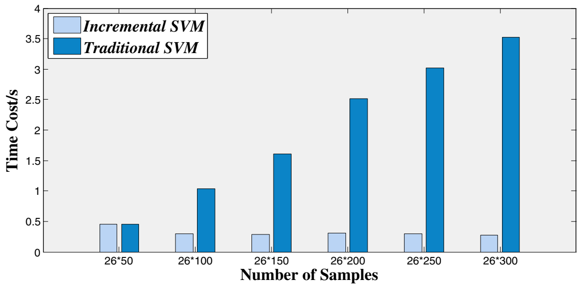

Figure 4. Comparison to time efficiency when sample sizes increase between incremental SVM and traditional SVM. As the sample size increase by the step of 50 for each identity, the time comsume of training on incremental SVM decrease while the traditional time rise by a large margin.

Table 2. The accuracy between incremental SVM and traditional SVM on different sample sizes.

Sample Sizes	Incremental SVM	Traditional SVM
26*50	96.9%	96.9%
26*100	98.9%	99.0%
26*150	99.80%	99.84%
26*200	99.80%	99.85%
26*250	99.81%	99.85%
26*300	99.81%	99.85%

the difference between them is not very large and is acceptable. The incremental algorithm also shows an upward trend in accuracy with the increase of training data. Figure 5 illustrates the confusion matrix of the incremental SVM classifier results with the number of training data get larger. We use the classifier trained with 26 identities containing 300 images each person. The results can suggest our application of incremental SVM is useful and can boost the face recognition performance. In conclusion, the experiment in

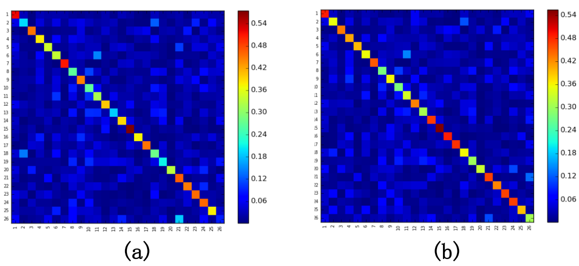

Figure 5. The confusion matrix of the accuracy on incremental SVM. (a) represents the confusion matrix on sample size of and (b) is the one one the sample size of , from which we can see the adaptation with the increase of sample size.

this subsection suggests that the self-learning method with incremental SVM algorithm applied in our system balances the accuracy and the real time performance of the system which is significant for stable operation of a system.

4 Conclusion

In this paper, we design a face recognition system based on OpenFace. And in our system we propose an intelligent classification training method S-DDL with incremental SVM classification training algorithm. Our method makes the system able to modify the classification model during the execution time and get more precise accuracy as well as less training time in face recognition. The results of experiments show the good performance of our system and prove the effetiveness of the incremental learning algorithm. In the future, we will concentrate on the feature extraction stage in which we prepare to apply transfer learning to improve CNNs models.

References

[1] Amos, Brandon, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. OpenFace: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[2] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[3] Torch7 : Collobert, Ronan, Koray Kavukcuoglu, and Clment Farabet. "Torch7: A matlab-like environment for machine learning." BigLearn, *NIPS Workshop*. No. EPFL-CONF-192376. 2011.

[4] Szegedy, C, et al. "Going deeper with convolutions." IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society, 2015:1-9.

[5] Schultz, Matthew, and T. Joachims. "Learning a distance metric from relative comparisons." *International Conference on Neural Information Processing Systems MIT Press*, 2003:41-48.

[6] Zhu, Zhenyao, et al. "Recover canonical-view faces in the wild with deep neural networks." arXiv preprint arXiv:1404.3543 (2014).

[7] Taigman, Y., et al. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification." *Conference on Computer Vision and Pattern Recognition* 2014:1701-1708.

[8] Sun, Yi, Xiaogang Wang, and Xiaoou Tang. "Deeply learned face representations are sparse, selective, and robust." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[9] Huang, Gary B., et al. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Vol. 1. No. 2. Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[10] Boukharouba, Khaled, et al. "Incremental and Decremental Multi-category Classification by Support Vector Machines." *International Conference on Machine Learning and Applications IEEE*, 2009:294-300.

[11] Syed, Nadeem Ahmed, H. Liu, and K. K. Sung. "Handling concept drifts in incremental learning with support vector machines." ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM, 1999:317-321.