IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

Analysis and Automation of Deep Face Recognition

Amani Ali Ahmed Ali[1], **Suresha M**[2]

Dept. of MCA & Computer Science, Kuvempu University, Shankaraghatta, Shimoga, India[1]

Assistant Professor, Dept. of MCA & Computer Science, Kuvempu University, Shankaraghatta, Shimoga, India[2]

Abstract: This paper presents a new approach for face recognition of single face,multi faces, and twins faces, based on Convolutional Neural Network (CNN). Analyzing and exploring essential parameters that can influence model performances the approach has been used in this work to recognize face. Furthermore different elegant prior contemporary models are recruited to introduce new leveraging model. Also show how a very large scale dataset (3M images over 3K people) can be assembled. The proposed method has been compared to the most contemporary approaches and conducted on proposed method with LFW, and YTF datasets and it shows an excellent performance with higher accuracy.

Keywords: Deep Face Recognition, Multiscale images.

I. INTRODUCTION

Recently face recognition becomes vital task using several methods. One of the most interesting used methods is using Convolutional Neural Network (CNN). CNN has been widely used in many real world applications, including image classification and recognition [1,2] and object detection [3] because it is one of the most efficient methods for extracting critical features for non-trivial tasks. CNN consists of a pipeline of alternative several different layers. Unlike neural network, CNN has three different types of layers which are considered a constituent element of CNN. Usually, Convolutional layer, subsampling layers, and fully connected layer are the main components of CNN. Also, there are some intermediate layers between those main layers that will be shown later. Then for a given task, images are passed into CNN to be processed. Passing images through several squish functions incorporated within CNN layers can lead to not leveraging some critical information used for recognition and some of the small features disappear after few layers. The reason for that is because the CNN architecture that implies like those restrictions. Specifically, both convolutional layers and max-pooling layers impose diminishing small features. However, since there are some tasks that have small features that are considered an essential part of a task, then classification using CNN is not efficient because most of those features diminish before reaching the final stage of classification. To implement a robust model, small features must survive for long stages of CNN. To alleviate weaknesses inherited from former CNN models, in this work, different parameters that can influence features surviving for longer distance are explored. Deeper analysis for convolutional and max-pooling layers are presented, and then introduced a model that has more chance for small featuresto survive until the final stage of CNN; specifically directly before fully connected layer.

In the world of face recognition, however, large scale public datasets have been lacking and, largely due to this factor, most of the recent advances in the community remain restricted to Internet giants such as Facebook and Google etc. For example, the most recent face recognition method by Google [4] was trained using 200 million images and eight million unique identities. The size of this dataset is almost three orders of magnitude larger than any publicly available face dataset. Needless to say, building a dataset this large is beyond the capabilities of most international research groups, particularly in academia.

This paper has two goals. The first one is to propose a procedure to create a reasonably large face dataset whilst requiring only a limited amount of person-power for annotation. Authors proposed a method for collecting face data using knowledge sources available on the web. Employed the procedure to build a dataset with over two million faces, and made this freely available to the research community. The second goal is to investigate various CNN architectures for face identification and verification, including exploring face alignment and metric learning, using the novel dataset for training. Many recent works on face recognition have proposed numerous variants of CNN architectures for faces, and assessed some of these modeling choices in order to filter what is important from irrelevant details. The outcome is a much simpler and yet effective network architecture achieving nearstate-of-the-art results on all popular image and video face recognition benchmarks.

II. RELATED WORK

The proposed work mainly concentrates on face recognition and the schematic diagram of general face recognition is shown in Figure 1.

DOI 10.17148/IJARCCE.2016.51299

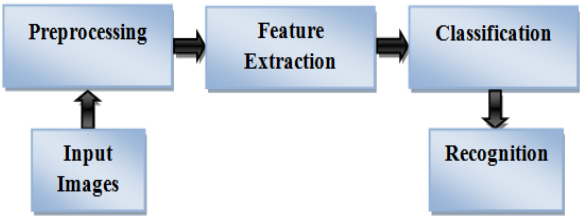

Fig.1.Schematic diagram of face identification system

This paper focuses on face recognition in images, a problem that has received significant attention in the recent past. Among the many methods proposed in the literature, distinguished the ones that do not use deep learning, whichrefer as shallow, from ones that do, that call deep. Shallow methods start by extracting a representation of the face image using handcrafted local image descriptors such as [5, 6, 9]; then they aggregate such local descriptors into an overall face descriptor by using a pooling mechanism, for example the Fisher Vector [10]. There are a large variety of such methods which cannot be described in detail here [10].

This work is concerned mainly with deep architectures for face recognition. The defining characteristic of such methods is the use of a CNN feature extractor, a learnable function obtained by composing several linear and non-linear operators. A representative system of this class of methods is DeepFace [11]. This method uses a deep CNN trained to classify faces using a dataset of 4 million examples spanning 4000 unique identities. It also uses a siamese network architecture, where the same CNN is applied to pairs of faces to obtain descriptors that are compared using the Euclidean distance. The goal of training is to minimize the distance between congruous pairs of faces (i.e. portraying the same identity) and maximise the distance between incongruous pairs, a form of metric learning. In addition to using a very large amount of training data, DeepFace uses an ensemble of CNNs, as well as a pre-processing phase in which face images are aligned to a canonical pose using a 3D model. When introduced, DeepFace achieved the best performance on the Labelled Faces in the Wild (LFW; [12]) benchmark as well as the YouTube Faces in the Wild (YFW; [9]) benchmark. The authors later extended this work in [14], by increasing the size of the dataset by two orders of magnitude, including 10 million identities and 50 images per identity. They proposed a bootstrapping strategy to select identities to train the network and showed that the generalization of the network can be improved by controlling the dimensionality of the fully connected layer.

The DeepFace work was extended by the DeepId series of papers [15, 16], each of which incrementally but steadily increased the performance on LFW and YFW. Compared to DeepFace, DeepID does not use 3D face alignment, but a simpler 2D affine alignment (as do in this paper) and trains on combination of CelebFaces [15] and WDRef [17]. However, the final model in [16] is quite complicated involving around 200 CNNs.

Very recently, researchers from Google [4] used a massive dataset of 200 million face identities and 800 million image face pairs to train a CNN similar to [18]. A point of difference is in their use of a “triplet-based” loss, where a pair of two congruous (a,b) and a third incongruous face c are compared. The goal is to make a closer to b than c; in other words, differently from other metric learning approaches, comparisons are always relative to a pivot face. This matches more closely how the metric is used in applications, where a query face is compared to a database of other faces to find the matching ones. In training this loss is applied at multiple layers, not just the final one. This method currently achieves the best performance on LFW and YTF.

III.DATA COLLECTION

In this section authors propose a multi-stage strategy to effectively collect a large face dataset containing hundreds of example images for thousands of unique identities. Individual stages are discussed in detail in the following paragraphs.

Stage1. Bootstrapping and filtering a list of candidate identity names. The first stage in building the dataset is to obtain a list of names of candidate identities for obtaining faces. The idea is to focus on celebrities and public figures, such as actors or politicians, so that a sufficient number of distinct images are likely to be found on the web, and also to avoid any privacy issue in downloading their images. An initial list of public figures is obtained by extracting males and females, ranked by popularity, from the Internet Movie Data Base (IMDB) celebrity list. This list, which contains mostly actors, is intersected with all the people in the Freebase knowledge graph [20], which has information on about 500K different identities, resulting in ranked lists of 2.5K males and 2.5K females. This forms a candidate list of 5K names which are known to be popular (from IMDB), and for which have attribute information such as ethnicity, age, kinship etc. (from the knowledge graph).
The total of 5K images was chosen to make the subsequent annotation process manageable for a small annotator team. The candidate list is then filtered to remove identities for which there are not enough distinct images, and to eliminate any overlap with standard benchmark datasets. To this end 350 images for each of the 5K names are then presented to human annotators (sequentially in four groups of 50) to determine which identities result in sufficient image purity. Specifically, annotators are asked to retain an identity only if the corresponding set of 350 images is roughly 96% pure. The lack of purity could be due to homonymy or image scarcity. This filtering step reduces the candidate list to 3,750 identities. Next, any names appearing in the LFW and YTF datasets are removed in order to make it possible to train on the new dataset and still evaluate fairly on those benchmarks. In this manner, a final list of 2,266 celebrity names is obtained.

DOI 10.17148/IJARCCE.2016.51299

Stage2. Collecting more images for each identity: Each of the 2,266 celebrity names is queried in both Google and Bing Image Search, and then again after appending the keyword “actor” to the names. This results in four queries per name and 500 results for each, obtaining 2,000 images for each identity.

Stage3. Improving purity with an automatic filter: The aim of this stage is to remove any erroneous faces in each set automatically using a classifier. To achieve this the top 50 images (based on Google search rank in the downloaded set) for each identity are used as positive training samples, and the top 50 images of all other identities are used as negative training samples. A one-vs-rest linear SVM is trained for each identity using the Fisher Vector Faces descriptor [10]. The linear SVM for each identity is then used to rank the 2,000 downloaded images for that identity, and the top 1,000 are retained (the threshold number of 1,000 was chosen to favour high precision in the positive predictions).

Stage4. Near duplicate removal: Exact duplicate images arising from the same image being found by two different search engines, or by copies of the same image being found at two different Internet locations, are removed. Near duplicates (e.g. images differing only in color balance, or with text superimposed) are also removed. This is done by computing the VLAD descriptor [21, 22] for each image, clustering such descriptors within the 1,000 image for each identity using a very tight threshold, and retaining a single element per cluster.

Stage5. Final manual filtering: At this point there are 2,266 identities and up to 1,000 images per identity. The aim of this final stage is to increase the purity (precision) of the data using human annotations. However, in order to make the annotation task less burdensome, and hence avoid high annotation costs, annotators are aided by using automatic ranking once more.
This time, however, a multi-way CNN is trained to discriminate between the 3,750 face identities using the AlexNet architecture of [23]; then the softmax scores are used to rank images within each identity set by decreasing likelihood of being an inlier. In order to accelerate the work of the annotators, the ranked images of each identity are displayed in blocks of 350 and annotators are asked to validate blocks as a whole. In particular, a block is declared good if approximate purity is greater than 99%.The process of which approximately 99% are frontal and 1% profile.

Discussion: Overall, this combination of using Internet search engines, filtering data using existing face recognition methods, and limited manual curation is able to produce an accurate large-scale dataset of faces labeled with their identities. The human annotation cost is quite small – the total amount of manual effort involved is only around 15 days, and only four days up to stage 4. Table 5 compares our dataset to several existing ones.

A number of design choices have been made in the process above. Here suggest some alternatives and extensions. The Freebase source can be replaced by other similar sources such as DBPedia (Structured WikiPedia) and Google Knowledge Graph. In fact, Freebase will be shut down and replaced by Google Knowledge Graph very soon. On the image collection front, additional images can be collected from sources like Wikimedia Commons, IMDB and also from search engines like Baidu and Yandex.

The removal of identities overlapping with LFW and YTF in stage 1 could be removed in order to increase the number of people available for the subsequent stages. The order of the stages could be changed to remove near duplicates before stage 2. In terms of extensions, the first stage of collection can be automated by looking at the distribution of pairwise distances between the downloaded images. An image class with high purity should exhibit a fairly unimodal distribution.

Fig.2.Example images from our dataset for six identities.

IV. PROPOSED METHOD

This work principally is recruited two different deep neural network models named (Network In Network) NIN and SPPnet explored in [1, 24] respectively and implemented new unified model. Next sections start exploring in depth the influence and leveraging of incorporating both models on network architectures and how they can influence classification performance. Then the unified proposed model is an elegant model because it shortens some weaknesses inherited from former models. Thus exploring both architectures is accomplished next sections to show model’s robustness on image classification.

A. Pipeline Steps of Face Classification

Face classification is not trivial task which can be achieved using various approaches. However, recently deep learning has been successfully applied to a wide range of machine learning applications. Accordingly, in this work authors proposed a subtle Convolutional Neural Networks (CNNs) which is used to train and test our face. Constructing CNNs plays an essential role in justifying both performance and time consumption. Thus, in our implementation, Authors designed an elegant CNN after carefully investigating its parameters.

DOI 10.17148/IJARCCE.2016.51299

In general, using CNNs for face recognition consists of a certain number of steps described below:

Preparing patterns before feeding to the CNN. All images (either a single photograph or from a set of faces tracked in a video) are pre-processed before passing into the network. After preparing images, they are fed to the deep model to extract features. As demonstrated earlier a robust CNN is used in this experiment to extract robust features used in the final decision to justify the class to which they belong to. Different CNN architectures are used for training and extracting features. Specifically very contemporary works are recruited and incorporated for introducing new unified model which achieves the state-of-the-art image classification on the datasets used in this work. Furthermore, a robust CNN is proposed at the end of this work which accomplishes superior results comparing with recent works.

Finally, the final outputs of CNN are evaluated for final scoring results. There are different methods to score final results of CNN either using SVM or using CNN itself by using soft-max layer build on the top of CNN. Thus the last layer named softmax layer in this work to evaluate and score the final recognition results is used at the top of CNN to minimize the error soft-max layer are.

In this work, Authors carefully explored the architecture of CNN consisting of five layers as shown in figure 3. It is clear that the network has three main stages for classification and as described below:

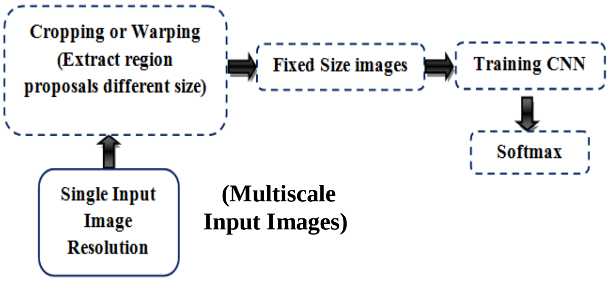
Cropping or Warping
(Extract region proposals different size)
Fixed Size images
Training CNN
Softmax
Single Input Image Resolution
(Multiscale Input Images)

Fig.3.classification steps of our face dataset

B. Exploring Different CNN Architectures

It is obvious that the proposed network in fig. 3 achieves competitive results comparing to prior works. In addition, it accomplishes results which outperform accomplished work in [13] specifically it dominants over deep neural network approaches. Moreover, it achieves competitive results to many other approaches. The stimulating results are supportive to dig deeper and to investigate influential parameters and explore more robust model. In this part, recruited models will be used for further investigation and more effort will be put to explore more appropriate architecture for image classification. Leveraging CNN architecture is proposed in this section used for image recognition. It achieves state-of-the-art results on given benchmarks. Consequently, more parameters that can influence model performance are discussed next.

This work proposes a new topology for CNN architecture. Fig. 4 depicts the proposed model and it has drastically changes comparing with one implemented and explored in fig. 3. The proposed model inherits some leverage points from NIN. Instead of using conventional connection between convolutional layers as describe in [7]. The robust connection proposed in NIN is incorporated in this work to increase and gain more accuracy on image classification. The size of CNN is kept the same as depicted in fig.3. The merit of this CNN architecture combines more than one elegant method such as multi-scale input images and nonlinear transformation between convolutional layers as demonstrated in [1] as shown in fig. 4.

To look deeper inside CNN and investigate the most critical parameters that can influence model performance. Fig.5 shows both convolutional and sub-sampling layers of CNN.

It is clear the subsequent of alternative between these kinds of layers; it quickly diminishes the input images after few stages of CNN leading to losing vital information useful for final stage of classification. Specifically this work is dealing with small image sizes as will be obtained later. All the benchmarks used in this work have image sizes of 32x32 pixels. Consequently the small features will be not available after few stages. Therefore an elegant model of CNN architecture is proposed in this work as shown in fig. 6. It is clear that new model propose different connection than standard connection of conventional CNN.

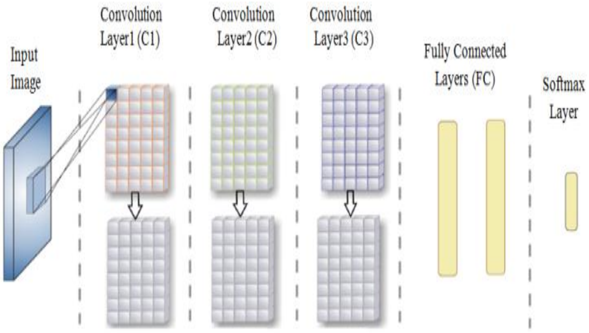

Fig. 4.CNN incorporated with two robust models

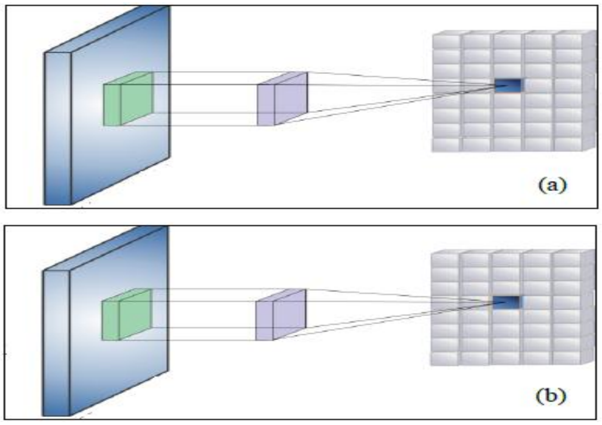

Fig. 5.CNN's layers (a) convolutional (b) max-pooling

Some layers are received their connections not only directly from the layer below but also from two and three layers below. The reason for this kind of connections because small features within the input images can survive longer and will be part of the final scoring detection results. Furthermore, the first layers of CNN extract global features of input objects but as the images advance toward final fully connect layers, more accurate features are extracted.

C. Exploring Different CNN Sizes

In order to precisely analyze the influence of different CNN architectures, a new CNN architecture is proposed and carefully selected their parameter because same CNN architecture might work sufficiently for some tasks and inadequately for other tasks. Hence, in this part different deep model architectures is investigated that can fit for image recognition. Accordingly, CNN architectures are explored to be best suited for image classification. There are two model architectures are used in our experiments. They are shown in table I. In addition to the structure obtained in table 1, each network has more additional two fully connected layers build on the top of the final max-pooling layer. Then finally, soft-max layer is built on the top of final fully connect layer used for final scoring results. It is clear that there are two CNN architectures detailed in table1 called Network1 and Network2. It is obvious that network1 is smaller than the network2. Where, network1 consists of three convolutional layer and three max-pooling layers.

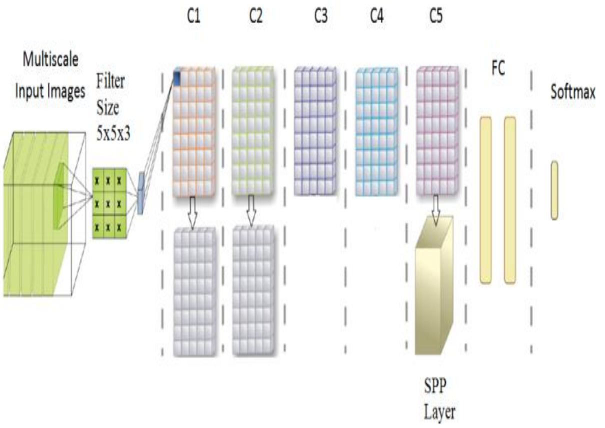

Fig. 6.CNN with five convolutional layers

TABLE I. TWO CNN ARCHITECTURES. THE ABBREVIATION C REFERS TO CONVOLUTION. LRN AND RELU ARE ABBREVIATION FOR LOCAL RESPONSE NORMALIZATION AND RECTIFIED LINEAR UNIT RESPECTIVELY

Model name	Input size	C1/ p	C2 /p	C3/p	C4/p	C5/p
Network1	32x32	192x5x5,str:1, ReLU	256x5x5,str:1, ReLU	192x3x3,str:1, ReLU		
		2x2, LRN	2x2, LRN	2x2, LRN		
Network2	32x32	192x5x5,str:1, ReLU	256x1x1,str:1, ReLU	384x1x1,str:1, ReLU	256x1x1, str:1, ReLU	192x3x3, str:1, ReLU
		3x2	3x2			Spp layer

V. EXPERIMENTATION

In the following firsttested several variations of the proposed models and training datausing ours dataset to test performance and compare it with other datasets.Then, compared the performance of the bestsetups with state-of-the-art methods on LFW and YTF.

Faces are detected using the method described in [19]. Ifface alignment is used, thenfacial landmarks are computed using the method of [8] and a 2D similarity transformation isapplied to map the face to a canonical position.

For the YTF videos, K face descriptors are obtained foreach video by ordering thefaces by their facial landmark confidence score and selecting the top K. Frontal faces are2D aligned, but no alignment is used for profiles. Finally, the video is represented by theaverage of the K face descriptors.

The following evaluates the effect of different components of the system when training a network for face verification and evaluating it on the LFW data. Table 2 summaries the results.

A. Dataset Curation

First Authors analyse the curation effort for its effect on the performance of the network. Authors use dataset snapshots at stages 3 and 5 i.e. before and after curation, and test them using configuration A. The reason for selecting this configuration is that the networks can be trained from scratch. As can be seen, performance before curation is better (Table 2 rows 1 and 2). There are probably two reasons for this: first, it is better to have more data, even with label noise; and second, a more subtle point, some of the hard positives present in the stage 3 data get removed as a side effect of the curation process, and so the stage 5 data training does not benefit from these.

As can be seen from Table 2 rows 2 and 3, using 2D alignment on test images do improve the performance, but performing 2D alignment on the training data does not provide an additional boost – see (Table 2 rows 4 and 5).

Architecture: Next authors vary the architecture of the network,observed a slight boost in performance from configuration A to B (Table 2 rows 3 and 4) while configuration D fails to improve the results over configuration B (Table 2 rows 4 and 6). There are several possible reasons for this: the number of parameters in config. D is much more than B, due to the greater number of convolution layers. Also since network D is trained using fine-tuning of the new layers, setting parameters like learning rate and momentum becomes critical. Training the network from scratch is also an option which needs to be investigated in the future. Triplet-loss embedding: Learning a discriminative metric by minimizing the triplet loss further improves performance by 1:9% (Table 2 row 7 vs. row 4).Note that this amounts to reducing the error rate.

Table 2 shows Training on the full dataset (F, stage 3), leads to a better performance than training on the curated dataset (C, stage 5). 2D alignment at the test time slightly improves the performance. Learning embedding for verification significantly boosts the performance. All results are obtained using l2 distance measure between the test samples.

B. Comparison with the state-of-the-art

LFW: Table 3 compares our results with the best results on LFW dataset, and also shows these as ROC curves. It can be observed that achieve comparable results to the state of the art using much less data and much simpler network architecture. YTF: Table 4 shows the performance on the YTF dataset. Authors achieved the state of the art performance using our triplet loss embedding method.

Table 3: LFW unrestricted setting. Left: achieve comparable results to the state of the art.

TABLE 2: PERFORMANCE EVALUATION ON LFW, UNRESTRICTED SETTING.

No.	Config.	Data	Train Align.	Test Align.	Embedding	100% - EER
1	A	C	NO	NO	NO	95.22
2	A	F	NO	NO	NO	96.70
3	A	F	NO	YES	NO	98.99
4	B	F	NO	YES	NO	98.88
5	B	F	YES	YES	NO	97.32
6	D	F	NO	YES	NO	97.86
7	B	F	NO	YES	YES	99.99

TABLE 3: LFW UNRESTRICTED SETTING. LEFT: ACHIEVE COMPARABLE RESULTS TO THE STATE OF THE ART

Method	Ref. #	Test Accuracy
Fisher Vector Faces	[25]	93.10 %
DeepFace	[11]	97.35%
Fusion	[14]	98.37%
FaceNet	[4]	98.87%
FaceNet + Alignment	[4]	99.63
Network1	ours	99.96%
Network2	ours	99%

TABLE 4: RESULTS ON THE YOUTUBE FACES DATASET, UNRESTRICTED SETTING. THE VALUE OF K INDICATES THE NUMBER OF FACES USED TO REPRESENT EACH VIDEO.

Method	Ref. #	Test Accuracy
Video Fisher Vector Faces	[10]	93.10 %
DeepFace	[11]	97.35%
FaceNet + Alignment	[4]	99.63%
Network1	ours	99.95%
Network2	ours	99.99%

Provided a few examples of both twins' images misclassificationsin Figures 7.This show that many of the mistakes made by our system are due to extremely challenging viewing conditions of some of the twins and Adience benchmark images. Most notable are mistakes caused by blur or low resolution

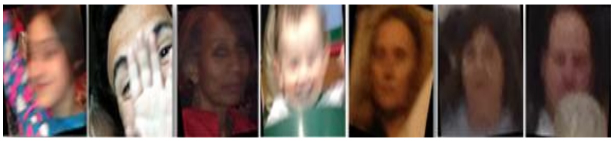

Fig. 7.misclassification images

DOI 10.17148/IJARCCE.2016.51299

TABLE 5: DATASET COMPARISONS: OUR DATASET HAS THE LARGEST COLLECTION OF FACE IMAGES OUTSIDE INDUSTRIAL DATASETS BY GOOLE, FACEBOOK, OR BAIDU, WHICH ARE NOT PUBLICLY AVAILABLE.

Dataset	Ref. #	Identities	Images
LFW		5,749	13,233
WDRef	[17]	2,995	99,773
CelebFaces	[15]	10,177	202,599
FaceBook	[11]	4,030	4.4M
Google	[4]	8M	200M
Ours		2,266	3M

VI. CONCLUSION

In this work, Authors have made two contributions: first, designed a procedure that is able to assemble a large scale dataset, with small label noise, whilst minimizing the amount of manual annotation involved. One of the key ideas was to use weaker classifiers to rank the data presented to the annotators. This procedure has been developed for faces, but is evidently suitable for other object classes as well as fine grained tasks. The second contribution was to show that image recognition using the deep neural network is introduced. Different model architectures are proposed by incorporating different prior elegant CNNs. Specifically both NIN and SPPnet are incorporated in a single unified model that achieves superior results comparing to former results. Then a new model is presented and outperforms prior work and accomplishes state-of-the-art results on the datasets. Also, different model architectures are introduced, and extensive parameters are discussed that can influence model performance. Deeper exploring different parameters that can be suited for CNN recognition model are presented as well. For evaluation, the experiments are conducted on challenge datasets.

REFERENCES

[1] Min Lin,Qiang Chen, and Shuicheng Yan, "Network In Network",arXiv 1312.4400v3,4 Mar 2014 .

[2] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, YoshuaBengio, "Maxout Networks", ICML 2013 .

[3] DumitruErhan, Christian Szegedy, Alexander Toshev, and DragomirAnguelov, "Scalable Object Detection using Deep Neural Networks", CVPR, 2014.

[4] F. Schroff, D. Kalenichenko, and J. Philbin. "Facenet: A unified embedding for face recognition and clustering". In Proc. CVPR, 2015.

[5] R. G. Cinbis, J. J. Verbeek, and C. Schmid. "Unsupervised metric learning for face identification in TV video". In Proc. ICCV, pages 1559–1566, 2011.

[6] C. Lu and X. Tang. "Surpassing human-level face verification performance on lfw with gaussianface". AAAI, 2015.

[7] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and YoshuaBengio. "Maxout networks". arXiv preprint arXiv:1302.4389, 2013.

[8] M. Everingham, J. Sivic, and A. Zisserman. "Taking the bite out of automatic naming of characters in TV video". Image and Vision Computing, 27(5), 2009.

[9] L. Wolf, Tal. Hassner, and I. Maoz. "Face recognition in unconstrained videos with matched background similarity". In Proc. CVPR, 2011.

[10] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman."A compact anddiscriminative face track descriptor". In Proc. CVPR, 2014.

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. "Deep-Face: Closing the gap to human-level performance in face verification". In Proc. CVPR, 2014.

[12] G. B. Huang, M. Ramesh, T. Berg, and E. "Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments". Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[13] Hayder M. Albehadili and NazIslam,"Hybrid Algorithm For the Optimization of Training Convolutional Neural Network". Volume 6 IJACSA,No. 10, October 2015 .

[14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. "Web-scale training for face identification". In Proc. CVPR, 2015.

[15] Y. Sun, X.Wang, and X. Tang." Deep learning facerepresentation from predicting 10,000 classes". In Proc. CVPR, 2014.

[16] Y. Sun, L. Ding, X. Wang, and X. Tang. "Deepid3: Face recognition with very deep neural networks". CoRR, abs/1502.00873, 2015.

[17] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. "Bayesian face revisited: A joint formulation". In Proc. ECCV, pages 566–579, 2012.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". CoRR, abs/1409.4842, 2014.

[19] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. "Face detection without bells and whistles". In Proc. ECCV, 2014.

[20] Freebase. http://www.freebase.com/.

[21] R. Arandjelovi´c and A. Zisserman. "All about VLAD". In Proc. CVPR, 2013.

[22] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. P'erez, and C. Schmid. "Aggregating local image descriptors into compact codes". IEEE PAMI, 2011.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks". In NIPS, pages 1106–1114, 2012.

[24] Kaiming, He and Xiangyu, Zhang and Shaoqing, Ren and Jian Sun,"Spatial pyramid pooling in deep convolutional networks for visual recognition", European Conference on Computer Vision, 2014.

[25] K. Simonyan, A. Vedaldi, and A. Zisserman. "Learning local feature descriptors using convexOptimization". IEEE PAMI, 2014.

DOI 10.17148/IJARCCE.2016.51299