IOP Conf. Series: Materials Science and Engineering **353** (2018) 012022 doi:10.1088/1757-899X/353/1/012022

Pubface: Celebrity face identification based on deep learning

H Ouanan[1*], M Ouanan[1] and B Aksasse[1]

[1]Dept. of Computer Science, ASIA Team, M2I Laboratory, Faculty of science and techniques, Errachidia, Morocco

Email: *ham.ouanan@gmail.com

Abstract. In this paper, we describe a new real time application called PubFace, which allows to recognize celebrities in public spaces by employs a new pose invariant face recognition deep neural network algorithm with an extremely low error rate. To build this application, we make the following contributions: firstly, we build a novel dataset with over five million faces labelled. Secondly, we fine tuning the deep convolutional neural network (CNN) VGG-16 architecture on our new dataset that we have built. Finally, we deploy this model on the Raspberry Pi 3 model B using the OpenCv dnn module (OpenCV 3.3).

1. Introduction

The human face plays an important role in our social interaction, carrying information about people's identity. Due to recent progress in performance, the use of facial recognition technology has grown significantly in the past several years due to its potential for a wide variety of applications in both commercial applications and security. Compared to other biometric technology, face recognition techniques does not require the collaboration of the subject. It enables faces to identify a person in real-time.

Recently face recognition systems was known extraordinary breakthrough. Much of this progress can be attributed to the success of deep learning based techniques, which are actively being used in many areas where traditional algorithmic solutions don't work well or don't work at all. These techniques perform an assumption that by collecting massive training sets, deep networks will have sufficient examples of both inter-subject and intra-subject appearance variations. From these variations, artificial neural networks can learn to generate discriminative features, which amplify subject identity and suppress other, confounding appearance variations. So, to train face recognition systems based on deep CNNs, very large training sets are needed with millions of labeled images. Recently, the number of face images has been growing exponentially on social network such as Facebook and Twitter. As an example, the director of Facebook AI Research Yann LeCun has said, "almost 1 billion new photos were uploaded each day on Facebook in 2016". However, large training datasets are not publicly available and very difficult to collect. In addition, to build large dataset by downloading the images from search engine is very difficult and most financially challenging. One way to get around a lack of large face datasets is to augment face datasets by synthesizing new possible views of the face images

IOP Conf. Series: Materials Science and Engineering **353** (2018) 012022 doi:10.1088/1757-899X/353/1/012022

they contain. Thereby, an existing face set is expanded to many times its size by introducing supplementary intra-subject appearance variations such as pose variations.
The remainder of the paper is as follows. In Section 2 we talks about the related works, Section 3 introduce our new proposed method (Pubface). In Section 4, we present the experiments and discuss our method. Section 5 concludes the paper.

2. Related work

The problem with small datasets is that deep learning models particularly Convolutional Neural Networks (CNNs) trained with them do not generalize well data from the validation and test set. Hence, these models suffer from the problem of overfitting. Therefore, to combat overfitting, many techniques have been proposed for example, dropout [1] and batch normalization [2]. Dropout works by probabilistically removing a neuron from designated layers during training or by dropping certain connection. Batch normalization method, allows us to train the normalization weights and normalizes layers. Batch normalization can be applied to any layer within the net and hence is very effective. Another popular method is data augmentation, where we use the data you currently have and manipulating them to produce more altered versions of the same image, to increase the variety of data seen during training. Data augmentation has shown effective in image classification [3]. Various smart data augmentation methods have been proposed include geometric augmentations such as oversampling [4], rotating [5], mirroring [6] and photometric transformations [7]. In addition to the typical transformations, face image was augmented with Hairstyles synthesis [8], Glasses synthesis [9], Illuminations synthesis [10]. Another framework named Hyper-class augmented and regularized deep learning [11]. This exposes the CNNs models to additional variations without the cost of collecting and annotating more data. This can have the effect of reducing overfitting and improving the model's ability to generalize.

3. Pubface

3.1. Dataset Collection

We have proposed a novel semi-automatic approach to build very large training datasets of synthetic images by compositing real face images in a small dataset with minimal human intervention. Specifically, we present a multi-stage method proposed to build a large synthetic face dataset with over five million faces labelled for identity by employing a smart synthesis augmented approach based on rendering pipeline to increase the pose and lighting variability. The different steps of the proposed approach are summarized below in Table 1; next we provide details about each step.

Step 1: Finding names of celebrities.
Step 2: Finding popular celebrities.
Step 3: Remove overlap with LFW benchmark.
Step 4: Collect representative images for each celebrity
Step 5: Face images were detected using a robust face detector.
Step 6: Remove the copies images of the same image.
Step 7: Annotation process.
Step 8: Synthesizing faces.

Figure 1. Main stages of the dataset building process

- The first step in building the face dataset is to get an initial list of public figures and popular celebrity's names i.e. the people with high representation on search engines, such as actors, football players and politicians to avoid any privacy issue in downloading their pictures. This is done by using the Internet Movie Data Base (IMDB) celebrity list, which contains the most of public figures ranked by popularity.

- The second step is the application of the filtering process to eliminate any overlap with the standard LFW [12] benchmark by removed any celebrity name appearing in the LFW face dataset in order to make it possible to train on the new dataset and still evaluate fairly on LFW benchmark, resulting in a lists of 4000 identities.
- The third step of the process consist in the collect a set of representative images for each celebrity of the candidate list from sources like Wikimedia Commons, and search engines like Yandex, Google Image Search and Facebook Graph Search.
- The four step of the process consist the running the Viola-Jones face detector [13] on all images collected in the previous stage to obtain the location of the face in each given image and then cropped and rescaled to a standard coordinate system. In order to align the faces we apply the face alignment method [14] to rotate them accordingly to an up-frontal position.
- The five step is to remove the duplicate images of the same image being found at two different Internet locations. This is done by computing one of the descriptors presented on [15] [16] [17] [18] for all image per each identity.
- The sex step in building the face dataset is the manual annotation process of all face images obtained in the previous stage.
- Finally, in order to augment the size of our dataset, by produce multiple views of face images with more intra-subject appearances variations such as pose variations. We consider an extension of the above-proposed method to produce frontal views to minimize variability for better alignment (Test phase) while we do this to augment variability for better capture intra-class appearance variations (Training phase). Therefore, to produce multiple views of each face image I, we begin by applying the facial landmark detector proposed in [19]. Given these detected facial key points we estimate the 6 DOF for the face in the given image I using correspondences between the facial key points $p_i \epsilon IR^2$ and points $P_i \epsilon IR^3$, labeled on a 3D generic face model S. Here, i indexes specific facial landmarks in I and the 3D shape S. we use PnP [20] to estimate extrinsic camera parameters, which gives us a perspective camera model mapping the generic 3D shape S on the image I. As well as the we make new suitable rotation matrices R for novel multiple views by sampling different yaw angles

$$p_i \sim MP_i \tag{1}$$

$$M = A\ [R\ t] \tag{2}$$

3.2. Network architecture

To perform face recognition, we considered the configuration D of the CNN architecture proposed in [21] in our investigation, which is referred as VGG16 trained on ImageNet [4] for image classification as well as for face recognition [22]. It comprise 11 blocks. The eight first blocks are said to be convolutional. All the convolution layers are followed a rectified unit layers and max pooling. The last three blocks are said to be fully connected layers. The output layer is adapted to 4000 output neurons corresponding to the number of classes of the corresponding Input layer. The number of layers in between the Input layer and the final dense layer denotes the depth of the DL model. To benefit from the representation power of the CNN, fine-tuning of this pre-trained model with our new dataset is a necessary step. We then deploy this model on the Raspberry Pi 3, model B (hardware with limited computational performance).

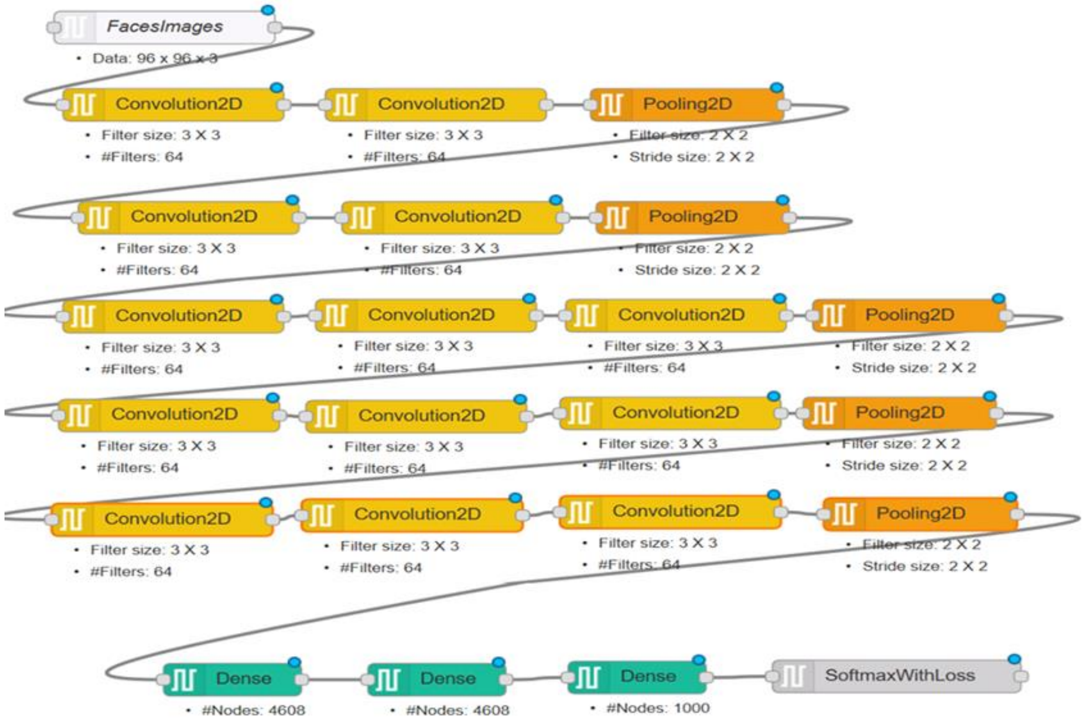

Figure 2. VGG16 architecture

4. Experiments and results

We call the generic dataset building as Puball-dataset, which all the images have the size of 96x96x3 pixels. Table 1 gives some statistical information on the larger face datasets public and private. The whole dataset is split as 60% for training, 20% validation, and 20% for testing.

Table 1. Dataset comparison

Dataset	Identities	Images
Facebook [23]	4,030	4.4M
Google [24]	8M	200M
MegaFace [25]	690,572	1.02M
LFW [12]	5,749	13,233
CelebFaces [26]	10,177	202,599
Puball-dataset (ours)	4000	5M

The entire CNN model was implemented in python using the deep-learning framework TensorFlow. The use of the pre-processing and alignienement procedures prior to classification increased the recognition rate from 90% to 98%. As benchmark we compare the proposed approach to OpenCV classifier was achieved the highest performance known as Fisherface.

Table 2. Benchmark Accuracy rate and Classification time for the proposed model and Fisherface

Method	Accuracy Rate (%)	Classification time (ms)
Fisherface	96	420
Our proposed model (ours)	98	90

IOP Conf. Series: Materials Science and Engineering **353** (2018) 012022 doi:10.1088/1757-899X/353/1/012022

Figure 3. Example of face images from our new dataset

5. Conclusion

In conclusion, our deep-learning based approach called Pubface lead to impressive results on hardware with limited resources with respect to both speed and accuracy compared to traditional OpenCV classifier known as fisherface which was achieved the highest performance. Pubface can recognize faces with 98% accuracy in real time, which is pretty much as good as humans can do. It's pretty amazing technology.

References

[1] G. E. HINTON, N. SRIVASTAVA, A. KIRZHEVSKY, I. SUTSKEVER and R. R. Salakhutdinov, "Improving neural networks by preventing co-daptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.

[2] Y. MA and D. KLABJAN, "Convergence analysis of batch normalization for deep neural nets," CoRR, abs/1705.08011, 2017.

[3] S. C. WONG, A. GATT, V. STAMATESCU and M. D. MCDONNEL, "Understanding data augmentation for classification: when to warp?," CoRR, abs/1609.08764, 2016.

[4] A. KRIZHEVSKY, I. SUTSKEVER and G. E. HINTON, "Imagenet classification with deep convolutional neural networks," Neural Inform. Process. Syst., p. 1097–1105, 2012.

[5] S. XIE and Z. TU, "Holistically-nested edge detection," Proc. Int. Conf. Comput. Vision, 2015.

[6] H. YANG and I. PATRAS, "Mirror, mirror on the wall, tell me, is the error small?," Proc. Conf. Comput. Vision Pattern Recognition, 2015.

[7] K. HE, X. ZHANG, S. REN and J. SUN, "Spatial pyramid pooling in deep convolutional networks for visual recognition," arXiv:1406.4729 [cs.CV], 2015.

[8] L. Luoqi , X. Hui , X. Junliang , L. Si , Z. Xi and S. Yan, "Wow! you are so beautiful today!," Proceedings of the 21st ACM international conference on Multimedia, pp. 3-12, 2013.

[9] W. Yandong , L. Weiyang , Y. Meng , F. Yuli , X. Youjun and H. Rui , "Structured occlusion coding for robust face recognition," Neurocomputing, p. 11–24, 2016.

[10] J. Dalong, H. Yuxiao, Y. Shuicheng, Z. Lei, Z. Hongjiang and G. Wen, "Efficient 3D

reconstruction for face recognition," Pattern Recognition , vol. 38, no. 6, pp. 787-798, 2005.

[11] S. XIE, T. YANG, X. WANG and Y. LIN, "Hyper-class augmented and regularized deep learning for fine-grained image classification," Proc. Conf. Comput. Vision Pattern Recognition, p. 2645–2654, 2015.

[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," University of Massachusetts, Amherst, Technical Report, pp. 07-49, 2007.

[13] H. Ouanan, M. Ouanan and B. Aksasse, "Implementation and optimization of face detection frameworkbased on OpenCV library on mobile platforms using Davinci's technology," International Journal of Imaging and Robotics, vol. 16, no. 2, pp. 118-131, 2016.

[14] H. Ouanan, M. Ouanan and B. Aksasse, "Novel approach to pose invariant face recognition," Procedia Computer Science, 2017.

[15] H. Ouanan, M. Ouanan and B. Aksasse, "Myface: Unconstrained Face Recognition," Lecture Notes in Networks and Systems, 2017.

[16] H. Ouanan, M. Ouanan and B. Aksasse, "Gabor-HOG Features based Face Recognition Scheme," TELKOMNIKA Indonesian Journal of Electrical Engineering, vol. 15, no. 2, pp. 331-335, 2015.

[17] H. OUANAN, M. OUANAN and B. Aksasse, "Gabor-Zernike Features based Face Recognition Scheme," International Journal of Imaging and Robotics, vol. 16, no. 2, pp. 118-131, 2016.

[18] H. OUANAN, M. OUANAN and B. AKSASSE, "Face Recognition by Neural Networks based on Gabor filters and Random Projection," International Journal of Mathematics and Computation, vol. 26, no. 2, pp. 30-42, 2015.

[19] H. Ouanan, M. Ouanan and B. Aksasse, "Facial landmark localization: Past, present and future," 4th IEEE International Colloquium on Information Science and Technology (CIST), 2016.

[20] T. Taketomi, K. Okada, G. Yamamoto, J. Miyazaki and H. Kato, "Camera pose estimation under dynamic intrinsic parameter change for augmented reality," Computers & Graphics, vol. 44, pp. 11-19, 2014.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[22] H. Ouanan, M. Ouanan and B. Aksasse, "Face Recognition Using Deep Features," Lecture Notes in Networks and Systems , 2017.

[23] T. Yaniv , Y. Ming , R. Marc'Aurelio and W. Lior , "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1701-1708, 2014.

[24] F. Schroff, D. Kalenichenko and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[25] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4873-4882, 2016.

[26] Y. Sun, X. Wang and X. Tang, "Deep learning face representation from predicting 10,000 classes," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[27] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov , "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv, pp. 01-18, 2012.