2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th Intl. Conference on Data Science and Systems

Tackling Rare False-Positives in Face Recognition: a Case Study

1st Sepehr Meshkinfamfard
University of Leicester
Leicester, UK
Visual Management Systems Ltd
Glasgow, UK
sm901@le.ac.uk

2nd Alexander N Gorban
University of Leicester
Leicester, UK
and Lobachevsky University
Nizhni Novgorod, Russia
a.n.gorban@le.ac.uk

3rd Ivan Y. Tyukin
University of Leicester
Leicester, UK
Lobachevsky University
Nizhny Novgorod, Rissia
St-Petersburg State Electrotechnical University
Saint-Petersburg, Rissia
I.Tyukin@le.ac.uk

***Abstract*—In this study, we take on one of the most common challenges in facial recognition, *i.e.* reducing the False Positives in the recognition phase, through studying performance of a standard Deep Learning Convolutional network in a real-life, real-time, and large-scale identity surveillance application. This application involved designing a queue management system that uses facial recognition, for an airport in the UK. Our approach was to capture the faces of passengers as they enter through Boarding Pass Gates (BPG) and as they exit Security Gates (SG). Thereafter, we compare the faces captured, within a fifteen minute window, from BPG against the ones from SG. When there is a match, we are able to calculate the time that someone has spent inside the security area, using the capture time of matched face. We call this the security queue time. Like any other facial recognition application, we have to deal with reducing the number of false positives, *i.e.* incorrectly matched faces. In this application false positives are statistically** *rare events*. **That is, the same or similar pair of images is unlikely to occur in a foreseeable time. To deal with this problem, we utilized several approaches including applying a second layer of detection using the Dlib library [3] to improve the quality of the detected faces. Specifically, by taking advantage of Dlibs Facial Landmarks, we created a scoring system similar to Dlibs, to choose the best frontal pose from amongst all faces attributed to a single person. Our large-scale trials show that this approach does measurably reduce the rate of false positives in such systems.**

***Index Terms*—facial recognition, queue management system, Dlib library, facial landmarks**

I. Introduction

Face recognition has become one of the most researched applications in the field of computer vision, specifically over the past few years. The reason for attracting such attention, from several disciplines like machine learning and image processing, is due to face recognition's ever-growing application in various areas [4]. Applications include user-friendly systems, information security, law enforcement and surveillance and entertainment, to name but a few. Facial recognition can also be used in applications such as managing staff and employees or students. In this respect, in real scenarios, efficiency of a system can depend on the environment (controlled or uncontrolled), positioning and the quality of detection equipment and the volume of people. For example, in [9], authors address a potential replacement, of a biometric-based student attendance monitor, for one based on facial recognition. Speaking about QMS (Queue Management System), There has been numerous research carried out using more conventional methods mostly utilizing basic sensors and devices constituting the IoT (Internet of Things) [1], [2]. To the best of our knowledge, there are not so many published articles on using facial recognition for queue management, and most of the state-of-the-art systems, advertised by companies, do not have open source documentation.

Generally speaking, any generic face recognition system, including this research, consists of three common phases, namely face detection, feature extraction and face recognition. In early studies [6], [7], the focus was mainly on frontal faces and how to improve detection/recognition using different classifiers as well as trained neural networks. However, with rapidly growing usage of facial recognition based technologies, designing systems capable of capturing and recognizing a wider range of faces is inevitable. For such a system to work consistently, we need to tackle several challenges related to each of these phases. These challenges include different lighting, different poses and occlusion. There are numerous studies carried out to tackle the aforementioned challenges. For example, in [8], the authors consider image blur, pose variation and occlusion.

One of the most prominent challenges that is broadly universal and applies to both the detection and recognition phases is the issue of false positives. A false positive in face detection arises when a pattern/object is recognized as a face and it is not, or when an individual is mismatched for another person. The most recent example of errors of this type has been in the focus of major public attention [5].

All Artificial Intelligence systems, commercial or open-source, are expected to have a margin of error, including false positives. Several approaches exist to date that are aimed at tackling AI mistakes. Altering training data, improving design procedures [12], [13], [14], [15], AI knowledge transfer,

978-1-5386-6614-2/18/$31.00 ©2018 IEEE
DOI 10.1109/HPCC/SmartCity/DSS.2018.00260

transfer learning [16], [17], [18], and privileged learning [19] are amongst most popular tools. Other techniques to tackle errors invoke various concentration of measure ideas [20]. In the domain of face recognition, approaches to resolve/address the issue of false positives have been studied in [6], [21], [22]. Notwithstanding the value and practical relevance of these results, they all invoke AI training which, one way or another, implies additional "learning" or up-training of the existing AI to perform better over time.

In the current application, we were faced with a different set of assumptions. In this application errors were assumed to be rare and singular events without any expectations for repetition. This rules out learning in the conventional sense, and as such requires finding alternatives to current state-of-the-art. As a possible way to overcome the problem a combination of a dedicated real-time filtering mechanism, data pre-processing and analysis is proposed. The idea was tested in a large-scale trial in a real-life setting $24/7$ over a period of 3 months and involved many thousands of individuals. The proposed solution has been confirmed as an effective tool to resolve the problem. In what follows we present the findings in the context of an airport queue management system.

II. The Queue Management System

A. System overview

All UK airports are expected to ensure that the average passenger spends no longer than twelve minutes going through the security area. The current boarding gates can measure the number of passengers coming through the security area via passengers boarding passes. However there is no way of determining whether an individual passenger has left the security area. The current solution involves a member of staff manually keeping track of a small sample of passengers passing from the boarding gates to the security scanners and logging the time taken. Knowing about the length of these queues, as well as the number of passengers getting through the airport, would help the staff to manage their resources in an efficient way. For example, it gives them a clear idea of how many security stations should be working based on how large/small the queues are. Also, passengers would know about the time they can expect to spend inside the queues and therefore, they can manage their time inside the airport more efficiently. Hence, knowing the time taken for a passenger since entering boarding-pass gates until leaving security gates, in almost real time, would be very beneficial. However, the current approach to this specific problem is far from being efficient.

Our Queue Management System (QMS) consists of two components. The hardware that is in charge of detecting faces, the back-end server that processes the data streamed from the hardware unit and calculates the average, fastest and slowest security queue times. A front-end web-page that is served by the back-end server to display the aforementioned statistical data as well as producing historical reports. As mentioned before, security queue time is the time a passenger spends in queues for BPGs and SGs. In the rest, we describe the function of each component in more detail.

For the hardware, shown in Figure 1, we utilise two (or more) HD cameras, one for BPG and one for SG, connected to two *Litso Cognitive Processing Module* ^{TM} (CPM) (prototype) via two mini PC's.

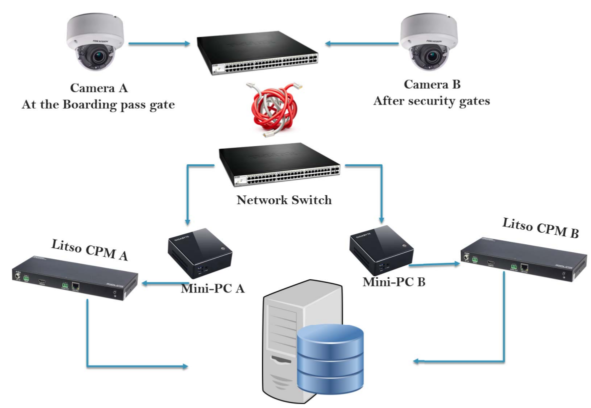

Fig. 1: Queue Management System

Each CPM is fed by a HDMI port and is able to detect and track as many as 200 faces per a frame within a 30 frame per second rate. More specifically, the CPM is capable of tracking people within several frames, capturing their faces and producing a unique track ID for each individual. Each track may contain several faces of an individual. The output of each CPM are XML files that contain information (metadata) such as frame ID, unique track ID for each individual, the time-stamp and coordinates of the face and finally $224 * 224$ Bitmap thumbnails showing the head and shoulders and/or face. The XML files are then received by the back-end server for processing. The back-end includes four different sections; parsing, image pre-processing, feature extraction and the data base. In the parsing section, we extract the encoded thumbnail together with all other metadata. Next, in the pre-processing unit each thumbnail is encoded to JPG format and saved to the local hard disk before passing through the Dlib library to be scored (face score). Then, the thumbnail will pass through a pre-trained Convolutional Neural Network (CNN), VGG-16 [23], for feature extraction. Finally, features of the thumbnail together with all related metadata will be stored in the database.

B. Face identification metrics

With the database being populated, the server makes a query to the database and takes out all tracks with face score above a defined threshold and recorded within the last fifteen minutes. Soon after, two separate (feature) lists of all these tracks are created, one for tracks from BPGs and another for the ones from SGs. The feature data in each list is then centered and normalised. We also reduce the dimension of each feature vector by applying Principal Component Analysis via Singular Vector Decomposition. Thereafter, these two lists are compared against each other and all comparisons are scored. Our scoring mechanism was based on the inner product-induced similarity measure.

More specifically, let $\boldsymbol{x}_i \in \mathbb{R}^n$ and $\boldsymbol{x}_j \in \mathbb{R}^n$ be the feature vectors of two faces captured form BPG and SG, respectively. The generated score for these two was defined as

$$\left\langle \frac{\boldsymbol{x}_i - \overline{\boldsymbol{x}}}{\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}\|}, \frac{\boldsymbol{x}_j - \overline{\boldsymbol{x}}}{\|\boldsymbol{x}_j - \overline{\boldsymbol{x}}\|} \right\rangle, \tag{1}$$

where $\overline{\boldsymbol{x}}$ is the (empirical) mean of the set. If the score exceeds a certain level then a match is reported in the system. The advantage of this similarity measure is at two-fold. First, it allows a simple and easy interpretation as a mere correlation coefficient between facial features. Second, the behavior of this measure is easy to predict and understand. The latter property is due to the Stochastic Separation Theorem [10] of which the statement is provided below (further details and generalizations can be found in [11]).

Since we are interested in assessing similarity between "random" faces let us first define what does such randomness mean in the context of this work. We assume that elementary features in a face can be modeled by independent and bounded random variables $X_1, \dots, X_n$ ($0 \le X_i \le 1$) with expectations $\overline{X}_i$ and variances $\sigma_i^2 > \sigma_0^2 > 0$. The independence assumption is not too restrictive since one can always run ICA or PCA prior to any comparisons. Let $\overline{\boldsymbol{x}}$ be a vector with coordinates $\overline{X}_i$, and let

$$R_0^2 = \sum_{i=1}^{n} \sigma_i^2.$$

The following result holds

Theorem 1: Let $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\}$ be i.i.d. random points from the product distribution in a unit cube, $0 < \delta < 2/3$. Then

$$\begin{aligned} &\mathbf{P}\left(1 - \delta \le \frac{\|\boldsymbol{x}_j - \overline{\boldsymbol{x}}\|^2}{R_0^2} \le 1 + \delta \text{ and} \right.\\ &\left. \left\langle \frac{\boldsymbol{x}_i - \overline{\boldsymbol{x}}}{R_0}, \frac{\boldsymbol{x}_M - \overline{\boldsymbol{x}}}{\|\boldsymbol{x}_M - \overline{\boldsymbol{x}}\|} \right\rangle < \sqrt{1-\delta} \text{ for all } i, j,\ i \neq M \right) \\ &\ge 1 - 2Me^{\left(-\frac{2\delta^2 R_0^4}{n}\right)} - (M-1)e^{\left(-\frac{2R_0^4(2-3\delta)^2}{n}\right)}; \end{aligned} \tag{2}$$

$$\begin{aligned} &\mathbf{P}\left(1 - \delta \le \frac{\|\boldsymbol{x}_j - \overline{\boldsymbol{x}}\|^2}{R_0^2} \le 1 + \delta \text{ and} \right.\\ &\left. \left\langle \frac{\boldsymbol{x}_i - \overline{\boldsymbol{x}}}{R_0}, \frac{\boldsymbol{x}_j - \overline{\boldsymbol{x}}}{\|\boldsymbol{x}_j - \overline{\boldsymbol{x}}\|} \right\rangle < \sqrt{1-\delta} \text{ for all } i, j,\ i \neq j \right) \\ &\ge 1 - 2Me^{\left(-\frac{2\delta^2 R_0^4}{n}\right)} - M(M-1)e^{\left(-\frac{2R_0^4(2-3\delta)^2}{n}\right)}. \end{aligned} \tag{3}$$

Remark 1: Equation (2) in the statement suggests that, for exponentially large in n sets of faces and adequate threshold parameter $\sqrt{1-\delta}$, the probability that a spurious match occurs between an image chosen at random (i.e. element $\boldsymbol{x}_M$) and the rest of the images is negligibly small. Indeed, if $\sigma_0^2 = \min_i\{\sigma_i^2\} > 0$ then $R_0^2 > n\sigma_0^2$, and the exponentials in (2), (3) are asymptomatically vanishing to zero as n grows. Equation (3) reflects that the same property holds for all elements in the sample.

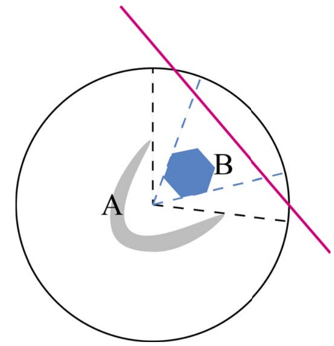

Fig. 2: Linear separability via projection onto a sphere. Sets A (shown as a shaded gray domain) and B (depicted as a blue filled hexagon) are not linearly separable in the original coordinates. Their projections onto the sphere, however, are separable (the separating hyperplane is shown in red).

Remark 2: Note that extreme selectivity property of the inner product in high dimension (2) persists when the test point $\boldsymbol{x}_M$ is slightly perturbed. This enables us to assign images with correlated features to a single cluster and hence determining an identity on this basis.

Remark 3: Apart from extreme selectivity in high dimension, normalization operation employed similarity metrics (1) has an additional regularization property, as illustrated with Fig. 2. Note, however, that elements of the set B in Fig. 2 are not guaranteed to be linearly separable even after the projection. Moreover, presence of clustering in the data may impede performance of similarity measure (1).

In addition to similarity measure (1) we experimented with standard L_2 and L_1 norms too. Remarkably though, on our current dataset the inner product (1) outperformed others in classifying matches.

When two faces are qualified as a match, based on its score, the server can find their associated time-stamps and calculate the queue time. These queue times are then sent to the front-end, to be displayed, see Fig. 3.

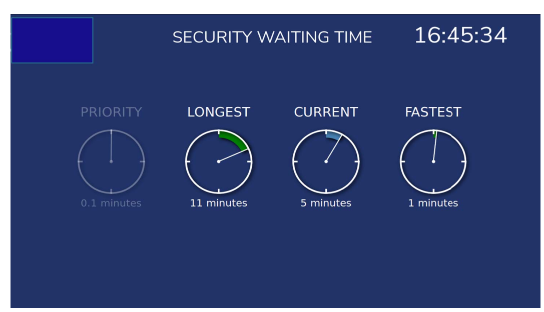

Fig. 3: Security queue time display in the airport

III. The Source of False Positives

In this case study, *i.e.* QMS in an airport, two different types of images are a major cause of false positive in the recognition stage. Firstly, although very rare, some patterns are detected by the CPM that are not faces (nonface as we call them), see Fig. 4. Secondly, some detected faces within a certain track have too acute/wide yaw/pitch angle and therefore are not suitable for the matching process. All in all, we need an additional image pre-processing stage to filter out patterns detected as faces and impose a penalty to the face score of thumbnails that are of low quality. However, in order to reduce the number of false positives some human analysis is inevitable.

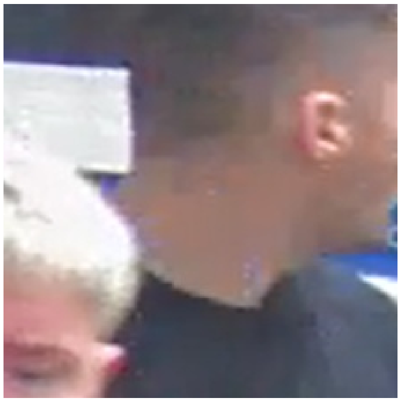

Fig. 4: An example of a nonface

IV. Dealing with One-Off, Rare False Positives

A. Our Approach

Our approach to minimising the number of false positives involved making changes within the image pre-processing section. These changes were made during different stages of the QMS trial at the airport, as two Proof of Concept (PoC) trials. In the rest of this paper we will describe the improvements made to the system. These improvements occurred over several stages before, during and after the two PoC trials.

B. Data pre-processing

The data has been initially split into a training set and a testing set, with the training set being a sample of feature vectors already processed by the system.

1) *Data Centering.* All measured feature vectors are centralized by subtracting empirical mean $\overline{\boldsymbol{x}}$ (determined from the training set) from each data point $\boldsymbol{x}_i$.
2) *Regularization.* Training data set has been subjected to PCA, and an effective data dimensionality is determined in accordance to the standard Kaiser-Guttman criterion. All measured feature vectors are then projected onto the relevant principal components.
3) *Projection.* Regularized data is projected onto the unit sphere.
4) *Clustering (optional).* For the regularized training data set (step 2 of the pre-processing), determine presence of any clustering structure. For each cluster, perform steps 1–3.

C. Before the First PoC Trial

Our system was fully designed before the first PoC trial. Having tested the system using footage from the airport, our first issue is concerned with the CNN we utilise. Our first satisfactory implementation of feature extraction (together with matching) was in MATLAB using VGG-face. However, after moving to Python, we were faced with different versions of VGG-face that were implemented using various platforms, such as Theano or Tensorflow, and sometimes various wrappers, such as Keras or Pytorch. Here, the main challenge was with different versions producing different results, which were often not as accurate as the MATLAB version. We tested several of these implementations to find a version that replicated the MATLAB results satisfactorily. After applying PCA on the (centered and normalised) data, we use an inner product as the distance between any two features. For reducing false positives at this stage, we found that head and shoulder thumbnails are not the best choice of thumbnail to work with as the environment around the face, including the clothes, adds noise to the feature, such as in Fig. 5. Because of this, we decided to progress using face-only thumbnails. Thereafter, we observed another issue in the false detection of patterns that are not faces. To overcome this, we started a secondary detection round, after the CPM detection, using the Dlib library. Dlib can recognize thumbnails which contain faces. At this stage the aim was to keep the rate of false positives at less than 30% of all matches.

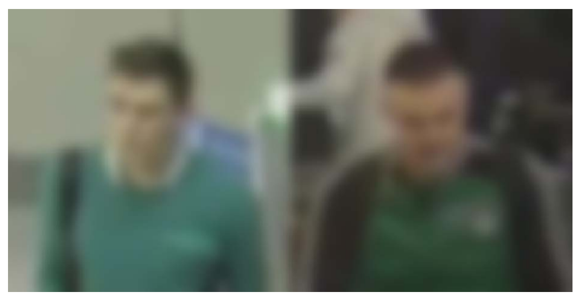

Fig. 5: An example of head and shoulder false positive

D. Observations After the First PoC trial

We trialled the system for three weeks in the airport. After analyzing the data collected from this first PoC, we then began to improve on the system. The first challenge we encountered was in the positioning of the SG camera, as people did not approach the camera face-on, as shown in Fig. 8. We changed the camera position to rectify this, as shown in Fig. 9), so images were captured in a frontal pose. We discovered that the direction of faces we received from both cameras affected the amount of false positives obtained in the system. In other words, Dlib face detection was not enough to lower the number of false positives.

At this point, we decided to include additional metadata to each thumbnail, in the form of the face score. Dlib library has the ability to project 68 points (see Fig. 6) on a face (we

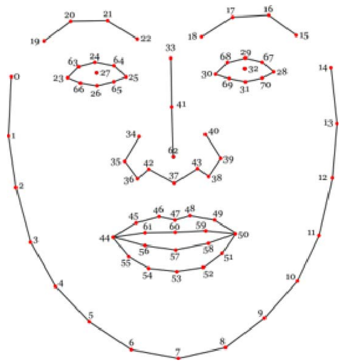

Fig. 6: Dlib 68 facial landmarks

switched to the 5 points version later on). Using this ability, we defined a scoring system based on the coordinates of 6 of these points that are the left (right) corner of left (right) eye, nose, tip of nose and tip of chin, as shown in Fig. 7. In practice these 6 point are enough to for calculating yaw, pitch and tilt of a face and generate a face score that tells us how frontal a face is. the score range is between 0 to 1.5.

Fig. 7: Six points we use for face score

A penalty is imposed on the score, proportionate to any changes in these values. We then pick a face from each track based on their score. In another words, among all captured faces of an individual, we pick the one that is closest to the frontal pose according to the new scoring system. After applying this selection process, we reduced the number of false positives from 30% to 10% for all matches.

E. Observations After the Second PoC trial

After the changes mentioned previously were made, the second PoC trial was run. After three months, we are able to analyse the additional data collected during this period. The number of false positives encountered reduced to 10% as we expected. By changing the position of the SG camera, the captured images and the matches made were much improved. When observing a number of false positives removed from the collected data, we discovered another source of false positives. Here, we discovered some anomalies in faces which

Fig. 8: Camera position in the first PoC

Fig. 9: Camera position in the second PoC

Fig. 10: Dlib miss-detections

had passed the scoring test with a substantially high score. These included faces in a side pose with only one eye visible, as shown in Fig. 10. To rectify this problem, we ran several experiments on thumbnails that were passed through the Dlib filter. One of the conventional ways to decrease computation cost is to reformat the thumbnail to grayscale prior to applying Dlib detection. However, based on our experience this would change the quality of Dlib landmark detection and fail the scoring as a result. By removing the grayscale reformatting, we resolved the issue to some extent. However, we still had to deal with side-posed faces that were mistakenly categorized as frontal. One feature that could be noticed about most thumbnails of this kind is that the nose tip was usually close to the right/left border of the picture. Therefore, by forcing the nose tip to be inside a confidence range, we could reduce these anomalies to a significant extent.

V. Analysis

A. Effect of Dlib scoring

The scoring system is a result of passing all thumbnails through the Dlib face detection. In case of failing such a test, *i.e.* no face is detected in the thumbnail, we score it zero. When a thumbnail passes the detection test, we then run the second test which checks whether or not the nose tip is inside the confidence range. If positive, we score the thumbnail according to the scoring system mentioned previously. If negative, the score again will be zero. Our analysis is based on a random collection of samples of data during a busy period at the airport. This data-set includes 28000 images of 500 individuals. We ran three different tests on the data-set. In the first experiment, we ran the recognition without using any filters, *i.e.* including all thumbnails with a face score greater than or equal to zero. For the second test we only considered thumbnails with a reaslonably high score of > 0.9. Finally our third analysis was based on thumbnails after applying all filters, *i.e.* the nose tip confidence range together with a high face score.

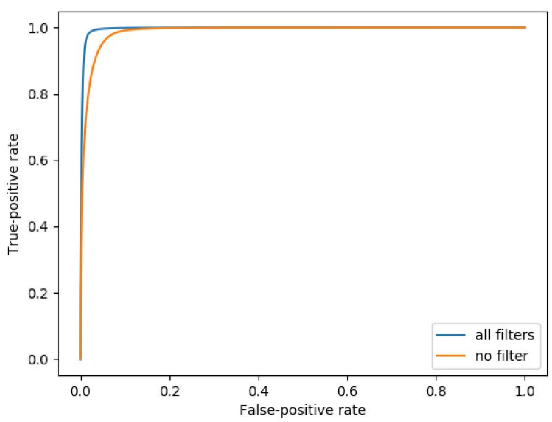

Fig. 11: The ROC curves after and before applying all filters

In Fig. 11, the blue line represents the ROC curve when applying Dlib, high score filter as well as nose tip confidence range filter and the orange line shows the ROC curve after switching off all filters. As you can see, applying our scoring system together with nose tip confidence range would improve the ROC curve.

In Fig. 12, the blue line is the ROC curve having applied the high score filter and the orange line is the curve after switching off all filters. Here, we observed a slight improvement when using high score filter.

B. Accounting for clustering structure in the data

Data pre-processing used in the experiments summarized in the previous section did not include the optional clustering step. Running k-means clustering algorithm with $k = 5$ on the training set, consists of 28000 images of 500 individuals, revealed marked clustering structure in the data. Fig. 13 shows performance of the original system on each of the identified clusters (after data pre-processing with the optional clustering step). No filtering was applied at this stage. Observe presence of a single cluster dragging the overall performance down.

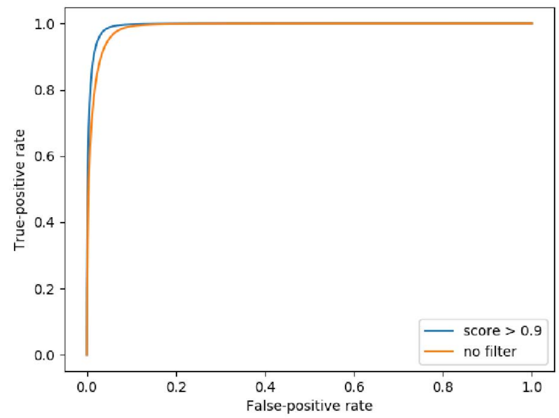

Fig. 12: The ROC curves after and before applying high score filter

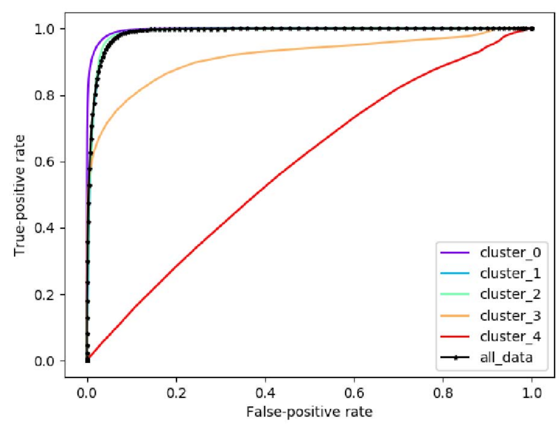

Fig. 13: The ROC curves for different clusters in the data (no filtering applied).

To see if and how the picture changes after Dlib filtering we repeated the procedure on the training set that has been subjected to high score filter. The results are shown in Fig. 14. The figure suggests that the system performance can be further improved if an additional clustering step is introduced into the data pre-processing. This indicates a possibility for creation of an adaptive unsupervised algorithm determining various comparison metrics further improving the overall matching performance. Furthermore, applying the high score filter helped to form almost identical sized clusters whereas in Fig. 13, cluster one includes 17000 data points whilst cluster 4 includes only 270.

VI. Conclusion

In this work, through a large-scale trial of a queue management system in a real-life settings involving thousands

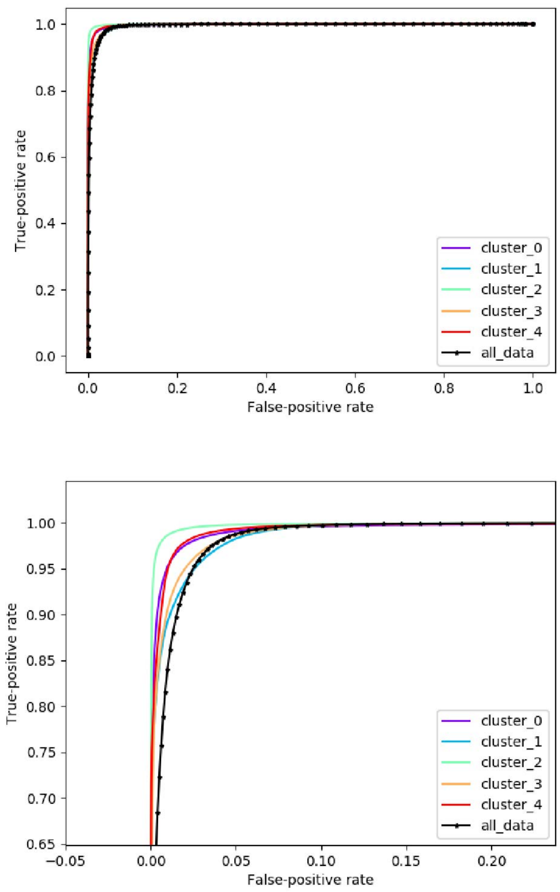

Fig. 14: The ROC curves for different clusters in the data (after filtering). Top panel shows the curves in the original scale. Bottom panel provides a zoomed-in view of the figure.

of real images of individuals, we assessed a possibility to reduce the level of false-positives via a dedicated filtering procedure. As a filtering rationale we used a face scoring system rejecting images that are too far off some nominal position. The trial, lasting $24/7$ over the time-span of more than 3 months revealed that the method works surprisingly well reducing the number of false positives from 30% to the figures below 10%. This has been achieved without any re-training and with an off-the-shelf convolutional VGG-type feature generator. Combining this method with on-line learning such as e.g. [20], should this be viable in a chosen setting, is likely to bring additional boosts in recognition accuracy.

ACKNOWLEDGMENT

The project was supported by a Knowledge Transfer Partnership KTP10522 between University of Leicester and Visual Management Systems Ltd. which is funded by Innovate UK, Visual Management Systems Ltd. and the Ministry of Education and Science of Russian Federation (Project No. 14.Y26.31.0022).

REFERENCES

[1] M. Ghazal, R. Hamouda and S. Ali, ”An IoT Smart Queue Management System with Real-Time Queue Tracking,” 2015 Fifth International Conference on e-Learning (econf), Manama, 2015, pp. 257-262.

[2] R. Klimek, ”Context-aware and pro-active queue management systems in intelligent environments,” 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, 2017, pp. 1077-1084.

[3] Davis E. King, Dlib-ml: A Machine Learning Toolkit, Journal of Machine Learning Research, 2009, 10. 1755-1758

[4] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. 2003. Face recognition: A literature survey. ACM Comput. Surv. 35, 4 (December 2003), 399-458.

[5] C. Foxx. Face recognition police tools “staggeringly inaccurate”, http://www.bbc.co.uk/news/technology-44089161, 2018.

[6] H. A. Rowley, S. Baluja and T. Kanade, ”Neural network-based face detection,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, Jan 1998.

[7] Human Face Detection in Visual Scenes, Henry A. Rowley and Baluja, Shumeet and Takeo Kanade, Advances in Neural Information Processing Systems 8, D. S. Touretzky and M. C. Mozer and M. E. Hasselmo, 875–881, 1996, MIT Press.

[8] C. Ding, D. Tao, Trunk-Branch Ensemble Convolutional Neural Networks for Video-based Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 10.1109/TPAMI.2017.2700390.

[9] S. Borra, K. J. Nazare, S.V. Raju, N. Dey, Attendance Recording System Using Partial Face Recognition Algorithm. 660. . 10.1007/978-3-319-44790-214.

[10] A. N. Gorban, I.Y. Tyukin, “Stochastic separation theorems.” Neural Netw. 94, pp. 255–259, 2017.

[11] A. N. Gorban, B. Grechuck, I.Y. Tyukin, “Augmented Artificial Intelligence.” arXiv preprint arXiv:1802.02172, 2018.

[12] A. Kuznetsova, S. Hwang, B. Rosenhahn, and L. Sigal, “Expanding object detectors horizon: Incremental learning framework for object detection in videos,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 28–36.

[13] I. Misra, A. Shrivastava, and M. Hebert, “Semi-supervised learning for object detectors from video,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3594–3602.

[14] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, “Learning object class detectors from weakly annotated video,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3282–3289.

[15] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, “Improving the robustness of deep neural networks via stability training,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] L. Pratt, “Discriminability-based transfer between neural networks,” *Advances in Neural Information Processing*, no. 5, pp. 204–211, 1992.

[17] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[18] T. Chen, I. Goodfellow, and J. Shlens, “Net2net: Accelerating learning via knowledge transfer,” *ICLR 2016*, 2015.

[19] V. Vapnik and R. Izmailov, “Knowledge transfer in svm and neural networks,” *Annals of Mathematics and Artificial Intelligence*, pp. 1–17, 2017.

[20] A. Gorban, R. Burton, I. Romanenko, and T. I., “One-trial correction of legacy ai systems and stochastic separation theorems, 2016.

[21] S. Yang, A. Wiliem and B. C. Lovell, ”To face or not to face: Towards reducing false positive of face detection,” 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), Palmerston North, 2016, pp. 1-6.

[22] P. Viola and M. Jones, ”Rapid object detection using a boosted cascade of simple features,” Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-511-I-518 vol.1.

[23] Simonyan, K. and Zisserman, A., Very Deep Convolutional Networks for Large-Scale Image Recognition, CoRR, abs/1409.1556, 2014.