PAPER • **OPEN ACCESS**

Face Validation Based Anomaly Detection Using Variational Autoencoder

To cite this article: B Zeno *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **618** 012011

IOP Conf. Series: Materials Science and Engineering **618** (2019) 012011 doi:10.1088/1757-899X/618/1/012011

Face Validation Based Anomaly Detection Using Variational Autoencoder

B Zeno[1], Yu Matveev[2], B Alkhatib[3]

[1,2] ITMO University
49 Kronverksky Pr., St. Petersburg, 197101,Russia
[3] Damascus University

[1]basilzeno@gmail.ruline, [2]matveev@mail.ifmo.ru, [3]t_balkhatib@svuonline.org

Abstract. In unconstrained facial images, large visual variations concerning pose, scale, the presence of occlusions, expressions and lighting usually cause difficulties in discriminating faces from the background accurately. As a result, some non-face regions are recognized as faces (false positive) and that influences the effectiveness of face detection algorithms which is characterized by low false positive (FP) rate, high detection rate and high speed of processing. In order to reduce these non-face regions, they are considered as anomalies and then try to detect them. In this paper, we propose an anomaly detection method using reconstruction error from variational autoencoder (VAE), which is a generative machine learning model. We train VAE to learn reconstructing faces that are close to its original input faces using FDDB dataset, then the difference between the original input face and the reconstructed output is measured to obtain the reconstruction error which can be used as an anomaly score. Consequently, the regions resulting from faces detection algorithm with high reconstruction error are defined as anomalies or false positives.

1. Introduction

Face detection is one of the mostly studied problems in vision, it has been actively researched for over two decades [1]. Face detection is considered as a problem of single-class object detection, and it is an important field of research in computer vision, because it forms a necessary first step for many face processing systems such as face recognition, face tracking, face verification and identification or facial expression analysis. Most famous of face detection algorithms have been based on cascading approach [2,3,4,5], Deformable Parts Models (DPMs) [6,7,8,9] and deep convolutional neural networks [10,11,12,13,14], detect faces within the rectangular search region specified by ROI (region of interest), which may contain many non-face regions or false positives that must be rejected.

This work was financially supported by Government of Russian Federation (Grant 08-08).

Many approaches have been proposed to reject non-face regions, that lead to improve the overall detection precision. Some of them based on eyes detection using support vector machine [15,16]. In [17,18,19] skin color methods are proposed to filter non-faces, where YCbCr color space, statistical data in different color spaces and some morphologic operations are used to achieve that.

The nature of non-face data makes its detection very challenging, where non-face data is far greater than that of face and the distribution of non-face pattern is very widespread. Moreover, non-face data are rare and, in most cases, non-existent. This makes the non-face detection a semi-supervised learning problem.

In this study, we try to solve this problem using generative machine learning methods, which recently have become popular due to their ability to model input data distributions and generate new examples from those distributions.
In section II, we briefly present some background about the anomaly detection and generative learning models, and in section III, we present the proposed anomaly detection method. Implementation is presented in section IV. Then we present the conclusion.

2. Ease of Use BACKGROUND

2.1. Anomaly Detection

Anomaly detection refers to the problem of finding patterns in data significantly deviated from expected normal behaviour. It is applied in many areas such as credit card fraud detection, network intrusion detection, sensor network fault detection, medical diagnosis and video surveillance [21].
Many anomaly detection algorithms have been proposed, and they can be further categorized as generative or discriminative approaches. A generative approach such as a Gaussian-mixture model, Variation Autoencoder, or Adversarial Autoencoders (e.g., [22]) builds a model solely based on normal training data and evaluates each testing case to see how well it fits the model. A discriminative approach (e.g., [23]), on the other hand, utilizes the data from normal and abnormal classes to create an explicit decision boundary separating them. The output of an anomaly detection algorithm can be one of two types:

- Anomaly scores: It is computed by evaluating the quality of the fit between the data point and the model. In other words, it quantifies the level of anomaly of each data point.
- Binary labels: A second type of output is a binary label indicating whether a data point is an abnormal or not. Although some algorithms might directly return binary labels, anomaly scores can also be converted into binary labels. This is typically achieved by imposing thresholds on anomaly scores, and the threshold is chosen based on the statistical distribution of the scores. A binary labeling contains less information than a scoring mechanism, but it is the final result that is often needed for decision making in practical applications.

In [22] anomalous event detection approach in images was proposed. It is Adversarial Autoencoders method, which is based on autoencoders in combination with Generative Adversarial Networks. The adversarial error of the learned autoencoder was used as anomaly score, since it is low for regular events and high for irregular events, such as walking on grass, bike, cart, etc. In [24] an anomaly detection method using variational autoencoders (VAE) was proposed. It used the reconstruction probability as anomaly score. Experimental results show that this method outperformed autoencoder based and PCA based methods.

2.2. Variational Autoencoder

Variational Autoencoder (VAE) [27] is generative model that belongs to the field of representation learning in Artificial Intelligence where an input is mapped into hidden representations. In a VAE we have examples X that are distributed according to some unknown distribution Pgen(X), which is specified over latent variables, and the goal is to learn a model P, such that P is as similar as possible to Pgen . Then data is generated or reconstructed from input by mapping the latent variables through a non-linear function implemented by a neural network. In the VAE, the highest layer of the directed graphical model z is treated as the latent variable where the generative process starts. g(z) represents the complex process of data generation that results in the data x, which is modeled in the structure of a neural network. The objective function of a VAE is the variational lower bound of the marginal likelihood of data, since the marginal likelihood is intractable. The marginal likelihood is the sum over the marginal likelihood of individual data points:

$$\log p_\theta(x^{(1)},\ldots,x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)}) \tag{1}$$

where the marginal likelihood of individual data points can be rewritten as follows.

$$\log p_\theta(x^{(i)}) = D_{KL}(q_\phi(z \mid x) \,\|\, p_\theta(z)) + L(\theta, \phi; x^{(i)}) \tag{2}$$

$q_\phi(z|x)$ is the approximate posterior and $p_\theta(z)$ is the prior distribution of the latent variable z. The first term of the right-hand side of equation (2) is the KL divergence of the approximate posterior and the prior. The second term of the right-hand side of equation (2) is the variational lower bound on the marginal likelihood of the data point i. Since the KL divergence term is always bigger than 0, equation (2) can be rewritten as follows.

$$\log p_\theta(x^{(i)}) = -D_{KL}(q_\phi(z|x) \| p_\theta(z)) + E_{q_\phi(z|x^{(i)})}[\log p_\theta(x|z)] \quad (3)$$

$p_\theta(x|z)$ is the likelihood of the data x given the latent variable z. The first term of equation (3) is the KL divergence between the approximate posterior and the prior of the latent variable z. This term forces the posterior distribution to be similar to the prior distribution, working as a regularization term. The second term of equation (3) can be understood in terms of the reconstruction of x through the posterior distribution $q_\phi(z|x)$ and the likelihood $p_\theta(x|z)$. The VAE models the parameters of the approximate posterior $q_\phi(z|x)$ and by using a neural network. This is where the VAE can relate to the autoencoder. As shown in figure 1.

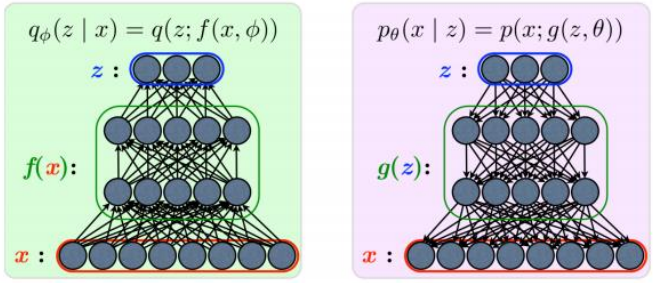

Figure 1. Neural networks in variational autoencoder

3. THE PROPOSED METHOD

We begin by presenting the problem statement, then the proposed method to detect anomalies.

3.1. Problem statement

Let $X = \{x_1, x_2, ..., x_n\}, x_i \in X$, set of detected faces from a face detection algorithm, which contains false-positive detections or non-face regions that effects on False Alarm Rate. To decrease this rate, we consider the non-faces as abnormalities and faces as normal data, then we apply anomaly detection algorithm f after face detection. See figure 2.

$$f(x_i) = \begin{cases} face(normal) & score \geq \varphi \\ non-face(abnormal) & score < \varphi \end{cases} \quad (4)$$

where, f anomaly detection algorithm, x_i: detected face, score: anomaly score, φ: the anomaly threshold, determined empirically to achieve best false alarm rate.

3.2. The proposed method

Our system consists of two components as shown in figure 3, an autoencoder network including an encoder network E(x) and decoder network D(z), and an anomaly detection. An input image x is encoded as a latent vector z = E(x), which will be decoded back to image space $\bar{x} = D(z)$. In order to train a VAE, we need two losses, one is KL divergence, which is used to make sure that the latent vector z is an independent unit. The other is reconstruction loss, which is that directly comparing the input image and the generated image in the pixel space. We choose mean square error (MSE) as reconstruction loss and we consider the reconstruction loss as anomaly score. Finally,

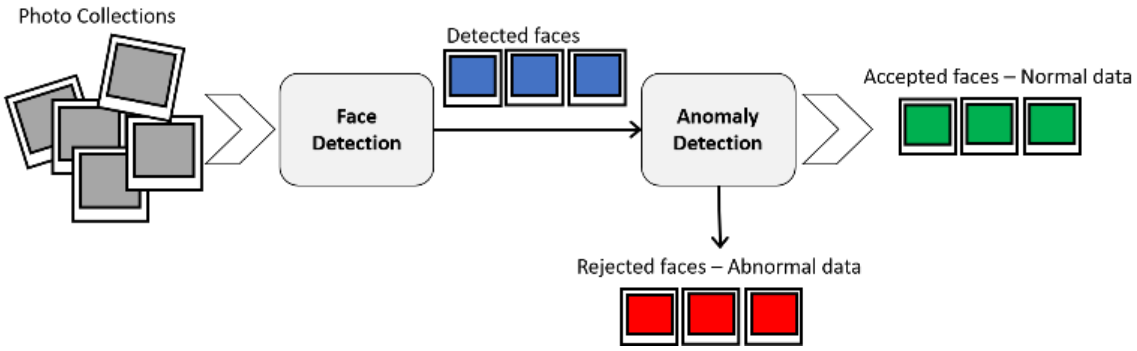

Figure 2. Anomaly detection system

the VAE model can be trained by optimizing the sum of the reconstruction loss (L_{rec}) and KL divergence loss (L_{kl}) by stochastic gradient descent.

$$\text{anomaly score} = L_{rec} = -E_{q_\phi(z|x^{(i)})}[\log p_\theta(x|z)] \quad (5)$$

$$L_{kl} = D_{KL}(q_\phi(z|x) \| p_\theta(z)) \quad (6)$$

$$L_{vae} = L_{kl} + L_{rec} \quad (7)$$

Both encoder and decoder network are based on deep CNN like AlexNet [28] and VGGNet [29]. We used the architecture that proposed in [30].

The algorithm of the proposed method is in Algorithm 1. The anomaly detection task is conducted a semi-supervised framework, using only data of normal instances for training the VAE. We use the autoencoder architecture that is proposed in [30].

Algorithm 1 Variational autoencoder based anomaly detection algorithm

INPUT: Normal dataset X, Anomalous dataset $x^{(i)} i = 1,..., N$, threshold α

OUTPUT: reconstruction loss L_{rec}

$\phi, \theta \leftarrow$ train a variational autoencoder using the normal dataset X

for i = 1 **to** N **do**

$\mu_z(i), \sigma_z(i) = f_\theta(z|x^{(i)})$

draw one sample $\bar{x}$ from z ~ N ($\mu_z(i), \sigma_z(i)$)

$L_{rec}{}^{(i)} = MSE(x^{(i)}, \bar{x}^{(i)}) = \frac{1}{L}\sum_{i=1}^{L}(x^{(i)} - \bar{x}^{(i)})^2$,

L: number of pixels.

if $L_{rec}{}^{(i)} >$ α **then**

$x^{(i)}$ is an anomaly

else

$x^{(i)}$ is not an anomaly

end if

end for

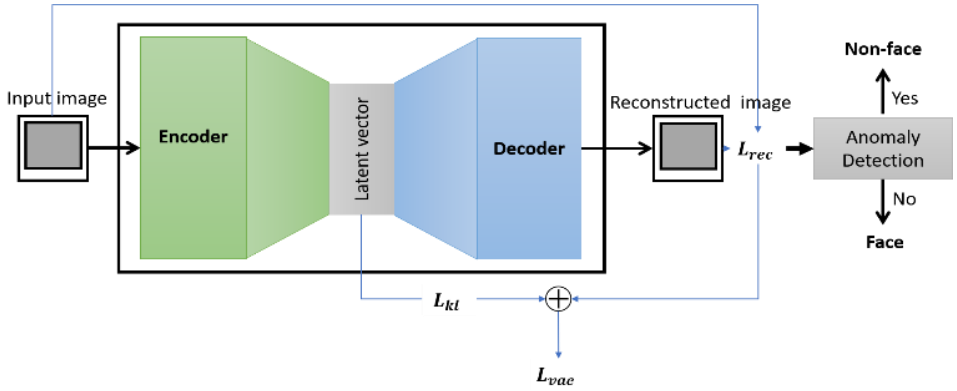

Figure 3. Variational autoencoder based anomaly detection model

4. IMPLEMENTATION

We conduct two experiments, in the first one, our system is trained on LWFcrop dataset, and the second one on Pubfig83-lfw dataset, while FDDB dataset is chosen for testing.

4.1. Baseline Methods

Viola Jones Face Detector: Viola and Jones proposed an efficient cascade framework [25] that rapidly discards negatives and spends more time in positive candidates. The cascade framework is one of the most successful practical products of vision research. This algorithm minimizes computation time while achieving high detection accuracy. It is 15 times quicker than any technique at the time of release with 95% accuracy at around 17 fps. Viola Jones algorithm has four stages: Haar feature selection, creation of an integral image, Adaboost training, and cascaded classification. We use this face detector as a first step before anomaly detection.

Hog-SVM Classifier using MMOD method as an anomaly detection algorithm: To detect abnormalities in set of detected faces that are obtained from the Haar-Cascade Face Detector, we use a fast detector based on Maximum Margin Object Detection (MMOD) method [26]. This method does not perform any sub-sampling, but instead optimizes over all candidate windows. The scoring function that used in MMOD's parameters outputs anomaly score, which is converted to anomaly labels by thresholding empirically.

4.2. Datasets

LFWcrop: It is a cropped version of the Labeled Faces in the Wild (LFW) [31], contains 3694 colored cropped faces with size 64x64 pixels.

Pubfig83-lfw: It is a combination of PubFig83 and the LFW dataset for open-universe face identification. Faces are pre-aligned by the eye positions as reported by PittPatt-reported fiducials [32].

We evaluate our system on the Face Detection Data Set and Benchmark (FDDB) challenge [33]. This challenging dataset contains images of human faces in multiple poses captured in indoor and outdoor settings. There are 2845 images with a total of 5171 faces contain a wide range of difficulties including occlusions, difficult poses, and low resolution and out-of-focus faces.

a) Reconstructed faces (Right) of faces from LFWcrop

b) Reconstructed faces (Right) of faces from Pubfig83-lfw

Figure 4. Examples of reconstruction faces from validation dataset

We preprocess the images by applying standardization technique as the following:
new_image = (image – db_mean) / db_stddev

4.3. Hardware Specifications
We used virtual machine instance in Google Cloud with the following details: operating system is Ubuntu 16.04, 6 cores CPU, 10GB RAM and GPU is Tesla K80, 12GB.

4.4. Training on LFWCrop and Pubfig83-lfw
In the first experiment, VAE model is trained on LFWcrop with 2000 images for training and 500 images for validation, while in the second experiment, is trained on Pubfig83-lfw with 5000 images and 2000 images for validation. In both experiments, we scale images to 64x64 and use batch size equals to 16. The size of latent vector z is set to 100 also. Figure 4. shows examples of reconstructed faces from validation datasets.

4.5. Experimental Results on FDDB

After training the models for 5000 epochs, we test them on FDDB dataset. figure 5. shows examples of reconstructed faces. Table. 1 presents the evaluation results of our VAE model-based anomaly detection in comparison with results of Hog-SVM results. We note that the first experiment, when the VAE model is trained on LFWcrop dataset gives better results than the second experiment when trained on Pubfig83-lfw dataset. In addition, the performance of our system achieves the performance of Hog-SVM anomaly detection model. Since, the false positives decreased from 421 to 128 and 135 for Hog-SVM classifier and out method respectively.

c) Reconstructed faces (Right) of faces from LFWcrop after 1st experiment

d) Reconstructed faces (Right) of faces from FDDB after 2nd experiment

Figure 5. Examples of reconstruction faces from FDDB dataset

Table 1. Evaluation Results of Proposed Method

Face detection step			
	Min window size	***Recall***	***FP***
Haar-Cascading OpenCV	80 x 80	64%	421
Anomaly detection step			
	Threshold φ	***Recall***	***FP***
Hog-SVM	>= 1	29 %	86
Hog-SVM	>= 0	55 %	121
Hog-SVM	>= -1	**55 %**	**128**
Ours (Exp1)	<=3500,000	34 %	94
Ours (Exp1)	<=4500,000	**54 %**	**135**
Ours (Exp2)	<=2500,000	18 %	102
Ours (Exp2)	<=3500,000	41 %	188

Acknowledgments
This work was financially supported by the Government of the Russian Federation (Grant 08-08).

References

[1] Zhang and Z. Zhang. A survey of recent advances in face detection. Microsoft Research, Technical Report 2010, no. MSR-TR-2010-66, 1 p.
[2] Viola P and Jones MJ. “Robust real-time face detection”, Int J Comp Vision, 2004, vol. 57, no. 2, pp. 137–154.
[3] H. Yang, X.Wang. “Cascade classifier for face detection”, Journal of Algorithms & Computational Technology, 2016, vol. 10, no. 3, pp. 187 – 197.
[4] R. Mohan, N. Sudha, “Fast face detection using boosted eigenfaces”, IEEE Symposium on Industrial Electronics & Applications, 2009, vol. 2, pp. 1002-1006.
[5] Chen , S. Ren, Y. Wei, X. Cao, J. Sun. “Joint cascade face detection and alignment”, In: Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 109–122.
[6] X. Zhu and D. Ramanan. “Face detection, pose estimation, and landmark localization in the wild”, Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2879–2886.
[7] J. Yan, X. Zhang, Z. Lei, and S. Z. Li. “Face detection by structural models”, Image and Vision Computing, 2014, vol. 32, no. 10, pp. 790–799.
[8] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. “Face detection without bells and whistles”, Lecture Notes in Computer Science, 2014, vol. 8692, pp. 720–735.
[9] T. Alafif, Z. Hailat, M. Aslan, and X. Chen. “On detecting partially occluded faces with pose variations, in proceedings of the 14th Inter- national Symposium on Pervasive Systems, Algorithms, and Networks (I-SPAN)”, IEEE Computer Society Conference Publishing Services (CPS), 2017.
[10] S. S. Farfade, M. J. Saberian, L. Li. Multi-view Face Detection Using Deep Convolutional Neural Networks. ICMR '15 Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015, vol. 2, pp. 643-650.
[11] H. Jiang, E. Learned-Miller. Face Detection with the Faster R-CNN. ArXiv e-prints:1606.03473. 2016.
[12] S. Sarkar, V. M. Patel, and R. Chellappa. Deep feature-based face detection on mobile devices. ArXiv e-prints, abs/1602.04868, 2016.
[13] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection, In Biometrics Theory, Applications and Systems (BTAS), 2015, pp 1–8.
[14] S. Yang, P. Luo, C. C. Loy, X. Tang. From facial parts responses to face detection: A deep learning approach. 2015, pp. 3676–3684.

[15] L. Nanni, A. Lumini, Sh. Brahnam. Ensemble of Face/Eye Detectors For Accurate Automatic Face Detection, International Journal of Latest Research in Science and Technology, 2015, vol. 4, no. 3, pp.8-18.
[16] L. Nanni, A. Lumini. Combining Face and Eye Detectors in a High- Performance Face-Detection System, 2012, vol. 19, no. 4, 3 p.
[17] J. Prinosil, J. Vlach. Face detection in image with complex background, IFIP — The International Federation for Information Processing, 2007, vol. 245, pp 533-544.
[18] Nefian. Georgia Tech face database, URL: http://www.anefian.com/research/face_reco.htm
[19] Md. Hafizur, J. Afrin. Human Face Detection in Color Images with Complex Background using Triangular Approach, Global Journal of Computer Science and Technology Graphics & Vision, 2013, vol. 13, no. 4. 2 p.
[20] F. Alizadeh, S. Nalousi, C. Savari. Face Detection in Color Images using Color Features of Skin, 2011, vol. 5, no. 4, pp. 368-369.
[21] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3):15, 2009.
[22] Dimokranitou A. Adversarial Autoencoders For Anomalous Event Detection In Images. Thesis Submitted to the Faculty of Purdue University. 2017.
[23] Qian J., Root J., Saligrama V., and Chen Y. A Rank-SVM Approach to Anomaly Detection, 2014. https://arxiv.org/abs/1405.0530.
[24] An, J. and Cho, S. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. 2015.
[25] Viola, P and Jones, M.J. Robust real-time face detection. Int J Comp Vision, 2004, vol. 57, no. 2, pp. 137–154.
[26] E. King. Max-Margin Object Detection. in Computer Vision and Pattern Recognition, 2015, arXiv:1502.00046.
[27] Kingma, M. Welling. Auto-encoding variational bayes. 12 2013.
[28] Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014
[30] X. Hou, L. Shen, K. Sun, G. Qiu. Deep Feature Consistent Variational Autoencoder. 2016.
[31] B. Huang, M. Marwan, L. Honglak, and L. Erik. Learning to align from scratch. In NIPS, 2012.
[32] http://www.briancbecker.com/blog/research/pubfig83-lfw-dataset/
[33] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.