Research Article

Liveness Control in Face Recognition With Deep Learning Methods

Nader Ebrahimpour[1*], Mustafa Arda Ayden[2], Banu Altay[3]

[1] Papilon Savunma, 0000-0003-1189-3054, naderebrahimpour@papilon.com.tr
[2] Papilon Savunma, 0000-0003-1112-0887, ardaayden@papilon.com.tr
[3] Papilon Savunma, 0000-0002-4788-5315, banualtay@papilon.com.tr

[*] Correspondence: naderebrahimpour@papilon.com.tr;

(First received March 19, 2022 and in final form June 06, 2022)

Reference: Ebrahimpour, N., Ayden, M. A. ., & Altay , B. . Liveness control in face recognition with deep learning methods. *The European Journal of Research and Development*, *2*(2), 92–101.

Abstract

Today, automatic identification of individuals from biometric features is widely used in identification and authentication, security, and monitoring applications. Since facial recognition is a more user-friendly and comfortable method than other biometric methods, it has grown rapidly in recent years. However, most facial recognition systems are vulnerable to spoofing attacks. Therefore, face liveness detection (FLD) methods are of great importance. On the other hand, unlike traditional methods, deep learning techniques promise to significantly increase the accuracy of facial liveness detection systems and eliminate the difficulties of the real-world implementation of these systems. Therefore, in this paper, the application of some deep learning models to detect face liveness is reviewed and compared with each other.

Keywords: Liveness Detection, Face Recognition, Deep Learning

1. Introduction

A face recognition system is a type of biometric system that uses intelligent methods to identify or verify a person's identity based on the physiological characteristics of their face (Anand & Shah, 2016). Face recognition systems consist of image recognition and comparison algorithms. The basis of these algorithms is based on the analysis of features related to the size, shape and location of facial organs such as eyes, nose, and cheeks in individuals (Vazquez-Fernandez & Gonzalez-Jimenez, 2016).

Facial recognition technology is one of the most commonly used biometric methods. It offers contactless biometric analysis and does not require additional hardware other than easily accessible sensors (RGB camera, etc.) (Vazquez-Fernandez & Gonzalez-Jimenez, 2016). Therefore, this technology has been considered in industrial and scientific fields in recent years (Adjabi, Ouahabi, Benzaoui, & Taleb-Ahmed, 2020). Facial recognition systems have many applications, such as identifying criminals, verifying identity cards, passport and credit card holders, and biometric control in banks, stores, and airports (Adámek, Matýsek, & Neumann, 2015).

However, most known facial recognition systems are vulnerable to spoofing attacks. Today, attempts are made to deceive biometric face recognition systems by showing a fake face in front of the camera. For example, in a live demonstration during the International Biometrics Conference (ICB 2013), a woman with special make-up cheated on the face recognition system and hacked into the system (Boulkenafet, Komulainen, & Hadid, 2016). Many such examples illustrate the vulnerability of facial recognition systems to spoofing attacks. Therefore, to be protected against such deceptions, a face recognition system must be able to detect live and spoof face images and have the ability to distinguish these images from each other. In facial recognition systems, the usual attack methods fall into several categories. These categories are based on the type of image provided to deceive facial recognition systems. Image attacks with printed images, video attacks with a recorded video, and mask attacks using three-dimensional face models are the main types of attacks (Fatemifar, Arashloo, Awais, & Kittler, 2019). Face images obtained by these deception methods are defined as spoof face photos, and images obtained by taking a picture of a real face are defined as live face images.

Before the face recognition process starts, whether the face is real or fake is determined by the FLD system. Suspicious images are filtered and do not enter the face recognition system. Thus, the security performance of a biometric system is improved with the help of the FLD system. FLD methods can be classified as hardware and software-based methods. In hardware-based methods, additional equipment such as depth and temperature sensors are used to obtain additional liveness information from the face in front of the camera. Therefore, hardware-based methods incur additional costs and make face recognition systems more difficult (Galbally, Marcel, & Fierrez, 2014). On the other hand, software-based methods evaluate whether a face is live or not based on the texture, structure, and visual quality elements in RGB images. These methods are preferred because they do not require additional equipment and are more cost-effective (Galbally et al., 2014).

Unlike the features determined by the system developers, the features obtained by deep feature learning techniques used in FLD systems can be more easily adapted to changes in external environments and different deception techniques and have a stable structure. Thus, such systems promise a dramatic increase in the accuracy of FLD systems, overcoming the key challenges faced by other systems. A wide variety of FLD methods and techniques have been used in industrial and academic studies to increase accuracy and reduce error. Early attempts at FLD systems relied on handcrafted features from raw face images. Examples of such traditional approaches are Local Binary Pattern (LBP) (Ojala, Pietikainen, & Harwood, 1994), Local Phase Quantization/LPQ (Ojansivu & Heikkilä, 2008), Histogram of Oriented Gradients/HOG (Freeman & Roth, 1995). After getting features from such methods, machine learning classifiers such as a Support Vector Machine (SVM) (Cristianini & Shawe-Taylor, 2000) are used in order to classify the inputs. However, these approaches can extract low-level features and they are not considered sufficient for current FLD systems. Recent achievements of deep learning in various fields of computer vision have paved the way for the use of deep neural network models for FLD. Deep learning models are the dominant approach in face detection systems and outperform traditional image processing algorithms. Such systems are being developed to extract features of the relevant face image using convolutional neural networks (CNNs) to detect fraudulent attacks (Sabaghi, Oghbaie, Hashemifard, & Akbari, 2021).

This paper examines and compares the performance of different models of convolutional neural networks for FLD systems.

2. Background

2.1.Convolutional Neural Networks

One of the most famous and successful deep learning architectures in machine learning is the convolutional neural network, introduced to the literature in the 1980s. However, in these years, due to the insufficiency of computing resources, the use of these networks in machine vision tasks took the 1990s. One of the first applications in this field is the LeNet network, which was introduced in 1998 to classify handwritten figures. Various architectures have been proposed for convolutional neural networks, consisting of three main layers: convolution, pooling, and fully connected. The convolution layer contains a set of filters (or kernels) whose parameters will be learned during training. For convolution, the filter is shifted across the height and width of the image, and the dot product between each element of the filter and the input is calculated. The pooling layer is used to reduce the computational load of convolutional layers. The purpose of placing this layer is to reduce the resolution of the desired feature map to obtain an invariant area. The fully connected layer is used for the classification and predicts the labels (Wu, 2017).

2.2.Convolutional Neural Network Models

In this section, some models of convolutional neural networks are reviewed.

2.2.1. AlexNet

AlexNet was introduced in 2012 by Alex Krishfsky. This network has 8 layers (five convolution and three fully connected layers), putting it in the shallow networks category. The structure of this network is such that it uses convolution layers of different sizes to extract features from the input images. Generally, the first two layers are convolution, followed by a max-pooling layer. The third, fourth, and fifth layers are convolution; after the fifth layer, there is a maximum pooling, then there are two fully connected layers. Finally, the softmax classification layer determines which label the given photo belongs to (Krizhevsky, Sutskever, & Hinton, 2012).

2.2.2. VGGNet

The VGG deep neural network has been proposed at the University of Oxford. This network is less complex than AlexNet as it reduces the number of parameters and is very popular because of this simplicity. The VGG network consists of two layers of convolution, followed by a max-pooling layer. In addition to sampling, this layer is also responsible for halving the feature size. Then the other two convolution layers and a max pool layer are placed similarly. Then there are three convolution layers and a max-pooling layer repeated two more times. Finally, there are two fully connected layers and an entire fully connected layer corresponding to the number of application classes (Simonyan & Zisserman, 2014).

2.2.3. ResNet

The residual neural network is one of the most attractive networks first proposed by several researchers at the Microsoft Research Institute. The main idea of the residual block in this network is that a convolution block processes the input, the result of these transformations produces a function, and the result of this function is added to the previous layer input to get the final output (He, Zhang, Ren, & Sun, 2016).

2.2.4. MobileNet

Among the various models of convolutional neural networks, the gap between small, high-speed networks that could be used in robotics, mini-computer boards, and mobile phones was felt. Accordingly, researchers created a new class of light convolutional networks with low parameters, high execution speed, and acceptable accuracy. One of the most prominent models of such networks is MobileNet neural network. Google researchers have proposed this network to design efficient, lightweight, fast and accurate networks (Howard et al., 2017).

2.2.5. DensNet

This model is a convolutional neural network that uses dense connections between layers via Dense Blocks, where all layers (with matching feature map dimensions) are directly interconnected. To preserve the feedforward nature of the network, each layer in the network receives additional inputs from all previous layers and transmits its feature maps to all subsequent layers (Huang, Liu, Van Der Maaten, & Weinberger, 2017).

2.2.6. EfficientNet

This model is a convolutional neural network architecture and scaling method that uniformly scales all depth/width/resolution dimensions. Unlike traditional practice that scales these factors on demand, the EfficientNet scaling method evenly scales network width, depth, and resolution with fixed scaling coefficients (Tan & Le, 2020).

3. Proposed Dataset and Methods for FLD

This section examines how to prepare data and use convolutional neural network models for FLD.

3.1.Dataset

In this study, selfie photos taken from the cameras of iPhone, Xiaomi and Samsung mobile phones were brought together to prepare the dataset and these images were labeled as original or live images. Later, the images above were displayed on the screens of different models of computers (eg Macbook PRO, HP, Monster) and the images displayed on the monitor were photographed with the cameras of Phone, Xiaomi and Samsung mobile phones. These photos have been lebeled as fake photos. In order to detect the attack of images printed on paper, live pictures were printed on paper, and photographs were taken again on paper and added to fake pictures. In order not to have problems in detecting the liveness of people with glasses, the photographs taken include pictures of faces with glasses. An example live and fake image can be found in the dataset shown in Figure 1. In this study, a data set was created with 13000 live face images and 13000 fack face images.

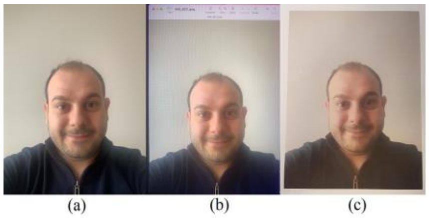

Figure 1 (a) live (b) spoof photographed from screen (c) spoof photographed from printed paper images.

3.2.Face Liveness Detection

The general steps of face liveness detection are shown in Figure 2. In this system, first of all, the places of the faces in the image were detected using the MTCNN (Zhang, Zhang, Li, & Qiao, 2016) face detection algorithm, and then the face regions are cropped from the picture. Facial cropped areas from the image with their labels are fed to the convolutional neural network model to train live and spoof images. And this process is continued until the model reached the desired validation accuracy threshold. The model's last layer is a Softmax layer used for classification. The probability of being live or spoofing an image is obtained through this layer. The face detection process is also carried out for the dataset used for testing purposes. After completing the model training, the test dataset is used to test the model. The dataset is randomly divided at the rates of 0.75, 0.15 and 0.15 for training, validation and testing subsets, respectively and the model training and testing processes are done using these sub-datasets.

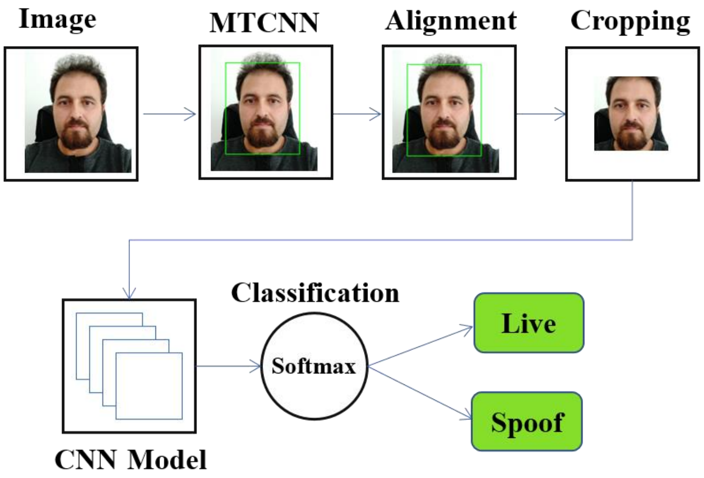

Figure 2 The steps of the face liveness detection system.

4. Results and Discussion

Keras (Silaparasetty, 2020) library based on TensorFlow (Chollet, 2018) open-source platform is used to implement the models. Keras is an open-source Python library that enables the fast implementation of deep learning models. In addition, a computer with 8 core Intel i7 CPU, 16 GB RAM, 256 GB SSD hard disk and NVIDIA GeForce GTX 1060 Ti graphics card was used for simulation. All data for training, evaluation and testing of the model is divided into three parts: training, evaluation and test data. The selection of training, evaluation and testing data from all data is completely random.

Accuracy, precision, and recall criteria were used to evaluate the models. These criteria are measured by equations (1) to (3) (Olson & Delen, 2008).

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

These equations mentioned TP, TN, FP, and FN values mean true-positive, true-negative, false-positive, and false-negative. Accuracy is defined as the percentage of correct predictions for test data. Recall measures the proportion of true positive labels correctly identified by the model. Precision is the quality of a machine learning model's positive prediction.

In this simulation, different models are trained under equal conditions. As the optimization algorithm in the experiment, Adam's optimization algorithm is used with a training rate of 0.0001, and model training is carried out with 30 training repetitions.

Tables 1 and 2 show the results based on recall and precision parameters, respectively. It has been observed that the EfficientNetB0 method has a better result than other methods when the recall and precision values of both classes are considered. In addition, the comparison of the mentioned models according to the accuracy parameter is shown in Figure 3. As can be seen from this figure, the models with the best accuracy for the application are EfficientNetB0, MobileNet, and DensNet169, respectively.

Table 1 Comparison of results by recall parameter.

Used Models	Classes	
	Spoof	Live
ResNet50	0.9981	0.9523
ResNet152	0.9321	0.9974
VggNet16	0.9573	0.9968
VggNet19	0.9820	0.9895
DensNet121	0.9956	0.9702
DensNet169	0.9884	0.9963
MobileNet	0.9949	0.9961
EfficientNetB0	0.9942	0.9972

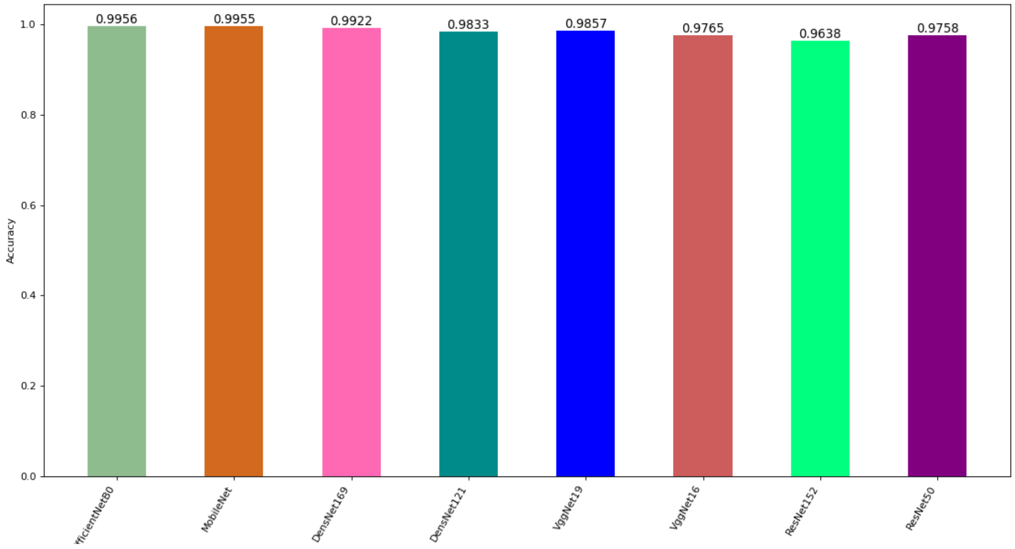

Figure 3 Comparison of results by accuracy parameter.

Table 2 Comparison of results by precision parameter.

Used Models	Classes	
	Spoof	Live
ResNet50	0.9569	0.9978
ResNet152	0.9974	0.9326
VggNet16	0.9969	0.9565
VggNet19	0.9900	0.9811
DensNet121	0.9726	0.9952
DensNet169	0.9964	0.9878
MobileNet	0.9963	0.9946
EfficientNetB0	0.9973	0.9938

5. Conclusions

In this study, a suitable dataset for the FLD system was created, and this dataset was used in the training of the CNN models as mentioned above. The outputs are shown in the tables and figures mentioned above. According to the obtained results, it has been revealed that the EfficientNetB0, MobileNet, and DensNet169 models are more efficient and have acceptable accuracy for this application, respectively.

References

[1] Adjabi, I., Ouahabi, A., Benzaoui, A., & Taleb-Ahmed, A. (2020). Past, present, and future of face recognition: A review. Electronics, 9(8), 1188.

[2] Adámek, M., Matýsek, M., & Neumann, P. (2015). Security of biometric systems. Procedia Engineering, 100, 169-176.

[3] Anand, B., & Shah, P. K. (2016). Face recognition using SURF features and SVM classifier. International Journal of Electronics Engineering Research, 8(1), 1-8.

4[] Boulkenafet, Z., Komulainen, J., & Hadid, A. (2016). Face spoofing detection using colour texture analysis. IEEE Transactions on Information Forensics and Security, 11(8), 1818-1830.

[5] Chollet, F. (2018). Keras: The python deep learning library. Astrophysics source code library, ascl: 1806.1022.

[6] Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods: Cambridge university press.

[7] Fatemifar, S., Arashloo, S. R., Awais, M., & Kittler, J. (2019). Spoofing attack detection by anomaly detection. Paper presented at the ICASSP 2019-2019 IEEE International Conference on Acoustics, [] Speech and Signal Processing (ICASSP).

[8] Freeman, W. T., & Roth, M. (1995). Orientation histograms for hand gesture recognition. Paper presented at the International workshop on automatic face and gesture recognition.

[9] Galbally, J., Marcel, S., & Fierrez, J. (2014). Biometric antispoofing methods: A survey in face recognition. IEEE Access, 2, 1530-1552.

[10] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[11] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[12] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[13] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[14] Ojala, T., Pietikainen, M., & Harwood, D. (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. Paper presented at the Proceedings of 12th international conference on pattern recognition.

[15] Ojansivu, V., & Heikkilä, J. (2008). Blur insensitive texture classification using local phase quantization. Paper presented at the International conference on image and signal processing.

[16] Olson, D. L., & Delen, D. (2008). Advanced data mining techniques: Springer Science & Business Media.

[17] Sabaghi, A., Oghbaie, M., Hashemifard, K., & Akbari, M. (2021). Deep Learning meets Liveness Detection: Recent Advancements and Challenges. arXiv preprint arXiv:2112.14796.

[18] Silaparasetty, N. (2020). The Tensorflow Machine Learning Library. In Machine Learning Concepts with Python and the Jupyter Notebook Environment (pp. 149-171): Springer.

[19] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[20] Tan, M., & Le, Q. E. (2020). Rethinking model scaling for convolutional neural networks. arXiv 2019. arXiv preprint arXiv:1905.11946.

[21] Vazquez-Fernandez, E., & Gonzalez-Jimenez, D. (2016). Face recognition for authentication on mobile devices. Image and Vision Computing, 55, 31-33.

[22] Wu, J. (2017). Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China, 5(23), 495.

[23] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 23(10), 1499-1503.