# Face Liveness Detection Using a sequential CNN technique

Abdelrahamn Ashraf Mohamed, Marwan Mohamed Nagah, Mohamed Gamal Abdelmonem,
Mohamed Yasser Ahmed, Mahmoud El-Sahhar, Fatma Helmy Ismail
*Faculty of Computer Science*
*Misr International University, Cario, Egypt*
abdelrahman1711335, marwan1709805, mohamed1709263,
mohamed1709620, mahmoud.ezzat, fatma.helmy {@miuegypt.edu.eg}

2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) | 978-1-6654-1490-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCWC51732.2021.9376030

***Abstract*—One of the most widely used biometric approaches is face recognition. Face recognition is used in many Fields. One of these fields is mobile devices authentication. While the number of mobile device users increasing year after year, the need for mobile security is also gaining ground. However, face recognition can be easily attacked by a malicious face spoofing. That is intended to deceive the face recognition system by facial pictures obtained from images or videos. Other cheaters show the mask of an authorized person to fool the recognition camera into a real person. Liveness detection is an important research topic to detect face spoofing. The proposed approach in this paper is a deep learning technique which is a sequential CNN (convolution Neural Network) divided into a feature extraction stage and a classification stage. The dataset used is CelebA-Spoof (2020) collected to recognize live and non-live faces. The experiment is performed on a part of the CelebA-Spoof dataset. The performance of the proposed approach is measured in terms of accuracy. The accuracy of testing the system on unseen data is 87% and the area under ROC curve is 0.535. there are many new techniques are intended to be used in future work such as capsule neural networks is expected to improves our results.**

***Index Terms*—deep learning, convolution neural network, face liveness detection, face spoofing, keras and tensorflow.**

## I. Introduction

Face recognition is a biometric system that is working on taking features from someone's face then it compares these features with the data of people in a database of known faces. Researchers have developed different methods to recognize the person's face and they managed to outcome the obstacles they faced such as different facial expressions, different angles, and bad illumination. It spreads very rapidly in the last decade. It has been used in many fields such as mobile device authentication [1], payments, companies used face recognition in attendance systems, also used in forensics and security access [2].

One of the problems that is facing developers when implementing a face recognition system is Face spoofing. Face spoofing is when an attacker tries to breach a face recognition system. The most famous face spoofing ways are printed images, Videos, and 3D Masks as shown in Figure (1) which an attacker can gain an illegal access to an authorized person and breach the face recognition system. Face spoofing was mentioned for the first time as a potential threat for face biometric systems was by Hoogsteden in 1992. After that, a detailed study was given by Doctor Stephanie A. C. Schuckers in 2002 on spoofing and anti-spoofing [3]. K.Kollreider proposed that in 2005 the structure tensor of the face image is the first technique that the liveness detection based on. The literature survey indicates that since 2005 face spoofing has shown a great interest through the researcher's community as shown in Figure (2). As a result that, since 2010 developers and researchers have been trying to develop a method that protects face recognition systems from face spoofing attacks.

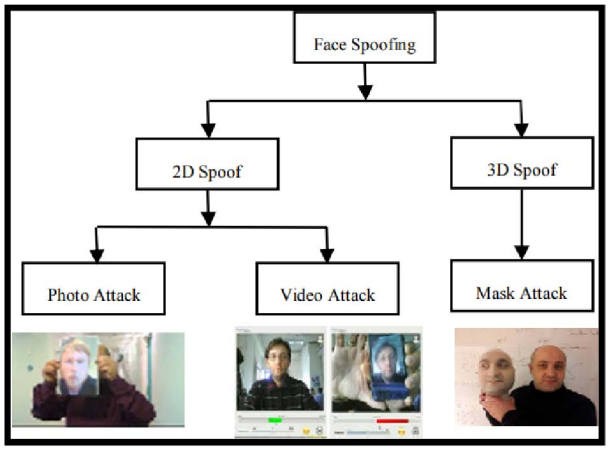


Fig. 1: Face spoofing types [3]

Face liveness detection is one of many methods that are used to prevent face spoofing attacks. It is relatively new to sense face liveness since the common security methods are fingerprints and passwords. However, many companies are in a bad need to detect face spoofing in order to prevent any illegal access to their systems. Cheaters can gain illegal access by showing an image (or a video or 3D-mask) of an authorized person standing in front of the security camera. The role of the security system is to offer access to the faces that belong to living persons. Therefore, detecting face liveness will play an important role in preventing face spoofing attacks.

978-0-7381-4394-1/21/$31.00 ©2021 IEEE

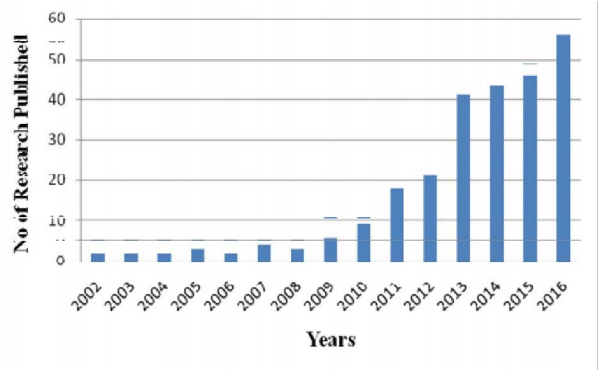


Fig. 2: Face spoofing researches [3]

The approach is done using a dataset called CelebA Anti spoofing dataset. It is new dataset that is published in 2020. It consists of 625,537 pictures of 10,177 subjects. So, It is big enough to train on and to have a very good results from it. We used 2000 pictures for training and 200 for testing. We put in consideration that the used dataset is balanced to prevent overfitting and to get the best results possible.

Our experiment is done using sequential CNN. So, It will be divided into several parts. First is the pre-processing part which is responsible of the part of feature extraction. Then augmentation will be done. Then the data will be passed to the CNN layers to train. After that the model will be able to classify whether the image is live or spoof. Cross-Validation is also used to test the model's ability to predict new data that was not used in estimating it in order to mark problems that can happen in the model like overfitting or selection bias.

## II. Related work

Face liveness detection is an important task to prevent the face spoofing problem, many approaches have been proposed that used traditional machine learning algorithms and deep learning algorithms to address the problem of face spoofing, machine learning consists of sequential steps including processing, segmentation, feature extraction, and classifications. Traditional machine learning is applied to face images to extract features such as support vector machine (SVM) [4] and viola jones algorithm [5] which can achieve good classification. However, there are some things that could happen when using machine learning such as errors, information loss, and poor results. Therefore, we intend to use deep learning techniques such as Convolutional neural network (CNN) [6] to get better accuracy and better results.

Sengur et al. [7], presents a convolutional neural network (CNN) approach to detect face liveness detection and it was applied on the NUAA dataset that was divided into 5761 images for test and 3491 images for train, this model achieved an accuracy of 83.38%. Akbulut et al. [6] presented as shown in figure 3 below CNN implementation also on the NUAA dataset but they divided the dataset into 5761 images for test and 1748 images for train and the model achieved an accuracy of 84.04% but when they use the local receptive fields (LRF)-ELM they achieved an accuracy of 76.31% on the same dataset. Komulainen and Pietikainen [5] used the same dataset NUAA and divided into 3362 test samples and 5761 train samples but this model used local binary pattern(LBP) based on micro-texture analysis and this depends on some features such as Gabor wavelet features and Histogram of oriented Gradients. This representation makes it easy for them to use fast linear support vector machine (SVM) classifiers, the results of the proposed system are calculated using equal error rate (EER) and area under the curve (AUC). So as a result of their work there was an improvement in EER from 2.8% to 1.1% and in AUC from 0.995 to 0.999.

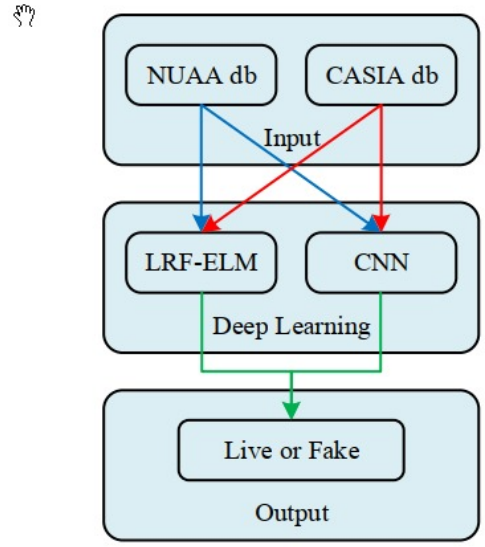


Fig. 3: CNN and LRF-ELM on the NUAA Dataset [6]

Wen et al. [8] proposed an effective system algorithm based on image distortion analysis, the system extract four important features from the IDA feature vector which are blurriness, chromatic moment, specular reflection, and color diversity, the dataset they use is the replay attack dataset which consists of 1,300 video recordings of both real-access and attack attempts of 50 different subjects, this model achieved an accuracy of (average TPR=90.5% & FAR=0.01). Liu et al. [9] used the same dataset Replay-attack but they use a Deep tree network (DTN) algorithm as shown in Figure 4 below, as it results for an overall accuracy of 95.5%. The Replay-attack dataset was used by Ito et al. [4] as they use the CNN algorithm to extract features from images and classified them into real and fake by support vector machine (SVM), the model runs on the Replay-attack dataset that consists of a set of videos sequences taken from 50 subjects of both real and fake scenarios. As a result, of that the error rate (EER) of using these methods is 2.3% and 0.75& respectively.

Guo et al. [10] proposed a system that can divide virtual fake data in order to get a solution for the face spoofing problem, the method relies on virtual synthesis then to use CNN to train

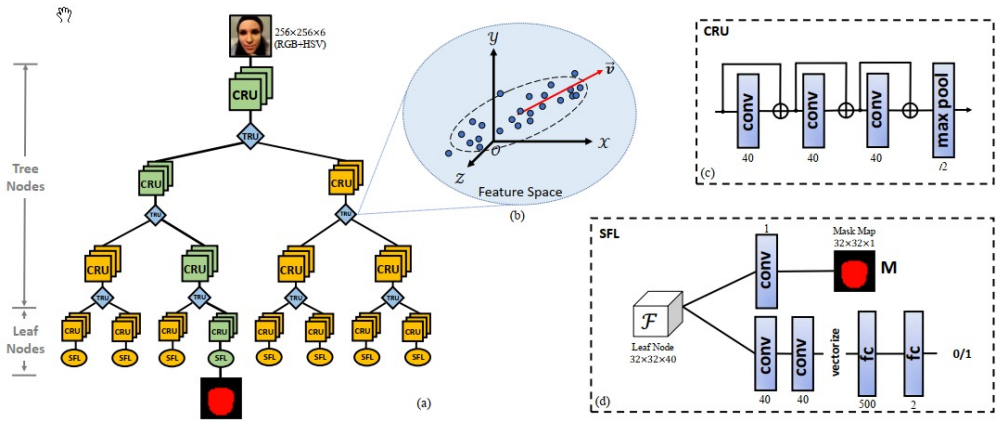


Fig. 4: Deep tree network algorithm [9]

the large scale synthetic spoof samples, for testing, CASIA-MFSD database is used and CASIA-RFS database is used for training. So as a result of their work there is a remarkable improvement over the weak perspective projection with an EER 2.22% and an HTER 1.67%. Moreover, ACER improves from 3.33% to 2.22% and Top-1 accuracy rises from 97.78% to 98.61%.

Pinto et al. [11] presented a new approach is used by using a technique that takes advantage of the noise signatures that is generated by the recaptured video, This method also use Fourier spectrum followed by the computation of the visual rhythm that captures the noise that is produced from the recorded videos. The used dataset in this paper is the Print Attack Database. It consists of 200 videos of 50 different users and 200 videos of spoof attacks. Our criticize for this paper is that they need to test their approach on larger video databases.

## III. System Architecture

The proposed model uses the sequential deep learning architecture in which layers of convolution and pooling are stacked from input to output. Figure 5 shows the proposed system architecture. The system consists of the following stages:

### A. *input data preprocessing stage*

Data augmentation [13] is a crucial part in training deep networks because they need large amounts of data to achieve high accuracy. Therefore, multiple augmentation techniques are applied, including shear range, zoom range, and horizontal flip as shown in figure (6).

### B. *feature extraction [14] stage*

The proposed CNN architecture consists of two operations: convolution operation, maxpooling and relu operations, which are repeated three times as shown in figure 5.

- convolution operation: Filters of size (3X3) are passed over the input images of size (150x150) to obtain the features that are important for classifying input images and discard the features that are not. In other words, filters convolve the images.
- maxpooling operation: this operation performs feature reduction on the output of the convolution layer, while preserving the unique features of images.
- relu activation function: relu function pseudo code can be represented by a simple if statement as follows:

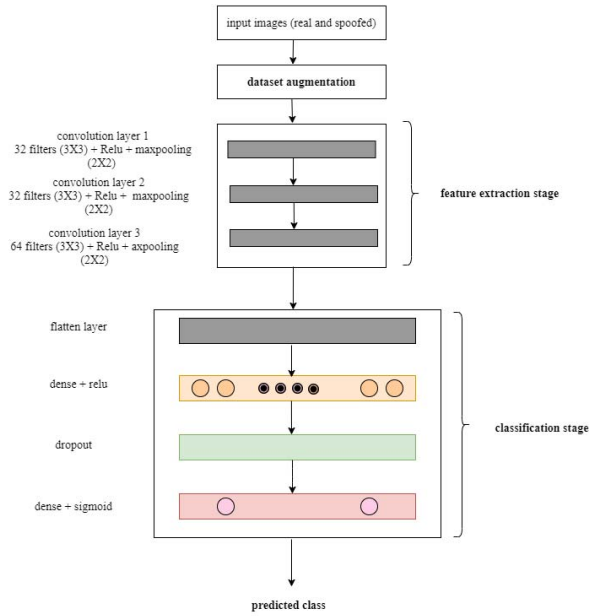


Fig. 5: proposed system architecture [12]

**if** $input > 0$ **then**
  $return\ input$
**else**
  $return 0$
**end if**

The function relu converts all negative inputs to zeros to

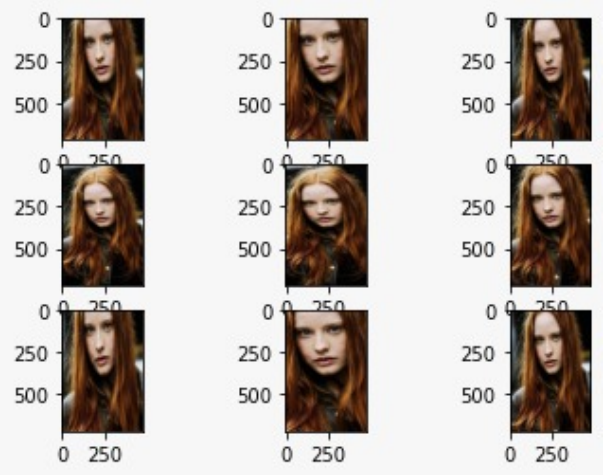

Fig. 6: Zoom augmentation results

appear as dots in the manipulated images. That makes images becomes clearer.

### C. classification stage

- flatten layer: the function of this layer is to structure the feature maps into a single vector suitable to be processed by the next neural network layer.
- dense and relu activation function: dense function in keras builds a neural network with a specific number of hidden nodes and a relu activation function. It takes the flattened vector to be as an input to the neural network.
- dropout: its function is to prevent overfitting.
- dense and sigmoid activation function: it constitutes the output layer with sigmoid function since it is a binary classification problem (live or non_live face)

### D. compiling the CNN

to compile the constructed CNN, the function compile is used with three parameters: the optimizer is rmsprop, the loss function is binary_crossentropy and the performance metric is accuracy. The optimizer rmsprop is an adaptive learning rate method proposed in [15]. The calculation of binary_crossentropy loss function is shown in equation 1

$$Loss = -\frac{1}{output}\sum_{size}^{output} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i) \quad (1)$$

where $\hat{y}_i$ is the $i$ -th scalar value in the model output, $y_i$ is the corresponding target value, and output size is the number of scalar values in the model output. The accuracy is calculated as shown in equation 2

$$Accuracy = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \quad (2)$$

## IV. Experimental Results

This section introduces the outcome of the experiments performed using a sequential CNN as described in the previous section.

### A. Setup of the Experiment

The experiments were done on a machine equipped with an Intel Core i7-4790 @ 3.6GHZ, 16GB of memory, and an NVIDIA GTX1060 GPU card. The machine runs on Windows10. The experiments were done using python programming language and keras framework. The experiment were done using a dataset called CelebA dataset. 2000 pictures were used for training and 200 pictures were used for testing divided equally between live and spoof to be balanced.

### B. Dataset description

A novel anti-spoofing dataset CelebA-Spoof has been collected to recognize live and non-live faces [16]. The main advantages of CelebA-Spoof are the following:

- Quantity: CelebA-Spoof contains 625,537 pictures of 10,177 subjects, which is larger than any existing dataset related to this field.
- Diversity: The spoofed images are taken from 8 scenes (2 environments * 4 illumination conditions) with more than 10 sensors.
- Annotation Richness: CelebA-Spoof contains 10 spoof type annotations, as well as the 40 attribute annotations inherited from the original CelebA dataset [17]
- Date: The dataset was made and published in 2020 and it is the newest dataset related to this field.

The proposed model is Sequential CNN. It has been tested and cross validated using a part of CelebA-Spoof dataset.

### C. Training and testing

The data set has been read using ImageDataGenerator [18] which is Keras function that will be used in fit generator function of the sequential model in Keras that is used for training images. The ImageDataGenerator generates classes automatically from the number of folders existing in the path given to it and fit generator [18] function knows how to work with the information given. This is used in case of testing.

ImageDataGenerator is also used in augmentation [13] part to increase the size of the training data and to avoid overfitting. Augmentation methods used are rescale to make normalization, Shear to make the image distorted along an axis, zoom to put in consideration the distance from the camera to the face, and horizontal flip.

In cross-validation, we could not use ImageDataGenerator with K-Fold methods to dividing the dataset into train and test randomly each time with a different output of images each time. So, We had to convert the object of ImageDataGenerator that contains the dataset into two arrays. The first array contains the images as an array of pixels 3 channels for each and the other array contains the label for each image which represents its class whether live or spoof. Then it has been sent to a loop of K-fold time in our case 5 times every time

TABLE I: Evaluation Measures Results

| Model | Type | Measure | | | |
|---|---|---|---|---|---|
| | | Accuracy | Precision | Recall | F-measure |
| CNN | Train | 96.9% | 96.9% | 96.4% | 88.2% |
| Naive Bayes | | 80.2% | 76.6% | 78.2% | 82.3% |
| CNN | **Test** | **87%** | **93.6%** | **78.2%** | **86.8%** |
| Naive Bayes | | **81.2%** | **78.2%** | **83.0%** | **83.2%** |
| CNN | Cross-Validation | 94.7% | 50.9% | 81.9% | 52.9% |
| Naive Bayes | | 82% | 79.6% | 86.3% | 83.4% |

there is a new test and train set of images divided by the k fold function and then it has been tested using fit function. Then in both ways normal and cross-validation we used the evaluate function to test our model with a totally different set of images.

### *D. Training and testing Results*

The dataset is divided into 2000 images for training and 200 images for testing. The number of epochs is 50. We used 50 epochs to train the model. Figure 7 shows the test accuracy and the loss per epoch. It is noticed that the accuracy increases and the loss decreases. ROC Curve [19] is produced also as shown in the figure 8. The area under curve is 0.535.

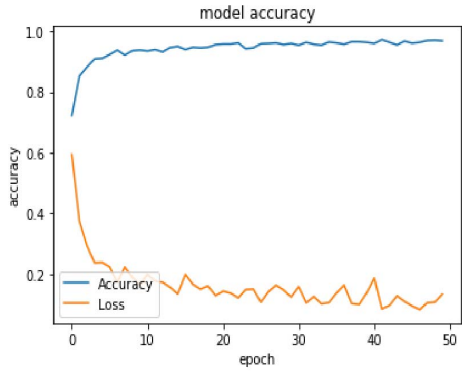


Fig. 7: Testing accuracy and loss per epoch

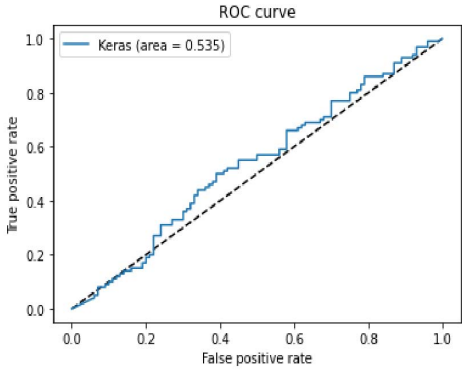


Fig. 8: ROC curve for testing

### *E. Cross Validation results*

We used 5-Fold Cross Validation as shown in figure 9.

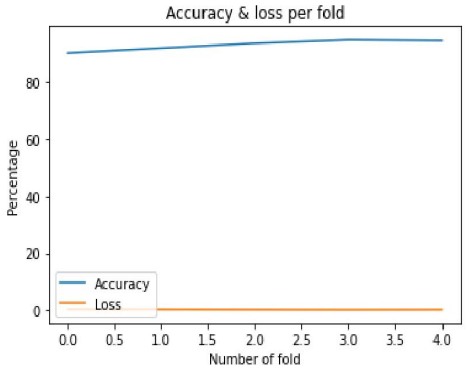


Fig. 9: cross validation accuracy and loss per epoch

Table (I) shows the accuracy, precision, recall, and F1-measure of training, testing, and cross-validation. The results show an accuracy of 87% during testing while it shows an accuracy of 96.9 % while training and 94.7% during cross-validation. The difference between testing and training accuracy is not that large. However, Other measures (such as precision, recall, and F1-measure) are higher in testing than in cross-validation. The justification is that cross-validation tests the system on a part of the data that is seen before. The low values of precision, recall, and F1-measure indicate that the classifier needs more data to be well trained during cross-validation. Moreover, the holdout cross validation is intended to be used rather than 5 fold cross validation.

## V. Conclusion and Future Work

Many security systems have taken care of Face liveness detection to prevent face spoofing. The proposed CNN approach achieved a relatively acceptable accuracy for testing which is 87% and for cross validation 94.7%. The system consists of two stages which are the feature extraction stage and the classification stage. The dataset used is CelebA-Spoof which appears in 2020. The training set contains 2000 images divided into 1000 live and 1000 spoofed to obtain a balanced set and the testing set contains 200 images. However, many new techniques are intended to be used in future work such as capsule neural networks [20] . Other models of CNN such as pre-trained ones will be compared with regular CNN to achieve the highest performance. Another type of face spoofing will be used like videos using deep learning algorithms to detect face spoofing in videos such as RNN which is an important face spoofing topic.

## REFERENCES


[1] G. Hassan and K. Elgazzar, "The case of face recognition on mobile devices," 2016.

[2] D. N. Parmar and B. B. Mehta, "Face recognition methods & applications," *arXiv preprint arXiv:1403.0485*, vol. 4, 2013.

[3] J. K. Sandeep Kumar and S. Singh, "A comparative study on face spoofing attacks," *International Conference on Computing, Communication and Automation (ICCCA)*, 2017.

[4] T. O. Koichi Ito and T. Aoki, "Recent advances in biometric security: A case study of liveness detection in face recognition," *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017.

[5] J. Komulainen and M. Pietikainen, "Face spoofing detection from single images using texture and local shape analysis," *International Conference on Computing, Communication and Automation (ICCCA)*, vol. 1, pp. 3–10, 2012.

[6] B. Yaman Akbulut, Abdulkadir Sengur and S. Ekici, "Deep learning based face liveness detection in videos," *international Artificial Intelligence and Data Processing Symposium (IDAP)*, 2017.

[7] Y. A. Abdulkadir Sengur, Zahid Akhtar and S. Ekici, "Deep feature extraction for face liveness detection," *International Conference on Artificial Intelligence and Data Processing (IDAP)*, 2018.

[8] H. H. Di Wen and A. K. Jaini, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, 2015.

[9] A. J. Yaojie Liu, Joel Stehouwer and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] J. X. Jianzhu Guo, Xiangyu Zhu and S. Z. Li, "Improving face anti-spoofing by 3d virtual synthesis," *International Conference on Biometrics (ICB)*, 2019.

[11] W. R. S. Allan da Silva Pinto, Helio Pedrini and A. Rocha, "Video-based face spoofing detection through visual rhythm analysis," *25th SIBGRAPI Conference on Graphics, Patterns and Images*, 2012.

[12] J. H. M. T. Husein Perez and A. Mosavi, "Deep learning for detecting building defects using convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, 2019.

[13] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," vol. 143, 2017.

[14] C. L. Yushi Chen, Hanlu Jiang and X. Jia, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," 2016.

[15] Y. Bengio and M. CA, "Rmsprop and equilibrated adaptive learning rates for nonconvex optimization," *corr abs/1502.04390*, 2015.

[16] Y. Zhang, Z. Yin, Y. Li, G. Yin, J. Yan, J. Shao, and Z. Liu, "Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations," *arXiv preprint arXiv:2007.12342*, 2020.

[17] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August*, vol. 15, p. 2018, 2018.

[18] A. Gulli and S. Pal, *Deep Learning with Keras*. Packt Publishing Ltd, 2017.

[19] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve," vol. 143, 1982.

[20] J. Y. Huy H. Nguyen and I. Echizen, "Use of a capsule network to detect fake images and videos," 2019.