Intelligent Systems with Applications 16 (2022) 200107

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

# Face presentation attack identification optimization with adjusting convolution blocks in VGG networks

Sudeep D. Thepade *, Mayuresh Dindorkar, Piyush Chaudhari, Shalakha Bang

*Computer Engineering Department, Pimpri Chinchwad College of Engineering, SPPU, Pune, India*

ARTICLE INFO

*Keywords:*
Face liveness detection
Transfer learning
VGG16
VGG19
DensNet121
Xception
MobileNet
InceptionV3

ABSTRACT

Advancement in deep learning is mapping to every field of life and applying it to almost all research problems. Numerous Deep Convolutional Neural Network (DCNN) architectures are being proposed, giving different results based on the depth and value of hyperparameters. The entire development of such DCNN architectures from scratch needs a lot of effort, and such architectures may not be used for other applications than the one they are structured for. Transfer learning is a way to modify these pre-trained networks to make them suitable for newer diverse applications. This paper attempts to empirically assess the performance and suitability of existing pre-trained DCNN architectures for human face liveness detection. Due to the advent of ambient computing for contactless identification of humans using their biometric traits, human face liveness detection proves to be an important research area. Six pre-trained DCNN models, alias VGG16, VGG19, DensNet121, Xception, MobileNet, and InceptionV3, are considered for empirical assessment in human face liveness detection. The method is explored using two face liveness detection datasets - NUAA and Replay-Attack. Face Liveness Accuracy, and Half Total Error Rate (HTER) are considered prime performance evaluation metrics. At a learning rate of $10^{-4}$, the VGG19 network with scenario "Original VGG" gives the highest face liveness detection accuracy, which is the outcome of current research.

## 1. Introduction

In the modern era of pervasive computing, an individual's security is reliant mainly on their biometric traits such as the face (Kekre et al., 2010a; Kekre et al., 2010b), iris (Kekre et al., 2011a; Thepade & Bidwai, 2013; Thepade & Mandal, 2014), fingerprints (Khade et al., 2018; Thepade & Khade, 2018), etc. Face recognition has become the most vital physiological biometric modality used in data security owing to rich features exhibited by the face and ease of handling the facial recognition systems (Kekre et al. 2010b,Kekre et al. 2011b). Attackers mislead these systems by spoofing attacks like photos, videos, cut-photo, mask attacks, etc. To mitigate these attacks existing face recognition systems should be improvised. Existing anti-spoofing methods make use of motion-based, shape-based, depth-based, color-based, texture-based, and deep learning-based features for spoof identification. A subset of deep neural networks (NN) called Convolutional Neural Network (CNN) is widely explored in various image processing areas. Several pre-trained deep CNN models like VGG16, VGG19, InceptionV3, etc., are available that are trained on large datasets for image-classification jobs.

Large annotated datasets are the prime requirement of a highly accurate model. To get such large datasets for every domain is not easy. To minimize this requirement, an approach called transfer learning came into existence. It is a method that reuses an existing pre-trained model to solve a different but related problem. Generally, the method which modifies the previously trained model developed for a particular task for performing some other task is called Transfer Learning. The transfer learning approach increases the speed and improves the performance of new models. Different transfer learning techniques are used based on the research area and available data.

The key contributions of the work presented in the paper are as followse.

- The paper has attempted the empirical performance assessment of six pre-trained deep CNN architectures (VGG16, VGG19, InceptionV3, DenseNet121, MobileNet, and Xception) for the human face liveness detection application.

* Corresponding author.
*E-mail address:* sudeepthepade@gmail.com (S.D. Thepade).

https://doi.org/10.1016/j.iswa.2022.200107
Received 11 October 2021; Received in revised form 20 July 2022; Accepted 6 August 2022
Available online 7 August 2022

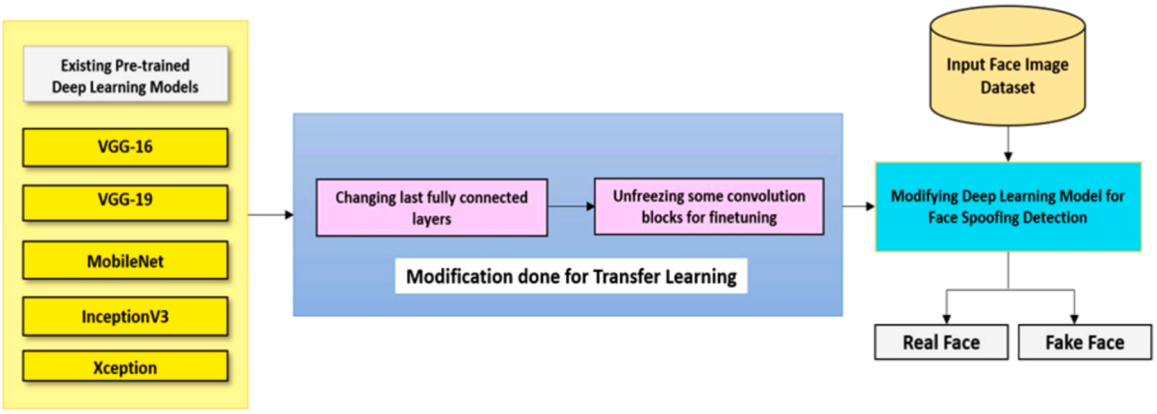


**Fig. 1.** Flowchart illustrating modifications done in the existing pre-trained deep CNN models for the task of face liveness detection.

**Table 1**
The number of training and testing face images taken from both datasets for the proposed empirical performance assessment of DCNN models.

| Face Class | Replay-Attack | | NUAA | |
|---|---|---|---|---|
| | Train | Test | Train | Test |
| Live Face | 10,800 | 13,656 | 1743 | 3362 |
| Spoofed Face | 32,280 | 42,809 | 1748 | 5761 |
| | Spoofed with four types | | Spoofed using only one type | |

**Table 2**
The optimizer parameters set during the performance assessment of existing DCNN models used in transfer learning for human face liveness detection.

| Parameter | Value |
|---|---|
| Optimizer | Adam |
| Learning rate | $10^{-3}$, $10^{-4}$, $10^{-5}$ |
| Batch-size | $10^{2}$ |
| Beta-1 | 0.9 |
| Beta-2 | 0.999 |
| Epsilon | $1 * 10^{-8}$ |

- Through the finetuning of the last two convolution blocks of VGG16 and VGG19 at the learning rate of 10–5, the improvised test accuracy is attempted.
- The complexity of VGG networks is reduced by decreasing the number of convolution blocks in explored scenarios (approaches)
- Experimental explorations are done on two standard datasets: the NUAA dataset and the Replay-Attack dataset.

## 2. Literature survey

Some of the recent attempts implementing pre-trained CNN models for face liveness detection have achieved prominent results. A brief review of these attempts is summarized here.

"FASNet," a CNN architecture that is an altered version of VGG16, is explored (Lucena et al., 2017). Using back-propagation, authors have finetuned the weights of the FASNet – from the fourth convolution block to the topmost layers. 3DMAD & Replay-Attack datasets are used to evaluate the performance of the proposed method.

A feature fusion approach that combines ResNet18 and Rotation Invariant Local Binary Pattern (RILBP) features is explored by Chen et al. (2019). Further, Support Vector Machine (SVM) is trained on these features to perform binary classification. Improved performance is observed in both intra and cross-database testing.

ResNet-18 is fed with three different types of features, namely color, temporal, and patch-based (Tang et al., 2018), to obtain class probabilities. A class-probability vector is generated by combining these feature class probabilities. Further, this class-probability vector is given to SVM for performing face anti-spoofing tasks.

In Tu & Fang, (2017), pertinent hidden features from the input face image are located by ResNet50. To learn temporal features, LSTM is applied on the top of ResNet50.

These features are utilized to determine whether the face is spoofed or live.

A simple face anti-spoofing technique is proposed in Das et al., (2019). Authors have extracted global deep features and handcrafted features from VGG16 and LBP descriptors, respectively. The proposed method has experimented on SSIJRI, Replay-Attack, Replay-Mobile and, 3DMAD datasets.

Two face anti-spoofing techniques are explored (Elloumi et al., 2020). The first technique states LBP histogram calculation & VGG16 finetuning. The second technique utilizes Image Quality Measures (IQM).

## 3. Proposed method

The pre-trained model is the saved previously trained network on a massive dataset for broad-scale image classification. In the transfer learning process, the knowledge gained by the machine learning (ML) model from one dataset is used on another. Two approaches of the transfer-learning are: (a) Using the pre-trained model as the ready-made feature extractor for a specific image-classification job. (b) Finetuning the pre-trained model, a whole or a portion of the model is retrained on new data. Transfer learning is usually employed to evade overfitting when the amount of data is less. Saving time and computational resources required during the training phase is a major advantage of using transfer learning.

In the proposed work here (Fig. 1), for detection of a human face, the input face image is first converted to Grayscale and then passed through the Haarcascade classifier to obtain Region-Of-Interest (ROI). ROI of input RGB image is resized to the window of dimension 96 * 96 pixels. After this preprocessing, the proposed methodology uses finetuning approach on different pre-trained deep learning models for face liveness detection. For every pre-trained model, the last fully connected layers (FC) are substituted by a classification head composed of two FC layers of size 256 and 1, respectively. After the first FC layer, a dropout layer is added to shun overfitting. Most models overfit for the learning rate '1 * $10^{-3}$'; hence, results for the learning rate of $1 * 10^{-4}$ and $1 * 10^{-5}$ are depicted in this paper. Adam optimizer is accustomed to learning rates and weight decay $1 * 10^{-6}$ (refer to Table 1). Further, sigmoid is used as a decision function suited for binary classification instead of softmax.

Finetuning the model with an arbitrarily initialized classification head may cause the pre-trained base model to forget its learning due to large gradient updates. Thus, we set the pre-trained model as non-

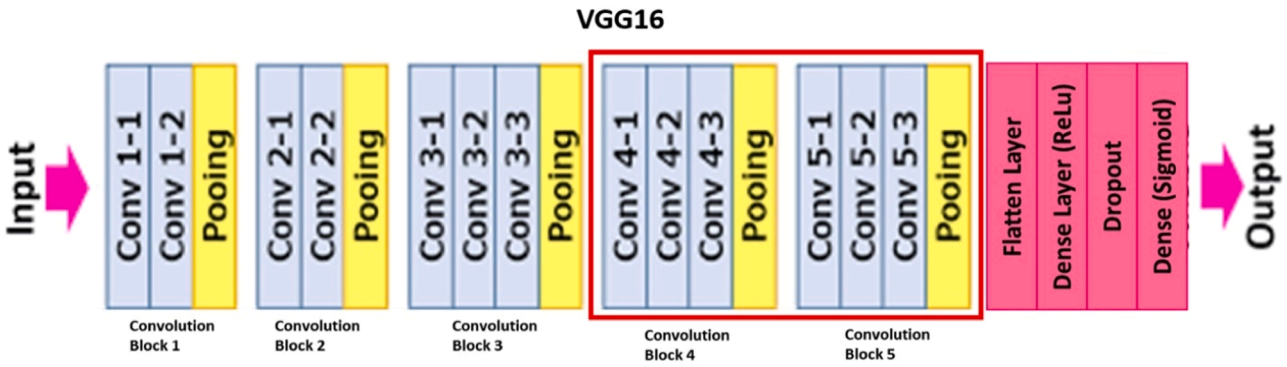


**Fig. 2.** Modified VGG16 architecture showing the finetuned convolution blocks by red region.

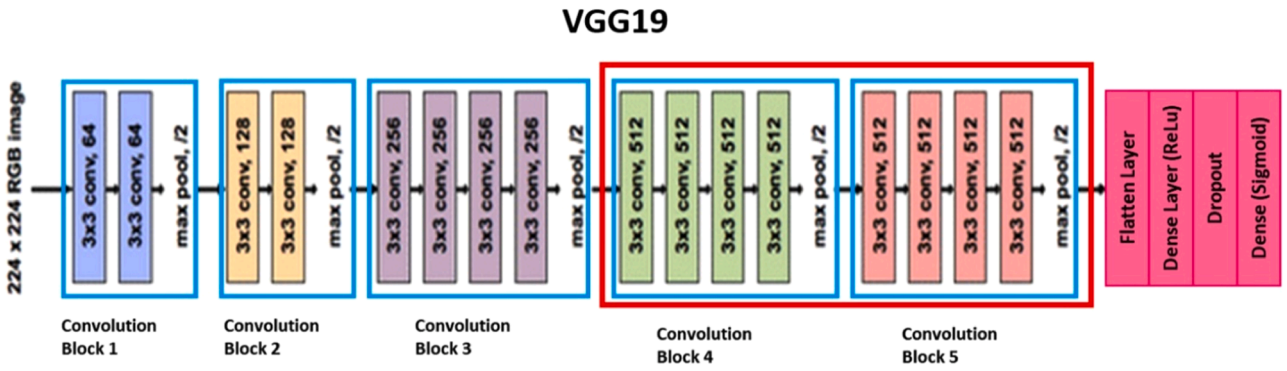


**Fig. 3.** Modified VGG19 architecture showing the finetuned convolution blocks by red region.

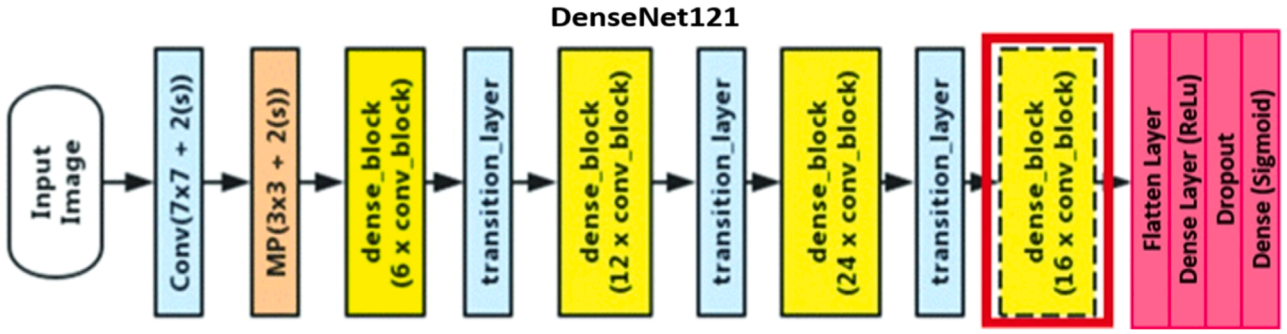


**Fig. 4.** Modified DenseNet121 architecture showing the finetuned convolution blocks by red region.

trainable while enabling the classification head to be trainable during training for the first five epochs. Later on, the last two convolution blocks of the pre-trained base model, which are immediately before the classification head, are set trainable (unfrozen). Then both the base model and classification head are trained jointly for the next 30 epochs. For VGG16, VGG19 & DenseNet121 last two convolution blocks enclosed in the red boundary are finetuned as shown in Figs. 2, 3, and 4, respectively.

Incase of InceptionV3, the last two inception blocks are finetuned, as shown in Fig. 5. The last two separable convolution blocks are finetuned in the case of Xception (refer to Fig. 6). Each separable convolution block includes depthwise and pointwise convolutions. For MobileNet, the last two depthwise separable convolution blocks are finetuned, as shown in Fig. 7.

From Tables 3 and 4, it is observed that VGG networks (VGG16 and VGG19) are performing better compared to other considered finetuned DCNNs hence these two networks are used for further explorations at the learning rate of $10^{-4}$ and $10^{-5}$. In further exploration, an attempt is made to reduce the complexity of these VGG networks, which can be inferred from Fig. 8. Following are the reasons to select only VGG networks for further explorations:

- Compact architecture
- Potential to give higher face liveness detection accuracy.

To reduce the complexity of the VGG networks, we have considered five different scenarios, alias Original-VGG, Scenario-D, Scenario-C, Scenario-B, and Scenario-A (refer to Fig. 8). The complexity is reduced in each scenario by removing the last convolution block from the VGG network; further, the remnant network is trained for 30 epochs, and obtained results are justified in the "Results and Discussion" section.

### 3.1. Existing popular DCNN models

Here the empirical performance-appraise of six popular DCNN models, alias VGG16, VGG19, InceptionV3, DenseNet121, Xception, and

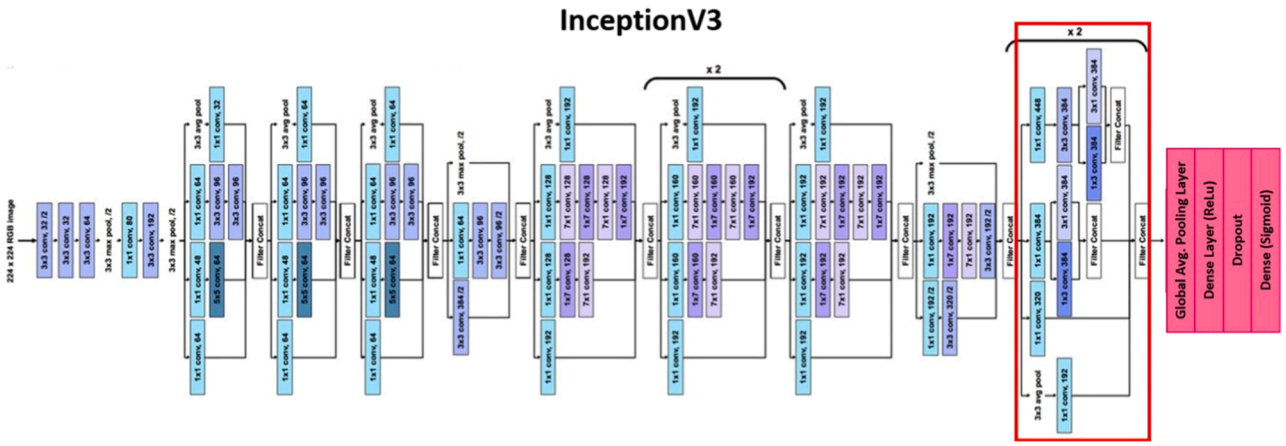


**Fig. 5.** Modified InceptionV3 architecture showing the finetuned convolution blocks by red region.

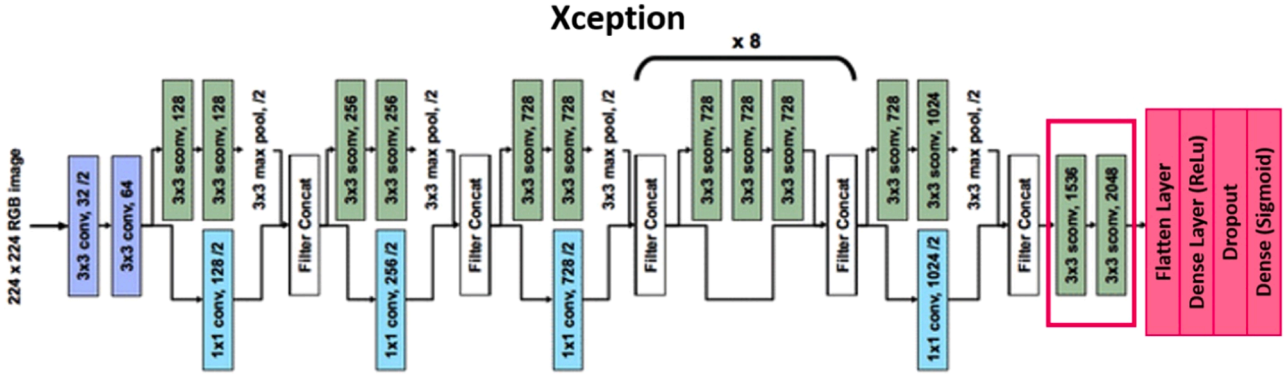


**Fig. 6.** Modified Xception architecture showing the finetuned convolution blocks by red region.

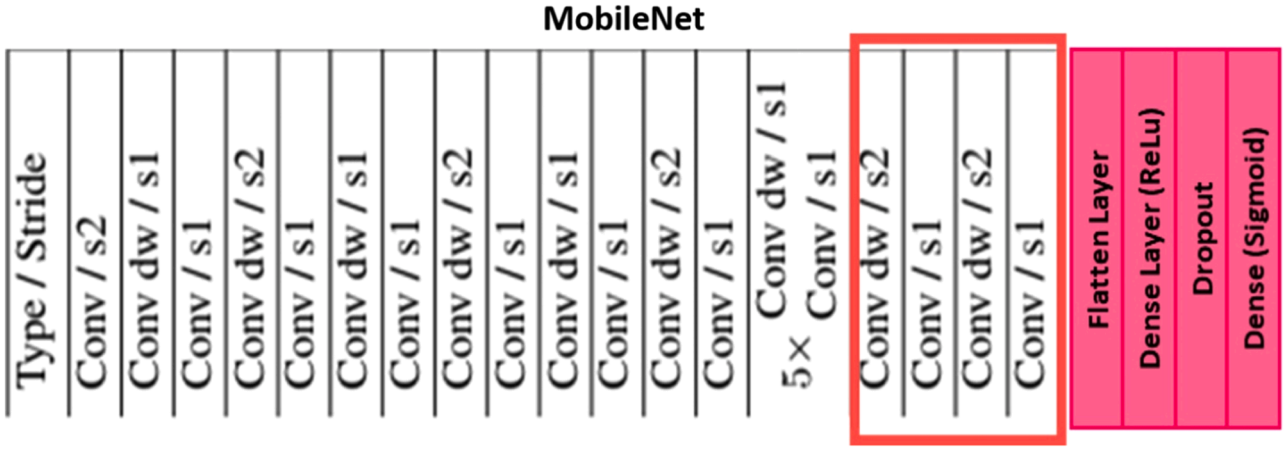


**Fig. 7.** Modified MobileNet architecture showing the finetuned convolution blocks by red region.

MobileNet, is done. These admired pre-trained DCNN models are fine-tuned using a transfer learning approach for face liveness detection.

A CNN model referred to as VGG16 is proposed by Simonyan & Zisserman (2015). It is trained on the ImageNet dataset composed of nearly 14 million images of 1000 classes using NVIDIA's Titan Black GPU. Its architecture consists of 16 layers: 13 convolution layers and three fully connected layers.

VGG19 (Simonyan & Zisserman, 2015) comprises 19 layers, including 16 convolution layers and three fully connected layers. A remarkable improvement in the existing VGG network has been made by increasing the depth of layers from 16 to 19.

With 42 layers of deep architecture (Szegedy et al., 2016), a low error rate is achieved, and hence it was awarded 1st runner-up ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2015. The computation cost of InceptionV3 is just 2.5 times higher than GoogleNet (InceptionV1), but it is still efficient compared to VGGNet.

DenseNet has 4 versions namely DenseNet121, DenseNet161, DenseNet169 and DenseNet201. In DenseNet, each layer is connected in a feed-forward fashion (Huang et al., 2017). Reducing vanishing gradient problems, encouraging feature reuse, etc., are the prime advantages of

**Table 3**
Testing accuracies obtained for considered finetuned DCNN models over 30 epochs.

| Finetuned DCNN | Learning Rate ($10^{-4}$) Test accuracy (%) | | ($10^{-5}$) | |
|---|---|---|---|---|
| | Replay-Attack | NUAA | Replay-Attack | NUAA |
| VGG16 | 99.99 | 67.14 | 100 | 73.89 |
| VGG19 | 99.95 | 68.43 | 100 | 68.30 |
| Mobilenet | 93.18 | 68.87 | 94.70 | 56.80 |
| DenseNet121 | 91.79 | 65.37 | 98.29 | 62.00 |
| Xception | 99.86 | 69.71 | 98.96 | 65.79 |
| InceptionV3 | 96.82 | 63.35 | 94.10 | 62.03 |

**Table 4**
HTER obtained for considered finetuned DCNN models over 30 epochs.

| Finetuned DCNN | Learning Rate ($10^{-4}$) HTER (%) | | ($10^{-5}$) | |
|---|---|---|---|---|
| | Replay-Attack | NUAA | Replay-Attack | NUAA |
| VGG16 | 0.015 | 37.54 | 0 | 33.14 |
| VGG19 | 0.1 | 28.66 | 0 | 41.345 |
| Mobilenet | 11.08 | 39.895 | 7.345 | 49.61 |
| DenseNet121 | 16.33 | 41.545 | 2.025 | 43.65 |
| Xception | 0.12 | 26.605 | 1.51 | 35.335 |
| InceptionV3 | 5.57 | 45.855 | 8.72 | 41 |

these networks. DenseNet121 is 121 layers of deep architecture.

Xception (Chollet, 2017) has 71 layers of deep architecture. Xception's architecture is motivated by Inception, where inception modules got substituted with depth-wise detachable convolutions. Xception has the same number of parameters as InceptionV3.

MobileNet has an organization of detachable convolution modules, formed by depthwise and pointwise convolution being done in the form of piles in the MobileNet network (Howard et al., 2017). It has 28 layer-deep architecture considering depthwise and pointwise convolutions as an individual layer.

### 3.2. Experimentation environment

A platform provided by Kaggle is used for experimental assessment here. It is an online data science community for data scientists and machine-learning practitioners. The Python code scribbled in the Kaggle notebook was executed using GPU as an accelerator. Pretrained DCNN models provided by the Keras library are used for experimentation.

Two standard and publically available datasets used in face liveness detection, alias 'Replay-Attack dataset' and 'NUAA dataset,' are explored here to validate the performance appraise of existing DCNN models after transfer learning. The testing accuracy and HTER values for each modified DCNN model are used as performance measures.

IDIAP Research Institute has constituted the Replay-Attack dataset (Chingovska et al., 2012), having 1300 videos of 50 persons with two diverse luminance conditions set up as adverse controlled during the video acquisition refer to Fig. 2. All videos in the dataset are filmed at a rate of 25 Hertz. We have extracted half of the frames per second (FPS)

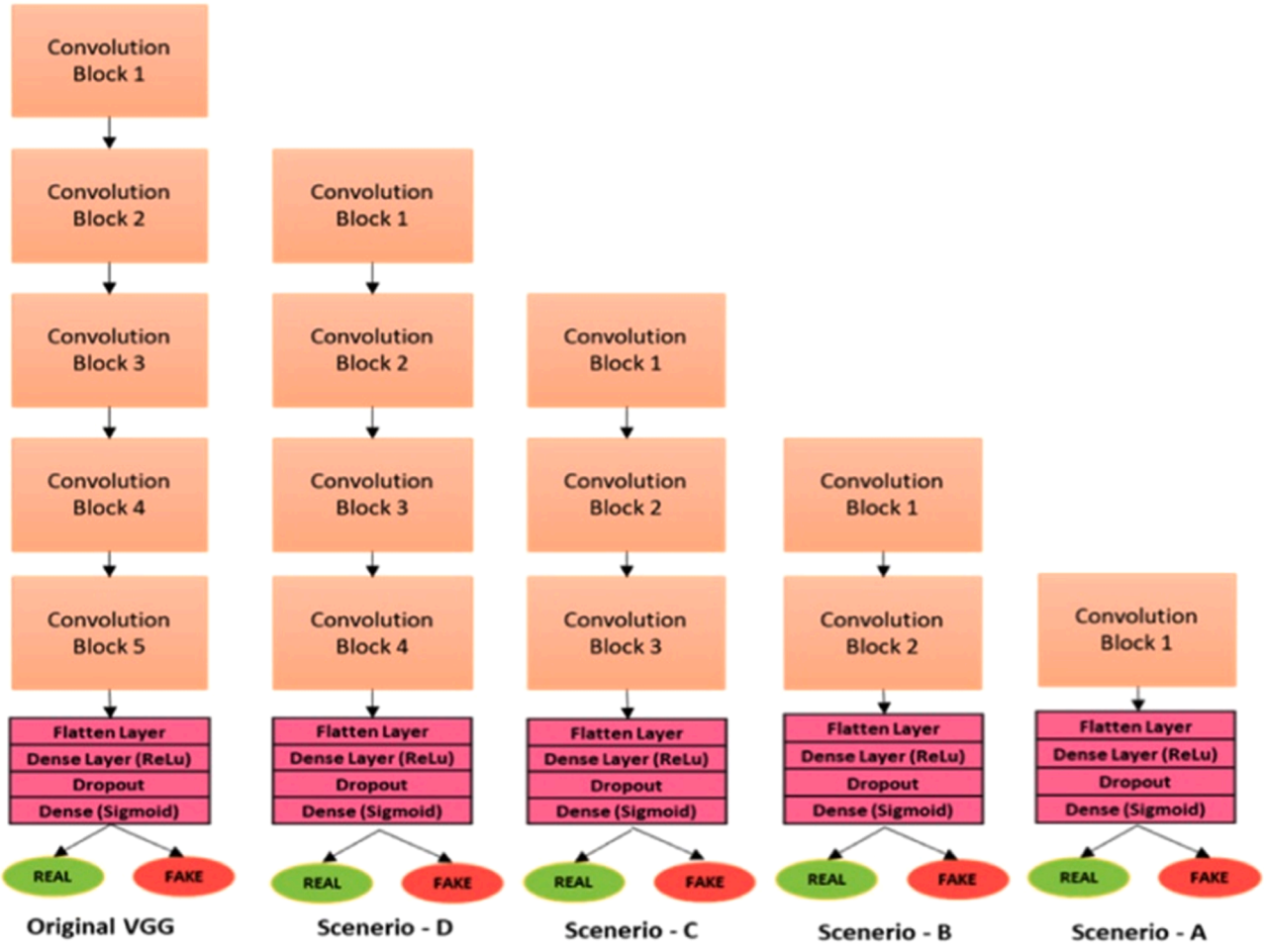


**Fig. 8.** Reducing the complexity of VGG16 and VGG19.

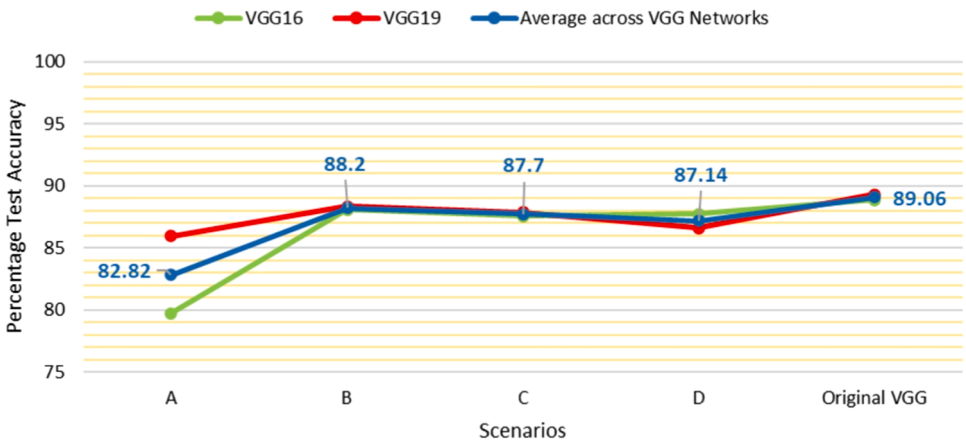


**Fig. 9.** Average percentage test accuracy of VGG16 and VGG19 across both NUAA & Replay-Attack datasets for learning rate of $10^{-4}$.

**Table 5**
Performance metrics observed for VGG16 over 30 epochs on NUAA dataset for learning rate of $10^{-4}$.

| Scenario | Test Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FAR (%) | FRR (%) | HTER (%) |
|---|---|---|---|---|---|---|---|
| A | 60.05 | 47.26 | 72.64 | 57.26 | 47.3 | 27.36 | 37.33 |
| B | 76.13 | 62.26 | 89.44 | 73.42 | 31.64 | 10.56 | 21.1 |
| C | 75.12 | 71.03 | 54.85 | 61.9 | 13.05 | 45.15 | 29.1 |
| D | 75.39 | 80.21 | 44.11 | 56.92 | 6.35 | 55.89 | 31.12 |
| Original VGG16 | 77.65 | 63.85 | 90.72 | 74.95 | 29.98 | 9.28 | 19.63 |

**Table 6**
Performance metrics observed for VGG16 over 30 epochs on Replay-Attack dataset for learning rate of $10^{-4}$.

| Scenario | Test Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FAR (%) | FRR (%) | HTER (%) |
|---|---|---|---|---|---|---|---|
| A | 99.34 | 98.87 | 98.4 | 98.63 | 0.36 | 1.6 | 0.98 |
| B | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| C | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| D | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| Original VGG16 | 100 | 100 | 100 | 100 | 0 | 0 | 0 |

**Table 7**
Performance metrics observed for VGG19 over 30 epochs on NUAA dataset for learning rate of $10^{-4}$.

| Scenario | Test Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FAR (%) | FRR (%) | HTER (%) |
|---|---|---|---|---|---|---|---|
| A | 72.308 | 59.33 | 79.42 | 67.92 | 31.77 | 20.58 | 26.175 |
| B | 76.626 | 62.08 | 94.38 | 74.9 | 33.64 | 5.62 | 19.63 |
| C | 75.681 | 60.49 | 98.51 | 74.95 | 37.55 | 1.49 | 19.52 |
| D | 73.154 | 63.99 | 62.37 | 63.17 | 20.48 | 37.63 | 29.055 |
| Original VGG19 | 78.56 | 64.85 | 91.58 | 75.93 | 28.97 | 8.42 | 18.695 |

**Table 8**
Performance metrics observed for VGG19 over 30 epochs on Replay-Attack dataset for learning rate of $10^{-4}$.

| Scenario | Test Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FAR (%) | FRR (%) | HTER (%) |
|---|---|---|---|---|---|---|---|
| A | 99.56 | 99.5 | 98.72 | 99.11 | 0.17 | 1.28 | 0.725 |
| B | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| C | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| D | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| Original VGG19 | 100 | 100 | 100 | 100 | 0 | 0 | 0 |

(i.e., 25/2 = 12) from each video.

Nanjing University of Aeronautics and Astronautics has invented the NUAA dataset (Tan et al., 2010). Real and Spoofed face images of 15 persons are acquired by a web camera. The dataset consists of 7509 fake and 5105 real face images, segregated into training and testing sets. Every face image has dimensions of 640 * 480 pixels. The database includes only one type of spoofing attack, a 'photo-attack.' The database possesses appearance variations like gender, with-or-without glasses, and light. Few sample faces from the NUAA dataset are given in Fig. 3. The number of training and testing samples taken across live and spoofed face images for both 'Replay-Attack' and 'NUAA' datasets are shown in Table 1.

The performance metrics used in the paper are as follows:

**Table 9**
Average percentage test accuracy of VGG networks across both the datasets for each scenario considering learning rate of $10^{-4}$.

| Scenario | VGG16 | VGG19 |
|---|---|---|
| A | 79.7 | 85.94 |
| B | 88.07 | 88.32 |
| C | 87.56 | 87.84 |
| D | 87.7 | 86.58 |
| Original VGG | 88.83 | 89.28 |
| Average over Scenarios | 86.37 | 87.59 |

$$FAR\ (False\ Acceptance\ Rate) = \frac{FP}{TN + FP} \tag{1}$$

$$FRR\ (False\ Rejection\ Rate) = \frac{FN}{TP + FN} \tag{2}$$

$$HTER\ (Half\ Total\ Error\ Rate) = \left(\frac{FAR + FRR}{2}\right) * 100 \tag{3}$$

$$Precision\ or\ Positive\ Predictive\ Value\ (PPV) = \frac{TP}{TP + FP} \tag{4}$$

$$Recall\ or\ Sensitivity = \frac{TP}{TP + FN} \tag{5}$$

**Table 14**
Average test percentage accuracy of VGG networks across both the datasets for each scenario considering learning rate of $10^{-5}$.

| Scenario | VGG16 | VGG19 |
|---|---|---|
| A | 82.57 | 81.07 |
| B | 81.94 | 83.95 |
| C | 87.09 | 87.12 |
| D | 86.26 | 87.59 |
| Original VGG | 87.21 | 84.67 |
| Average over Scenarios | 85.01 | 84.88 |

**Table 10**
Performance metrics observed for VGG16 over 30 epochs on NUAA dataset for learning rate of $10^{-5}$.

| Scenario | Test Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FAR (%) | FRR (%) | HTER (%) |
|---|---|---|---|---|---|---|---|
| A | 65.32 | 52.52 | 61.30 | 56.57 | 32.34 | 38.69 | 35.52 |
| B | 63.87 | 51.00 | 50.27 | 50.63 | 28.19 | 49.73 | 38.96 |
| C | 74.18 | 60.41 | 86.85 | 71.26 | 33.22 | 13.15 | 23.185 |
| D | 72.51 | 63.47 | 59.85 | 61.61 | 20.10 | 40.15 | 30.125 |
| Original VGG16 | 74.41 | 76.70 | 43.87 | 55.82 | 7.78 | 56.13 | 31.955 |

**Table 11**
Performance metrics observed for VGG16 over 30 epochs on Replay-Attack dataset for learning rate of $10^{-5}$.

| Scenario | Test Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FAR (%) | FRR (%) | HTER (%) |
|---|---|---|---|---|---|---|---|
| A | 99.81 | 99.79 | 99.40 | 99.59 | 0.065 | 0.6 | 0.3325 |
| B | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| C | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| D | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| Original VGG16 | 100 | 100 | 100 | 100 | 0 | 0 | 0 |

**Table 12**
Performance metrics observed for VGG19 over 30 epochs on NUAA dataset for learning rate of $10^{-5}$.

| Scenario | Test Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FAR (%) | FRR (%) | HTER (%) |
|---|---|---|---|---|---|---|---|
| A | 62.14 | 49.11 | 75.16 | 59.40 | 45.46 | 24.84 | 35.15 |
| B | 67.89 | 54.00 | 87.00 | 66.64 | 43.26 | 13.00 | 28.13 |
| C | 74.23 | 60.89 | 84.06 | 70.62 | 31.50 | 15.94 | 23.72 |
| D | 75.18 | 61.81 | 84.98 | 71.57 | 30.64 | 15.02 | 22.83 |
| Original VGG19 | 69.34 | 59.3 | 53.54 | 56.27 | 21.44 | 46.46 | 33.95 |

**Table 13**
Performance metrics observed for VGG19 over 30 epochs on Replay-Attack dataset for learning rate of $10^{-5}$.

| Scenario | Test Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FAR (%) | FRR (%) | HTER (%) |
|---|---|---|---|---|---|---|---|
| A | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| B | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| C | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| D | 100 | 100 | 100 | 100 | 0 | 0 | 0 |
| Original VGG19 | 100 | 100 | 100 | 100 | 0 | 0 | 0 |

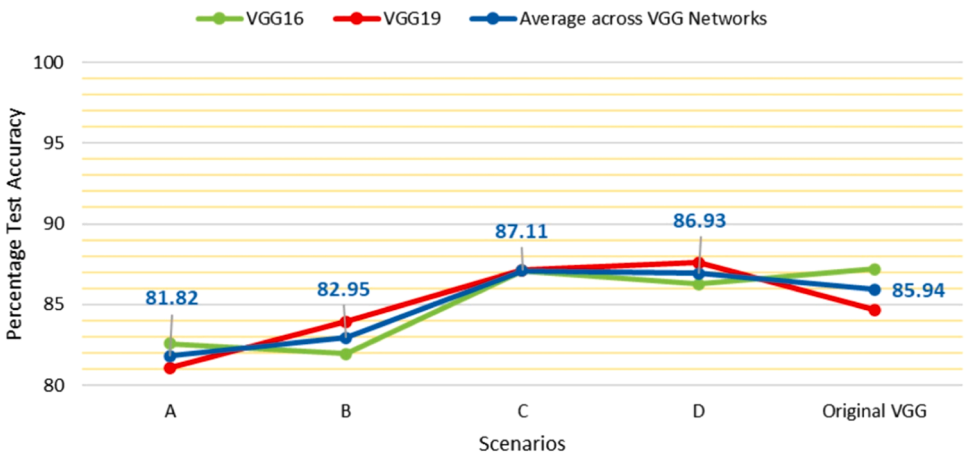


**Fig. 10.** Average percentage test accuracy of VGG16 and VGG19 across both NUAA & Replay-Attack datasets for learning rate of $10^{-5}$.

**Table 15**
Comparison of the existing DCNN models of literature survey with the best performing finetuned DCNN model observed in the current empirical assessment.

| Face Anti-spoofing Method | Pre-trained DCNN model | Performance Measure | Dataset Used | Test Accuracy (%) | HTER/ACER (%) | EER (%) |
|---|---|---|---|---|---|---|
| VGG16 (Lucena et al., 2017) | VGG16 | HTER, Test Accuracy | Replay-Attack | 99.04 | 1.20 | – |
| | | | 3DMAD | 100 | 0.00 | – |
| ResNet50 + RI-LBP (Chen et al., 2019) | ResNet-50 | HTER, EER | Replay-Attack | – | 2.6 | 2.3 |
| | | | NUAA | – | – | 0.5 |
| | | | CASIA-FASD | – | – | 4.4 |
| | | | MSU-MFSD. | – | – | 3.1 |
| Multiple Deep Features (Tang et al., 2018) | ResNet-18 | EER, ACER | CASIA-FASD | – | – | 2.2 |
| | | | Replay-Mobile | – | 0.0 | – |
| | | | OULU-NPU | – | 2.4 | – |
| ResNet50 + LSTM (Tu & Fang, 2017) | ResNet-50 | HTER, EER | CASIA-FASD | – | 1.22 | 1.0 |
| | | | Replay-Attack | – | 1.18 | 1.03 |
| VGG16 + LBP (Das et al., 2019) | VGG16 | Test Accuracy | SSIJRI | 92.05 | – | – |
| | | | Replay-Attack | 75.25 | – | – |
| | | | Replay-Mobile | 90.52 | – | – |
| | | | 3DMAD | 96.97 | – | – |
| VGG16 (Elloumi et al., 2020) | VGG16 | HTER | Replay-Attack | – | 2.5 | – |
| | | | CASIA-FASD | – | 0.0 | – |
| | | | Replay-Mobile | – | 0.0 | – |
| Proposed method VGG19 (Learning rate = $10^{-4}$, Scenario = "Original VGG") | VGG19 | Test Accuracy, HTER | Replay-Attack | 100 | 0.00 | – |
| | | | NUAA | 78.56 | 18.7 | – |

$$F1 - Score = \left(\frac{2 * Precision * Recall}{Precision + Recall}\right) \quad (6)$$

$$Accuracy = \left(\frac{TN + TP}{TN + TP + FN + FP}\right) * 100 \quad (7)$$

Where,
TP (True Positive) = Number of Real Faces predicted as Real.
TN (True Negative) = Number of Fake Faces predicted as Fake.
FP (False Positive) = Number of Fake Faces predicted as Real.
FN (False Negative) = Number of Real faces predicted as Fake

## 4. Results and discussion

For each scenario, we evaluated the model over 30 epochs and selected the model of a particular epoch having maximum test accuracy for comparing with models of other corresponding scenarios. Learning rates of $10^{-4}$ and $10^{-5}$ are employed for all scenarios. Performance metrics used to evaluate the scenarios are Test accuracy, Precision, Recall, F1-score, FAR, FRR, and HTER.

The graph in Fig. 9 depicts that for the learning rate of $10^{-4}$, scenario "Original VGG" outputs the highest accuracy among the considered scenarios. Thus, from Table 9 and Fig. 9, it can be jointly concluded that a combination of VGG19 and scenario "Original VGG" results in the best performance for face liveness detection, considering a learning rate of $10^{-4}$.

Similarly, for a learning rate of $10^{-5}$, from Table 14 and Fig. 10, it can be concluded that a combination of VGG16 and scenario "C" outputs second-best results compared to a combination of VGG19 and scenario "original VGG" with a learning rate of $10^{-4}$. By comparing graphs drawn in Figs. 9 and 10, it can be concluded that for all scenarios, higher test accuracy is obtained at a learning rate of $10^{-4}$ compared to the learning rate of $10^{-5}$.

Table 15 shows the comparison of a few of the existing DCNN models used in face spoofing detection methods proposed in the literature survey with the best-observed performance in the empirical assessment done in this paper with the help of two datasets. The performance comparison is not easy as the experimentation environment used in each existing method is different as well as the performance measures and datasets also differ. But if we compare the explorations done using

Replay-Attack datasets (as shown in Table 15), the testing accuracy and HTER observed in the proposed work presented here with VGG19 (with the scenario "Original VGG") is superior to the considered existing similar attempts from the literature (Chen et al., 2019; Das et al., 2019; Elloumi et al., 2020; Lucena et al., 2017; Tang et al., 2018; Tu & Fang, 2017).

## 5. Conclusion

Human face liveness detection is important in today's ambient computing scenario, where user identity is verified through the contactless biometric traits acquired. The paper has attempted the empirical performance assessment of six pre-trained deep CNN architectures (VGG16, VGG19, InceptionV3, DenseNet121, MobileNet, and Xception) for the human face liveness detection application. All pre-trained DCNN models are modified to make them suitable for human face liveness detection. The testing accuracy & HTER observed over a diverse number of epochs are used as the performance parameters for assessing the pre-trained DCNN models. Firstly, by finetuning the last two convolution blocks of VGG16 and VGG19 at the learning rate of $10^{-5}$, we obtained test accuracy of 100%, which is the highest for the Replay-Attack dataset compared to methods mentioned in Table 15.

Secondly, the aim to reduce the complexity of VGG networks was fulfilled by decreasing the number of convolution blocks in each scenario, as shown in Fig. 8. In this approach, from results, it is evident that the VGG19 network with the scenario "Original VGG" at a learning rate of $10^{-4}$ gives remarkable overall results in terms of test accuracy of 78.56% on the NUAA dataset and 100% for the Replay-Attack dataset. Closely following this result, the VGG16 network with scenario "C" at a learning rate of $10^{-5}$ outputs second-best results. Lastly, for all considered scenarios, higher test accuracies are obtained for the learning rate of $10^{-4}$. In the present research work, the proposed method has experimented on the datasets that contain only the photo attack and the video attack. To enrich the conclusion regarding the robustness level of the current finding, it can experiment on datasets containing mask attacks in the future.

## Author declaration

*[Instructions: Please check all applicable boxes and provide additional information as requested.]*

## Funding

No funding was received for this work.

a Dr. Sudeep D. Thepade
https://orcid.org/0000–0001–7809–4148

b Piyush Chaudhari
https://orcid.org/0000–0001–9023–0157

c Mayuresh Dindorkar
https://orcid.org/0000–0001–8089–3301

d Shalakha Bang
https://orcid.org/0000–0003–3472–9636

## CRediT authorship contribution statement

**Sudeep D. Thepade:** Conceptualization, Methodology, Writing – original draft, Supervision. **Mayuresh Dindorkar:** Data curation, Investigation, Visualization, Software, Validation. **Piyush Chaudhari:** Data curation, Investigation, Writing – original draft. **Shalakha Bang:** Data curation, Investigation, Visualization, Software, Validation.

## Declaration of Competing Interest

No conflict of interest exists.

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

## References

Chen, F., Wen, C., Xie, K., Wen, F., Sheng, G., & Tang, X. (2019). Face liveness detection: Fusing colour texture feature and deep feature. *IET Biometrics, 8*(6), 369–377. https://doi.org/10.1049/iet-bmt.2018.5235

Chingovska, I., Anjos, A., & Marcel, S. (2012). On the effectiveness of local binary patterns in face anti-spoofing. In A. Bromme, & C. Busch (Eds.), *Proceedings of the BIOSIG - international conference of biometrics special interest group* (pp. 1–7). IEEE.

Chollet, F. (2017). Xception: deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1800–1807). https://doi.org/10.1109/CVPR.2017.195, 2017.

Das, P. K., Hu, B., Liu, C., Cui, K., Ranjan, P., & Xiong, G. (2019). A new approach for face anti-spoofing using handcrafted and deep network features. In *Proceedings of the IEEE international conference on service operations and logistics, and informatics (SOLI)* (pp. 33–38). IEEE Press. https://doi.org/10.1109/SOLI48380.2019.8955089.

Elloumi, W., Chetouani, A., Charrada, T. B., & Fourati, E. (2020). *Anti-Spoofing in face recognition: Deep learning and image quality assessment-based approaches* (pp. 51–69). Cham: Deep Biometrics. Unsupervised and semi-supervised learning. Springer. https://doi.org/10.1007/978-3-030-32583-1_4

Howard, A. G., Menglong, Z., Chen, B., Dmitry, K., Wang, W., Weyand, T., et al. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. ArXiv abs/1704.04861.

Huang, G., Liu, Z., Maaten, L. V. D., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2261–2269). https://doi.org/10.1109/CVPR.2017.243, 2017.

Kekre, H. B., Thepade, S. D., & Chopra, T. (2010a). Face and gender recognition using principal component analysis. *International Journal on Computer Science and Engineering (IJCSE), 2*(4), 959–964.

Kekre, H. B., Thepade, S. D., Jain, J., & Agrawal, N. (2011a). Iris recognition using texture features extracted from walshlet pyramid. In B. K. Mishra (Ed.), *Proceedings of the ACM international conference & workshop on emerging trends in technology (ICWET '11)* (pp. 76–81). Association for Computing Machinery. https://doi.org/10.1145/1980022.1980038.

Kekre, H. B., Thepade, S. D., & Maloo, A. (2010b). Eigenvectors of covariance matrix using row mean and column mean sequences for face recognition. *International Journal of Biometrics and Bioinformatics (IJBB), 4*(2), 42–50.

Kekre, H. B., Thepade, S. D., & Maloo, A. (2011b). Face recognition using texture features extracted from Walshlet pyramid. *ACEEE International Journal on Recent Trends in Engineering and Technology (IJRTET), 5*(1).

Khade, S., Thepade, S. D., & Ambedkar, A. (2018). Fingerprint liveness detection using directional ridge frequency with machine learning classifiers. In *Proceedings of the fourth international conference on computing communication control and automation (ICCUBEA)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICCUBEA.2018.8697895.

Lucena, O., Junior, A., Moia, V., Souza, R., Valle, E.&, & Lotufo, R. (2017). *Transfer learning using convolutional neural networks for face anti-spoofing, 10317* pp. 27–34). Springer Cham. https://doi.org/10.1007/978-3-319-59876-5_4

Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. arXiv 1409.1556.

Szegedy, C., Vanhoucke, V., Loffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2818–2826). IEEE. https://doi.org/10.1109/CVPR.2016.308, 2016.

Tan, X., Li, Y., Liu, J., & Jiang, L. (2010). Face Liveness Detection from a single image with sparse low rank bilinear discriminative model. In *Computer vision – eccv 2010, 6316* pp. 504–517). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15567-3_37

Tang, Y., Wang, X., Jia, X., & Shen, L. (2018). Fusing multiple deep features for face anti-spoofing. In *Biometric recognition. CCBR 2018, 10996* pp. 321–330). https://doi.org/10.1007/978-3-319-97909-0_35

Thepade, S. D., & Bidwai, P. (2013). Iris recognition using fractional coefficients of transforms, wavelet transforms and hybrid wavelet transforms. In *Proceedings of the IEEE international conference on control, computing, communication and materials (ICCCCM)* (pp. 1–5). https://doi.org/10.1109/ICCCCM.2013.6648921

Thepade, S. D., & Khade, S. (2018). Novel fingerprint liveness detection with fractional energy of cosine transformed fingerprint images and machine learning classifiers. In *Proceedings of the IEEE Punecon*. IEEE,. https://doi.org/10.1109/PUNECON.2018.8745407.

Thepade, S. D., & Mandal, P. R. (2014). Novel iris recognition technique using fractional energies of transformed iris images using haar and kekre transforms. *International Journal of Scientific & Engineering Research, 5*(4), 305–308.

Tu, X., & Fang, Y. (2017). Ultra-deep neural network for face anti-spoofing. In *, 10635. Proceedings of the neural information processing. ICONIP* (pp. 686–695). Springer, Cham. https://doi.org/10.1007/978-3-319-70096-0_70.