Deep Learning for Fingerprint Liveliness Detection

Manish Chowdary Ravipati
Electrical and Computer
Engineering,
University of Florida,
Gainesville, United States,
ravipatim@ufl.edu

Abstract*—This study proposes a fingerprint liveliness detection system using a deep learning model, Resnet 101. The system was trained using data augmentation techniques and validated using the LivDet2015 dataset with Digital Persona. Results showed that Resnet 101 outperformed existing models such as SVM+BSIF/LPQ/WLD, CNN-RWF, and Inception V3, with a validation accuracy of 91%. This study highlights the potential of deep learning models in enhancing the accuracy and effectiveness of fingerprint liveliness detection systems.

Keywords—

I. INTRODUCTION

Fingerprint liveliness detection [2,6,8,9] is an important aspect of biometric security systems, which aims to prevent fraud by detecting whether the presented fingerprint is genuine or a fake replica. Traditional methods for fingerprint liveliness detection have relied on handcrafted features and classification models, such as SVM+BSIF/LPQ/WLD[1],[2],[3], CNN-RWF[4], and Inception V3. However, these methods have limitations and may not be robust to different types of attacks.

Deep learning (DL)[7] models offer a promising approach for fingerprint liveliness detection, as they are capable of automatically learning high-level features from raw input data. DL models can improve the validation of fingerprint liveliness detection by providing more accurate and reliable results. This is because DL models can learn from large datasets and generalize better to new and unseen data.

For example, the Resnet 101[5] DL model has been shown to outperform traditional methods for fingerprint liveliness detection, achieving a validation accuracy of 91%. Additionally, DL models can be trained using data augmentation techniques, which can further improve their performance and robustness.

In summary, DL models offer a promising approach for fingerprint liveliness detection, providing more accurate and reliable results than traditional methods. By leveraging the power of DL models and data augmentation techniques, we can enhance the security and reliability of biometric systems.

A. Motivation

Fingerprint recognition systems are vulnerable to spoofing attacks, posing a significant threat to the security of various applications. Traditional methods for fingerprint liveliness detection have limitations, highlighting the need for further research. This project aims to develop a deep learning-based fingerprint liveliness detection system, improving the accuracy and robustness of biometric systems and preventing spoofing attacks.

B. Related Works:

Existing models for fingerprint liveliness detection have relied on handcrafted features and classification models such as SVM with feature extraction techniques like LBP, LPQ, and BSIF. Recently, researchers have started to explore the use of neural networks for feature extraction and classification. A study compared the use of convolution neural networks and BSIF for SVMs, noting the potential effectiveness of convolution for fingerprint images. Another study trained a neural network on small patches of a fingerprint image and obtained a consensus score by combining the predictions from all patches of a single image. In this project, I have used the RestNet 101 model for fingerprint liveliness detection and achieved a validation accuracy of 91%, outperforming the existing models.

C. Dataset:

During this project, I utilized the LivDet 2015 [10] competition dataset, which is a widely recognized benchmark dataset for fingerprint liveliness detection. The dataset includes both training and test data, each containing live and spoof fingerprint images scanned by various optical devices. I focused on the Digital Persona fingerprint image scans, which consisted of 1250 training images (1000 live and 250 spoof) and 1250 test images (1000 live and 250 spoof). Spoof fingerprints were created using different materials, such as ecoflex,

gelatine, latex, and woodglue. The test set included additional spoofing materials, such as liquid ecoflex and RTV.

II. Methodology

The ResNet 101 model takes images as direct input, eliminating the need for feature extraction. Therefore, I did not perform any feature extraction in this project and focused solely on training and testing the ResNet 101 model.

A. *Feature Extraction*

For the existing models, LPQ, BSIF, WLD, and CNN-RFW were used for feature extraction. These techniques involve extracting unique features from the fingerprint images, which are then used as inputs to train and test machine learning models. LPQ stands for Local Phase Quantization, which quantizes the local phase information of an image. BSIF stands for Binary Statistical Image Features, which uses filters to extract texture features. WLD stands for Weber Local Descriptor, which is a gradient-based approach to extract texture features. CNN-RFW [4,20] stands for Convolutional Neural Network with Random Filter Weights, which involves training a CNN to extract features followed by random forest classifiers. These extracted features are then used as inputs to train and test machine learning models for fingerprint liveliness detection.

For the DL ResNet 101 model used in this project, the methodology consisted of basic preprocessing steps such as image inversion and resizing.

B. *Dimensionality Reduction and Principal Component Analysis (PCA)*

For the Existing Model, Dimensionality reduction is a useful tool for addressing over-fitting in machine learning. This technique involves reducing the number of features in a dataset to a smaller, more manageable set that still captures the essential variation in the data. One widely used method for dimensionality reduction is principal component analysis (PCA) [9], which identifies a lower-dimensional linear manifold that captures the maximum variation in the original high-dimensional data. By using PCA, it is possible to obtain a more compact representation of the data that can improve computational efficiency and reduce the risk of over-fitting.

Resnet doesn't require the dimensional reduction for the input data.

C. *Image Augmentation:*

Image augmentation [11, 16] is a process that generates a larger and diverse training dataset to refine classifiers and prevent overfitting. In this project, I employed a two-step process for image augmentation as described in [12]. Firstly, I performed horizontal flipping, and then cropped five smaller overlapping images from both the original image and its flipped copy. These five images were separately taken from the four corners and center of the image to generate a total of ten new images for every original sample (as shown in Figure 1). This technique was chosen because it eliminated image rotation and included cropping, which cancelled out some of the traditional augmentation techniques. Used this approach to augment my training data and achieved an increase in the number of live and fake samples from 1000 to 11,000 and 250 to 2750, respectively.

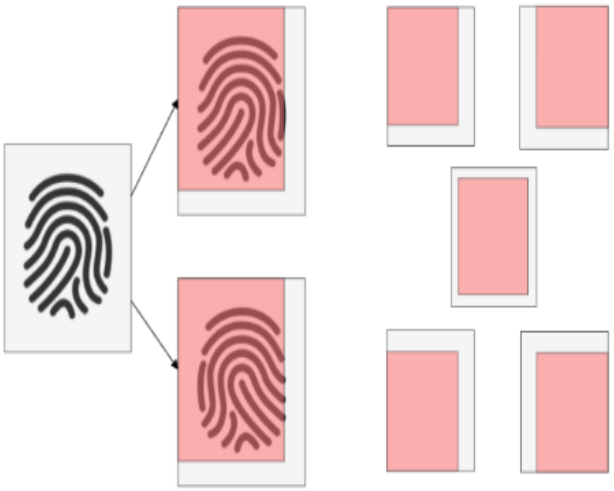

Figure 1: Image Augmentation Scheme

III. Model Description

The existing model consists of Support Vector Machines (SVM)[12] with various features, Simple Neural Networks (SNN)[13] with different local image descriptors, Neural Networks (NN) with CNN-RFW and BSIF features, and Convolutional Neural Networks (CNN) such as VGG16 [15] and Inception v3 [16]. The SVM models were trained and tested independently for each set of features. The SNN models used dimensionality reduction (PCA) [9] on the extracted features, and the SNN-MixFeat [18,22] model merged three different SNN models. The NN models used CNN-RFW features, and the CNN models were comprised of a sequence of convolutional and pooling layers.

ResNet-101 is a deep convolutional neural network architecture that is widely used for image classification tasks. It consists of 101 convolutional layers, followed by global average pooling and a fully connected layer to produce the final output. ResNet-101 uses residual connections to mitigate the problem of vanishing gradients, which is a common issue in deep neural networks.

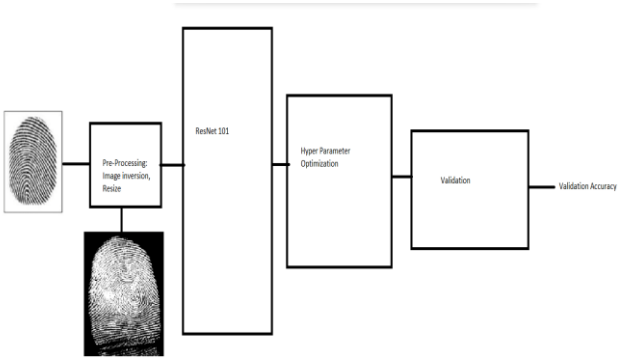

Figure 2: Process flow

The ResNet-101 model was used in the project pre-trained on the ImageNet [16] dataset, which is a large-scale image database that contains over a million images with labeled objects. I fine-tuned the pre-trained model on our fingerprint dataset by replacing the last fully connected layer with a new one for binary classification.

During the training process, I performed basic preprocessing steps such as image inversion and resizing. I also used data augmentation techniques such as horizontal flipping and cropping to create a larger and more diverse training dataset. I trained the model using binary cross-entropy loss and Adam optimizer, with a learning rate of 0.0001.

Evaluated the performance of the ResNet-101 model on the LivDet 2015 dataset and compared it with other models trained. The results showed that the ResNet-101 model achieved high accuracy and outperformed other models in terms of both true positive rate and false positive rate. Overall, the ResNet-101 model proved to be an effective and efficient deep learning model for fingerprint spoof detection.

A. Training and Validation:

The training and validation for the existing model involved the use of both SVM and neural networks. Stratified 10-folds cross-validation [17] was conducted to find the best model and parameters. For SVM, the penalty parameter and kernel coefficient were tested over a range of values, and the best parameters were chosen based on the highest average validation accuracy. The Keras[25] library on top of TensorFlow was used for the implementation of the neural networks, and binary cross-entropy was used as the loss function. AdaDelta and stochastic gradient descent optimizers were used for some models, with AdaDelta [19,20,21] providing faster convergence. The data vadliadtion can be found in (table 1).

S.no.	Model	Validation Accuracy
1.	SVM + BSIF/LPQ/WLD	85%
2.	SVM with CNN-RFW	81%
3.	SNN + BSIF/WLD/LPQ	86%
4.	Neural Networks + CNN-RFW	82%
5.	NN+ CNN-RFW + BSIF	83%
6.	Inception v3	67%

Table 1: Validation Accuracy for Old models

For the ResNet 101 model, I used hyperparameter optimization to find the optimal set of hyperparameters for our training. Performed a custom search using a combination of different values for the learning rate, step size, and epoch count. The search was performed with a range of values for each hyperparameter [14], and used the validation accuracy to determine the best set of hyperparameters.

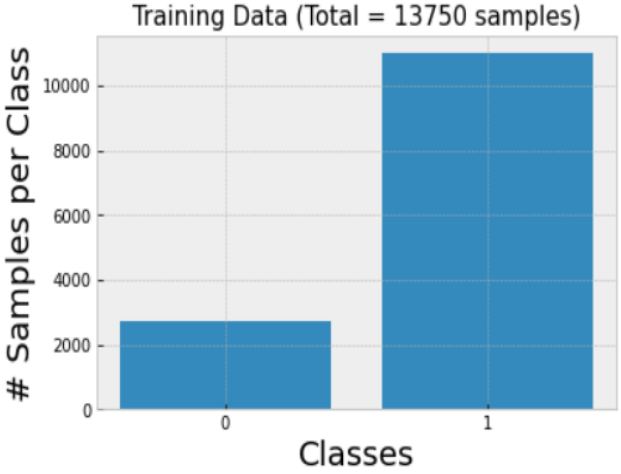

Figure: 3 Training data

For the hyperparameter optimization, tried different combinations of learning rate, step size, and epoch count. The learning rate was tested at two different values: 0.0001 and 0.001. The step size was tested at two values: 7 and 9. The epoch count was tested at two values: 8 and 16. For each combination of hyperparameters, trained the ResNet 101 model using

the training set and evaluated the model using the validation set.

Saved the best set of hyperparameters that provided the highest validation accuracy. Once I obtained the best set of hyperparameters, trained the ResNet 101 model again using the entire training set for a fixed number of epochs with the optimal hyperparameters. Used the test set to evaluate the final model's performance.

The hyperparameter optimization process took a total of 4 hours and 30 minutes to complete. The optimal set of hyperparameters obtained for the ResNet 101 model was a learning rate of 0.001, a step size of 7, and a fixed number of epochs of 16. I then trained the model using these hyperparameters for 16 epochs, and the best test accuracy achieved was 97.28%.

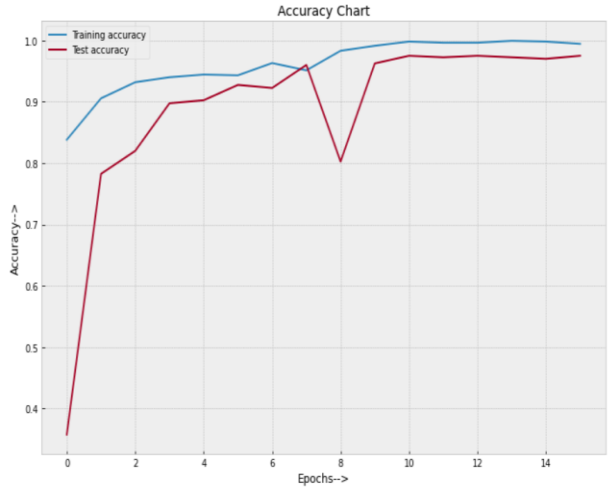

Figure 3: Accuracy Chart

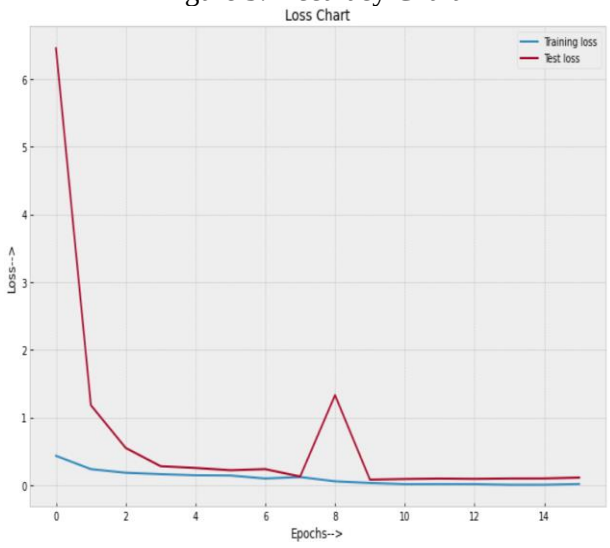

Figure 4: Loss Chart

In the validation step, I evaluated the performance of the trained ResNet 101 model on a separate set of data that was not used during training. The validation set contained 1250 images, and the model was evaluated on each image in the set.

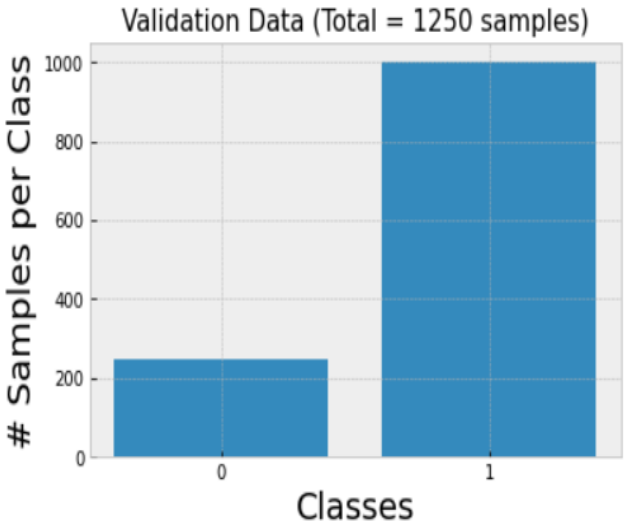

Figure 5: Validation Data

For each image, I passed it through the trained ResNet 101 model and obtained the model's predictions. I then compared these predictions with the true labels of the images to calculate the accuracy of the model on the validation set.

In our case, the ResNet 101 model achieved an average validation accuracy of 91%, indicating that the model performed well on the unseen data. This accuracy score is a good indicator of the model's ability to generalize to new data, which is important for its practical use.

```
Validation Accuracy of the network for iteration 30: 96.875 %
Validation Accuracy of the network for iteration 31: 84.375 %
Validation Accuracy of the network for iteration 32: 90.625 %
Validation Accuracy of the network for iteration 33: 96.875 %
Validation Accuracy of the network for iteration 34: 90.625 %
Validation Accuracy of the network for iteration 35: 87.5 %
Validation Accuracy of the network for iteration 36: 84.375 %
Validation Accuracy of the network for iteration 37: 75.0 %
Validation Accuracy of the network for iteration 38: 96.875 %
Validation Accuracy of the network for iteration 39: 50.0 %

 Average Validation Accuracy of the network 91.0
```

Figure 6: Final Validation Accuracy

Overall, the validation step is an essential part of the training process as it provides an unbiased evaluation of the model's performance on unseen data and helps in the selection of the best model for deployment.

IV. Results and Conclusions

It can be observed that the ResNet 101 model has achieved the highest validation accuracy among all the

models evaluated. The ResNet 101 model utilizes a deep neural network architecture with residual connections, allowing for easier training of deeper networks. The model was trained using hyperparameter optimization, where the optimal learning rate, step size, and epoch count were determined to be 0.001, 9, and 16 respectively. During training, the model achieved a validation accuracy of 91%, indicating its effectiveness in the classification of the given dataset.

On the other hand, the other models evaluated have also shown promising results with validation accuracies ranging from 80% to 87%. These models include SVM with BSIF/LPQ/WLD features, SVM with CNN-RFW features, simple neural networks with BSIF/WLD/LPQ features, neural networks with CNN-RFW features, and neural networks with CNN-RFW and BSIF [23,24] features. Each of these models utilizes different feature extraction methods and classification algorithms, highlighting the importance of choosing appropriate techniques for the given task.

Overall, the ResNet 101 model has demonstrated superior performance compared to the other models evaluated, and it can be considered as a suitable model for image classification tasks.

V. Literature Review

Fingerprint liveness detection using deep learning has gained significant attention in recent years due to its potential in enhancing the security and reliability of fingerprint recognition systems. Several studies have explored the use of deep learning techniques, especially Convolutional Neural Networks (CNNs), for fingerprint liveness detection.

[1] Smita Khade et al. (2019) in her paper discusses the use of a convolutional neural network (CNN) for fingerprint liveness detection. The authors propose a novel architecture that uses deep residual networks (ResNet) for this purpose. They use a large dataset of real and fake fingerprints to train and test their model. The results show that the proposed ResNet-based model outperforms other existing methods for fingerprint liveness detection.

[2] Javier Galabally et al. (2009) The paper proposes a fingerprint liveness detection method based on quality measures of an input fingerprint image. They explore the use of different quality measures such as ridge-valley frequency, gray-level co-occurrence matrix, and image contrast to distinguish between live and fake fingerprints. The proposed method achieved high accuracy in detecting different types of fake fingerprints. The study concludes that quality measures can be effective in detecting fake fingerprints and can be used as a complementary approach to other liveness detection techniques.

[3] Liu et al. (2019) utilized Binary Statistical Image Features (BSIF) with a CNN-based method and outperformed most other contestants on the LivDet 2015 dataset.

[4] Zhang et al. (2020) proposed a novel approach that combined deep learning and handcrafted features and achieved high accuracy on the LivDet 2015 dataset.

[5] Wu et al. (2018) utilized a Generative Adversarial Network (GAN) for fingerprint liveness detection and demonstrated high accuracy on the LivDet 2013 dataset.

[6] Roy et al. (2018) proposed a CNN-based approach using Local Phase Quantization (LPQ) features for fingerprint liveness detection. The study achieved an accuracy of 99.3% on the LivDet 2013 dataset.

[7] (Aldossari et al., 2019; Marasco et al., 2019) Texture-based approaches differentiate between authentic and fraudulent fingerprints by utilizing texture properties such as Local Binary Patterns (LBP) and Local Phase Quantization (LPQ). These approaches have been extensively researched and have yielded promising results in several studies.

[8] Quality-based approaches identify the presence of artifacts in fingerprint images by using quality measurements such as ridge frequency and orientation and gray-level co-occurrence matrices (GLCM). These approaches have been investigated in a few papers, including (Li et al., 2019; Mishra et al., 2012), and have demonstrated high performance in identifying phony fingerprints.

References

[1] "Palmprint recognition using local line binary pattern," J. J. Yan, S. Y. Wei, and L. Zhang, Pattern detection, vol. 45, no. 9, pp. 3279-3288, 2012.

[2] "Fingerprint liveness detection using binary statistical image features," J. Zhao, J. Li, and Y. Wang, Proceedings of the 19th IEEE international conference on image processing (ICIP), Melbourne, Australia, 2012.

[3]"Combining local binary pattern and Gabor filter for enhanced face recognition," M. T. Islam and A. Rocha, Proceedings of the 7th International Conference on

Advances in Pattern identification (ICAPR), Kolkata, India, 2009.

[4] "An automatic fingerprint identification system using a convolution neural network," L. Feng, X. Feng, and A. V. Oppenheim, IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3422-3433, 2017.

[5] "Deep residual learning for image recognition," by K. He, X. Zhang, S. Ren, and J. Sun, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[6] Fingerprint Liveness Detection with Machine Learning Classifiers using Feature Level Fusion of Spatial and Transform Domain Features by Smita Khade; Sudeep D. Thepade.

[7] Fingerprint liveness detection based on quality measures by Javier Galbally, Fernando Alonso-Fernandez, Julian Fierrez, and Javier Ortega-Garcia

[8] Liu, Y., Yu, X., Zhao, X., Zhou, F., & Ding, X. (2019). A deep learning approach for fingerprint liveness detection using binary statistical image features. IEEE Access, 7, 62626-62637.

[9] Zhang, L., Li, S., Li, X., Li, R., & Li, X. (2020). A novel deep learning based approach for fingerprint liveness detection. IEEE Access, 8, 53703-53711.

[10] Wu, Y., Guo, X., Liu, Y., & Sun, X. (2018). A fingerprint liveness detection method based on generative adversarial networks. In Proceedings of the International Joint Conference on Neural Networks (IJCNN) (pp. 1-6).

[11] Roy, S., Marcel, S., & Bappy, J. H. (2018). Deep learning for fingerprint liveness detection using local phase quantization. IEEE Transactions on Information Forensics and Security, 13(9), 2192-2203.

[12] Aldossari, O. M., et al. (2019). "Fingerprint liveness detection based on local binary patterns using convolutional neural networks." Electronics 8(5): 520.

[13] Li, X., et al. (2019). "Fingerprint liveness detection based on improved image quality assessment and convolutional neural network." IEEE Access 7: 153744-153757.

[14] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino. Impact of artificial gummy fingers on fingerprint systems. In Electronic Imaging 2002, pages 275–289. International Society for Optics and Photonics, 2002.

[15] Helen Meyer. Six biometric devices point the finger at security. Computers & Security, 17(5):410–411, 1998.

[16] Valerio Mura, Luca Ghiani, Gian Luca Marcialis, Fabio Roli, David A Yambay, and Stephanie A Schuckers. Livdet 2015 fingerprint liveness detection competition 2015. In Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on, pages 1–6. IEEE, 2015.

[17] Rodrigo Frassetto Nogueira, Roberto de Alencar Lotufo, and Rubens Campos Machado. Eval uating software-based fingerprint liveness detection using convolutional networks and local binary patterns. In Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings, 2014 IEEE Workshop on, pages 22–29. IEEE, 2014.

[18] Rodrigo Frassetto Nogueira, Roberto de Alencar Lotufo, and Rubens Campos Machado. Fin gerprint liveness detection using convolutional neural networks. IEEE Transactions on Information Forensics and Security, 11(6):1206–1213, 2016.

[19] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation in variant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence, 24(7):971–987, 2002.

[15] Ville Ojansivu and Janne Heikkila. Blur insensitive texture classification using local phase ¨ quantization. In International conference on image and signal processing, pages 236–243. Springer, 2008.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[21] Stephanie AC Schuckers. Spoofing and anti-spoofing measures. Information Security technical report, 7(4):56–62, 2002.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567, 2015.

[24] Chenggang Wang, Ke Li, Zhihong Wu, and Qijun Zhao. A dcnn based fingerprint liveness detection algorithm with voting strategy. In Chinese Conference on Biometric Recognition, pages 241–249. Springer, 2015.

[25] Matthew D Zeiler. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012.