JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 605-616 (2021)
DOI: 10.6688/JISE.202105_37(3).0007

# Anti-Spoofing of Live Face Authentication on Smartphone

TZ-CHIA TSENG[1], TENG-FU SHIH[2] AND CHIOU-SHANN FUH[2]
[1]*Graduate Institute of Biomedical Electronics and Bioinformatics*
*College of Electrical Engineering and Computer Science*
[2]*Department of Computer Science and Information Engineering*
*National Taiwan University*
*Taipei, 106 Taiwan*
*E-mail: r05945052@ntu.edu.tw; g104018004@smail.nchu.edu.tw; fuh@csie.ntu.edu.tw*

Our proposed method is capable of authenticating the input image is from real user or spoofing attack, including paper photograph, digital photograph, and video, using only the Red, Green, Blue (RGB) frontal camera of common smart phone, without the help of depth camera or infrared thermal sensor. We first capture live faces in each frame of input video streams by single shot multi-box detector then feed into our designed convolution neural network after certain data augmentation and finally obtain a well-trained spoof face classifier. Finally, we compared to Parkin and Grinchuk's results, using dataset CASIA-SURF [1], and compare the result of vgg16, InceptionNet, ResNet, DenseNet and MobileNet in CASIA-SURFT dataset.

***Keywords:*** spoofing attack, single shot multi-box detector, data augmentation, VGG-16, stereo matching

## 1. INTRODUCTION

User authentication is a fundamental security mechanism. However, the most widely used certificate for authentication, passwords, have widely known drawbacks in security and usability: strong passwords are difficult to memorize, whereas convenient ones provide only weak protection; other certificates also popular, such as keys and cards, are troublesome to bring in hand but hassle-free to lose at anywhere [2].

Among biometric recognition, face recognition is the most common and widely used biometric features as information from the face can be extracted easily without any physical contact, and almost all smart phones are equipped with a front-facing camera.

In contrast with other biometrics (*e.g.* iris recognition) that are difficult for adversaries to acquire and duplicate, human faces can be easily captured and reproduced, which makes face authentication systems vulnerable to attacks. Since traditional face recognition systems do not consider the existence of an adversary, many studies have revealed that these systems are vulnerable to spoofing attacks [3]. A well-known example is a 2D spoofing attack, which misleads a system by using a 2D facial duplicate of a valid user. Whereas multiple cameras can compute disparity and thus depth, single camera cannot distinguish between a high-definition replication of a subject picture and the live subject himself. Geometrically, those two situations are mathematically equivalent from the point of view of a single camera [3]. As an image or a video of a person is easily obtainable and highly reproducible [4], 2D spoofing attack is one of the most common attacks. There are three types of 2D spoofing attacks, namely photo attack, video attack, and mimic mask attack.

Received July 2, 2019; revised September 7, 2019; accepted October 14, 2019.
Communicated by Chu-Song Chen.

Photo attack evades the detection by using a picture of a legitimate user on a piece of paper, or an electronic screen, while video attack misleads the system by using a video of an authorized person on electronic devices [5]. In mimic mask attack, an adversary camouflages as an authorized person by wearing a 3D mask [4]. 3D masks are expensive to build and are rare in real applications. Hence, in this paper, we focus on 2D presentation attacks including prints, photographs, and videos.

## 2. RELATED WORK

### 2.1 Overview

As far as we reviewed, prior face anti-spoofing works can be categorized into three groups: texture-based methods, motion-based methods, and deep learning methods.

Since most face recognition systems adopt only RGB cameras, using texture information has been a natural approach to tackling face anti-spoofing. Texture analysis methods explore patterns that depict the quality of image data. In [6], Local Binary Patterns (LBP) are used for the texture features to analyze unnatural patterns in spoof samples. Other common local features that have been used in prior work include HOG, DoG, SIFT and SURF [7]. Since the technique is specific to just photos and does not address video based or electronic screen attack [8] use Moire patterns generated as the results of recapture of videos and photos to discriminate the live face vs a spoof one in a video replay attack scenario only.

Some approaches attempt to leverage the spontaneous face motions. Motion-based methods use movement information such as eye blinking and lip movement to distinguish a real face from spoof. For example, [9] explores conditional random field to model different stages of eye blinking. The underlying assumption is that real faces show different motion patterns compared to a spoof one. Additionally, eye-blinking is one cue proposed in [10], to detect spoof attacks such as paper attack. In [11], Kollreider *et al.* use lip motion to monitor the face liveness. Bao *et al.* [12] present a countermeasure using optical flow fields that estimate the difference between 2D photograph attacks and 3D real faces.

Several other saliency-based methods [13] use attributes to differentiate between the 3D real face and a 2D fake face for spoofing detection. However advanced masked attacks can pose a fake face as a 3D real face and can be very challenging for the salience detection schemes to detect.

Based on two main assumptions that photograph is always flat and a spoof photo is always smaller than a real face, [14] proposed an authentication method using frequency analysis. Because of the variations of expressions and movements, the frequency components of a real face are higher than the spoof photo. Those assumptions however do not hold in case of CASIA and REPLAY ATTACK datasets where even the spoof photos are of same size as the real face. Further, video attacks and warping eliminate the possibility of detecting liveness based on the flatness conditions [15].

In addition, some works have combined different concepts together. In [16], texture, motion and liveness features are combined together to detect spoofing attacks on PRINT-ATTACK database. In [17], it is shown that a combination of spatial and temporal processing of videos can amplify subtle variations in the faces sufficient to detect liveness.

They propose a multiscale approach, Eulerian-based method, to magnify motion without feature tracking or motion estimation.

Most of the above anti-spoofing techniques concentrate on a particular category of attacks and thus the information on the kind of attack is to be inferred in prior. An anti-spoofing algorithm based on Haralick texture features proposed in [18] addresses 3D masks, print attacks and replay attacks but lacks the results of cross-database experiments.

In the recent, Yaojie proposed to use spatial information of living have more effective features than prosthetic [19] and use deep tree learning in an unsupervised fashion [20].

### 2.2 Stereo Matching for Disparity and Depth

The stereo matching component uses the rectified stereo-image pair and computes disparity, error, and confidence images. Using the stereo images, we can calculate the disparity map by estimating the distance moved by a particular point in the left and right images.

#### 2.2.1 Local methods

Current FPGA [21] technology offers thousands of small logic blocks embedded in the connection matrix. This allows arbitrary computation blocks to be constructed from basic computing blocks through parallel circuit connections [22].

The local methods search for a point by exploiting the surroundings of a point, usually done via a block (5×5, 7×7 pixels). The whole block is matched with similar sized blocks on the horizontal axis of the second image and the block with the greatest similarity provides us with the point's location in the two images. Usually many optimization and color correlation functions are used for matching the windows. Some of the popular functions for matching are Hamming's Distance [23], computation on Census Transformed images, weights windows, adaptive weighted windows. One of the methods with low computation time and higher accuracy is the Multi-Block matching [24].

#### 2.2.2 Global methods

Global methods differ from local ones in that they express the smoothness assumption explicitly via a smoothness term. These methods involve an energy function $E(D)$ that measures the quality of the disparity map $D$. Later, the energy function is optimized to find the disparity map with the lowest energy and of the highest quality. The optimization of the energy function can be done via several ways – dynamic programming, graph cuts, message passing, and so on. The algorithms in themselves have high complexity due to the high dimensional variable incorporated in the energy function, since each pixel disparity value is a variable. The best methods of global matching are approximately 100 times slower than those of local matching [25].

#### 2.2.3 Semi-global methods

Semi-Global Matching [26] successfully combines concepts of global and local stereo methods for accurate, pixel-wise matching at low runtime. The method reduces the multi-

dimension approach to one dimension. This reduces the computational complexity of the global methods with path-wise optimization many folds while retaining the accuracy of the global methods.

## 3. METHODOLOGY

We first collect the regions of interest (ROI) of spoof and real images using SSD face detector from input videos through the webcam. After several data augmentation steps, we then feed the data into our CNN model derived from VGG16. Finally, we can use this model to authenticate images input from the webcam.

### 3.1 Single Shot Multi-Box Detector (SSD) [27-29]

SSD (Single Shot Multi-Box Detector) is a popular algorithm in object detection. A typical CNN [30, 31] network gradually shrinks the feature map size and increases the depth as it goes to the deeper layers. The deep layers cover larger receptive fields and construct more abstract representation, while the shallow layers cover smaller receptive fields. It is a common trick use in YOLO [32] and fast RCNN.

#### 3.1.1 Training

For every feature point, we generate a number of priors, which are then used to match ground truth boxes to determine the labels and bounding boxes.

The loss function is the combination of classification loss and regression loss. The regression loss used here is Smooth-$L_1$ loss, which is the same as Faster RCNN and Fast RCNN. Pytorch has documentation for Smooth-$L_1$ Loss.

According to [33], Each training image is randomly sampled by:

1. Entire original input image
2. Sample a patch so that the overlap with objects is 0.1, 0.3, 0.5, 0.7 or 0.9
3. Randomly sample a patch

The size of each sampled patch is [0.1, 1] or original image size, and aspect ratio from 1/2 to 2. After the above steps, each sampled patch will be resized to fixed size and maybe horizontally flipped with probability of 0.5, in addition to some photo-metric distortions.

#### 3.1.2 Prediction

Prediction is simple. By feeding an image into the network, every prior will have a set of bounding boxes and labels. Remember we boost the number of positive priors by matching one object to multiple priors? Now we have multiple priors to predict the same object. To remove the duplicates, NMS (Non-Maximum Suppression) [34] used.

#### 3.1.3 The drawbacks

Shallow layers in a neural network may not generate enough high level features to

predict for small objects. Therefore, SSD performs worse for smaller objects than larger objects [28].

The need of complex data augmentation also suggests it needs a large number of data to train. For example, SSD does better for Pascal VOC if the model is pre-trained on Common Objects in COntext (COCO) [35] dataset so make sure our model is pre-trained on big datasets such as Pascal VOC, COCO and Open Images [36] before training it on our own data.

### 3.2 Histogram Equalization

Histogram Equalization [37] is a computer image processing technique used to improve contrast in images.

Histogram equalization cannot be applied separately to the Red, Green, and Blue components of the image as it leads to dramatic changes in the image's color balance. However, if the image is first converted to another color space, like Hue, Saturation, Lightness / Hue, Saturation, Value (HSL/HSV) color space, then the algorithm can be applied to the luminance or value channel without resulting in changes to the hue and saturation of the image [9].

The self-quotient [38] image has been proposed as an illumination invariant feature, it is another popular preprocessing technique.

Adaptive histogram equalization [39] differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.

### 3.3 Data Augmentation [40]

The issue we wanted to deal with the most by data augmentation is the illumination variation in different situations. We use the function ***imageDataGenerator()*** in Keras to implement data augmentation. The parameters we set:

rotation_range = 20, zoom_range = 0.15, width_shift_range = 0.2, height_shift_range = 0.2, shear_range = 0.15, horizontal_flip = True, fill_mode = "nearest".

### 3.4 Neural Network

The remarkable thing about VGG-16 [41] net is that they said, "Instead having so many hyper parameters, let's use a much simpler network where you focus on just having convolution [42] layers that are just 3 by 3 filters with stride 1 and always use the SAME padding, and make all your max polling layers 2 by 2 with a stride of 2".

Our NN is composed of 16 convolution layers, 3 fully connected layer, and a softmax [43] layer. We first set the size of kernel to 5 to capture larger features, and then to 3 after several layers to capture subtler feature (Fig. 1).

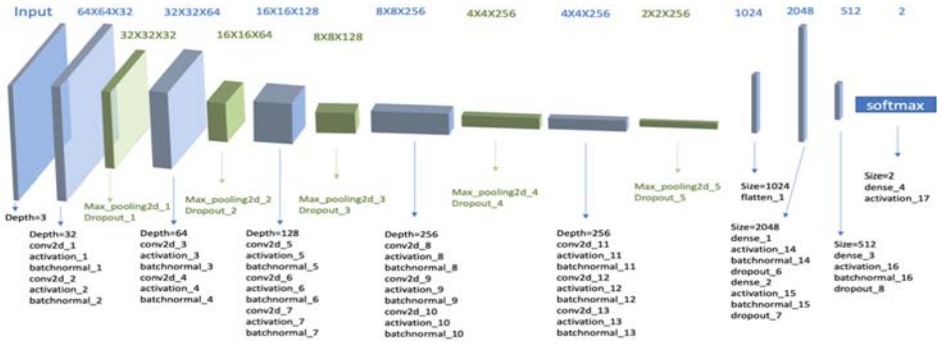


Fig. 1. The architecture of our proposed neural network.

## 4. EXPERIMENTAL RESULT

The authenticating system is built on the environment in Table 4.1.

We use OpenCV to implement most computer vision tasks and employ Keras as our deep learning framework to predict the probability of live face.

**Table 4.1. The environment to develop and conduct experiments is shown in this table.**

| | |
|---|---|
| Operating System | Ubuntu 18.04 |
| Central Processing Unit | Intel® Core™ i7-8550U CPU @ 1.80GHz × 8 |
| Programming Language | Python with OpenCV 4.0.0 |
| Graphic Processing Unit | Nvidia GeForce GTX 1050/PCIe/SSE2 |
| Deep Learning Framework | Keras 2.2.4 |

### 4.1 Evaluation

Fig. 2 is the training loss and accuracy on dataset collected from various scenarios including our laboratory, office, classrooms, wild, and so forth.

Table 4.2 is the testing result of our proposed model. Though the sensitivity and specificity are not perfect, our proposed model truly do a good job in several usability tests.

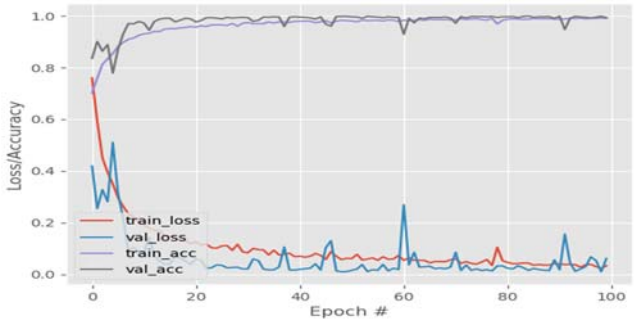


Fig. 2. The training loss and accuracy on dataset.

**Table 4.2. The testing result: percentage (number of frames / number of total frames).**

| | Predicted Real | Predicted Spoof |
|---|---|---|
| Actual Real | 93.73% (= 13,639/14,551) | 6.27% (= 912/14,551) |
| Actual Spoof | 4.79% (= 1,313/27,411) | 95.21% (= 26,098/27,411) |

## 4.2 Results and Observations

Compared to Parkin and Grinchuk's results [1], though our true positive rate (TPR) does not achieve 100.00% at $10^{-2}$ FPR (their model trained with RGB+IR+Depth); however, while taking only RGB into consideration, our TPR (Fig. 3) 99.69% outperforms their 71.74% at $10^{-2}$ FPR (Table 4-3). Result of InceptionV3, ResNet, DenseNet and MobileNet in CASIA-SURFT dataset (Fig. 4). The result of self-quotient image preprocessing (Fig. 5) and the result of original image in SiW-M dataset (Fig. 6).

**Table 4.3. TPR at FPR (Parkin and Grinchuk's result).**

| Modality | TPR at FPR | | |
|---|---|---|---|
| | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ |
| RGB | 71.74 | 22.34 | 7.85 |
| IR | 91.82 | 72.25 | 57.41 |
| Depth | 100.00 | 99.77 | 98.40 |
| RGB+IR+Depth | 100.00 | 100.00 | 99.87 |

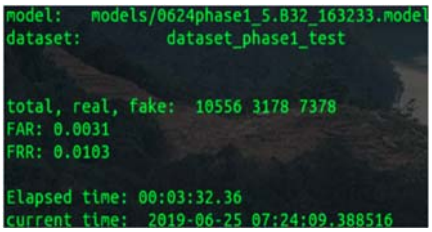


Fig. 3. Result of our method in CASIA-SURFT dataset.

| Confusion matrix (InceptionV3) | | predict | |
|---|---|---|---|
| | | fake | real |
| true | fake | 2046 | 7 |
| | real | 6 | 868 |

| Confusion matrix (ResNet50) | | predict | |
|---|---|---|---|
| | | fake | real |
| true | fake | 2042 | 11 |
| | real | 2 | 872 |

| Confusion matrix (DenseNet121) | | predict | |
|---|---|---|---|
| | | fake | real |
| true | fake | 2052 | 1 |
| | real | 10 | 864 |

| Confusion matrix (MobileNet) | | predict | |
|---|---|---|---|
| | | fake | real |
| true | fake | 2030 | 23 |
| | real | 80 | 794 |

Fig. 4. Result of InceptionV3, ResNet, DenseNet and MobileNet in CASIA-SURFT dataset.

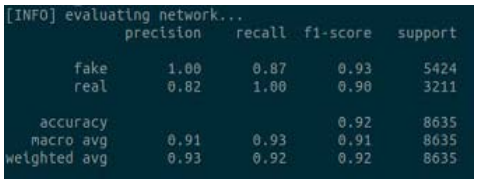


Fig. 5. Result of our method in SiW-M dataset.

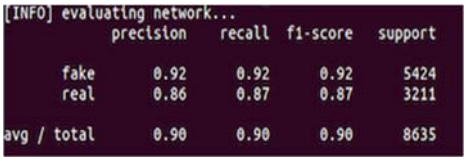


Fig. 6. Result of our method (self-quotient) in SiW-M dataset.

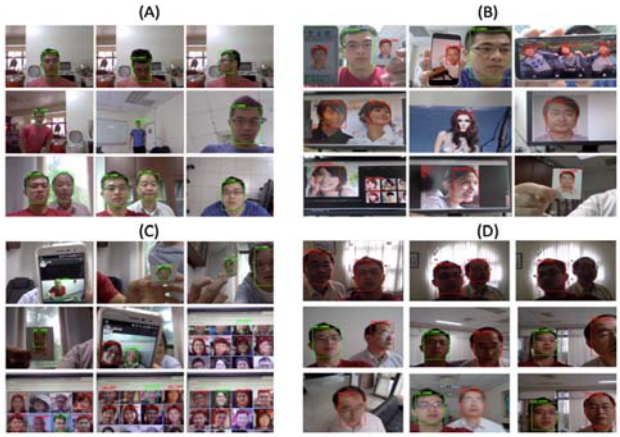


Fig. 7. Snapshots of testing results in (a) true positive group; (b) true negative group; (c) miss detection group; (d) false alarm group.

Our proposed model is invariant to perspectives of authenticating live face and whatever media (Fig. 7).

## 5. CONCLUSION AND FUTURE WORK

In this paper, we develop a real-time live face authentication system, which can authenticate live face from spoof one in nearly every scenario successfully. The authentication algorithm can handle general issues, such as the target face is occluded by other person or barriers; or brightness and illumination variance.

However, there are some known drawbacks that our algorithm is not perfect under limited data and the difference between training and testing scenario. These issues are caused by the difficulty of deriving other perspectives from the original angle and simulating environments that are not in the training data.

## REFERENCES

1. A. Parkin and O. Grinchuk, "Recognizing multi-modal face spoofing with face recognition networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
2. P. P. Chan, W. Liu, D. Chen, D. S. Yeung, F. Zhang, X. Wang, and C. C. Hsu, "Face liveness detection using a flash against 2D spoofing attack," *IEEE Transactions on*

*Information Forensics and Security*, Vol. 13, 2017, pp. 521-534.
3. D. T. Nguyen, T. D. Pham, N. R. Baek, and K. R. Park, "Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors," *Sensors*, Vol. 18, 2018, p. 699.
4. X. Song, X. Zhao, L. Fang, and T. Lin, "Discriminative representation combinations for accurate face spoofing detection," *Pattern Recognition*, Vol. 85, 2019, pp. 220-231.
5. D. Tang, Z. Zhou, Y. Zhang, and K. Zhang, "Face flashing: a secure liveness detection protocol based on light reflections," *arXiv preprint*, 2018 arXiv:1801.01949.
6. I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proceedings of IEEE SIG International Conference of Biometrics*, 2012, pp. 1-7.
7. A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: a public database and a baseline," in *Proceedings of IEEE International Joint Conference on Biometrics*, 2011, pp. 1-7.
8. T. de Freitas Pereira, J. Komulainen, A. Anjos, J. M. De Martino, A. Hadid, M. Pietikäinen, and S. Marcel, "Face liveness detection using dynamic texture," *EURASIP Journal on Image and Video Processing*, Vol. 2014, 2014, Article No. 2.
9. G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *Proceedings of IEEE International Conference on Computer Vision*, 2007, pp. 1-8.
10. A. Anjos, J. Komulainen, S. Marcel, A. Hadid, and M. Pietikäinen, "Face anti-spoofing: Visual approach," *Handbook of Biometric Anti-Spoofing*, Springer, London, pp. 65-82.
11. K. Kollreider, H. Fronthaler, and J. Bigun, "Evaluating liveness by face images and the structure tensor," in *Proceedings of the 4th IEEE Workshop on Automatic Identification Advanced Technologies Location*, 2005, pp. 75-80.
12. W. Bao, H. Li, N. Li, and W. Jiang, "A liveness detection method for face recognition based on optical flow field," in *Proceedings of IEEE International Conference on Image Analysis and Signal Processing*, 2009, pp. 233-236.
13. W. Kim, S. Suh, and J. J. Han, "Face liveness detection from a single image via diffusion speed model," *IEEE Transactions on Image Processing*, 2015, Vol. 24, 2456-2465.
14. J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," *Biometric Technology for Human Identification*, Vol. 5404, 2004 pp. 296-303.
15. N. N. Lakshminarayana, N. Narayan, N. Napp, S. Setlur, and V. Govindaraju, "A discriminative spatio-temporal mapping of face for liveness detection," in *Proceedings of IEEE International Conference on Identity, Security and Behavior Analysis*, 2017, pp. 1-7.
16. R. Tronci, D. Muntoni, G. Fadda, M. Pili, N. Sirena, G. Murgia, and F. Roli, "Fusion of multiple clues for photo-attack detection in face recognition systems," in *Proceedings of IEEE International Joint Conference on Biometrics*, 2011, pp. 1-6.
17. J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *Proceedings of IEEE International Joint Conference on Biometrics*, 2011, pp. 1-7.

18. A. Agarwal, R. Singh, and M. Vatsa, "Face anti-spoofing using Haralick features," in *Proceedings of IEEE 8th International Conference on Biometrics Theory, Applications and Systems*, 2016, pp. 1-6.
19. Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4680-4689.
20. Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 389-398.
21. Wikipedia, "Field-programmable gate array," https://en.wikipedia.org/wiki/Field-programmable_gate_array, 2019.
22. R. A. Hamzah and H. Ibrahim, "Literature survey on stereo vision disparity map algorithms," https://www.hindawi.com/journals/js/2016/8742920/, 2016.
23. Wikipedia, "Hamming distance," https://en.wikipedia.org/wiki/Hamming_distance, 2019
24. N. Einecke and J. Eggert, "A multi-block-matching approach for stereo," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Vol. IV, 2015, pp. 585-592.
25. P. Bhandari, "Depth estimation techniques," https://medium.com/@piyush1995bhandari/depth-estimation-techniques-830ffd297245, 2018.
26. H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, 2007, pp. 328-341.
27. W. Forson, "Understanding SSD multibox − Real-time object detection in deep learning," https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab, 2019.
28. H. Gao, *et al.*, "Understand single shot multibox detector (SSD) and implement it in pytorch," https://medium.com/@smallfishbigsea/understand-ssd-and-implement-your-own-caa3232cd6ad, 2019.
29. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 21-37.
30. Kaggle, "How to choose CNN architecture MNIST," https://www.kaggle.com/cdeotte/how-to-choose-cnn-architecture-mnist, 2019.
31. UFLDL Tutorial, "Convolutional neural network," http://deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/, 2019.
32. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.
33. S. H. Tsang, "Review: SSD − Single Shot Detector (Object Detection)," https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11, 2019.
34. DeepLearning.Hub, "Learning non-maximum suppression," https://twitter.com/DLdotHub/status/861966336527872000, 2019.
35. Common Objects in Context, http://cocodataset.org/#home, 2019.
36. Google Open Source, "Open images dataset," https://opensource.google.com/projects/open-images-dataset, 2019.

37. S. Sudhakar, "Histogram equalization," https://towardsdatascience.com/@shree6791, 2017.
38. H. Wang, S. Z. Li, Y. Wang, and J. Zhang, "Self quotient image for face recognition," in *Proceedings of IEEE International Conference on Image Processing*, 2004, Vol. 2, pp. 1397-1400.
39. Wikipedia, "Adaptive histogram equalization," https://en.wikipedia.org/wiki/Adaptive_histogram_equalization, 2019.
40. P. Pai, "Data augmentation techniques in CNN using tensorflow," https://medium.com/ymedialabs-innovation/data-augmentation-techniques-in-cnn-using-tensorflow-371ae43d5be9, 2019.
41. S. T. Tsang, "Review: VGGNet – 1st Runner-Up (Image Classification), Winner (Localization) in ILSVRC 2014," https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvlc-2014-image-classification-d02355543a11, 2018.
42. V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint*, 2016 arXiv:1603.07285.
43. X. Qi, T. Wang, and J. Liu, "Comparison of support vector machine and softmax classifiers in computer vision," in *Proceedings of IEEE 2nd International Conference on Mechanical, Control and Computer Engineering*, 2017, pp. 151-155.

**Tz-Chia Tseng (曾子家)** received the M.S. degree in Graduate Institute of Biomedical Electronics and Bioinformatics from National Taiwan University, Taiwan. His research interests include object detection and deep learning.

**Teng-Fu Shih (施登富)** received the M.S. degree in Graduate Institute of Statistics from National Chung Hsing University, Taiwan. His research interests include Mandarin monosyllable recognition and deep learning.

**Chiou-Shann Fuh (傅楸善)** received the BS degree in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan, in 1983, the MS degree in Computer Science from the Pennsylvania State University, University Park, PA, in 1987, and the Ph.D. degree in Computer Science from Harvard University, Cambridge, MA, in 1992. He was with AT&T Bell Laboratories and engaged in performance monitoring of switching networks from 1992 to 1993. He was an Associate Professor in Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan from 1993 to 2000 and then promoted to a Full Professor. His current research interests include digital image processing, computer vision, pattern recognition, mathematical morphology, artificial intelligence, deep learning, chatbot, big data analysis, cloud computing, and their applications to defect inspection, automatic optical inspection, industrial automation, digital still camera, digital video camcorder, surveillance camera, and camera module such as color interpolation, auto exposure, auto focus, auto white balance, color calibration, and color management.