Development of a Bi-level Web Connected Home Access System using Multi-Deep Learning Neural Networks

K. Y. Tham, T. W. Cheam, H. L. Wong and M. F. A. Fauzi

Faculty of Engineering, Multimedia University, Persiaran Multimedia,
63100 Cyberjaya, Selangor, Malaysia
hlwong@mmu.edu.my

Abstract. Home entrance is a vital entry point that should be secured at all times. A bi-level home access system was designed and developed using face authentication and hand gesture recognition. The system's mainframe runs on a Raspberry Pi 3 minicomputer. The board serves as the computing platform to process various deep learning algorithms for face authentication and hand gesture recognition. It also serves as a communication hub which allows registered users to communicate with the system remotely via mobile application. Home occupants may also register emergency contacts such as their neighbours' for quick response at their property. An Android mobile application was developed for remote user interface. Google Firebase platform was used to store user profile and historical data. The face authentication consists of two steps, namely face detection and face recognition. The Multi-task Cascaded Convolutional Neural Network (MTCNN) was employed for face detection, while the Inception ResNet was used for face recognition. Upon successful face authentication, the system proceeds to read the user's hand gesture. First, the system detects the hand using Single Shot MultiBox Detector (SSD) that runs on a Convolutional Neural Network (CNN). Next, a sequence of hand pose is recognised using the conventional CNN method. Based on experiments, the average detection/recognition accuracy under normal operating conditions using real face and real-video captured by the system is approximately 95.7%. An occupant needs approximately 7s to complete the process to enter the house.

Keywords: Face Authentication, Hand Gesture Recognition, Deep Learning, Embedded System, Raspberry Pi, Android Mobile Apps, Google Firebase.

1 Introduction

Security is always a concern to the house occupants. With the advancement of technology, we can utilize artificial intelligence and the internet to enhance home security. Based on Malaysia crime and safety report [1], Kuala Lumpur was identified as a high-threat city for crime. Those crimes include snatch theft, burglary and abductions. The frequency of residential breaks-ins was also high. In urban area, most people work during the day and leave their houses unguarded. The residents usually return home at different times. Due to the high cost of living, some parents have to allow their teenage

R. Alfred et al. (eds.), *Computational Science and Technology*,
Lecture Notes in Electrical Engineering 603,
https://doi.org/10.1007/978-981-15-0058-9_22

school or college-going children to return home by themselves. Nevertheless, older children may also be vulnerable to indirect or direct threats [2].

A bi-level home entry security system was proposed to deter break-ins and prevent trespassing through close monitoring and faster response by the locally connected community. The system can be installed at the main door of the house. With such a system, the house occupant can monitor the returning of other occupants remotely. Parents will have more peace of mind in letting their teenage child to return home alone, as they could monitor the child remotely. In an event that a returnee is forced by knife-point to open the door, the returnee can wave a secret panic gesture without alarming the abductor. Then, a warning alert is sent to the nearby contacts, which may include the neighbours, security guards or police.

The paper is organised as follows. A background study is presented in section 2. Then, the system design which includes the hardware, the networks employed and the firmware will be discussed in section 3. The results are presented and discussed in section 4. Finally, the conclusions and future recommendations are given in section 5.

2 Background Study

Eight types of lock system were discussed in [3]. The mechanical lock system, password lock system and Radio Frequency Identification (RFID) lock system requires keys or card, which do not verify the authority of the person holding it. Thus, the security level is low. Lock systems that operate using One Time Password (OTP) or Near Field Communication (NFC) are slightly more secure. OTP-based systems require a mobile device to connect to a server in order to obtain the generated OTP while the NFC method requires the mobile phone to have the feature itself [4]. These mechanisms are harder to be hacked. However, the need of carrying a mobile device may pose as a downside because one may not carry a mobile phone at all times, especially children. Biometric lock system grants access by verifying the user's unique physical or behavioural characteristics. However, physical characteristics such as fingerprints may be duplicated, and it remains unique throughout one's life. On the other hand, face appearance can change, and hand gesture can be altered. Development of home access system integrated with face authentication and hand gesture recognition on embedded systems is still at its infancy.

The results from different classification techniques for face recognition, namely the Support Vector Machine (SVM), Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) with Bag of Words (BoW), Histogram of Gradients (HOG) and image pixels were studied by [5]. From the literature, image pixels with CNN and ANN yielded the highest accuracy. With suitable training, direct face image combined with neural network methods also gave the lowest time spent. Hand gesture recognition is primarily used to understand sign language and is also applied in human-computer/robot interaction. Hands can be segmented by using hand contour and its silhouette [6]. Convexity defects and convex hull have been employed by [7] for hand gesture recognition. Both methods require a clean binary image of the hand gesture in order to capture the concavity and convexity locations. The locations are modelled mathematically

in order to classify different hand gestures. Even though the accuracy reported is on the high-side of 98%, the need to define each hand pose can be tedious if more gestures have to be added in a long run. Feedforward multilayer ANN with back-propagation training was developed to distinguish four different gestures of post-processed hand images [8]. The method yielded an accuracy of 88.7%. Max-pooling CNN with six hidden layers for hand gesture recognition was presented by [9]. The authors also compared it with SVM classifier and a tiled-CNN. The proposed max-pooling CNN achieved the highest classification rate of 96.8%.

A recent review on Deep Learning (DL) based vision system was presented by [10]. The authors discussed various vision applications, which include object detection, recognition and tracking. They discussed the following DL architectures: Deep Boltzmann Machine (DBM), Deep Belief Networks (DBN), Stacked Auto-Encoders, CNN and Deep Residual Learning (DRL). According to their survey, CNN implementations remained the most popular at 66.7%. Although DL accuracies can be high, the challenges will be setting-up dataset for efficient network training as well as achieving computation efficiency for real-time system applications.

Our proposed design is closely related to the concept given in Faceture ID [11]. A Leap motion controller (LMC), which is mainly used for virtual reality, was employed. It was used in recognising hand gesture motion that represents digits. The digits are OTP sent to the user after the user's face was recognised. CNN was employed in their recognition processes. However, their proof of concept was done on a computer. The LMC alone is an expensive device compared to our proposed system. Moreover, the Faceture ID required one to have a mobile device at the entry point in order to receive the OTP.

3 System Design

The goal of this project was to design and develop a full ecosystem for a home entrance access system that allows interconnectivity with home occupants and the local community with real-time updates through mobile application and a cloud database. Fig. 1 shows an overview of the system.

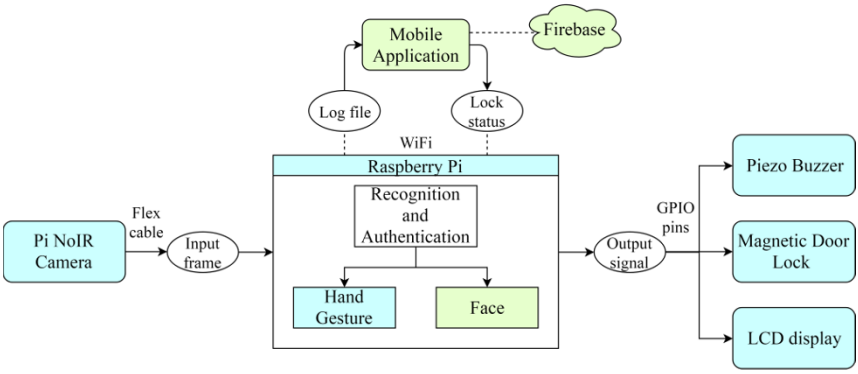


Fig. 1. System overview of the home access system

The Raspberry Pi 3 Model B+ is a single board computer employed as the central processing unit and the communication hub. It runs on quad-core with a 64-bit SoC processor at 1.4GHz with 1GB SDRAM. Its powerful processing capability allows execution of machine learning libraries, which were used in this project. Besides that, the board has built-in 802.11ac Wi-Fi that helps interconnectivity with the mobile application and cloud database. The board also comes with multi-purpose GPIO pin-outs, which were used with on-site interfacing units. They are the piezo buzzer, a 16 x 2 LCD display module and a relay switch. The Pi board can only drive 3.3V and 5V. Thus, the relay was applied to control the electromagnetic door lock which was driven by a 9V supply. A Raspberry Pi NoIR camera module was connected through the CSI port to capture the image frames for face authentication and hand gesture recognition. The CSI bus is capable of transferring data at extremely high speed, up to 2Gbps. It has a fixed focused lens with native resolution of 5 megapixels. Table 1 gives the summary of the GPIO connections.

Table 1. Raspberry Pi 3 GPIO connection to hardware peripherals.

GPIO number	Connection	Function
2	Vcc pin of LCD module and relay	5V power supply
39	Ground pin of LCD module, relay and negative terminal of piezo buzzer	Ground
3	SDA of LCD module	I^2C communication
5	SCL of LCD module	I^2C communication
16	Signal pin of relay	GPIO
18	Positive terminal of piezo buzzer	GPIO

With a small form factor of 85mm x 56mm x 20mm, weighing at only 50g, the Raspberry Pi 3 can be easily packaged together with the other hardware peripherals and installed at the door entrance. The system's flow is described in Fig. 2.

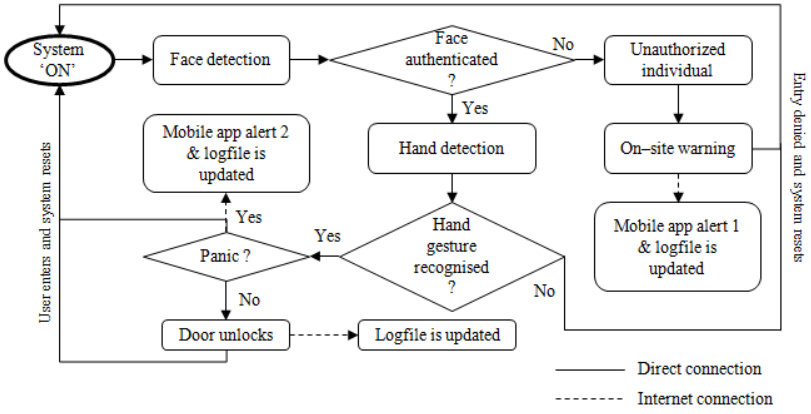


Fig. 2. System flow of the home access system

3.1 Face Authentication

The face authentication consists of two parts, namely the face detection and its recognition.

Initially, the system actively scans for a face. Each frame is converted from RGB to grayscale before the detection commences. The Multi-task Cascaded Convolutional Neural Network (MTCNN) [12] was employed for the purpose of face detection and alignment. MTCNN is a deep cascaded multi-task framework that explores the inherent correlation between detection and alignment to optimize its performance. It consists of three layers, which are the proposal network (P), refinement network (R) and output network (O). Firstly, the face is detected through the P-network. Then, rejection of a large number of false candidates is performed. Calibration with bounding box regression is done in R-network. Finally, the O-network identifies the face bounding box and its facial landmarks. 3x3 kernels were used in the 2D-convolution filters and max pooling except for their last layers in the R-network and O-network, which utilized 2x2 kernels. Finally, the detected face is segmented and normalized to 162 x 162 pixels while its features are preserved in a 128-dimensional feature vector.

The Inception ResNet v1 CNN classifier [13] was applied to identify the house occupants. Initially, a segmented face goes through a series of 2D-convolution layers and max pooling with 3x3 kernels before being fed into three Inception ResNet blocks. Each Inception network has residual connections between each layer of output that reduces the loss of computation resources between layers. These networks utilized combinations of 1x1, 1x3, 3x1, 3x3, 1x7, 7x1 convolution kernels. Later, average pooling was done using a kernel size of 8x8 before being fed into the dropout layer. Dropout is used to prevent model over-fitting and the value employed was 0.2. Finally, the softmax layer enables the model to distinguish more than two different users. The trained classifier output (q) is compared with the numerical result from MTCNN (p) using equation (1). The acceptance threshold was set at 95%.

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{1}$$

3.2 Hand Gesture Recognition

Hand gesture recognition consists of two parts, namely the hand detection and hand pose recognition. A sequence of correct hand poses will yield the correct hand gesture recognition as entry or panic password.

An open source real-time hand detection using neural networks from [14] was adopted. Single Shot MultiBox Detector (SSD) [15] approach was implemented in the algorithm. The SSD approach is based on a feed-forward CNN that produces bounding boxes and scores for the presence of an object class. Thus, recognition of multiple desired objects in a single frame is possible. In this case, the network was train to detect the human hand. The convolution process decreases the input size progressively to allow detections at multiple scales. 3x3 sized kernels were used for its 2D-convolution

and max pooling layers. The model used for transfer learning of the neural network is the SSD Mobilenet v1 Common Objects in Context (COCO) model.

Later, the hand is segmented and normalized to 200 x 200 pixels. The background is subtracted by detecting skin color via thresholding done in the HSV color space, namely the hue (H), saturation (S) and value (V) components. The threshold employed in this project is given in equation (2). The binary image then goes through a series of morphological process to reduce the noise caused by threshold as shown in Fig. 3.

$$\begin{aligned} 0 \le H \le 30 \\ 50 \le S \le 200 \\ 80 \le V \le 255 \end{aligned} \tag{2}$$

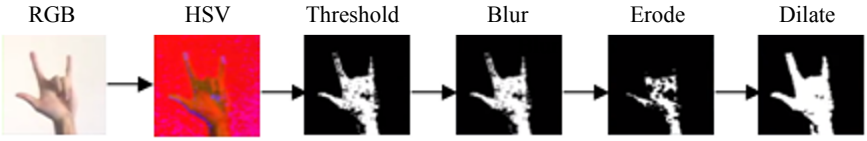


Fig. 3. Pre-processing steps for hand contour segmentation

The Convolutional Neural Network (CNN) [16] and adaptation given in [17] was employed to recognise static hand pose. The CNN is originally made to recognise digits and handwritings. Since the segmented hands important features are their edges and positions, the same network is suitable to recognise hand pose as well. The post-processed binary image is fed into the CNN, which consists of twelve layers. 3x3 kernels for 2D-convolution and 2x2 kernels for max pooling were implemented. The dropout value applied was 0.5. Eight classes of hand pose were trained, as tabulated in Fig. 4. All classes except 'nothing' were used to recognise hand gesture. The 'nothing' class denotes binary images which do not fall under the seven other classes. In this project, a sequence of three hand-poses makes up for a set of hand gesture password for entry or panic code. The sequence can be defined by the user.

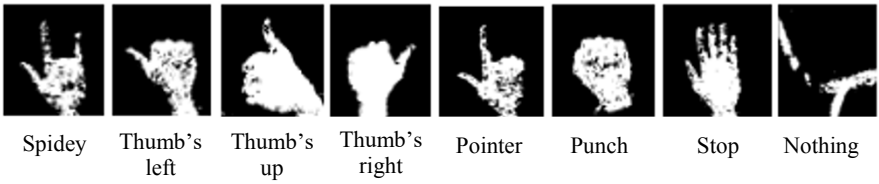


Fig. 4. Hand pose classes trained by CNN

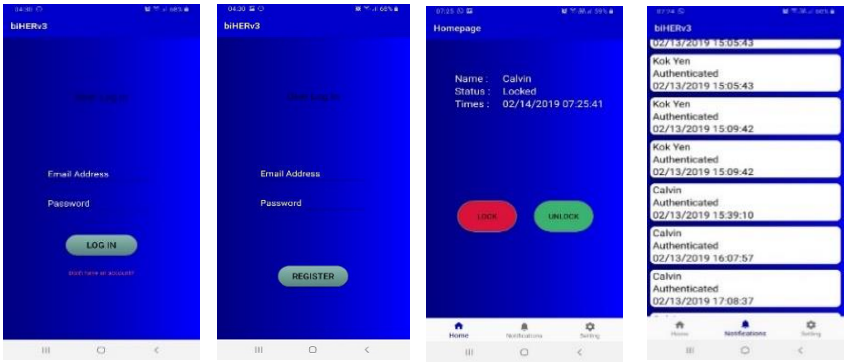
3.3 Firmware

Raspbian 9 serves as the operating system for the Raspberry Pi 3 board. The OpenCV image processing library was installed on the board and used for preprocessing steps such as color conversions and morphological operations. Besides that, the machine

learning library, Keras, which is backed by TensorFlow, was also installed on the board. The Python programming language was used to develop the codes for image processing and deep learning. For developmental purpose, a similar setup was put in place on a notebook running on Ubuntu. CNN models training and data collection were done using the notebook. Then, the trained models were exported to the Raspberry Pi 3 board for implementation and their processing speeds were observed.

Google Firebase. The services utilized in this project are Firebase authentication, real-time database and cloud messaging. All user login credentials are stored in Firebase authentication. This allows users to access the home system using their mobile devices. An authorized user can also register a new user by entering the email address and password. The Firebase database is used to store all historical data of the home access system. During a successful or failed attempt, a real-time data stamp along with its status and user ID is logged. The data can be checked via a mobile application. The database was divided into three tables. All activities are stored in the first table. Second table holds this latest entry data. A user can send signal to lock or unlock the door remotely through a third table. The Firebase cloud messaging is used to send notification or alert to the mobile application. A python script was created to invoke the messaging service during an emergency, such as foiled attempt to enter the house or panic gesture signaled by an occupant.

Mobile Application. For proof of concept, an Android-based mobile application was developed. The application communicates with Firebase and Raspberry Pi 3 board. Login page, new user registration page, historical data browser and door lock/unlock interface were created. Therefore, the system's activities logged in Firebase can be viewed remotely. Two Python scripts were coded for communication. Firstly, is to update the status of the Firebase database whenever there is an attempted entry. Secondly, is to allow the Raspberry Pi 3 to read the database and respond accordingly. Fig. 5 shows screenshots of the mobile interface developed.



(a) (b) (c) (d)

Fig. 5. Mobile application interface (a) Login page; (b) New user registration; (c) Remote lock/unlock buttons and (d) View historical data

4 Results & Discussions

The face authentication was tested with the LFW face image database, which consists of 13233 images of 5749 different human subjects [18]. The dataset was divided into two parts. 80% of the dataset were used for model training while the remaining 20% were used for validation. Validation showed an accuracy of 100% for face detection using MTCNN and 98.6% for face recognition using the Inception ResNet v1.

Later, the face authentication was tested using real-video of five individuals captured by the camera. To register a user for model training, the video of the user's face moving and tilting slightly was taken for 10s. Then, the frames grabbed from the video were used for training. The model was able to recognise individuals when the head is slightly tilted, different facial expressions and vaguely occluded. Few positive results are shown in Fig. 6. On the other hand, the recognition failed when the face is turned ~90° to the left or right and when the face is severely occluded. However, this case is unlikely in real-application scenario as an authorized user would naturally face the camera during the home access process.

Fig. 6. Sample frames of successful face recognitions

The hand pose recognition model using CNN was trained using a total of 23136 pre-processed binary hand segmented images, containing eight classes. The training set was randomly split into 80% for training and 20% for validation. Different rotation, translation and distance of hand pose images were introduced to the training set to increase its robustness. The validation results showed an accuracy of 99.7%. The hand pose recognition for the eight different classes mentioned was then tested on real-video that went through preprocessing on the spot. The hand pose recognition accuracy was slightly lower as tabulated in Table 2. The accuracy drop was due to lighting direction, hand angle and background variations invoked to the test video, which effect the binary image produced (*see* Fig. 7).

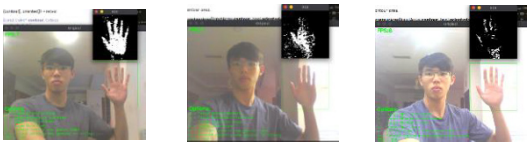


Fig. 7. Effects of lighting to the preprocessing stage.

Execution time for each process is evaluated on the Raspberry Pi 3 board as tabulated in Table 2. Accuracies of each process are shown as well. Since a sequence of three hand poses are needed to recognise a set of hand gesture password, the processing time for hand gesture recognition is threefold the time of a single hand pose recognition process. Based on the experimental data, a person needed about 7s to enter the house, assuming there is no error in between processes.

Table 2. Accuracy and speed for face authentication and hand pose recognition processes

Process	Data source	Number of training images	Accuracy	Time per frame
Face detection (MTCNN)	LFW	10586	100%	0.13s
Face recognition (Inception ResNet-CNN)	Self-captured	1912	95%	1s
Hand detection (SSD-CNN)	Self-captured	3840	94.4%	1s
Hand segmentation	Self-captured	-	-	0.04s
Hand pose recognition (CNN)	Self-captured	23136	93.4%	1s

The Raspberry Pi 3 minicomputer can perform extensive algorithm for home entry purpose. The board was also capable of driving the hardware peripherals, sending and retrieving data from cloud. However, the authors noticed that the board heated up easily during high computing operations. A heat sink can be attached to the board's main processor to reduce the temperature.

5 Conclusion & Future Work

A home access system with face authentication and hand gesture recognition had been designed and developed on the Raspberry Pi 3 single board computer. Besides being able to process computing intensive algorithms, the board can also communicate with Android-based mobile device and the Google Firebase cloud computing platform. The home access system was tested to simulate real-operating environment. Since the home occupants are able to set emergency contacts, the setup implicitly encourages neighbours to work with each other to keep the local community safe. Thus, the neighbours and local security could reach the property faster in the event of danger. Four CNN variants had been implemented on the Raspberry Pi 3. The MTCNN and Inception ResNet v1 were used for face authentication. The SSD-CNN and a conventional CNN were utilized for hand pose recognition. Those Deep Learning models yielded high accuracies and were able to perform at acceptable speed for home accessing purpose. In order to increase the robustness of the models, the training sets can be expanded by introducing more variations. Application of artificial image noise to increase lighting tone and direction variations may be used to expand the training set. On the other hand, the preprocessing before the face authentication and hand gesture recognition can be improved further through adaptive filtering based on the recent environment lighting condition.

References

1. Malaysia 2018 Crime & Safety Report, https://www.osac.gov/ , last accessed 2019/04/15.
2. Malaysian Police Reveals That On Average, 4 Children Go Missing Every Day in Our Country, https://www.worldofbuzz.com/, last accessed 2019/04/15.
3. Divya, R. S., Mathew, M.: Survey on various door lock access control mechanisms. In: 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1-3. IEEE, Kollam (2017).
4. Hung, C., Bai, Y., Ren, J.: Design and implementation of a door lock control based on a near field communication of a smartphone, In: 2015 IEEE International Conference on Consumer Electronics, pp. 45-46. IEEE, Taipei (2015).
5. Islam, K. T., Raj, R. G., Al-Murad, A.: Performance of SVM, CNN, and ANN with BoW, HOG, and image pixels in face recognition. In: 2nd International Conference on Electrical & Electronic Engineering, pp. 1-4. IEEE, Rajshahi, (2017).
6. Passarella, R., Fadli, M., Sutarno: Hand gesture recognition as password to open the door with camera and convexity defect method. In: 1st International Conference on Computer Science and Engineering, pp. 63-73. ICON-CSE, Palembang (2014).
7. Mesbahi, S. C., Mahraz, M. A., Riffi, J., Tairi, H.: Hand gesture recognition based on convexity approach and background subtraction. In: International Conference on Intelligent Systems and Computer Vision, pp. 1-5. IEEE, Fez (2018).
8. Tasnuva, A.: A neural network based real time hand gesture recognition system. International Journal of Computer Applications 59(4), 17-22 (2012).
9. Nagi, J.,Ducatelle, F.,Di Caro, G. A.,Cireşan, D., Meier, U., Giusti, A., Nagi, F.: Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: IEEE International Conference on Signal and Image Processing Applications, pp. 342-347. IEEE, Kuala Lumpur (2011).
10. Abbas, Q., Ibrahim, M. E.A., Jafar, M.A. A comprehensive review of recent advances on deep vision systems. Artificial Intelligence Review 52(1). 39-76 (2019).
11. Aleluya, E. R. and T. Vicente, C. : Faceture ID: face and hand gesture multi-factor authentication using deep learning. Procedia Computer Science 135, 147-154 (2018).
12. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters 23(10), 1499-1503 (2016).
13. Szegedy, C., Ioffe, S., Vincent. V.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In: 31st AAAI Conference on Artificial Intelligence, pp. 4278-4284. AAAI, San Francisco (2017).
14. Victor Dibia, Real-time Hand-Detection using Neural Networks (SSD) on Tensorflow, (2017), GitHub repository, https://github.com/victordibia/handtracking, last accessed 2019/03/30.
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C.: SSD: Single shot multibox detector. In: European Conference on Computer Vision, pp 21-37. Springer-LNCS, Amsterdam (2016).
16. LeCun, Y., Kavukcuoglu, K. and Farabet, C.: Convolutional networks and applications in vision. In: IEEE International Symposium on Circuits and Systems, Paris, pp. 253-256. IEEE, Paris (2010).
17. Keras Tutorial: The Ultimate Beginner's Guide to Deep Learning in Python, https://elitedatascience.com/keras-tutorial-deep-learning-in-python, last accessed 2018/11/05.
18. Labeled Faces in Wild Home, http://vis-www.cs.umass.edu/lfw/, last accessed 2018/11/30.