Convolutional Neural Network Using Kalman Filter for Human Detection and Tracking on RGB-D Video

Jovin Angelico[1] and Ken Ratri Retno Wardani[2]
[1−2]Faculty of Informatics Engineering, Institut Teknologi Harapan Bangsa
Bandung 40132, Indonesia
Email: [1]jovin.angelico@live.com, [2]ken_ratri@ithb.ac.id

***Abstract*—The computer ability to detect human being by computer vision is still being improved both in accuracy or computation time. In low-lighting condition, the detection accuracy is usually low. This research uses additional information, besides RGB channels, namely a depth map that shows objects' distance relative to the camera. This research integrates Cascade Classifier (CC) to localize the potential object, the Convolutional Neural Network (CNN) technique to identify the human and non-human image, and the Kalman filter technique to track human movement. For training and testing purposes, there are two kinds of RGB-D datasets used with different points of view and lighting conditions. Both datasets have been selected to remove images which contain a lot of noises and occlusions so that during the training process it will be more directed. Using these integrated techniques, detection and tracking accuracy reach 77.7%. The impact of using Kalman filter increases computation efficiency by 41%.**

***Index Terms*—Convolutional Neural Network, Human Detection, Tracking, RGB-D, Kalman Filter**

I. Introduction

COMPUTER vision is a field that aims to give computers the ability to interpret images like a human vision system. There are several implementations used in computer vision, such as feature extraction and matching, segmentation, and object detection and recognition. Object detection is a method to detect objects in images. For example, the objects can be bikes, cars, or humans.

The computer ability to detect humans in various conditions is still very interesting to be developed [1]. For example, detecting humans at night conditions will be more difficult than during daytime conditions. This is due to several factors such as eccentric rays, silhouettes, and dim light [2]. This ability has real applications, such as smart-car, virtual reality, surveillance system, smart robots, and others. All such applications require not only high accuracy but also high efficiency to operate [3]. To improve accuracy and computing efficiency, people can try to develop existing methods paired with additional devices to receive additional information, such as a depth sensor or heat sensor. The general modern camera tends to capture the image in Red, Green, and Blue (RGB) that can be interpreted easily by humans, but facing the dark condition, the image is not very meaningful. By using a depth camera, which can capture depth map, people can tell the distance between an object relative to the camera and not affected to lighting. A depth map is calculated by combining various techniques and technologies, such as infrared, structured light, and a stereo camera [4]. The advantage gained by using depth map information is the accuracy will be stable, despite the changes of light intenseness sensitivity to optical textures. Thus, it helps to increase accuracy in computer vision, including human detection [5].

Human detection can be done by detecting human entirely or partially. There is a drawback by detecting human entirely when a lots of noises or occlusions exist. The accuracy will be inaccurate. Detecting the human body entirely is also harder in a crowded environment in which people are only partially seen because of the condition of occlusion, clutter, and different postures [6]. Therefore, it is better to detect partially, normally head-shoulder only.

The previous research by Ref. [8] detected human being based on features extracted by using Histogram of Gradient (HoG), Block Orientation (BO), Histogram of Color (HoC), and Histogram of Bar-shaped (HoB). Reference [9] used Fused Phase, Gradient and Texture (FPGT) features and Center Symmetric Local Binary Pattern (CSLBP) to obtain the image texture. Then, they proceeded it with Principal Components Analysis (PCA) to reduce the dimensions of feature and group FPGT features using Support Vector Machine (SVM).

Received: Aug. 25, 2018; received in revised form: Oct. 11, 2018; accepted: Oct. 15, 2018; available online: Dec. 5, 2018.

Cite this article as: J. Angelico and K. R. R. Wardani, "Convolutional Neural Network Using Kalman Filter for Human Detection and Tracking on RGB-D Video", CommIT (Communication & Information Technology) Journal 12(2), 105–110, 2018.

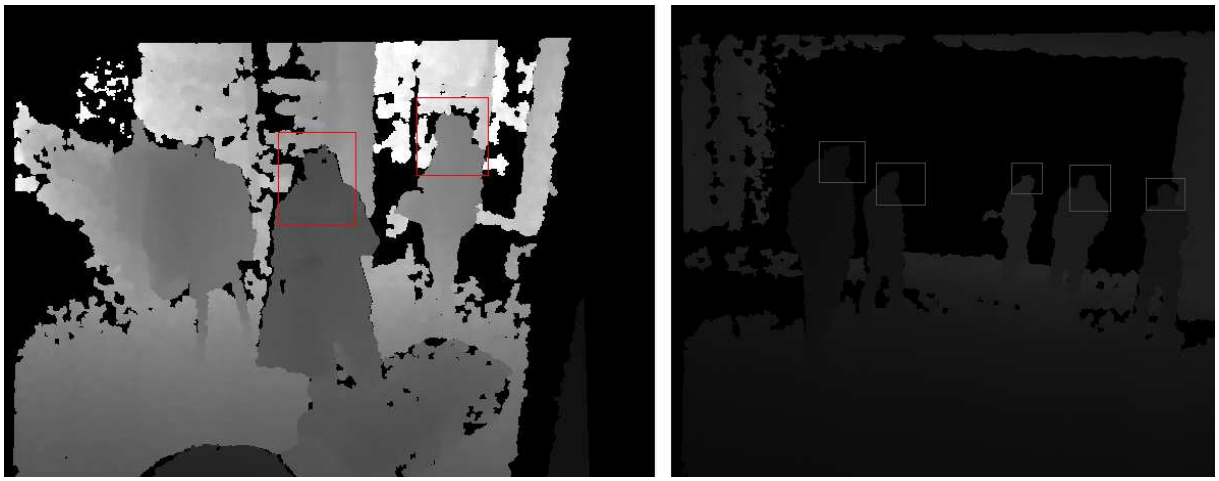

Fig. 1. Depth maps of Clothing Store dataset (left) and Outdoors dataset (right) [7].

The results of Refs. [8, 9] showed high accuracy but required adequate lighting conditions. Then, Ref. [10] used the Physical Radius-Depth (PRD) detector to detect human candidates quickly. Then, Convolutional Neural Network (CNN) for feature extraction and classification was also used. Without using the acceleration of the GPU, the methods could operate in real-time and detect under dim lighting conditions with RGB-D datasets. On the research of video surveillance for human detection and tracking, Ref. [11] used Gaussian Mixture Model (GMM) to detect person and Kalman filter to track the detected person. To reduce the processing time, the Papoulis-Gerchberg method used the down-sampled video quality. Kalman filter also reduced the computation time by predicting the location of the object in each frame.

Moreover, the neural network technique can help the computer to learn like how the human brain works. In many fields, neural networks were implemented to detect or recognize tasks, such as hand-gesture recognition [12]. It uses color detection for image segmentation and the Artificial Neural Network (ANN) for classifying. Furthermore, there is facial expression recognition [13] that used Gabor feature extraction and the single layer feed-forward neural network.

This research uses RGB-D datasets and combines CNN method with Kalman filter to detect and track humans. In computer vision, CNN had twice faster performance than ANN and Multilayer Perceptron (MLP) [14]. CNN is a technique that can process large data, such as weight sharing, subsampling, or pooling. The CNN method includes processes of feature extraction and classification. CNN has proven its advantages in classifying many complex features simultaneously [15]. Meanwhile, the Kalman filter method is chosen to improve computing efficiency to track humans.

Compared to Ref. [16] regarding real-time detection and tracking of the human face, there is a difference in this research.This research detects humans by head-shoulder using depth images in RGB-D datasets. Meanwhile, Ref. [16] tried to detect the face in video based on traditional RGB. Moreover, the parameters of the CNN architecture, such as initial image size, convolution filter size, and depth of the architecture are also different.

In this research, to measure accuracy, the researchers use confusion matrix and Jaccard similarity measure. The computational efficiency is measured in millisecond.

II. Research Method

A. Dataset RGB-D

There are two RGB-D datasets used, Clothing Store dataset and Outdoor dataset with the resolution of 640×480 pixels [7]. Both datasets are selected for the training process. The images that contain lots of noises or occlusions are removed. The example of the images used is shown in Fig. 1. The learning stage uses head-shoulder of human in the depth map channel. The ground-truth file has been provided on the datasets which explain the positions of head-shoulder for every person. Then, non-human images are taken at the random positions which do not intersect with the human region.

Clothing Store dataset is captured with adequate lighting. There are a few people sitting so they look skewed. Then, the Outdoor dataset is capture with dim lighting, and people are just passing over. The distance between the camera and humans in the Outdoor dataset

is more straight and uniform than that of the Clothing Store dataset.

B. Median Filter

All depth maps are processed using median filtering with the kernel size of 5×5 pixels. It is considered as the most suitable for the datasets. The median filter is useful for noise removal; thus, the kernel size is chosen by considering the noises of the images.

C. Cascade Classifier (CC)

Cascade Classifier (CC) is a method that learns a set of positive and negative images based on the existing feature type and searches object positions using the sliding window method. A set of features across the entire image will be aggregated incrementally. The learning process uses Haar classifier in 20 stages to get the CC model in XML format. With more number of stages, it will increase the combined number of features. The images have been selected from the corresponding dataset. The numbers of positive images use 33 pieces and 240 pieces for negative image with sample size width and height of 24 pixels. Meanwhile, the type of boosted classifier used is Gentle AdaBoost (GAB). The detection of the human candidate region is limited to the size between 40×40 pixels to 90×90 pixels by considering the average size of human head-shoulder in the datasets.

D. Scale Image

To be able to use the human candidate that has been localized by CC to CNN, all the candidate regions are scaled to 48×64 pixels.

E. Convolutional Neural Network (CNN)

CNN is a classification method with the final result of the probability of each member of the classes. CNN is rather similar to ANN. However, CNN uses weight sharing techniques where the only one kernel is used in each convolution layer and fully connected layer. Hence, the number of parameters is reduced and speeds up the calculation process. Moreover, there are subsampling or pooling techniques to reduce data dimensions by taking the most important information. For that reason, CNN is commonly used for processing high-dimensional data. In this research, CNN is used to identify humans. It classifies two classes which are human and non-human class.

There are several types of layers in CNN: convolution layer, subsampling/pooling layer, and fully connected layers. First, the convolution layer calculates input with the convolution process. The result may be used into the activation function such as Rectified Linear Unit (ReLU) function to have a certain range. Subsampling layer usually follows convolution layer to take the most important features. The common technique used in subsampling layer is max pooling by taking the highest value. The last layer before the logistic regression activation function is the fully connected layer.

The softmax function is commonly used as a logistic regression function in the last layer of the neural network. By using the softmax function, the probability value will be limited to the range between zero and one. The total value of all probability from every class is one.

The depth of the hidden layer type in CNN architecture may vary. The types of features handled on each layer have different complexities. The low-level hidden layer handles basic features such as lines and edges. While high-level hidden layer identifies complex features by combining common features to form features as the whole object. To distinguish different types of objects with similar features commonly, the researchers use a deeper number of hidden layers.

This research uses two pairs of convolution-subsampling layers and a fully connected layer. The convolution layer has a kernel size of 5×5 pixels and one stride. Meanwhile, the subsampling layer has a kernel size of 2×2 pixels and stride of two. The main architecture of CNN is seen in Fig. 2. The kernel value updating is performed at the backpropagation stage based on the gradient value of each layer. By updating kernel values, the result of the neural network technique will be closer to the desired value or also known as the target.

This research uses stochastic or online learning in CNN. The stochastic learning updates kernel value for each sample, in this case, is the image. With this method, the gradient value has high variance, and it is possible to find a new local minimum. Moreover, the learning process is faster, and memory consumption is lower than the batch learning method that updates the value per epoch.

In this research, the learning process of CNN is done using 1000 epochs at the learning rate of 1.0×10^{-3}. Generally, the learning rate values are within the range of 1.0×10^{-2} and 1.0×10^{-3}, but it can also be adjusted to the kernel value used. When the learning rate is too low compared to the kernel value used, the learning process takes time. On the other hand, when the learning rate is too high, the learning process becomes unstable.

There is also a learning rate divider which will divide the learning rate when the current epoch is some multiples of a certain epoch. The learning rate divider

Cite this article as: J. Angelico and K. R. R. Wardani, "Convolutional Neural Network Using Kalman Filter for Human Detection and Tracking on RGB-D Video", CommIT (Communication & Information Technology) Journal 12(2), 105–110, 2018.

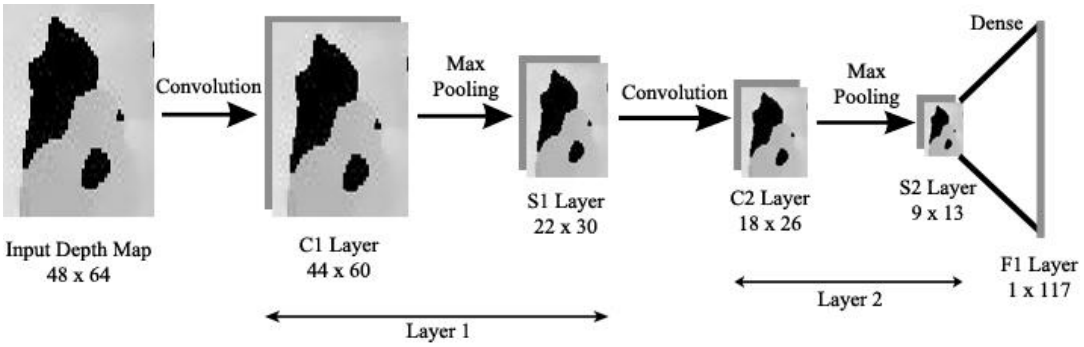

Fig. 2. Main architecture of CNN.

uses 1.001, and 1.1287 per 50 epochs. The CNN model is saved per 50 epochs or when it reaches the highest accuracy.

F. Kalman Filter

Kalman filter is an algorithm to estimate the value of unknown variables by connecting variables to produce more accurate estimation. This method is suitable for the environment that keeps changing. The advantage of Kalman filter is a simple calculation so it does not require much time or memory.

There are two main stages in the Kalman filter method that is prediction stage and update stage. The prediction stage will estimate variable values. Meanwhile, the update stage is to update information following the actual conditions.

There are five main equations in Kalman filter to predict and update. Equations (1) and (2) are used to predict the next variable value and Eqs. (3)–(5) update the value. Equation (3) calculates Kalman gain by considering the environment noises (R_k). The equations can be seen as follows:

$$\hat{x}_k = F_k x_{k-1} + B_k u_k \tag{1}$$

$$P_k = F_k P_{k-1} F_k^T + Q_k \tag{2}$$

$$K' = \frac{P_k H_k^T}{H_k P_k H_k^T + R_k} \tag{3}$$

$$x_k = \hat{x}_k + K'(z_k - H_k \hat{x}_k) \tag{4}$$

$$P_k = P_k(1 - K' H_k) \tag{5}$$

III. Results and Discussion

Testing is done per method and a combination of several methods as shown in Fig. 3.

The CNN testing measurement uses a confusion matrix including one of these: accuracy, precision, and recall. Accuracy shows the ratio of the true and the false classification as seen in Eq. (6). The precision is a ratio of the true positive and all positive results as seen in Eq. (7). The recall or sensitivity is a ratio of the true positive result with the union of the true positive and the false negative result as seen in Eq. (8).

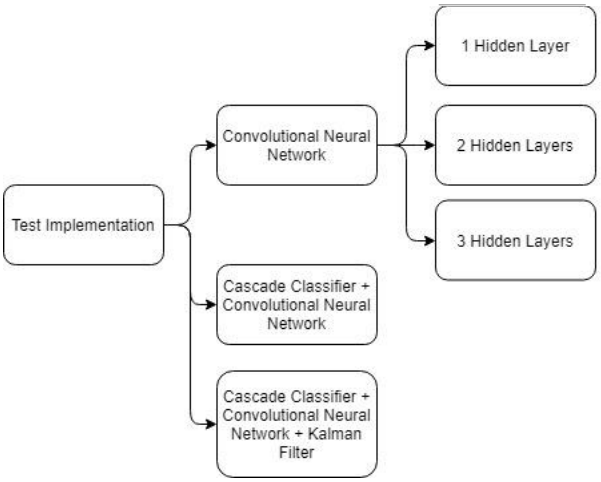

Fig. 3. Testing process.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{TN}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

For the Kalman filter, the researchers use Jaccard similarity or Intersection over Union (IoU). In this research, IoU compares two sets: ground-truth and predicted bounding boxes. The limitation of this measurement is when one of the sets has few members that will make the error high. The equation of IoU can be seen in Eq. (9).

$$\text{IoU} = \frac{X \bigcap Y}{X \bigcup Y} \tag{9}$$

Cite this article as: J. Angelico and K. R. R. Wardani, "Convolutional Neural Network Using Kalman Filter for Human Detection and Tracking on RGB-D Video", CommIT (Communication & Information Technology) Journal 12(2), 105–110, 2018.

TABLE I
ACCURACY OF CONVOLUTIONAL NEURAL NETWORK (CNN).

Hidden Layer	Outdoor/folder	Average Accuracy (%)	Max Accuracy (%)
1	Train 56	94.1	99.7
	Test 31	86.4	90.4
	Test 54	85.5	87.1
2	Train 56	85.9	93.1
	Test 31	82.6	88.1
	Test 54	81.6	87.1
3	Train 56	79.1	93.0
	Test 31	75.9	87.4
	Test 54	77.5	89.4

TABLE II
THE EFFECTS OF KALMAN FILTER TO THE ACCURACY, PRECISION, AND RECALL.

Methods	Average		
	Accuracy	Precision	Recall
CC, CNN	79.0	78.3	95.6
CC, CNN, Kalman filter	77.7	79.4	91.1

TABLE III
COMPARISON OF THE COMPUTATION TIME WITH AND WITHOUT KALMAN FILTER (KF) IN MILLISECOND.

Dataset Outdoor 31	With Median Filter		No Median Filter	
	CC, CNN	CC, CNN, KF	CC, CNN	CC, CNN, KF
Time for 47 images	21558	18451	3515	2049
Time/image	458.681	392.575	74.787	43.596
Efficiency	14.4%		41.7%	

The accuracy of CNN architecture testing that uses Outdoor dataset can be seen in Table I. Each dataset in the CNN model during the training process will be tested, so average accuracy and max accuracy are calculated from the first epoch until the 1000^{th} epoch.

Next, comparing the implementation of CC, CNN detects humans in all frames versus CC, CNN, Kalman filter, which is detecting and predicting alternately. The results are in Table II.

The computation time of using Kalman filter shows faster calculation about 41.7%. The computation efficiency calculates the delta time divided by the initial time without Kalman filter. The results can be seen in Table III.

The first testing, CNN architecture, shows that using one hidden layer (convolution and subsampling layer) reaches higher accuracy because the parameter is less than the others. One hidden layer learns faster. It can be seen from the average accuracy. Moreover, human shape in the depth map can be classified by CNN with one hidden layer, so with the same maximum epoch for all architecture, one hidden layer reaches the highest accuracy.

The Outdoor dataset in Test 54 has more abstract human shapes. So, by using a deeper neural network, three hidden layers, CNN can reach higher accuracy than one hidden layer. The deeper network will learn the higher level feature [17] and may reach higher accuracy if the epoch is extended.

Next, the test shows that without Kalman filter implementation, it generates higher accuracy and recall. However, the precision is lower than implementing Kalman filter. This occurs because using CC and CNN may miss detecting human and increasing false negative. On the other hand, Kalman filter will predict human positions from the previous information, so it will not cause miss detection, and the true positive will not turn into false positive. That is why using Kalman filter may reach higher precision.

IV. CONCLUSION

There are several conclusions of human detection and tracking system on RGB-D video. First, CNN can recognize humans although the small image that is 48×64 pixels by using depth map information. The architecture determines the accuracy of CNN. Using one hidden layer produces the highest accuracy of 90.4% in Test 31 of Outdoor datasets with 1000 epochs, the learning rate of 1.0×10^{-3}, and the learning rate divider of 1.001 per 50 epochs.

Second, the number of iteration and adaptation before predicting and R_k value influence the accuracy of Kalman filter prediction. Kalman filter uses simple calculation and reduces the computation time. Third, the difference of accuracy, precision, and recall in Kalman filter implementation is not significant, but it can increase computation efficiency to 41.71%.

Then, the system should be better to detect and track human more accurate by using the frontal point of view. The accuracy may be improved by developing the current method to Fast R-CNN or Faster R-CNN in the future.

ACKNOWLEDGEMENT

We would like to thank Lembaga Penelitian dan Pengabdian Masyarakat of Institut Teknologi Harapan Bangsa Bandung (LPPM ITHB) for supporting the funds of this research.

REFERENCES

[1] D. Tatarenkov and D. Podolsky, "The human detection in images using the depth map," in *Systems of Signal Synchronization, Generating and Processing in Telecommunications*

(SINKHROINFO). Kazan, Russia: IEEE, July 3–4, 2017, pp. 1–4.

[2] N. Sabri, Z. Ibrahim, M. M. Saad, N. N. A. Mangshor, and N. Jamil, "Human detection in video surveillance using texture features," in *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. IEEE, Nov. 25–27, 2016, pp. 45–50.

[3] Z.-J. Lin, W.-N. Chen, J. Zhang, and J.-J. Li, "Fast multiple human detection with neighborhood-based speciation differential evolution," in *Seventh International Conference on Information Science and Technology (ICIST)*. Da Nang, Vietnam: IEEE, April 16–19, 2017, pp. 200–207.

[4] T. Jia, Z. Zhou, and H. Gao, "Depth measurement based on infrared coded structured light," *Journal of Sensors*, vol. 2014, 2014.

[5] L. Tian, M. Li, G. Zhang, J. Zhao, and Y. Q. Chen, "Robust human detection with super-pixel segmentation and random ferns classification using RGB-D camera," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. Hong Kong, China: IEEE, July 10–14, 2017, pp. 1542–1547.

[6] B. Choi, C. Meriçli, J. Biswas, and M. Veloso, "Fast human detection for indoor mobile robots using depth images," in *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, May 6–10, 2013, pp. 1108–1113.

[7] Fudan University, "Clothing store RGBD dataset." [Online]. Available: https://cv.fudan.edu.cn/_upload/tpl/06/f4/1780/template1780/humandetection.htm

[8] S. Singh and S. C. Gupta, "Human object detection by HoG, HoB, HoC and BO features," in *Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*. Waknaghat, India: IEEE, Dec. 22–24, 2016, pp. 742–746.

[9] H. K. Ragb and V. K. Asari, "Multi-feature fusion and PCA based approach for efficient human detection," in *IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. Washington, DC, USA: IEEE, Oct. 18–20, 2016, pp. 1–6.

[10] J. Zhao, G. Zhang, L. Tian, and Y. Q. Chen, "Real-time human detection with depth camera via a physical radius-depth detector and a CNN descriptor," in *IEEE International Conference on Multimedia and Expo (ICME)*. Hong Kong, China: IEEE, July 10–14, 2017, pp. 1536–1541.

[11] V. Sriram, K and H. Havaldar, R, "Human detection and tracking in video surveillance system," in *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. Chennai, India: IEEE, Dec. 15–17, 2016, pp. 1–3.

[12] L. Yusnita, N. Hadisukmana, R. B. Wahyu, R. Roestam, Y. Wahyu *et al.*, "Implementation of real-time static hand gesture recognition using artificial neural network," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*. Kuta Bali, Indonesia: IEEE, Aug. 8–10, 2017, pp. 1–6.

[13] J. Cristanto and K. R. R. Wardani, "Penerapan metode single-layer feed-forward neural network menggunakan kernal gabor untuk pengenalan ekspresi wajah," *Jurnal Telematika*, vol. 12, no. 1, 2017.

[14] S. B. Driss, M. Soua, R. Kachouri, and M. Akil, "A comparison study between MPL and Convolutional Neural Network models for character recognition," in *SPIE Conference on Real-Time Image and Video Processing*, Anaheim, CA, United States, April 10–11, 2017.

[15] Stanford.edu, "Cs231n: Convolutional neural networks for visual recognition," syllabus. [Online]. Available: http://cs231n.github.io/convolutional-networks/

[16] Z. Ren, S. Yang, F. Zou, F. Yang, C. Luan, and K. Li, "A face tracking framework based on convolutional neural networks and kalman filter," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. Beijing, China: IEEE, Nov. 24–26, 2017, pp. 410–413.

[17] J. Jordan, "Setting the learning rate of your neural network," 2018. [Online]. Available: https://www.jeremyjordan.me/nn-learning-rate/