Face Recognition Using Faster R-CNN with Inception-V2 Architecture for CCTV Camera

Lavin J. Halawa
Department of Computer Science
Faculty of Mathematics and Science
Universitas Diponegoro
Semarang, Indonesia

Adi Wibowo
Department of Computer Science
Faculty of Mathematics and Science
Universitas Diponegoro
Semarang, Indonesia
bowo.adi@live.undip.ac.id

Ferda Ernawan
Faculty of Computing
College of Computing and
Applied Sciences
University Malaysia Pahang

***Abstract*— Detection and prevention of criminal incidents using CCTV are currently increasing trend, for example, car and motorcycle parking lot. However, not continuous people monitoring and careless of events produce useless CCTV function for the prevention of criminal incidents. In this paper, face recognition is used for the recognition of vehicle owners in parking lots that are CCTV installed. The Faster-RCNN method is used for face detection and also for face recognition. Inception V2 architecture is utilized due to has a high accuracy among Convolutional Neural Network architecture. The best learning rate and epoch parameters for the Faster R-CNN model are optimized to improve face recognition on CCTV. In this research, the dataset consists of 6 people images with 50 faces images for each people, which used as training data, testing data, and validation data.**

Keywords —Face Recognition, Convolutional Neural Network, Inception, Faster R-CNN

I. INTRODUCTION

Motor vehicle theft is a serious problem because there were more than 100,000 incidents in 2017 [1]. The frequent event of theft is by taking a vehicle in a parking lot [2]. The application of CCTV or live cameras is a mechanism for theft prevention and detection. However, continuous monitoring and vehicle owner recognition are required in the prevention and detection of crime. So even though CCTV was installed, the theft incident still occurred due to negligence in monitoring the parking area. Automotic CCTV monitoring using machine learning methods is solution for suspicious human activity detection on parking lot [3].

Research related to theft prevention in parking lots using detection automation in CCTV has been developed. Najla develops abnormal event recognition in the parking lot for theft preventing [4]. Sayma developed automatic notifications to security for vehicle movements [5]. The use of detection of moving objects can be late to provide theft prevention due to the thief is already in the vehicle. One way to prevent thieves from entering the vehicle is to identify the face of the vehicle owner. The face recognition can be used to check whether the driver is authenticated or not.

The increased interest in the face recognition field has explicitly a relationship with technology that has been available after 30 years of development [6]. Humans have an excellent ability to recognize a person's face regardless of age, lighting conditions, and diverse expressions. Research conducted in this field has a goal to design a facial recognition system that can rival or even exceed the level of human recognition ability, which reaches almost 97.5%. In its application, the methods for facial recognition are deep learning techniques [7].

Deep learning, specifically the Convolutional Neural Network (CNN) method, has achieved excellent results in the field of face recognition. Unlike traditional methods that are made manually, CNN learning features could handle complex intra-personal variations [8]. In face recognition research using a dataset from Labeled Faces in Wild (LFW), CNN has a high degree of accuracy, compared to Joint Bayesian and Local Binary Pattern (LBP) [9].

CNN has a high degree of accuracy using LFW data, but in practice, an image can have several face; therefore, a selective search is performed to determine the location and identification of face. Regional CNN (R-CNN) method is a viable method for a selective search. The results of each selective search were then carried out by CNN [10]. However, the R-CNN method has a weakness, where the computation process runs slowly because of the CNN process is repetitive. This problem is solved by doing CNN on the image first, then determining the location and label of the object. The development of this method is known as Fast R-CNN, and then F-R-CNN Faster. The Faster R-CNN is a development of the Fast R-CNN which adds a proposal region to the CNN method [11]. In the Faster R-CNN, the region of interest is formed using the Region Proposal Network (RPN), while in its predecessor the Fast R-CNN region of interest is formed using external methods such as selective search. The RPN provides more accurate region of interest results, reducing the number of inappropriate regions of interest, thus speeding up the model training process [12]. In its application, Faster R-CNN has various architectures, one of them used in this research is Inception V2. The architecture of Inception V2 is designed to reduce the complexity of CNN [13].

In this research, Faster R-CNN with Inception V2 architecture are used as a face recognition method on CCTV

978-1-7281-4610-2/19/$31.00 ©2019 IEEE

cameras installed in parking lots and optimized the accuracy based on the learning rate and epoch.

II. METHOD

A. Faster R-CNN

Faster R-CNN is a method developed by Shaoqin Ren. Faster R-CNN is a development of Fast R-CNN developed by Ross Girshick by adding Region Proposal Network to Fast R-CNN. Faster R-CNN consist of 2 modules. The first module is the Fully Convolutional Network (FCN) which is used to create a region proposal network (RPN), and the second module is the Fast R-CNN which is used as a detector based on the proposal region of the first module. The second module (RPN) help Fast R-CNN to find the region of interest, making the computation faster [12]. Faster R-CNN model shown at Fig. 1.

1) Region Proposal Network

Region Proposal Network (RPN receives the image as input and produces a set of squares with the proposal object (object location), each with an object score. RPN maps the last layer of CNN using a 3x3 sliding window to a smaller dimension to get the feature map [12].

The purpose of the RPN is to create a number of Regions of Interest (ROI) that have a high probability of covering an object [14]. Architecture of RPN is shown at Fig. 2.

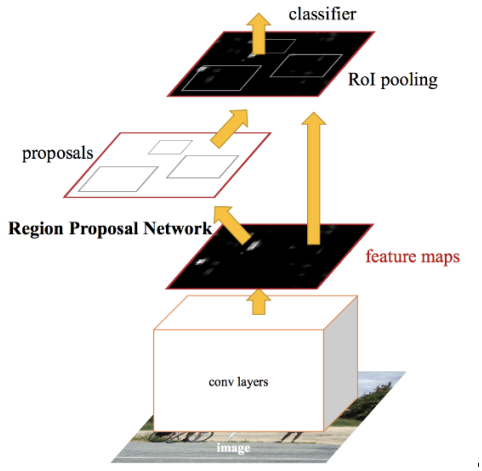

Fig. 1. Faster R-CNN [12]

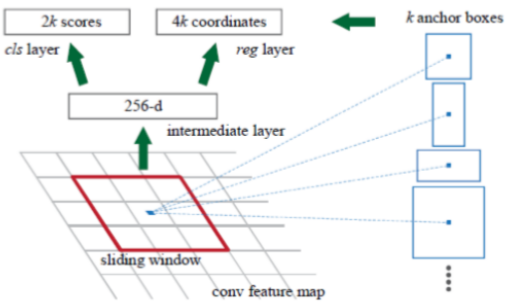

Fig. 2. Region Proposed Network [14]

The resulting feature map consists of 2 layers, namely the cls layer and the reg layer. Anchor is the central point of sliding windows. To facilitate diverse object sizes, anchor box dimensions vary from 1x1, 1x2 or 2x1. Cls layer carries 2k estimated value of possible objects for each proposal, where k is the sum of the total possible locations of objects, because a 3x3 matrix is used for sliding windows, the total likelihood is 9 pixels. The Reg layer carries 4k coordinates from the location of the k box. The 4 coordinates are in the middle box (x, y), width / width (w), and height / height (h) [12].

To reduce the number of anchor boxes, a Non Maximum Surpression is performed, where the intersecting anchor box will be deleted if it has a lower Intersection over Union value. The limit of Intersection over Union values used is more than 0.7 for objects (positive) and less than 0.3 for background (negative). Here is how to calculate Intersection over Union shown in (1).

$$IoU = \frac{Anchor\ box \cap Ground\ Truth\ box}{Anchor\ box \cup Ground\ Truth\ box} \quad (1)$$

RPN is an algorithm that needs to be trained, so RPN has a loss function, shown in (2)

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (2)$$

Description
p_i : Object possibility
t_i : $4k$ anchor coordinate
p_i^* : Ground truth label
t_i^* : Ground truth coordinate
L_{cls} : Classification Loss (Log loss)
L_{reg} : Regression Loss (Smooth L1 loss)
N_{cls} : Classification Normalization
N_{reg} : Regression Normalization
λ. : Balancing parameter

The calculation of anchor coordinate is shown at (3.1) to (3.8)

$$t_x = (x - x_a)/w_a \quad (3.1)$$
$$t_y = (y - y_a)/h_a \quad (3.2)$$
$$t_w = log(w/w_a) \quad (3.3)$$
$$t_h = log(h/h_a) \quad (3.4)$$
$$t_x^* = (x^* - x_a)/w_a \quad (3.5)$$
$$t_y^* = (y^* - y_a)/w_a \quad (3.6)$$
$$t_w^* = log(w^*/w_a) \quad (3.7)$$
$$t_h^* = log(h^*/h_a) \quad (3.8)$$

Description
y : y-axis prediction box coordinate
x : x-axis prediction box coordinate
h : prediction box height
w : prediction box width
a : anchor box
$*$: ground truth box

Loss Function classifier and Loss Function Bounding Box calculation is shown at (4) and (5).

$$L_{cls}(p_i, p_i^*) = -(p_i^* log(p_i) + (1 - p_i^*) log(1 - p_i)) \quad (4)$$

$$L_{reg}(t_i, t_i^*) = \sum_{i \in \{x,y,w,h\}} smooth_{L1}(t_i - t_i^*) \quad (5)$$

where

$$smooth_{L1}(t_i - t_i^*) = \begin{cases} 0.5x^2 & if\ |t_i - t_i^*| < 1 \\ |x| - 0.5 & other \end{cases} \quad (6)$$

Description
p_i : Object possibility
t_i : $4k$ anchor coordinate
p_i^* : Ground truth label
t_i^* : Ground truth coordinate
L_{cls} : Classification Loss (Log loss)
L_{reg} : Regression Loss (Smooth L1 loss)

2) Fast R-CNN

Fast R-CNN is used as a detection network to detect objects. Fast R-CNN is conducted on the proposed RPN [11]. The Fast R-CNN architecture can be seen in Fig. 3.

Fast R-CNN needs to be trained so that it can detect classes and bounding boxes correctly. Fast R-CNN training is carried out using backpropagation with changes in weights using Stochastic Gradient Descent. Calculation of loss function from Fast R-CNN can be seen in (7).

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda.[u \geq 1] L_{reg}(t^u, v) \quad (7)$$

Description
p : Object possibility
u : Classification class
t : Ground truth label
v : Ground truth coordinate for u class
L_{cls} : Loss function classification
L_{reg} : Loss function bounding box
λ. : Balancing parameter

L_{cls} calculation is shown at (8).

$$L_{cls}(p, u) = -\log\left(\frac{e^{p_u}}{\sum_{j=1}^{K} e^{p_j}}\right) \quad (8)$$

Description
p : Object possibility
u : Classification class
L_{cls} : Loss function classification
K : Number of class
L_{reg} can be calculated using (5) with t^u and v as input.

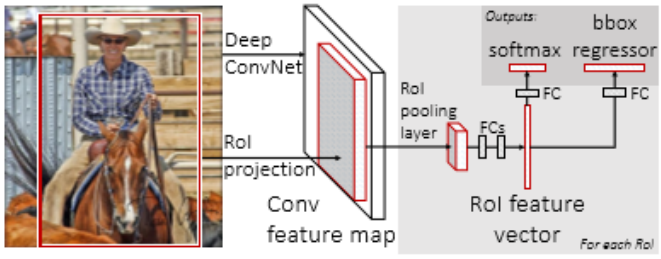

Fig. 3. Fast R-CNN [5]

B. Image Convolution

Image convolution is a convolution operation used to extract features in an image that are conceptually the same as a blur operation. Image convolution operations require kernels and strides. The kernel functions to reduce the size of the matrix from the original image. Stride is a parameter that gives an idea of how much shift from a kernel horizontally and vertically [15]. Convolution equation is shown by (9)

$$c(x, y) = \sum_{a=-m}^{m} \sum_{b=-n}^{n} f(a, b)\, k(x - a, y - b) \quad (9)$$

Description :
f : Input image
k : kernel
$c(x, y)$: result
m : x-axis kernel location
n : y-axis kernel location

C. Max Pooling

Max pooling is a sample based reduction process. The goal is to reduce input dimensions and make assumptions about the features contained in the sub-region. This is done to reduce computational requirements by reducing the number of parameters to be studied and providing basic translations into internal representations.

Max Pooling is done by applying max filters to sub-regions that do not overlap [16]. In the CNN process in general, max pooling is done using a 2x2 kernel size with stride 2. Max Pooling's illustration can be seen in Fig. 4.

Max pooling equation can be seen at (10).

$$c(x, y) = max((x, y), (x + 1, y), (x, y + 1), (x + 1, y + 1)) \quad (10)$$

Description :
max : Max Function
$c(x, y)$: result

D. Softmax Activation Function

Softmax activation function is a function used to calculate the probability distribution of vectors containing real numbers. This function produces an output ranging from 0 to 1 with a total probability equal to 1 [17]. The Softmax activation function can be seen in (11).

$$F(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \quad (11)$$

Description :
$F(x_i)$: Softmax function from x of i
K : total data
e : epsilon (2,7182)
x : input

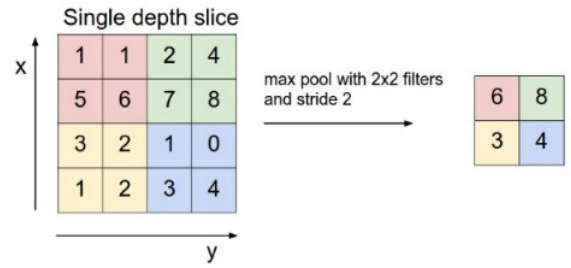

Fig. 4. Max Pooling

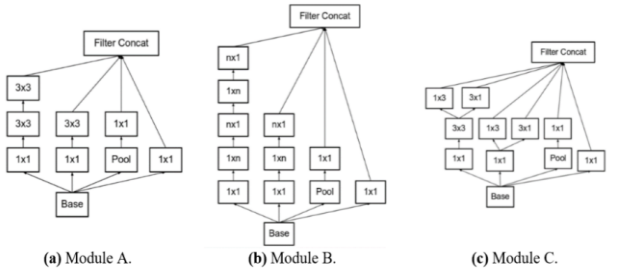

Fig. 5. Inception V2 Architecture [13]

$$x' = x - \alpha(\Delta f(x)) \tag{12}$$

Description:
x' = new value
x = old value
α = learning rate
$\Delta f(x)$ = difference in function $f(x)$

E. Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) method aims to minimize the empirical risk in a model by repeatedly calculating the derivative functions in a single training example and updating the corresponding model parameters [18]. The SGD function is shown by (12).

F. ReLU Activation Function

Rectified Linear Units (ReLU) is an activation function introduced by Richard HR Hahnloser in 2000. ReLU works by using a threshold value of 0 [19].This function is shown by (13).

$$f(x) = \max(0, x) \tag{13}$$

G. Inception V2

Inception V2 is one of the most commonly used CNN architectures. Inception V2 was chosen because it is one architecture that has a high degree of accuracy among other CNN architectures. The architecture of Inception V2 is designed to reduce the complexity of CNN, which is done by compiling an architecture that is wider than deep. Inception V2 has 3 modules shown in Fig. 5.

The first module replaces the 5 × 5 convolution into 3 × 3. Then the convolution factorization is carried out. Finally the modules are modified more broadly to reduce the complexity of convolution networks [13]. Inception V2 architecture can be seen in Fig. 5.

H. Batch Normalization

Batch normalization is a process of normalization by reducing the average value and dividing it by standard deviations. In each of the Inception V2 convolution processes Batch Normalization is applied because it accelerates model training [13]. The batch normalization equation is shown in (14)

III. RESULT AND DISCUSSION

Problem solving steps in this study are divided into several steps starting with data collection in the form of images, data distribution into test data, training data and validation data, determination of Ground Truth Boxes for test data and training data, training in deep learning models, testing the deep learning model using data validation, and model evaluation

$$\hat{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{14}$$

Description:
x_i = Input data
$\hat{x_i}$ = Normalized data x_i
μ = Average
σ^2 = Standard deviation
ϵ = epsilon (1 x 10^{-7})

A. Data collection

Dataset was collected manually by taking pictures of 6 Diponegoro University students with 50 images each, with a total of 500 images. Capturing images using the CCTV camera with a strength of 5 megapixels. Image dataset example can be seen on Fig. 6.

B. Faster R-CNN Training

The training process begins with changing the image size to 300x300. The next step is to do image convolution with the Inception V2 architecture as shown in Fig. 5. This process will produce a feature map.

The resulting feature map will then form anchors and anchor boxes which will be used to form the RPN. RPN is trained using end to end backpropagation and Stochastic Gradient Descent.

The first iteration of the RPN training will produce several proposal regions. The results of the process will then be trained with Fast R-CNN as initial input. This process starts by inserting an image with RPN into Inception V2, after which Region of Interest pooling (RoI pooling) is performed. RoI pooling is done to convert features in the proposal region into a small 7x7 feature map using max pooling. Then the feature map is mapped again using the Fully Connected layer, producing a RoI feature vector. The RoI feature vector will then be classified by the softmax function and also the bounding-box regressor for marking the location of the object. The Fast R-CNN training process is the same as RPN, using end to end backpropagation and Stochastic Gradient Descent. RPN training and Fast R-CNN are carried out alternately up to the specified epoch.

Fig. 6. Image Dataset

TABLE I. CONFUSION MATRIX

Ground Truth	*Detected (positive)*	*Rejected (Negative)*
Relevant	*True Positive (TP)*	*False Negative (FN)*
Non-relevant	*False Positive (FP)*	*True Negative (TN)*

TABLE II. TEST RESULT

Testing	***Learning Rate***	**Evaluation**		
		Accuracy	Sensitivity	Specificity
Static image	0,001	72,5%	80,03%	40%
	0,0001	60%	68,57%	0%
CCTV	0,001	30%	44,44%	27,27%
	0,0001	35%	65%	5%

C. Evaluation

Evaluation of results on a dataset that has many faces of many people, confusion matrix is the most optimal evaluation method. The standard method for evaluating face recognition revolves around determining positive or negative results [18], as shown in Table I.

Description
TP : Faces detected accordingly
FP : Faces detected but wrongly matched
FN : Faces recorded on dataset not detected
TN : Faces not recorded on dataset not detected

D. Results

In the first experiment, Faster R-CNN model was trained using two different learning rates; 0.001 and 0.0001 with 25000 epoch. The testing was conducted by test the image into the model. Based on this scenario, in a learning rate of 0.001, the accuracy value was 72.5% with 80.03% sensitivity and 40% specificity. Then, in learning rate of 0.0001 obtained an accuracy value of 60% with 68.57% sensitivity and 0% specificity. Test Scenario 2 and Discussion

In the second experiment, the best Faster R-CNN model are tested in real time camera connected to PC. Based on experiment results, in learning rate of 0.001, the accuracy value is 30% with 44.44% sensitivity, and 27.27% specificity. While the learning rate of 0.0001 obtained an accuracy value of 35% with 65% sensitivity and 5% specificity. The test results can be seen in Table 3. Based on the experiment results shows that there are external factors in real time camera recording such as lighting factors, and object distance from the camera.

IV. CONCLUSION

In this paper, Faster R-CNN is used for real time face detection on CCTV camera in parking lot. Based on exepiment results, the best Faster R-CNN model parameters are obtained at epoch 25,000 and learning rate 0.001 for real time detection. The number of non-face images detected as faces is very small, proving that Faster R-CNN is good at detecting an object in general, but unable to properly distinguish the same object with different features. On the real-time face recognithon there are still many faces untrained but still detected. This is due to the lack of Faster R-CNN ability in extracting facial features.

REFERENCES

[1] Badan Pusat Statistik. (2018). Statistik Kriminal 2018. Jakarta: Badan Pusat Statistik.

[2] Hartanto, E., Ablisar, M., Mulyadi, M., & Marlina. (2015). Kebijakan Kriminal Pencegahan Pencurian Kendaraan Bermotor. USU Law Journal, Vol.3.No.1, 101-112.

[3] Gowsikhaa, D., and S. Abirami. (2012).Suspicious Human Activity Detection from Surveillance Videos. International Journal on Internet & Distributed Computing Systems 2, no. 2.

[4] Hammami, Mohamed. (2018). Abnormal High-Level Event Recognition in Parking lot." In Intelligent Systems Design and Applications: 17th International Conference on Intelligent Systems Design and Applications (ISDA 2017) held in Delhi, India, December 14-16, 2017, vol. 736, p. 389. Springer,

[5] Shammi, S., Islam, S., Rahman, H.A. and Zaman, H.U.,(2018). An Automated Way of Vehicle Theft Detection in Parking Facilities by Identifying Moving Vehicles in CCTV Video Stream. In 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT) (pp. 36-41). IEEE.

[6] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face Recognition: A Literature Survey. ACM Computing Surveys.

[7] Gupta, D. P., Saxena, N., Sharma, M., & Tripathi, J. (2018). Deep Neural Network for Human Face Recognition. International Journal of Engineering and Manufacturing, 63-71.

[8] Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S. Z., & Hospedales, T. (2015). When Face Recognition Meets with Deep Learning: an Evaluation of Convolutional Neural Networks for Face Recognition. IEEE International Conference on Computer Vision Workshops, 384-392.

[9] Balaban, S. (2015). Deep Learning and Face Recognition: The State of the Art. arXiv:1902.03524v1 .

[10] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies For Accurate Object Detection And Semantic Segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587

[11] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[12] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE transactions on Pattern Analysis and Machine Intelegence, 1137-1149.

[13] Szegedy, C., Vanhoucke, V., Ioffe, S., & Shlens, J. (2016). Rethinking the Inception Architecture for Computer Vision. IEEE conference on Computer Vision and Pattern Recognition (CVPR). , 2818-2826.

[14] Goswami, S. (2018, July 11). A deeper look at how Faster-RCNN works. Retrieved from Medium: https://medium.com/@whatdhack/a-deeper-look-at-how-faster-rcnn-works-84081284e1cd. Accessed at 19 June 2019

[15] Murphy, J. (2016). An Overview of Convolutional Neural Network Architectures for Deep Learning. Microway, Inc.

[16] Scherer, D., Muller, A., & Behnke, S. (2010). Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. 20th International Conference on Artificial Neural Networks (ICANN).

[17] Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378.

[18] Hardt, M., Recht, B., & Singer, Y. (2015). Train faster, generalize better: Stability of stochastic gradient descent. arXiv preprint arXiv:1509.01240.

[19] Agarap, A. F. (2018). Deep Learning using Rectified Linear Units (ReLU). arXiv preprint arXiv:1803.08375.

[20] Sundaram, M., & Mani, A. (2016). Face Recognition: Demystification of Multifarious Aspect in Evaluation Metrics. In S. Ramakrishnan, Face Recognition - Semisupervised Classification, Subspace Projection and Evaluation Methods (pp. 76-92). IntechOpen.