IEEE
EECSI 2020
CONFERENCE
Electrical Engineering, Computer Science and Informatics
October 1-2, 2020 | Yogyakarta, Indonesia

Versi 1.0

The Improvement Impact Performance of Face Detection Using YOLO Algorithm

Institut Teknologi Sepuluh Nopember
Rakha Asyrofi, Yoni Azhar Winata

Index

1. Introduction
2. Literature Review
   a. Related Work
   b. The detection principle of YOLO
3. Implementation
   a. Face Detection using YOLOv3
   b. Augmentation Data
   c. Dataset
   d. Experimental Environment
   e. Image Manipulation for Data Augmentation
4. Conclusion

Introduction

1 Although in various studies, the detection of deep learning objects can overcome obstacles that occur in the feature extraction method, but in fact, deep learning also has its challenges

2 **First,** to work well, deep learning is often carried out on large numbers of interconnected neurons so that it will impact on the need for large resources too and usually requires a long time for training [4].

3 **Second**, problems with overfitting and underfitting are typical when creating learning models. **Third**, there is a lack of training data [5], [6]

4 **The main problem solved** in this paper is **to compare the appropriate manipulation techniques in face detection to add image data to the training model process** so **that it can identify the most suitable technique to use in this problem**. In addition, a program function has been created to implement image manipulation

Literature Review

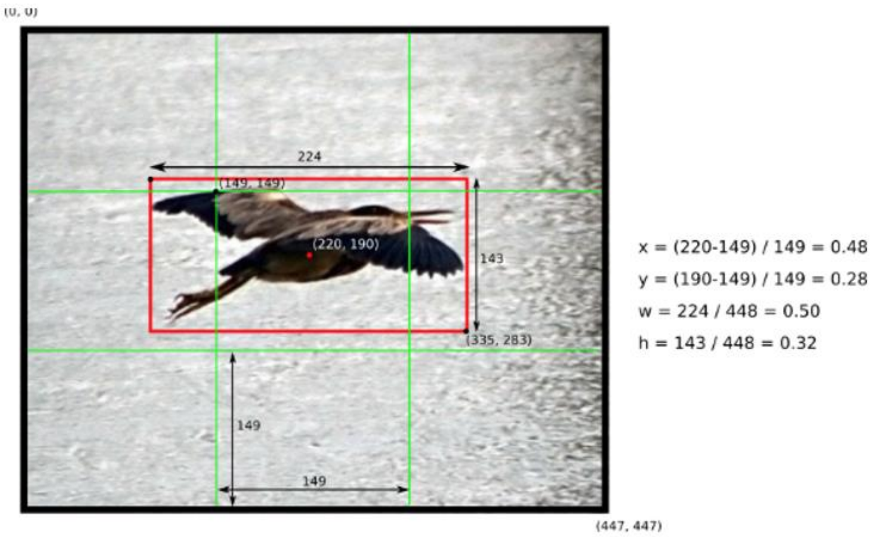


Figure 1. b_x and b_y are bounding annotation from the center of the object detected, b_h and b_w are bounding box height and weight annotations

In equation 1, confidence *C* represents the number of bboxes. Pr indicates whether there are objects in the cell. *IOU* is Intersection Over Union or overlapping rates. *C* will always be 0 when the grid cell is background.

In equation 2, the box area is the prediction of each box, if the value is low it will be deleted based on the threshold. **If there are 2 bboxes** referring to different classes or because there are overlapping objects, the tensor size results are (S * S * (*nB* * 5 + 2 * *nC*)) = (7 * 7 * (2 * 5 + 2 * 2)) = (7 * 7 * 14) tensors. Where, the 1st bbox and the 2nd bbox, refer to different object classes.

$$C(Object) = \Pr(Object) * IOU(Pred, Truth) \qquad (1)$$

$$IOU(Pred, Truth) = \frac{area(box_{truth}) \cap area(box_{pred})}{area(box_{truth}) \cup area(box_{pred})} \qquad (2)$$

The Detection Principle of YOLO

From **equation 3**, there is a Localization Loss where it is the size of the error of the predicted bbox location and size. In the detection, **only what is detected** is taken into account. when *j* from bbox in cell i detect objects in the grid. **If it does not exist**, then the value is set to 0. will increase when the burden of the loss in the coordinates of the bbox also increases.

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\|_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2] \qquad (3)$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\|_{ij}^{obj}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2]$$

When an object in the cell grid is detected then its Confidence Loss is measured in **equation 4**. is the socre confidence box of bbox j in cell grid i. when *j* from bbox in cell *i* detects an object in the grid. If there isn't, then the value is set to 0.

$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}\|_{ij}^{obj}(C_i-\hat{C}_i)^2 \qquad (4)$$

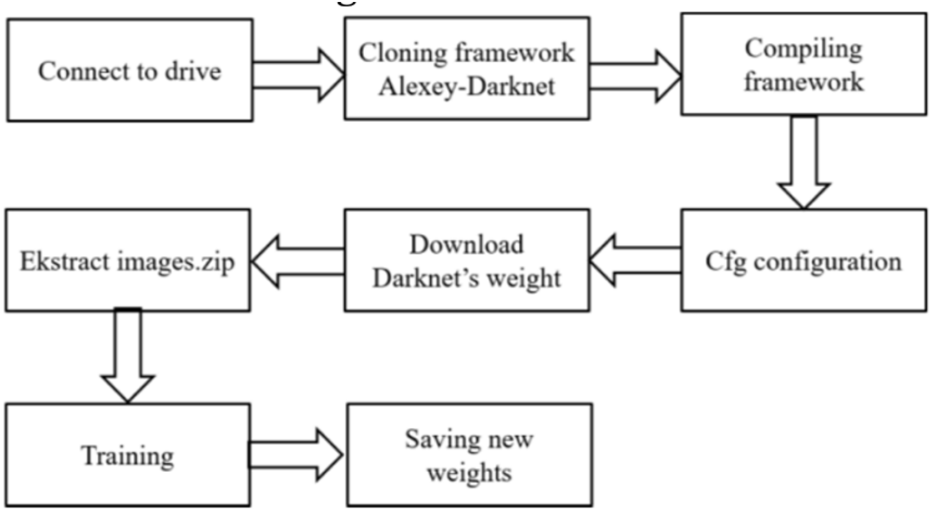
If the object is not detected, then according to **equation 5**, the value is where Ci is the box Confidence socre of box *j* in cell grid *i* and is the weight down and loss when detecting the background of the image [10], [13].

$$+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\|_{ij}^{obj}(C_i-\hat{C}_i)^2 \qquad (5)$$

$$+\sum_{i=0}^{S^2}\|_{i}^{obj}\sum_{C\in classes}(p_i(c)-\hat{p}_i(c))^2$$

Face Detection Using YOLOv3

In this study, **Figure 2** explain the deep learning architecture used is YOLOv3. YOLOv3 uses the Darknet variant, which initially has 53 network layers on Imagenet. For object detection, another 53 layers are stacked on top of it. An architecture consisting of 106 layers makes YOLOv3 fully convolutional. Insert image in YOLO architecture is an image with size 416x416. After the image is inserted, a grid will be created on the image with SxS size [12]. The grid of each cell consists of x, y, w, h and has the confidence of the object. If there is confidence, it is defined in equation (1).



Connect to drive

Cloning framework Alexey-Darknet

Compiling framework

Cfg configuration

Download Darknet's weight

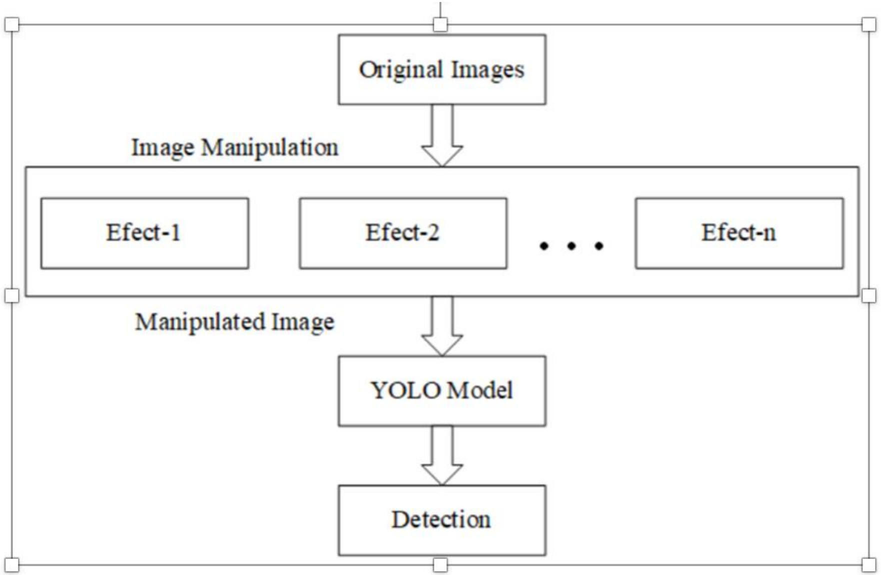
Ekstract images.zip

Training

Saving new weights

Augmentation Data

The implementation according to **Figure 3**, The image effect application has been implemented using libraries in Python. The results from the Image Data Generator have been used by inserting the original image into the generator with the aim of increasing the amount of data. The result of the implementation of this technique, the number of images with manipulated objects, is gradually increased, **from 500 training data to 1000 images**. For mixing images the effect is implemented by **detecting the eyes**, **nose and mouth** on the **face using Haar Cascade** then pasting cropped images such as facial accessories using the library from OpenCV.



Dataset

The experiment in this study was to use the **Celeb dataset** from the **Kaggle repository**. The dataset consists of more than **200000 human face data** where only 500 data will be taken. The dataset is divided into **70% for training**, **15% for validation**, and **15% for testing**. After that, for the data manipulation process, **70% of the 500 training data** will be implemented with some image effects. In training conducted, the image data manipulation algorithm was implemented. The **iteration** is carried out as many as **3000** for the **training process**. **The learning rate is 0.001**, and the learning level is reduced at **5000 and 20000 times.**

Experimental Environment

This research has been conducted in an **Experimental environment** using an Intel Core i5-8250U, CPU is 1.60GHz, GPU is MX250, 8GM RAM. The operating system running on a computer is Windows 10 Professional, 63-bit. While the architecture used is YOLO v3 with the framework is Darknet and the syntax is made using Python 3

Image Manipulation for Data Manipulation

Based on **experiment 1** where a dataset of **500 images without manipulating** the image, it can detect several facial images in the form of **paintings or comics** but often it is also wrong in detecting facial objects. While **experiment 2** with **500 images** of data and image manipulation to **1000 images**, has an increase in the detection rate of facial objects that is not too significant. Based on this experiment, according to **Figure 4 to Figure 6**, to **detect facial objects, whether original images, comics, or writing**, more than **1000 training** data is needed and the data must be varied and at least reflect what kind of facial object you want to get.

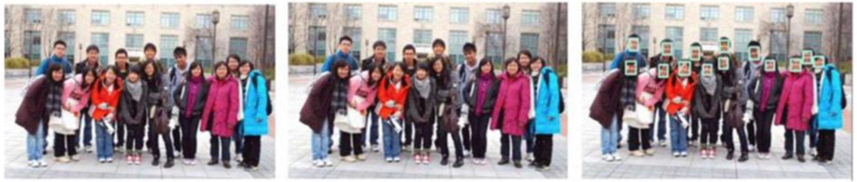


Figure 4. Original Image Experiment Result, Non Augmentation – Augmentation – WIscDER pre-trained

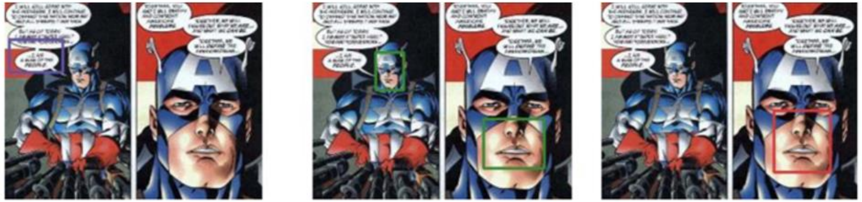


Figure 5. Comic Experiment Result, Non Augmentation – Augmentation – WIDER pre-trained

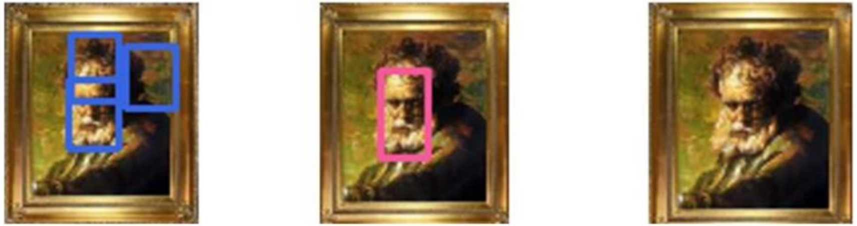


Figure 6. (a) Art Painting Experiment Result, Non Augmentation – Augmentation – WIDER pre-trained

Conclusion and Evaluation

The conclusions obtained are based on the confidence value of the testing results. The experimental results resulted in the fact that with data augmentation, the value of its confidence increased, although not significantly. Experiments using the Wider dataset tend to be able to detect real faces more accurately than using the Celeb dataset. This might happen because Wider has more variations in shooting angles than Celeb. Experiments using the Wider dataset can detect small objects, while using the Celeb dataset cannot detect small face objects.

Thank You

For your attention