Vehicle Detection Simulation using YOLOv4 on Autonomous Vehicle System

Josep Andre Ginting
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
josepandregintings@gmail.com

***Abstract*—Vehicle detection is one of the many tasks that must be completed by an autonomous vehicle system. Currently, the use of deep learning algorithms (deep learning) convolutional neural network (CNN) is very influential in the performance of object detection. YOLO object detector (You Only Look Once) is one of the best performing object detection systems today compared to other object detection systems that can detect objects in real-time. The main problem with real-time object detection is at the expense of accuracy. Reduction of resolution from input to model is usually done to increase speed, but will reduce accuracy due to missing object features. The model used to create the object detection model is the fourth version of YOLO which is trained using the MS COCO dataset so as to produce high mAP detection performance and run at real-time speed in detecting vehicles. The experimental results show that the model built using the fourth version of YOLO produces the most optimal detection model with a good mAP value of up to 71.08% at an IoU threshold of 0.5 and with a real-time detection speed of 30.8. FPS. However, if you want to apply object tracking to the fourth version of the YOLO object detection model which has been previously trained by applying the DeepSORT algorithm, it will slow down the processing speed from 30.8 FPS to 21.45 FPS.**

Keywords—vehicle detection; autonomous vehicle; YOLO; real-time; FPS.

I. INTRODUCTION (*Heading 1*)

Autonomous vehicle (AV) or also called self driving car is a land vehicle that can run autonomously. Autonomous vehicle is expected to be one of the solutions to reduce traffic problems such as traffic accidents due to human negligence as the driver. The number of traffic accidents in Indonesia itself from 2015 to 2019 increased by and average of 4.87 percent per year, and was followed by an increase in the number of deaths and minor injuries of 1.41 percent and 6.26 percent each year. An average of three people die every hour due to traffic accidents, of which 61 percent of accidents are caused by drivers [5].

Computer vision is a discipline in computer science that focuses on discussing and developing the ability of computers to imitate human visual abilities in perceiving an object seen either from images or digital videos [4]. Computer vision task include methods for obtaining, processing, analyzing and understanding digital images, and extracting high- dimensional data from the real world to produce numerical or symbolic information, for example in the form of decision [6].

Masmoudi et al. (2019) has tried to compare the performance of several different learning models in detecting objects for autonomous vehicles, that is Support Vector Machine (SVM), You Only Look Once (YOLO), and Single Shot Multi-box Detector (SSD) to be applied to autonomous vehicle. Based on that research, it is explained that there are two main models with fast object detection are SSD and YOLO. SSD has a better accuracy rate than YOLO but the detection speed is much slower than YOLO [8].

In recent years, SSD has not have an increase in its development, in contra to YOLO which always developed by other developers every year. In the fourth version of YOLO (YOLOv4), there are several additions from the previous version of YOLO such as improvements in the neural network, data augmentation, new activation function, and fast convergent loss function [1]. YOLO itselft is a real-time object detection system which is an object detection system that works at the actual time during the process of event that is happening. The use of YOLO is suitable to be applied in solving problems related to autonomous vehicles because YOLO can detect object in real-time.

The YOLO model used in this research isYOLOv4 (the fourth version of YOLO). The model built will be trained with training data in the form of images of vehicles that have been labeled based on existing vehicles in Indonesia such as car, motorcycle, bus and truck, so that model performance is expected to be better and more accurate than previous research.

II. LITERATURE REVIEW

A. Autonomous Vehicle

Autonomous vehicle (AV) is an unmanned land vehicle that can explore its environment and move without being diven by humans [3]. navigation in autonomous vehicle systems is usually handled with a sensor suite consisting of various types of sensor, including camera, wheel odometry ultrasonic and range sensor (SONAR, RAFAR, and LiDAR). Autonomous vehicle technology is expected to be one of the solution to reduce traffic problems such as traffic accident due to human negligence as the driver. According to the American National

highway Traffic Safety Administration (NHTSA), there are several benefits to the presence of an autonomous vehicle, including security,economic and social benefits, efficiency and comfort, and mobility.

B. Computer Vision

Computer vision is a discipline in computer science that focused on discussing and developing the ability of computers to imitate human visual abilities in perceiving and object seen either from a digital image or video [4]. Computer vision task include methods for obtaining, processing, analyzing and understanding digital images, and extracting high-dimensional data from the real world to produce numerical or symbolic information, for example in the form of decisions.

C. Object Detection

Object detection is a computer technology related to computer vision and image processing that deals with detecting examples of semantic objects of a certain class (such as people, buildings, or cars) in digital images or videos. Methods for object detection generally fall under machine learning based approaches or deep learning based approaches.

D. Machine Learning

Machine learning is the study of the performance of computer algorithms that improves automatically through experience. A computer program is said to learn from experience E with respect to some class of task T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E [9]. In general, there are three types of machine learning approaches, supervised learning, unsupervised learning, and reinforcement learning.

1) Artificial Neural Network

Artificial neural network (ANN) as the name suggest is a model inspired by the workings of neurons in the brain. ANN is a machine learning model capable of pattern recognition. ANN is also often referred to as a neural network. ANN has a several layers that are interconnected. The layer that receivers external data is the input layer, the layer that produces the final result is output layer, and the layer between the input and output layers is the hidden layer.

2) Convolutional Neural Network

Convolutional neural network (CNN) is one of the most popular categories of neural networks, especially for high-dimensional data such as images and videos [khan, s]. CNN is a further development of the multi-layer perceptron in ANN because it uses a similar method but with more dimensions. The advantage of CNN compared to ANN is that it uses larger dimensions so that it will affect the overall scale of an object.

E. You Only Look Once

YOLO or You Only Look Once is a real-time object detection system. YOLO is an object detection that works at the actual time during the process or event is happening [10]. In the fourth version of YOLO called YOLOv4 developed by Bochhkovskiey et al. (2020), added several combinations of features to increase CNN accuracy in the YOLO detection system.

F. DeepSORT

Simple Online and Real-time Tracking (SORT) is pragmatic approach to multiple object tracking with a focus on simple and effective algorithms. In deepSORT, to be able to do object tracking, there must be be object detection first, because object tracking in deepSORT is done by object detection method [11].

III. Problem Analysis and Solution Design

A. Problem Analysis

Vehicle detection on an autonomous vehicle is the main functionality that must be owned. If an autonomous vehicle fails to detect the vehicle in front of it, it will increase the probability of and accident on the road, which means it fails to achieve one of the main goals of the autonomous vehicle itself to reduce the rate of traffic accidents. Therefore, the accuracy of the vehicle detector that will be used by the autonomous vehicle itself must have high accuracy.

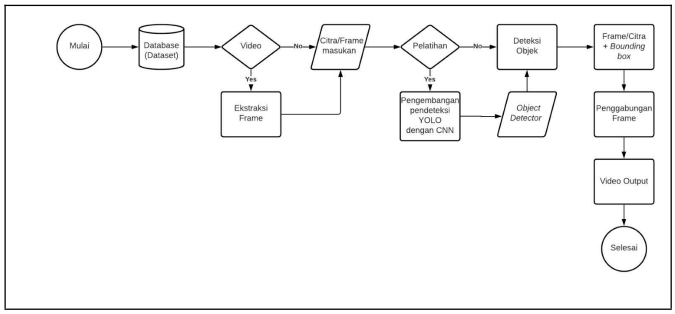
B. Solution Analysis

To solve the problems above, several processes are describes which are derived from related literature studies with the goal of improving model performance in terms of accuracy and processing speed performance. There are various possible solutions for object detection. The alternative solution chossen is based on the ultimate goal of model development is for object detection in autonomous vehicles which requires the model to be able to detect at real-time speed.

C. Solution Design

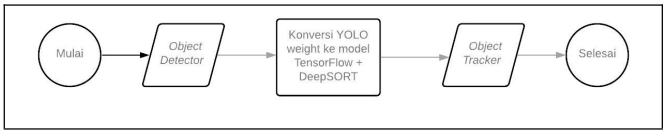
A process flow design will be developed from the initial dataset acquisition, model training, program code generation, to using a detection model to create bounding box. The solution used is YOLOv4 with CNN architecture CSPDarknet-53 which is the default YOLOv4 architecture which will produce a learning model that is used to detect vehicles on the highway. The overall steps of the development process are the process of extracting video dataset into frames/ images, model training, model development with CNN, to using a detection model to add bounding box detection result to frames/images. The problem solution is implemented according to the design described in Figure 1.

Fig. 1. *Vehicle object detection system development steps*



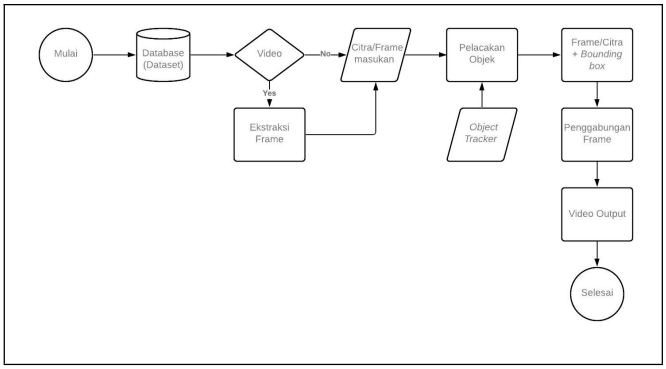
After the object detection model has ben trained and can detect it well, to be able to do object tracking, it is necessary to add a deepSORT algorithm. The implementation of deepSORT is done with the help of the library from TensorFlow. To b able to add deepSORT with the TensorFlow library, the object detection model that has been trained previously and stored in the YOLO weight format must first be converted to the TensorFlow model format. The conversion step can be seen in Figure 2.

Fig. 2. Steps to convert YOLO weight to TensorFlow model



After being converted into the TensorFlow model, the object detection model is added with the deepSORT algorithm to be able to perform object tracking so as to produce an object tracker model. The model can be used to track object on a video with steps similar to the object detection process, or more details can be seen in Figure 3.

Fig. 3. Object tracking process



1) Dataset Acquisition

In this study, a dataset sourced from MS COCO is used to do training and validation of the model created, and will take the dataset directly on roads in Indonesia for testing. The use of the MS COCO dataset is because this dataset is a goog dataset and is commonly used as a benchmark in computer vision research. The MS COCO dataset itself on the object detection feature consists of 121,408 images, 883,331 object notations, 80 classes, with a median image ratio 640 x 480. The vehicles used in the training itself only consist of four classes that is car, motorcycle, bus, and truck.

2) Object Detection Model Creation

The making of a vehicle object detection model with YOLOv4 is divided into three main stages, that is preparation, feature extraction and model training.

a) Preparation: Prepare the image to be used in the training phase. The dataset is data from the MS COCO dataset with classes consisting of cars, motorcycles, buses, and trucks. Each object in the training images has been labeled according to the type of vehicle in the image.

b) Feature Extraction: Feature extraction on the image to find out the important features of the image which will later be used as a learning feature in deep learning using CNN.

c) Model Training: The result of feature extraction on the training image is then used as input for the CNN deep learning algorithm with the CSPDarknet-53 backbone on YOLOv4 to generate a vehicle object detection model.

3) Convert YOLO weight to TensorFlow

The object detection model that has been trained using YOLOv4 will first be converted into TensorFlow model and the deepSORT algorithm added to be able to track objects. The object detection model that has been added with deepSORT algorithm will produce tracking model.

4) Test Design

The test is carried out by looking at the result of the model detection accuracy by calculating the mean average precision (mAP) and the frame per second (FPS) detection rate. The test was carried out by experimenting by testing the models made both object detection and object tracking directly on video recordings on highways in Indonesia. The result of the experiment will be compared with existing models in previous studies to test whether the detection produces better result than the existing vehicle object detection model.

IV. Implementation and Testing

A. Implementation Environment

The implementation of created model is carried out in a Cloud environment with the following specification: Inter® Xeon® CPU @ 2.30Ghs, Nvidia Tesla V100, 25.46 GB memory, 147.15 storage, Ubuntu 18.04.5 LTS operating system, Python 3.7.10, and some of libraries include OpenCV, darknet, TensorFlow, Numpy, Glob, Time and google colab library : drive. The Cloud environment used is Google Colaboratory (Colab) where the notebook will be connected to a virtual machine with a maximum program active period of twelve hours and automatically terminated after that. Cloud usage is required to be able to run the object detection model with the expected performance. This is because the model can run optimally with the Nvidia GPU (Graphics Processing Unit) which is provided free or paid by Google Colaboratory.

B. Object Detection Model Implementation

The implementation process of the vehicle object detection model is carried out in several stages, consisting of dataset preparation, environment preparatio, model configuration, model training, and also selecting the best model based on the best evaluation metrics.

1) Dataset Preparation

The MS COCO dataset has grouped the datasets into training and validation datasets with a comparison of training data and overall validation before being selected based on the class to be trained, which is 118K versus 5K [3]. After being taken based on the class that includes the vehicle to be trained, the number of images in the training dataset containing car objects is 12,251, images containing motorcycle objects are 3,502 images, images containing bus objects are 3,952 images,

and images containing truck objects are 3,952 images. 6127 images. For each existing image may contain objects of cars, motorcycles, buses, and trucks at the same time. The validation dataset consists of 535 images containing car objects, 159 motorcycle images, 189 bus images, and 250 truck images. For each existing image may contain objects of cars, motorcycles, buses, and trucks at the same time.

The bounding box annotation format in the MS COCO dataset is different from the bounding box annotation format used in the YOLO darknet training. In addition to the difference in the format of writing annotations, the annotations in the MS COCO dataset are also stored in a different file format, the MS COCO dataset is stored in JSON form called MS COCO JSON which stores all image annotations in a JSON file, while that used by the darknet YOLO structure saved in TXT file format where each TXT file only saves an annotation of an image. Because there are different annotation formats and also different annotation storage file formats, it is necessary to make adjustments first by converting the annotation format from MS COCO format to YOLO darknet format and saving the annotation data of each image into a TXT file before training.

For testing, data will be taken by recording the road conditions in the Berastagi District, Karo Regency, North Sumatra, Indonesia from the top of the car dashboard. The video recording is used because it contains the actual condition of the roads in Indonesia. Video recordings were taken in bright conditions during the day and there were no disturbances such as rain. The video captured and used as a test dataset can be accessed at the following link: https://drive.google.com/drive/folders/1VKJBCNNexRRUI3Hni3S- oyHgFGim1G_g?usp=sharing.

2) Training Environment Preparation

After the bounding box annotation format of the dataset matches the format used by YOLO darknet, then the development environment used by darknet is prepared. A detailed explanation of each file or folder is as follows.

a) Darknet: a folder that contains darknet program.

b) Dataset: a folder containing training, validation and testing datasets.

c) Training: a folder containing weights after completing the training process.

d) cfg: folder containing configuration data used on YOLO darknet.

e) train.txt: contains a list of images used for training.

f) val.txt: contains a list of images used for validation.

g) obj.data: contains data or information about the datasets used and the location of obj.names, directory of training, testing and checkpoints.

h) obj.names: contains the class names in the datasets.

The obj.data file will contain information about the dataset such as the number of classes, the location of the image list for training and validation, the location of the obj.names file, and the location of backup checkpoints during training. The obj.names file will contain the class names of the objects to be trained, namely cars, motorcycles, buses and trucks.

3) Model Configuration

The configuration used by the darknet model is to conduct training models on the MS COCO dataset, so there is no need to change the configuration too much. The configuration that needs to be changed is to adjust the number of classes from training to four, namely for car, motorcycle, bus and truck classes. Also made adjustments from the maximum number of batches to an amount or greater than the number of training data (30000 batches), and steps adjusted to the maximum number of batches (80% and 90% of the maximum number of batches), as well as changing the filter in each convolution layer before YOLO becomes

$$filter = (number\ of\ classes + 5) * 3 \quad (1)$$

Since the model will have four classes, namely cars, motorcycles, buses and trucks, the filter value will be 27.

TABLE I. PARAMETER CONFIGURATION ON YOLOV4 AND YOLOV4-TINY MODELS

	Parameter	**YOLOv4**	**YOLOv4-tiny**
1	Batch size	64	
2	Subdivision	16	
3	Network size	416x416, 384x384	
4	Channels	3	
5	Momentum	0.9	0.949
6	Decay	0.0005	
7	Angle	0	
8	Saturnation	1.5	
9	Exposure	1.5	
10	Hue	0.1	
11	Learning rate	0.001	0.00261
12	Burn in	1000	
13	Max batch	30000	
14	Steps	24000, 27000	
15	Scales	0.1, 0.1	
16	Activation function	Leaky, Linear, Mish	Leaky, Linear
17	IoU	0.5	

4) Model Training and Selection of the Best Model

The object detection model training is carried out by running the darknet program and entering the appropriate configuration according to the dataset used. The run program will save the training progress every 100 iterations and every 10000 iterations will be created a backup weight. After running the batch to the end, each model will be tested for its performance by checking the resulting mean average precision (mAP). The best results are taken as the model that will be used for detection. The test uses a YOLO darknet program with an argument detector map to test the average precision (AP) of each class, then calculates the average AP of each class. There will be four classes, namely car, motorcycle, truck and bus

classes, so the average of the four classes will be calculated to get the mAP.

C. Convert YOLO weight to TensorFlow model

After the model is trained, the model generated by training on YOLO can only perform object detection. To be able to perform object tracking, the previously trained YOLO model must be converted into another format, namely the TensorFlow model. In the TenforFlow model, a deepSORT algorithm will be added so that the model can perform object tracking. This TensorFlow model will be used to perform object detection and object tracking using the deepSORT algorithm.

D. Model Testing

The vehicle detection system that has been created and developed is then tested by taking into account accuracy metrics with mAP and also processing speed in FPS. This section will explain the testing process carried out on the object detection model that has been built.

1) Testing Purpose

The purpose of this test as a whole is as follows:

a) See the results of the accuracy and speed of the vehicle detection system that has been built.

b) Finding the best combination of parameters for the vehicle detection system developed by looking at the performance of accuracy and processing speed.

c) Comparing the differences in the results of object detection / object detection from the YOLO model with object tracking from the YOLO model that has been converted to the TensorFlow + deepSORT model.

d) Comparing the performance of the vehicle detection system model developed with the state-of-the-art model.

2) Test result

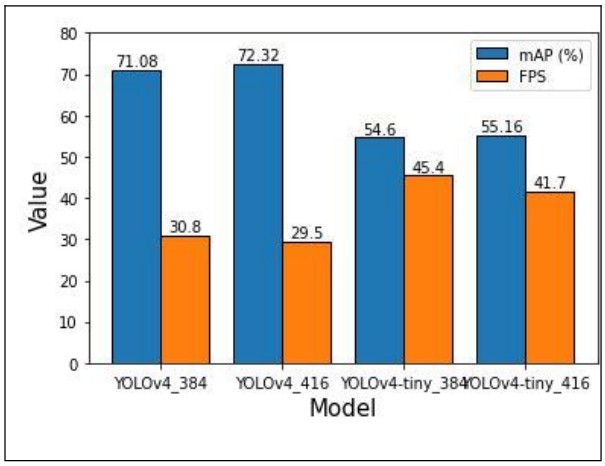
The making of a vehicle object detection model with YOLOv4 is divided into three main stages, that is preparation, feature extraction and model training.

TABLE II. DETECTION RESULT OF YOLOV4 AND YOLOV4-TINY MODELS WITH AN IOU THRESHOLD 0.5

Model	Network size	FPS	mAP (%)
YOLOv4	384 x 384	30.8	71.08
	416 x 416	29.5	72.32
YOLOv4-tiny	384 x 384	45.4	54.60
	416 x 416	41.7	55.16

Table II is the result of testing the YOLOv4 and YOLOv4-tiny models. It can be seen that the YOLOv4 model produces a slower detection speed than the YOLOv4-tiny model, but has a better mAP value. This is because the YOLOv4 model has a thicker architecture (more layers) than the smaller YOLOv4.

Fig. 4. Model comparison chart



The previously trained YOLO weight object detection model will be converted into a TensorFlow model, then added with the deepSORT algorithm to be able to perform object tracking. The result of the object tracking speed test can be seen in the following table.

TABLE III. COMPARISON OF THE OBJECT TRACKING SPEED OF THE MODEL

Model	Network size	FPS
YOLOv4	384 x 384	21.45
	416 x 416	20.13
YOLOv4-tiny	384 x 384	41.32
	416 x 416	40.76

Fig. 5. Comparison of object detection (top) and object tracking (bottom) on the YOLOv4 416x416 model

In Figure 5 can be seen the difference in the display of the results of object detection and object tracking. In object detection, an image will be generated along with the bounding box of the detected object, each vehicle class will have a different color bounding box and there is a class name on the top. Whereas in object tracking, each object in the video frame will be given a different color bounding box, as well as a class name at the top.

V. Conclusions and Future Work

A. Conclusions

The YOLOv4 detector is one of the best object detection systems currenly available that can perform the detection process in real-time. The metrics generated in the test are quite good. Based on the result of the analysis of the tests that have been carried out on developed system, the following conclusions can be drawn:

1) The YOLO object detection model with the CNN CSPDarknet53 backbone architecture with a combination of trained configuration successfully detects vehicle objects with real-time speed in the working environment.

2) The object detection model with YOLOv4 with a network size configuration of 416 x 416 at an IoU threshold of 0.5 has the best mean average precision (mAP) value of 72.32% compared to models with other network sizes or compared to the YOLOv4-tiny model.

3) The YOLOv4 object detection model coupled with the DeepSORT algorithm successfully detects and tracks vehicles on the highway well.

B. Future Work

The result of the study are quite good. However, there are still many tings that can be improved so that the detection performance can run more optimally. The following suggestions can be made to develop future research:

1) Using more datasets and with vehicle types that are more suitable for the location of the system test, in this case in Indonesia. The goal is because in the dataset that is currently used, the vehicle image in the training and validation data is an image of a vehicle from outside Indonesia, so there are several types of vehicles that are not very suitable for vehicles in Indonesia. With the use of a more appropriate dataset, of course, the system built can produce better detection and evaluation results of test data will be more representative of real conditions in the territory of Indonesia.

2) Using a dataset that has a more balanced distribution of the number of objects in the image between all objects to be detected in the case of this study, namely cars, motorcycles, buses, and trucks. The dataset used today is MS COCO, the distribution of the number of images containing objects of cars, motorcycles, buses, and trucks is still not evenly distributed.

3) Added a dataset for vehicle images taken from the rear. This is because in the autonomous vehicle module you will often find vehicle images taken from the rear, while in the MS COCO dataset used in this study very few vehicle images are taken from the back of the vehicle.

4) Conduct model training on training data that have more varied environmental conditions, such as rainy, snowy, morning, afternoon, evening or other conditions that may occur that can affect detection performance.

5) Using larger and unlimited computing resources so that the training process is not hampered by technical constraints, such as the training process being cut off due to the time limit of the use of the Google Collaboratory used in this study.

References

[1] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934..

[2] Common Objects in Context (2020). Adapted on August 18, 2021 from https://cocodataset.org/.

[3] Gehrig, S. K., & Stein, F. J. (1999, October). Dead reckoning and cartography using stereo vision for an autonomous car. In Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No. 99CH36289) (Vol. 3, pp. 1507-1512). IEEE.

[4] Huang, T. (1996). Computer vision: Evolution and promise.

[5] Kementerian Kesehatan Republik Indonesia (2017). Disadur pada tanggal 21 Novermber 2020 dari https://www.kemkes.go.id/article/print/17082100002/rata-rata-3-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan.html.

[6] Klette, R. (2014). Concise computer vision. Springer, London.

[7] Litman, T. (2020). Autonomous vehicle implementation predictions: Implications for transport planning

[8] Masmoudi, M., Ghazzai, H., Frikha, M., & Massoud, Y. (2019, September). Object detection learning techniques for autonomous vehicle applications. In 2019 IEEE International Conference of Vehicular Electronics and Safety (ICVES) (pp. 1-5). IEEE.

[9] Mitchell, T.M. (1997). Machine Learning. New York, NY: McGraw-Hill.

[10] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[11] Wojke, N., Bewley, A., & Paul, D. (2017, September). Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649). IEEE.