

Deteksi Sepeda Motor di Jalan Raya Menggunakan *Faster R-CNN* Berbasis *VGG16*

Detecting Motorcycle at Traffic Roads with Faster VGG16 based R-CNN

Moch Dian Lazuardi Yudha¹, Wawan Setiawan², Yaya Wihardi³

Departemen Pendidikan Ilmu Komputer
Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam
Universitas Pendidikan Indonesia
Bandung, Indonesia

¹dian_0856@student.upi.edu, ²wawans@upi.edu, ³yayawihardi@upi.edu

Abstrak—Penelitian ini bertujuan untuk mengembangkan sistem deteksi objek dengan masukan berupa data citra dalam ukuran bebas. Pengembangan model sistem deteksi objek dilakukan penerapan *Machine Learning* untuk mengatasi deteksi objek pada sebuah citra dengan metode *Faster R-CNN* yang berbasis algoritma *VGG16*. Sistem yang dikembangkan menghasilkan *bounding box* pada sebuah objek yang berada didalam citra. Pengembangan sistem dilakukan dalam bahasa pemrograman *Python* dengan memanfaatkan beberapa *library* seperti *Keras*. Eksperimen dilakukan dengan mengukur nilai *loss* dari data *training* yang dimasukkan kedalam sistem. Hasil eksperimen menunjukkan bahwa informasi yang dihasilkan terbukti dapat mendeteksi objek pada sebuah citra yang diberikan. Sistem ini dapat menghasilkan informasi berdasarkan data citra yang telah di-*training* dengan sistem ini.

Kata Kunci— *Computer Vision; Machine Learning; Faster RCNN; VGG16; Convolutional Neural Network; Regional Proposal Network; ROI Pooling*

I. PENDAHULUAN

Dengan perkembangan zaman yang semakin maju, kendaraan di dunia semakin modern dan semakin banyak [1]. Kendaraan tersebut dapat dideteksi melalui sistem dengan pengambilan data yang berasal dari *CCTV*. Sistem tersebut dapat memanfaatkan algoritma *Convolutional Neural Network* yang telah di-*bundle* dengan *library Keras* dan terbukti baik dalam segi performa [2].

Convolutional Neural Network adalah *Neural Network* yang memiliki banyak *layer* dan disetiap *layer* nya memiliki bidang dua dimensi dan masing-masing bidang terdiri dari beberapa *neuron independent* [3]. Umumnya *Convolutional Neural Network* digunakan untuk mengklasifikasi citra yang nantinya akan menghasilkan label kelas dan proses ini disebut dengan *machine learning* [4].

Machine Learning adalah bagian dari ilmu computer yang dapat membelajarkan computer sehingga memiliki kemampuan untuk belajar tanpa di-*program* secara eksplisit [6]. *Machine learning* terbaru menjadi 2 bagian, yaitu *supervised learning* dan *unsupervised learning* [5].

Convolutional Neural Network memiliki arsitektur yang dinamakan *VGG16* yang telah terbukti menghasilkan objek yang akurat [7]. Dikembangkan dengan arsitektur yang lebih kompleks, *VGG16* menjadi basis utama dari algoritma *Faster R-CNN* yang dapat menghasilkan 91.4% kebenaran objek pada suatu dataset [7].

Arsitektur *Faster R-CNN* memiliki tiga bagian utama (model *Convolutional Neural Network*, *Region Proposal Network*, dan *ROI Pooling Layer*) [8]. Pada bagian model *Convolutional Neural Network*, bisa diisi dengan berbagai arsitektur *Convolutional Neural Network* seperti *resnet* dan *vgg* [8]. Arsitektur *Faster R-CNN* dengan dasar algoritma *VGG16* membutuhkan GPU yang mempunyai kapasitas minimal 3GB untuk tahapan *training*, sedangkan arsitektur *YoloV3* membutuhkan GPU yang mempunyai kapasitas minimal 4GB [7].

Dalam pembangunan sistem pendeteksi motor di lalu lintas, penelitian ini memakai arsitektur *Faster RCNN* untuk mempelajari data berupa citra yang terdiri dari 600 data. Penulis juga memanfaatkan *library* yang terdapat dalam bahasa pemrograman *Python* seperti *Keras* sebagai *backend* dari sistem yang akan dikembangkan.

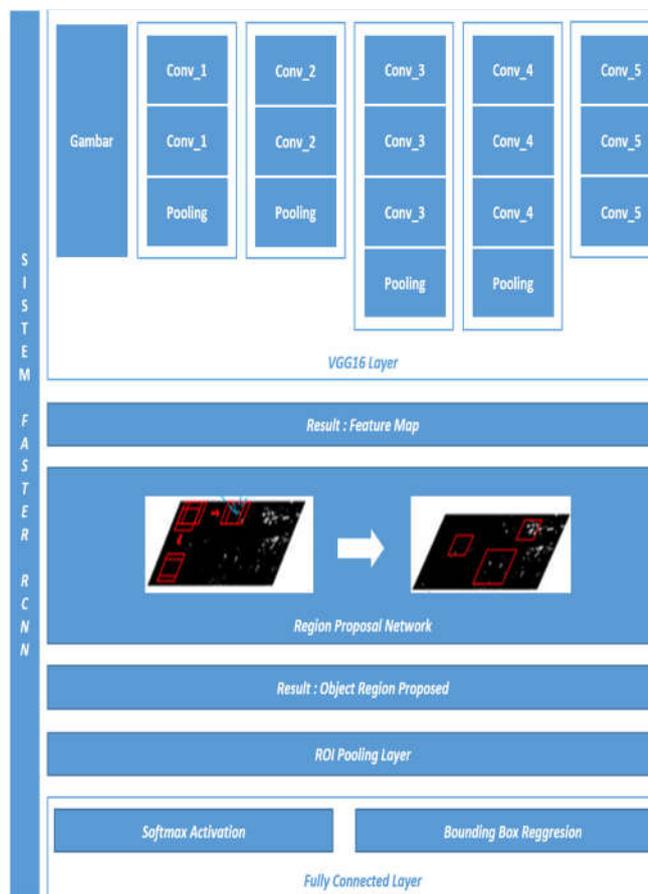
II. PENELITIAN TERKAIT

Pengembangan sistem deteksi menggunakan arsitektur *Faster RCNN* telah dibangun pada penelitian sebelumnya, diantaranya sistem deteksi buah kiwi yang mampu mendeteksi buah kiwi dengan bantuan arsitektur *Faster RCNN* yang berbasis arsitektur *VGG16* [7]. Terdapat penelitian lain juga yang dapat men-*tracking*

manusia dengan arsitektur *Faster RCNN* yang berbasis *VGG16* [9].

III. MODEL SISTEM ALGORITMA DETEKSI MOTOR DI JALAN RAYA

Faster R-CNN berbasis *VGG16* terbagi kedalam empat bagian utama yaitu dasar model algoritma *VGG16*, *Region Proposal Network*, *ROI Pooling Layer*, dan *Fully Connected Layer* [8]. Pada penelitian ini dilakukan konfigurasi dengan memasukkan learning rate $1e - 5$ dan 35 epoch. Tidak ada perubahan arsitektur algoritma pada penelitian ini. Model arsitektur ini menggambarkan ringkasan tahap pengembangan penelitian.



GAMBAR 1 MODEL ALGORITMA *FASTER RCNN* BERBASIS *VGG16*

A. *VGG16 Model Layer*

Data yang dapat diproses oleh tahap ini yaitu 600 x 800 Pixel. Oleh karena itu harus dilakukan *resize* pada sebuah gambar yang didapat. Tahapan ini bertujuan untuk *training* data yang telah melalui pra proses (*resize*,

labeling object). Hasil dari tahapan ini yaitu *feature map* yang akan digunakan untuk tahapan *region proposed network*.

B. *Region Proposed Network*

Pada proses ini dilakukan pengambilan objek yang lebih akurat. Untuk *generate* proposal *region*, dilakukan proses *sliding window* menggunakan anchor. Untuk model *VGG16*, dimensi anchornya 512 dimensi [10]. Dengan *sliding window* tersebut, pengklasifikasi menentukan probabilitas proposal yang memiliki objek yang sesungguhnya dan meregresi koordinat proposal. Anchor yang digunakan pada *Region Proposal Network* dalam penelitian ini memiliki skala 128, 256, dan 512 yang berfungsi untuk memproporsi *regional* pada suatu objek didalam gambar. Hasil dari tahap ini yaitu hilangnya *bounding box* yang tidak terpakai di suatu objek karena telah ditetapkan *bounding box* asli objek tersebut didalam suatu gambar. Dalam perhitungannya, *bounding box* yang memiliki nilai kurang dari 0.7 nantinya akan dihapus.

C. *ROI Pooling Layer*

Masukan dari hasil *Region Proposal Network* akan diolah oleh tahapan ini. Tahapan ini menggunakan *max pooling layer* untuk meng-convert fitur didalam *region* yang telah ditentukan kedalam fitur map yang lebih kecil atau mem-*fixing* fitur map tersebut [11]. Tahapan ini bekerja dengan membagi $h \times w$ *roi window* menjadi grid $H \times W$ dari sub *window* lalu *max pooling* nilai di masing-masing sub *window* ke dalam *grid cell* tersebut [11].

D. *Fully Connected Layer*

Tahapan ini adalah tahapan terakhir dari model arsitektur *Faster RCNN*. Dalam tahapan ini hanya terdapat softmax dan regresi *bounding box*. Dalam library tensorflow, penulis menggunakan aktivasi *softmax* dan *linear*. Tahapan ini bertujuan untuk prediksi objek kelas nya dan regresi *bounding box*.

IV. DESAIN EKSPERIMEN

Eksperimen dilakukan dengan cara membedakan dua data yang berjumlah 280 objek motor dan 620 objek motor. Kedua test case tersebut memiliki sumber yang sama yaitu *CCTV Online Dishub* Kab. Sukoharjo dengan pengambilan video pada pagi hari dan siang hari. Penelitian ini mengambil gambar pada jam tersebut dikarenakan cahaya yang cocok untuk pengambilan data dan dilakukan *training* oleh sistem. Penelitian ini menggunakan bahasa pemrograman *python* yang dibantu dengan library *keras* sebagai *backend* sistem, *opencv* sebagai pengolahan citra dan *labelimg* sebagai aplikasi untuk pelabelan objek pada tahap awal. Algoritma *Faster*

RCNN dengan berbasis VGG16 dibuat dengan bantuan library keras.

Tabel I
TEST CASE EXPERIENCE

Eksperimen	Jumlah Objek	Jumlah Epoch	Sumber
Eks1	280 Objek	37 Epoch	Website CCTV Dishub Kab.Sukoharjo
Eks2	620 Objek	35 Epoch	Website CCTV Dishub Kab.Sukoharjo

V. HASIL DAN PEMBAHASAN

A. Perbandingan Output Sistem dengan penelitian terkait

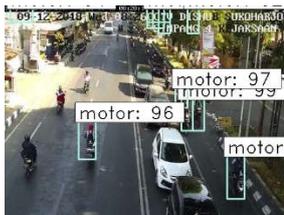
Untuk memperjelas bagaimana akurasi algoritma *Faster RCNN*, maka dari itu penulis membandingkan *output* dengan penelitian terkait seperti gambar 2 dan gambar 3, peneliti menggunakan data yang berbeda dari dua jurnal tersebut.



Gambar 2 Hasil buah kiwi pada pagi hari [3]



Gambar 3 Hasil tracking manusia dengan CCTV [4]



Gambar 4 Hasil deteksi motor pada sistem

Berdasarkan gambar 2, bisa dilihat bahwa buah kiwi dapat dideteksi dengan sangat jelas oleh algoritma *Faster RCNN*. Pada penelitian tersebut, penulis menggunakan dataset sebanyak 28800 objek buah kiwi [7]. Pada gambar 2, terdapat kesalahan deteksi yang ditandakan dengan lingkaran kuning, pada sistem tersebut, lingkaran kuning yaitu buah kiwi yang tidak terdeteksi oleh sistem. Gambar 3 menggambarkan tracking manusia dan bisa disimpulkan *Faster RCNN* dapat mendeteksi manusia

yang terdapat pada CCTV. Gambar 4 adalah hasil dari penitilian ini. Peneliti hanya ingin mendeteksi motor yang menghadap kebawah, tetapi sistem melakukan kesalahan dengan mendeteksi motor yang menghadap keatas. Pada penitilian ini, penulis menggunakan 620 objek motor untuk tahapan *training*.

B. Hasil Eksperimen

Penelitian ini terdapat 2 eksperimen yang berbeda jumlah data dengan menilai berdasarkan *output* nilai *loss* pada setiap epoch yang didapatkan. Sehingga didapatkan hasil dari dua eksperimen tersebut pada table II.

Tabel II

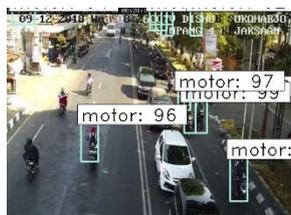
HASIL PENGUKURAN ASPEK READABILITY

Epoch	Eksperimen 1 (280 Objek)	Eksperimen 2 (620 Objek)
1	2.0636	2.9509
2	0.6355	1.0397
3	0.3967	0.6937
4	0.3227	0.5109
5	0.2881	0.4627
6	0.2262	0.3986
7	0.2053	0.3736
8	0.2405	0.3651
9	0.2683	0.3753
10	0.1695	0.2724
11	0.1388	0.2563
12	0.2088	0.2593
13	0.1463	0.2548
14	0.1481	0.2774
15	0.1732	0.2163
16	0.1348	0.2074
17	0.12	0.1916
18	0.1254	0.2225
19	0.1346	0.1901
20	0.0928	0.1654
21	0.1112	0.1586
22	0.1097	0.1776
23	0.087	0.1702
24	0.0907	0.1759
25	0.1218	0.1528
26	0.113	0.1422
27	0.0842	0.1521
28	0.0856	0.1423
29	0.0737	0.1691
30	0.0786	0.1311
31	0.0682	0.1355
32	0.0844	0.1324
33	0.0658	0.1211
34	0.0601	0.1307
35	0.0796	0.1211
36	0.0703	-
37	0.0627	-

Selain nilai *loss*, sistem sudah bisa mendeteksi objek dalam sebuah citra yang terlihat pada gambar 5 dan gambar 6.



Gambar 5 Hasil deteksi motor dalam eksperimen 1



Gambar 6 Hasil deteksi motor dalam eksperimen 2

VI. KESIMPULAN

Pembuatan sistem deteksi motor di lalu lintas menggunakan algoritma *Faster RCNN* bisa terbilang akurat. Sistem tersebut sudah bisa mendeteksi motor yang diinginkan oleh penulis dan dapat berjalan dengan berbeda *size bounding box* pada setiap objek. Algoritma ini bisa dikembangkan ke tujuan yang lain seperti deteksi pelanggaran di lalu lintas menggunakan CCTV, selain itu juga bisa digunakan untuk *tracking* objek pada video maupun CCTV. Tetapi pada penelitian ini penulis hanya menggunakan citra untuk tahapan *testing*.

Penelitian ini menjawab bahwa algoritma *Faster RCNN* masih dapat mendeteksi suatu objek bahkan dengan dataset yang terbilang kecil dan hanya dilakukan iterasi sebanyak 35 epoch. Tetapi jika dilihat pada hasil eksperimen bahwa dataset yang memiliki data yang lebih banyak akan mendeteksi objek yang lebih akurat dibandingkan dataset yang lebih kecil.

Manfaat dari penelitian ini yaitu penulis dapat mengetahui bagaimana *convolutional neural network* dapat berjalan dengan baik dengan arsitektur yang lebih kompleks. Karena, pada saat ini *convolutional neural network* dengan arsitektur *faster rcnn* banyak dipakai dibidang *computer vision*.

Kesimpulan dari keseluruhan hasil eksperimen yang dilakukan, keluaran dari sistem tidak sepenuhnya akurat tetapi bisa dibidang 95% objek yang terdeteksi sudah akurat. Nilai *loss* yang dipakai pada penelitian ini yaitu 0.0601 untuk eksperimen pertama dan 0.1211 untuk penelitian kedua. Maupun eksperimen kedua mendapatkan nilai *loss* yang lebih besar dari nilai pertama, tetapi objek yang terdeteksi lebih akurat dari eksperimen pertama. Kemungkinan besar dikarenakan data yang dipakai pada eksperimen kedua lebih besar maka dari itu sistem akan mempelajari lebih dalam terkait objek yang ingin dideteksi.

Untuk penelitian berikutnya dapat dilakukan pengembangan sistem untuk mendeteksi kendaraan yang kecelakaan dan kendaraan yang melanggar lalu lintas. Selain itu juga *faster rcnn* dapat diubah dasar model algoritma seperti memakai dasar algoritma *resnet*.

DAFTAR PUSTAKA

- [1] Zhu, Wentao, dkk, (2012), *Vehicle Detection in Driving Simulation Using Extreme Learning Machine*. Intitute of Computing Technology, China.
- [2] Bianco, Simone., Marco, B., Davide, M., Raimondo, S., (2016). *Deep Learning for Logo Recognition*. University Degli Study. Milano.
- [3] Zhan, Shu, Qin, T., Xiao, Li. (2015). *Face Detection Using Representation Learning*. Hefei University of Technology. China.
- [4] Wang, Yi, Zhiming, L., Pierre, J. (2016). *Interactive Deep Learning Method for Segmenting Moving Objects*. University of Sherbrooke. Canada.
- [5] Riza, L. S. (2015). *Data Science and Big Data Processing in R: Representations and Software*. Granada: Universidad de Granada.
- [6] Arthur, S. L. (1959). *Some Studies in Machine Learning Using the Game of Checkers*. IBM Journal of Research and Development.
- [7] Song, Zhenzhen., dkk, (2019), *Kiwifruit detection in field images using Faster R-CNN with VGG16*. Northwest A&F University, China.
- [8] Deng, Zhipeng, Hao Sun, Shilin Zhou, Juanping Zhao, Lin Lei, Huanxin Zou. (2018). *Multi-scale Object Detection in Remote Sensing Imagery with Convolutional Neural Network*. Changsha: College of Electronic Science.
- [9] Chahyati, Dina, Mohamad Ivan Fanany, Aniati Murni Arymurthy. (2017). *Tracking People by Detection Using CNN Features*. Universitas Indonesia, Indonesia.
- [10] Ren, Shaoqing, Kaiming He, Ross Girshick, Jian Sun. (2016). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. University of Science and Technology China. China.
- [11] Girshick, Ross, (2015). *Fast R-CNN*. Microsoft Research. Microsoft.