PROPOSAL SKRIPSI

SISTEM DETEKSI MANUSIA MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE (YOLO) V2* PADA CITRA PEJALAN KAKI

OLEH:

DIAN WIBOWO

(160411100044)

PROGRAM STUDI INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2019

Abstrak

Sekarang ini banyak berkembang berbagai metode pada recognition system untuk berbagai keperluan yang berbeda-beda. Khususnya implementasi computer vision (CV) yang dapat diterapkan dalam berbagai sistem otomatisasi. Setelah suksesnya algoritma untuk deteksi wajah kini berkembang lagi algoritma human detection (deteksi manusia) secara keseluruhan, jika sebelumnya hanya berfokus pada wajah saja, maka algoritma deteksi manusia dapat mengidentifikasi secara keseluruhan fitur yang ada pada tubuh manusia. Salah satu algoritma deteksi manusia yaitu YOLO (You Only Look Once) v2 sebagai pembaharuan dari algoritma sebelumnya yaitu YOLO. YOLO v2 merupakan cara deteksi citra menggunakan metode Convolutional Neural Network (CNN) tanpa menggunakan region proposal atau kandidat area, serta dapat memecahkan masalah deteksi objek dalam regresi tunggal, dimana regresi merupakan rentang nilai antar 0 sampai dengan 1 untuk mengalokasikan bounding box pada output. YOLO v2 digunakan untuk deteksi objek pada citra sehingga dapat memprediksi apa saja objek yang ada pada citra dan dimana lokasi objek itu berada berdasarkan koordinat bounding box dan probabilitas kelas yang memungkinkan merepresentasikan sebuah objek. Deteksi objek yang dilakukan oleh YOLO v2 adalah dengan cara membagi citra dengan beberapa grid , dimana proses deteksi objek dilakukan dengan CNN dilakukan pada tiap grid tersebut.

Kata kunci : deteksi manusia, YOLO v2 , Convolutional Neural Network, Computer Vision

BAB I PENDAHULUAN

* 1. Latar Belakang

Dalam beberapa tahun terakhir, deteksi objek telah menjadi signifikan bidang visi komputer. Tujuan deteksi objek adalah untuk mendeteksi dan mengklasifikasikan objek yang mengarah ke banyak bidang khusus dan aplikasi seperti deteksi wajah dan pengenalan wajah. Tujuannya bukan hanya kemampuan untuk melihat objek pada citra tetapi juga kemampuan untuk memahami dan menyimpulkan dari gambar itu terlihat. Kemampuan untuk mereplikasi visi di komputer adalah diperlukan untuk kemajuan teknologi sehari-hari. Deteksi objek mengatasi masalah ini dengan memprediksi lokasi objek melalui *bounding box* secara bersamaan serta mengklasifikasikan masing-masing objek dalam gambar yang diberikan [1], [2], [3]. Kemampuan sebuah *model* untuk melakukan deteksi objek secara *real-time* sangat diperlukan untuk mengakomodasi sistem *otonom car*. Algoritma deteksi objek yang efisien dan cepat adalah kunci keberhasilan *otonom car* [4], perangkat *augmented reality* [5], dan sistem kecerdasan intelektual lainnya.

Deteksi manusia pada citra pejalan kaki berdasarkan *shallow machine learning* dapat dibagi menjadi dua bagian kategori utama berdasarkan *global features* dan *human parts*. Deteksi manusia pada pejalan kaki berdasarkan *global features* memilih fitur yang dideskripsikan untuk diexpresikan pada karakteristik umum, seperti penampilan, tekstur citra dari pejalan kaki. Umumnya fitur deskripsi yang digunakan seperti Haar feature[6]. Deteksi citra pejalan kaki berdasarkan *global features* umumnya menjadi masalah yang sulit pada sebagian citra pelajan kaki di tempat kejadian yang sebenarnya. Sedangkan pada deteksi citra pejalan kaki berdasarkan *human parts* citra pada pejalan kaki dibagi kedalam empat beberapa komponen unit yang berbeda yaitu kepala dan bahu, kaki, lengan kiri, dan lengan kanan[7]

Teknologi deteksi citra pejalan kaki pada *deep learning* telah banyak menambah hasil akurasi deteksi dibandingkan dengan *shallow machine learning*  menggunakan *construction multi-layer neural network.* Diantara mereka, algoritma metode deteksi *convolutional neural network (CNN)* adalah yang paling banyak digunakan [8]. Ada banyak series dari algoritma deteksi objek *deep learning* yang berdasarkan CNN framework, diantaranya *region convolutional neural network (RCNN)*[8]. Fast-RCNN[9], Faster-RCNN berdasarkan *candidate region*[10], *You Only Look Once (YOLO)*[11], dan YOLOv2 yang memiliki kecepatan dan akurasi yang bagus dari metode sebelumnya.

Pada penelitian ini mengusulkan modifikasi metode algoritma YOLOv2 pada deteksi manusia pada citra pejalan kaki. Pertama, *account* dari citra pejalan kaki yang berbeda area dan aspek rasio di kluster untuk memaksimalkan proses kalibrasi data dan nilai *priori information* dari klaster diintegrasikan ke dalam proses prediksi. Kedua, hubungan *functional mapping* dari *output offset* kepada target scale ditetapkan dan di gabungkan dengan *loss function.* Ketiga, penyelarasan nilai jarak dari area dan aspek rasio ke kluster pejalan kaki, dan menyaring seluruh output *region proposal boxes.* Terakhir*,* duplikasi r*egion proposal boxes*  yang disaring akan dihapus oleh *non-maximal suppression.*

**1.2 Perumusan Masalah**

**1.2.1 Permasalahan (Research Problem)**

Terdapat berbagai bounding box yang berisi banyak region proposal namun tidak menunjukkan *class* pejalan kaki, serta duplikasi *region proposal* yang menyebabkan proses komputasi menjadi lama.

**1.2.2 Usulan Solusi : Metode**

Algoritma *You Only Look Once (YOLO)* v2 dengan *proposal box filtering.*

**1.2.3 Petanyaan Penelitian (Research Question)**

Bagaimana pengaruh *proposal box filtering* terhadap kinerja *You Only Look Once (YOLO)* v2 dalam deteksi citra pejalan kaki?

**1.3 Tujuan dan Manfaat Penelitian**

**1.3.1 Tujuan Penelitian**

Mengetahui pengaruh *proposal box filtering* terhadap kinerja *You Only Look Once (YOLO)* v2 dalam deteksi citra pejalan kaki.

**1.3.2 Manfaat Penelitian**

Manfaat dari hasil penelitian ini adalah diharapkan metode yang berhasil terpilih dapat diimplementasikan ke dalam sebuah sistem pengawasan visual.

**1.4 Batasan Masalah**

Dalam proposal ini terdapat beberapa batasan masalah diantaranya adalah :

1. Data yang digunakan dalam penelitian ini adalah set data KITTI dataset

2. Menggunakan framework TensorFlow.

**1.5 Sistematika Proposal**

BAB I PENDAHULUAN

Bab ini berisikan latar belakang masalah, rumusan masalah, tujuan dan

manfaat penelitian, batasan masalah dan sistematika penulisan laporan.

BAB II KAJIAN PUSTAKA

Pada bab ini dijelaskan mengenai materi-materi yang menjadi landasan

dalam penyusuan penelitian ini. Materi-materi tersebut diantaranya adalah

penelitan-penelitian terkait identifikasi atribut pejalan kaki, multi task learning

(MTL), convolution neural networks (CNN), Tensorflow.

BAB III METODE USULAN

Pada bab ini dijelaskan mengenai arsitektur sistem yang akan dibuat, set

data yang digunakan dan pengujian sistem.

BAB II KAJIAN PUSTAKA

**2.1 Convolutional Neural Network**

Convolutional Neural Network merupakan pengembangan dari Multi Layer Preceptron (MLP) yang didesain untuk mengelolah data dua dimensi, pada CNN, setiap neuoron dipresentasikan dalam bentuk dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena kedalaman jarigan yang tinggi dan banyak diaplikasikan pada suatu citra [12]. Seluruh Convolutional Neural Network menyatakan sebuah fungsi skor terdiferensiasi yang selanjutnya diikuti oleh fungsi Softmax [13]. Input data ke dalam Convolutional Neural Network adalah mengatur dalam bentuk lebar, tinggi dan kedalaman seperti yang ditunjukkan pada Gambar 2.1

CNN terdiri dari banyak lapisan, lapisan ini digunakan berulang kali, mengarah pada pembentukan Deep Neural Network. Jenis lapisan utama yang digunakan untuk membangun CNN adalah:

1. Input : Lapisan ini menyimpan nilai piksel mentah gambar.

2. Lapisan Konvolusi : Lapisan ini mendapatkan hasil lapisan neuron yang terhubung ke daerah input. tentukan jumlah filter yang akan digunakan dalam lapisan ini. Setiap filter merupakan jendela berukuran 5x5 yang dikalikan input data dan mendapatkan piksel dengan intensitas maksimum sebagai output.

3. Rectified Linear Unit (ReLU) : Lapisan ini menerapkan fungsi aktivasi yang digunakan untuk menghilangkan nilai piksel dibawah 0 pada data gambar. Di terapkannya fungsi ReLU untuk mempertahankan nilai yang sama dari piksel dan tidak diubah oleh propagasi.

4. Lapisan Pooling: Lapisan ini melakukan penurunan nilai citra hasil dari

konvolusi sesuai dengan filter yang digunakan.5. Lapisan Fully-Connected : Lapisan ini digunakan untuk menghitung kelas skor yaitu kelas yang memiliki skor maksimum sesuai dengan digit input.Arsitektur dasar Convolutional Neural Network dapat dilihat pada Gambar 2

**2.1.1 Lapisan Konvolusi**

Lapisan konvolusi dianggap sebagai blok bangunan inti dari asitektur CNN .Lapisan konvolusi akan menghitung produk baru diantara wilayah neuron didalam lapisan yang dimasukkan dan bobotnya tehubung secara local di lapisan output, output yang dihasilkan biasanya memiliki dimensi spasial yang sama. Konvolusi didefinisikan sebagai operasi matematika yang meggambarkan aturan untuk bagaimana menggabung dua set informasi. Lapisan konvolusi memiliki 4 parameter yang biasanya digunakan untuk mengatur output dari konvolusi, yaitu kernel/ filter, depth, stride dan zero-padding. Kernel adalah setiap kernel kecil spasial berhubungan dengan lebar dan tinggi ukuran kernel, depth adalah mengontrol jumlah neuron di lapisan konvolusi yang terhubung ke daerah yang sama dari volume input, stride adalah seberapa jauh melakukan pepindahan kernel, misalnya stride 1 berarti melakukan perpindahan kernel terhadap 1 piksel, Zero-padding adalah padding dengan nilai 0 yang ditambahkan untuk piksel yang menggelilingi citra input misalnya ukuran citra awal adalah 5x5, dengan menggunakan zero-padding 2, maka ukuran citra menjadi 7x7 [14]. Tujuan dilakukannya konvolusi pada citra adalah untuk mengekstrasi fitur dari data input citra. Konvolusi akan menghasilkan transformasi liniear dari data input sesuai

**2.1.2 Lapisan ReLU (Rectified Linear Unit)**

Pada Neural Network terdapat beberapa jenis fungsi aktivasi seperti tanh, Sigmoid, ReLU dll. Pada Convolutional Neural Network, fungsi aktivasi yang digunakan adalah Rectified Linear Unit (ReLU). Menurut [15] Menggunakan fungsi aktivasi Rectified Linear Unit (ReLU) bekerja 6 kali lebih cepat dibandingkan menggunakan aktivasi tanh. Secara matematis dapat didefinisikan sebagai berikut :

𝑓(𝑥) = 𝑚𝑎𝑥 (0, 𝑥) (2.1)

fungsi ReLU dapat dilihat pada Gambar 2.4.

Grafik pada Gambar 2.4 menunukkan bahwa fungsi ReLU bersifat nonlinier, fungsi ReLU mengambil nilai x, mengembalikan 0 jika x bernilai negatif dan mengembalikan x jika x bernilai positif. Penggunaan ReLU dapat diilustrasikan seperti Gambar 2.5.

**2.1.3 Lapisan Max-Pooling**

Lapisan Max-Pooling digunakan untuk mereduksi ukuran data citra output dari lapisan konvolusi. Max-Pooling bekerja dengan cara membagi output dari lapisan konvolusi menjadi beberapa grid kecil, kemudian mengambil nilai yang paling besar dari setiap grid untuk menyusun matriks citra yang telah direduksi. Gambar 2.6 menunjukkan contoh proses Max-Pooling yang didapatkan setelah proses konvolusi + ReLU dengan filter ukuan 2 x 2 dan stride 2 ( bergeser sebanyak 2 grid ), kemudian diambil nilai yang paling besar, sehingga dimensi rectified feature map berkurang [12]. Contoh dari Max-Pooling dapat dilihat pada Gambar 2.6.

**2.1.4 Lapisan Fully-Connected**

Lapisan Fully-Connected adalah Multi Layer Perceptron yang menggunakan fungsi softmax sebagai aktivasi neuron, Lapisan Fully-Conneted pada MLP memiliki tujuan untuk melakukan transformasi pada satu dimensi data agar dapat di klasifikasikan secara linear. Setiap neuron pada lapisan konvolusi perlu di ubah menjadi data satu dimensi sebelum dimasukkan pada lapisan Fully-Conneted, karena hal tesebut menyebabkan data kehilangan informasi spasialnya dan tidak reversible sehingga lapisan Fully-Conneted hanya dapat diimplementasikan di akhir jaringan

**2.2 YOLO**

Tidak seperti algoritma deteksi objek yang berdasarkan kandidat region, algoritma YOLO mengatu berdasarkan *global regression* yang dapat menyatukan kategori prediksi dan posisi pegresi di dalam satu *framework,* dan secara langsung mengatur *network* untuk mendirikan pemetaan hubungan antara hasil output dan citra input, membuat *network model* lebih tersaring dan memecahkan sebagian besar masalah kinerja yang buruk.

Matrik tiga dimensi mengatur kembali aturan dari YOLO untuk melakukan proses prediksi kategori dan lokasi. Implementasi yang spesifik ditampilkan pada Figure 2. Citra input akan dibagi menjadi kedalam SxS *grids,* dimana S adalah bilangan integer, dan setiap *grid*  akan bertugas untuk meprediksi B *bounding boxes.* YOLO menyelesaikan regresi setiap *bounding box*. Setiap bounding box memprediksi confidence value yang menunjukkan ada atau tidak adanya objek didalam bounding box. Jika data training dataset mengandung C tipe dari objek yang dideteksi, maka output dari demensi network berukuran SxS(Bx5+C)[13].

**2.2.1 Batch Normalization**

Batch normalization ditujukan untuk mengarah kepada peningkatan yang signifikan dalam nilai konvergensi. Dengan menambahkan batch normalization pada semua lapisan konvolusional di YOLO akan mendapatkan lebih dari 2% peningkatan dalam mAP. Bacth normalization juga membantu mengatur model. Dengan batch normalization, kita dapat menghapus dropout dari model tanpa overfitting[14].

2.2.2 High Resolution Classifier

Semua metode deteksi yang canggih menggunakan classifier yang sudah dilatih sebelumnya di ImageNet. Dimulai dengan AlexNet, yang sebagian besar pengklasifikasianya beroperasi pada citra input yang lebih kecil dari 256 × 256 [15]. Karena YOLO original melakukan trains classifier network pada 224 × 224 dan meningkatkan resolusi menjadi 448 untuk deteksi. Ini berarti network harus secara bersamaan beralih ke belajar mendeteksi objek dan menyesuaikan dengan resolusi input baru.

2.2.3 Convolutional with Anchor Box

YOLO memprediksi koordinat bounding box secara langsung menggunakan fully connected layer di ataskonvolusi feature extractor. Kami menghapus fully connected layer dari YOLO dan

menggunakan anchor box untuk memprediksi bounding box. Menghilangkan satu pooling layer untuk membuat output convolutional layer menjadi lebih tinggi. Dengan menggunakan Anchor box akan sedikit mengurangi nilai akurasi namun dapat memprediksi lebih dari seribu box per citra[16]

2.2.4 Dimension Cluster

Dalam Faster R-CNN network, area dan aspek rasion dipilih secara manual tanpa mengambil nilai prioro information kedalam account ketika menggunakan achor point untuk memprediksi candidate proposal boxes. Tapi YOLOv2 mengkleaster candidate proposal boxes ke dalam training dataset dengan overlap ratio antara candidate box dan calibration box sebagai measurement function, dan menggunkan pankang dan tinggi cluster tengah sebagai scale range dari candidate proposal box[13].

2.3. TensorFlow

Tensorflow merupakan suatu interface yang digunakan untuk mengimplementasikan algoritma machine learning dengan cara membuat operasi komputasi yang berinteraksi satu sama lain [13]. Interaksi dibentuk dalam bentuk grafik komputasi yang merepresentasikan arsitektur fungsional. Komputasi yang diekspresikan menggunakan TensorFlow dapat dieksekusi pada beragam sistem yang heterogen dengan sedikit atau bahkan tanpa perubahan mulai dari perangkat mobile sampai sistem terdistribusi berskala besar dari ratusan mesin dan ribuan perangkat komputasi seperti kartu GPU. Sistem ini fleksibel dan dapat digunakan untuk mengekspresikan berbagai macam algoritma, termasuk algoritma pelatihan dan prediksi untuk model deep neural network.

2.4. Penelitian Terkait

Dalam melakukan deteksi manusia, terdapat beberapa penelitian sebelumnya. Pada tahun 2014, Ross Girshick mengenalkan metode R-CNN dengan tiga model untuk deteksi obyek. Model pertama adalah region proposal yang menggunakan selective search dengan jumlah minimal 2000 region untuk menentukan set deteksi yang tersedia. Model ke-dua adalah CNN yang digunakan untuk melakukan ekstraksi fitur. Dan model ke-tiga adalah Support Vector Machine (SVM) yang digunakan untuk melakukan klasifikasi kelas dari masing masing obyek [2].

Pada tahun 2015, Yonglong Tian mengusulkan TA-CNN untuk melakukan klasifikasi pada fitur multi-task dan multiple data source pada deteksi pejalan kaki. Fitur multi-task pada penelitian terbagi menjadi dua, yaitu atribut pejalan kaki dan atribut lokasi pejalan kaki. Pada penelitian ini, didapatkan hasil dengan empat informasi yaitu, pedestrian classifier, pedestrian attributes, shared background attributes, dan unshared background attributes [5].

Selanjutnya pada tahun 2016, Joseph Redmon dkk melakukan penelitian pada deteksi obyek dengan menggunakan You Only Look Once (YOLO) dengan mengubah ukuran input image menjadi 448×448. YOLO memproses input image pada waktu sebenarnya dengan kecepatan 45 frames per detik. Penggunaan metode ini

menghasilkan tingkat akurasi sebesar 65.5% [16]

3.1. Dataset

Dataset yang digunakan pada penelitian ini adalah KITTI dataset yang dikumpulkan sebagai bagian dari penelitian tentang deteksi manusia dalam skenario mengendarai kendaraan tanpa pengemudi. KITTI dataset diperoleh dan dikumpulkan selama proses pengujian kendaraan tanpa pengemudi[13]. Dataset ini memiliki data training sebanyak 7.481 citra dan data testing dengan jumlah 7.518, dimana setiap citra bisa memuat 30 objek pejalan kaki.

Contoh dataset dapat dilihat pada Gambar 3.1

3.2 Arsitektur Sistem

Berikut Penjelasan setiap tahapan dari rancangan sistem pada gambar 3.2.

1. Set data dipisah menjadi dua yaitu data training dan data testing sesuai presentase yang telah ditentukan.
2. Dilakukan proses resize citra
3. Dilakukan training mengunakan algoritma YOLOv2 menggunakan arsitektur CNN yang telah dibangun untuk mendapatkan model.
4. Model yang telah didapatkan digunakan untuk testing.
5. Setalah dilakukan testing didapatkan hasil identifikasi atribut pejalan kaki.

[1] J. Schmidhuber, “Deep learning in neural networks: An overview,”

Neural networks, vol. 61, pp. 85–117, 2015. 1

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature

hierarchies for accurate object detection and semantic segmentation,”

in Proceedings of the IEEE conference on computer vision and pattern

recognition, 2014, pp. 580–587. 1

[3] C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” in Advances in neural information processing systems, 2013,pp. 2553–2561. 1

[4] L. Fridman, D. E. Brown, M. Glazer, W. Angell, S. Dodd, B. Jenik,

J. Terwilliger, J. Kindelsberger, L. Ding, S. Seaman et al., “Mit autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation,” arXiv

preprint arXiv:1711.06976, 2017. 1

[5] O. Akgul, H. I. Penekli, and Y. Genc, “Applying deep learning in

augmented reality tracking,” in Signal-Image Technology & Internet Based Systems (SITIS), 2016 12th International Conference on. IEEE, 2016, pp. 47–54. 1

1. Mohan A, Papageorgiou C, Poggio T. Example-based object detection in images by components. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2001, 23(4):349-361.
2. Felzenszwalb PF, Girschick R B, Mcallester D. cascade object detection with deformable part models. Communications of the Acm, 2010, 56(9):2241-2248.
3. Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Computer Science, 2013(10):580-587.
4. Girschick . Fast R-CNN. IEEE International conference on Computer Vision, 2015:1440.
5. Ren S, He K, Girshick R, at al. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Network. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015(1):1-6.
6. Redmon J, Divvala A, Girshick R, et al. You Only Look Once: Unified, Real-time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779-788.
7. P. W. S. Eka, A. Y. Wijaya and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network ( Cnn ) pada Caltech 101," Jurnal Teknik ITS, vol. 5, no. 1, 2016.
8. Ziwei Liu, Ying Shi, and Mingjun Sun,” A Pedestrian Detection Algorithm Based on Improve YOLO v2,” in Proceedings of CCIS 2018
9. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012
11. J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271, 2017.
12. I. L. Tom Hope, Yehezkel S. Resheff, Learning TensorFlow, vol. 521, no. 1. 2016