

Pemanfaatan *Deep Learning* pada Video *Dash Cam* untuk Deteksi Pengendara Sepeda Motor

Stephen Ekaputra Limantoro¹, Yosi Kristian², Devi Dwi Purwanto³

Abstract—The number of motorcyclists in Indonesia was 105.15 million in 2016. It made the Indonesian government difficult to monitor motorcyclists on the highways. Dash cam could be used as the alternative tool to detect motorcyclists when given the intelligence. One of the typical drawbacks in detecting objects is complex and varied feature. A convolutional neural networks (CNN) that was capable of detecting motorcyclists was proposed. CNN successfully classified the ship object with f1-score of 0.94. Sliding window and heat map were used in this paper to search the localization and region of motorcyclists. Two experiments had been done in this paper. The goal of this paper was to set the best combination of CNN architecture and parameter. The first experiment consisted of three trained weights while the second experiment consisted of one trained weight. Weight performances against test data in experiment 1 and experiment 2 were measured using f1-score of 0.977, 0.988, 0.989, and 0.986, respectively. From the experimental results using the sliding window, experiment 2 had a lower error rate to predict motorcyclists than experiment 1 because the training data on experiment 1 contained more and various images.

Intisari—Jumlah pengendara sepeda motor di Indonesia mencapai 105,15 juta di tahun 2016. Hal itu membuat pemerintah Indonesia mengalami kesulitan dalam memantau pengendara sepeda motor di jalan raya. *Dash cam* dapat digunakan sebagai alat alternatif untuk mendeteksi pengendara sepeda motor jika diberi kecerdasan. Salah satu kendala umum dalam mendeteksi objek adalah fitur yang kompleks dan variatif. Makalah ini menggunakan metode *convolutional neural networks* (CNN) yang mampu untuk mendeteksi pengendara sepeda motor. Metode CNN berhasil mengklasifikasikan objek kapal dengan *f1-score* 0,94. *Sliding window* dan *heat map* digunakan dalam makalah ini untuk mencari area/region pengendara sepeda motor. Dua eksperimen dilakukan dalam makalah ini. Tujuan dari makalah ini adalah mencari kombinasi arsitektur dan parameter CNN yang terbaik. Eksperimen pertama terdiri atas tiga bobot yang sudah dilatih sedangkan eksperimen kedua mengandung satu bobot yang sudah dilatih. Kinerja bobot terhadap data pengujian pada eksperimen 1 dan eksperimen 2 diukur melalui *f1-score* masing-masing 0,977, 0,988, 0,989, dan 0,986. Dari hasil percobaan menggunakan *sliding window*, eksperimen 2 memiliki tingkat *error* yang lebih rendah untuk memprediksi pengendara sepeda motor dibandingkan dengan eksperimen 1 karena data pelatihan pada

eksperimen 1 lebih banyak jumlahnya dan mengandung citra yang lebih variatif.

Kata Kunci—Deteksi, *Convolutional Neural Networks* (CNN), Sepeda Motor, *Deep Learning*, *Computer Vision*.

I. PENDAHULUAN

Teknologi informasi banyak digunakan manusia untuk mempermudah dan mempercepat pekerjaan manusia. Salah satu teknologi yang digunakan di dalam mobil adalah *dash cam*. *Dash cam* hanya digunakan untuk merekam video saja sehingga *dash cam* perlu diberi kecerdasan untuk mendeteksi objek. Deteksi kendaraan merupakan hal yang sangat penting untuk mengawasi lalu lintas [1], [2]. Jumlah pengendara sepeda motor di Indonesia meningkat 8,3% pada tahun 2015 dan mencapai 105,15 juta unit di tahun 2016 berdasarkan Badan Pusat Statistik Indonesia [3]. Banyaknya pengendara sepeda motor membuat pemerintah Indonesia mengalami kesulitan untuk memantau pengendara sepeda motor yang ada di jalan raya. Pemanfaatan makalah ini adalah untuk mendukung terbentuknya *self driving car*. Oleh sebab itu, teknologi deteksi pengendara sepeda motor diperlukan untuk membantu pemerintah Indonesia dalam pengawasan mobil menggunakan *dash cam* terhadap pengendara sepeda motor. Pada makalah ini, pengendara sepeda motor yang dimaksud adalah sepeda motor beserta pengendaranya.

Salah satu masalah utama dalam deteksi pengendara sepeda motor adalah proses klasifikasi yang melibatkan citra yang kompleks dan variatif, contohnya rumah, pejalan kaki, pohon, mobil, dan sebagainya. Semakin banyak jumlah objek dan variasinya membuat citra susah diprediksi [4], [5]. *Convolutional neural networks* (CNN) digunakan karena dapat menangani masalah yang kompleks tersebut dan memiliki kinerja yang baik dari hasil penelitian terkait [6]. *Deep learning* dibagi menjadi pembelajaran yang terawasi, pembelajaran yang tidak terawasi, dan *reinforcement learning* [7]. Proses klasifikasi pada makalah ini termasuk dalam pembelajaran yang terawasi, yaitu proses pembelajaran mesin yang memerlukan kelas/label [8].

CNN merupakan variasi dari jaringan saraf tiruan yang memiliki bobot dan beberapa lapisan tersembunyi yang disusun menjadi arsitektur [9]. Arsitektur CNN dibagi menjadi tiga bagian, yaitu lapisan konvolusional, fungsi aktivasi (Relu), dan lapisan *max pooling* [10]. Lapisan konvolusional berperan sebagai penghubung dengan lapisan setelahnya yang mengubah data masukan menjadi peta fitur yang sudah dikonvolusi dengan filter [11]. Fungsi aktivasi membedakan antar kelas dan melakukan transformasi nonlinier dari nilai masukan untuk menghasilkan keluaran. Lapisan *max pooling* mengurangi dimensi dari peta fitur dengan cara memilih angka piksel terbesar dari angka piksel tetangganya untuk

¹ Mahasiswa, Program Studi Teknologi Informasi, Sekolah Tinggi Teknik Surabaya, Jl. Ngagel Jaya Tengah 73-77, Surabaya, Jawa Timur, Indonesia 60284 (e-mail: stephenekaputra@gmail.com)

² Dosen, Program Studi Teknologi Informasi, Sekolah Tinggi Teknik Surabaya, Jl. Ngagel Jaya Tengah 73-77, Surabaya, Jawa Timur, Indonesia 60284 (e-mail: yosi@stts.edu)

³ Dosen, Program Studi Sistem Informasi, Sekolah Tinggi Teknik Surabaya, Jl. Ngagel Jaya Tengah 73-77, Surabaya, Jawa Timur, Indonesia 60284 (e-mail: devi@stts.edu)

dijadikan keluaran [12]. Selain itu, *dropout* perlu ditambahkan dalam arsitektur CNN untuk mencegah *overfitting* [6], [13].

Pada makalah ini, CNN digunakan untuk deteksi pengendara sepeda motor dari video *dash cam* yang beresolusi 1080 HD. *Dash cam* ditempelkan pada *dashboard* mobil dan merekam video di jalan raya Surabaya. Pengambilan citra pengendara sepeda motor dilakukan pada saat posisi pengendara sepeda motor tegak lurus ke depan dan pada saat cerah/siang hari. Makalah ini merupakan pengembangan dari penelitian sebelumnya [14].

Kontribusi dari makalah ini dapat dijabarkan sebagai berikut.

1. CNN, *sliding window*, dan *heat map* digunakan untuk mendeteksi pengendara sepeda motor pada video *dash cam*.
2. Dua eksperimen dilakukan dengan membandingkan dua jenis *data set* untuk mengetahui hasil kinerja detektor pengendara sepeda motor yang terbaik.
3. *Data set* diambil dari potongan video *dash cam* beresolusi 1080 HD yang berjumlah total 4.351 buah.

Makalah ini dibagi menjadi lima bagian. Bagian pertama berisi tentang pendahuluan. Pada bagian kedua, penelitian terkait dibahas. Bagian ketiga menjelaskan mengenai metodologi. Bagian keempat adalah hasil eksperimen dan pembahasan dari bagian ketiga. Kesimpulan dan saran dibahas di dalam bagian kelima.

II. KLASIFIKASI MENGGUNAKAN CNN

CNN banyak digunakan untuk klasifikasi karena tingkat kesuksesannya yang besar. Dalam sebuah penelitian, klasifikasi kapal pada citra TerraSar-X berhasil mencapai *f1-score* 0,94 [15]. Kinerja CNN akan bertambah baik jika masukan data mempunyai lebih banyak variasi. Proses pelatihan menggunakan ukuran *batch* 32 dan jumlah *epoch* menggunakan *early stopping* untuk mencari hasil yang terbaik.

Dalam sebuah artikel, tiga jenis arsitektur CNN dibandingkan, yaitu kecil, sedang, dan besar untuk mengklasifikasikan kerusakan rel kereta api [11]. Dari hasil perbandingan tersebut, arsitektur CNN besar memiliki kinerja yang lebih baik dibandingkan dengan arsitektur CNN kecil dan sedang dengan akurasi 92%. *Data set* yang digunakan berjumlah 22.408 dengan *colorspace grayscale* diambil dari rekaman kamera di atas rel kereta. Proses pelatihannya menggunakan parameter *learning rate* 0,01 dan *decay* 0,0001 serta *epoch* sebesar 40. Dalam eksperimennya, penggunaan fungsi aktivasi relu memiliki kinerja yang lebih baik dibandingkan penggunaan fungsi aktivasi tanh.

Hasil penelitian AlexNet memenangi kompetisi ILSVRC 2010 dan ILSVRC 2012 dengan tingkat *error* terendah 15,3% [6]. Sebanyak 1,2 juta data citra dengan 1.000 kelas dikoleksi dari situs web dan diberi label oleh manusia menggunakan *Amazon's Mechanical Turk crowd-sourcing tool*. Arsitektur CNN yang digunakan terdiri atas lima lapisan konvolusional, lima fungsi aktivasi relu, dua lapisan *max pooling*, dan tiga lapisan *fully-connected*. Proses pelatihan menggunakan *stochastic gradient descent* dengan parameter ukuran *batch* 128, *learning rate* 0,01, momentum 0,9, dan *decay* 0,0005.

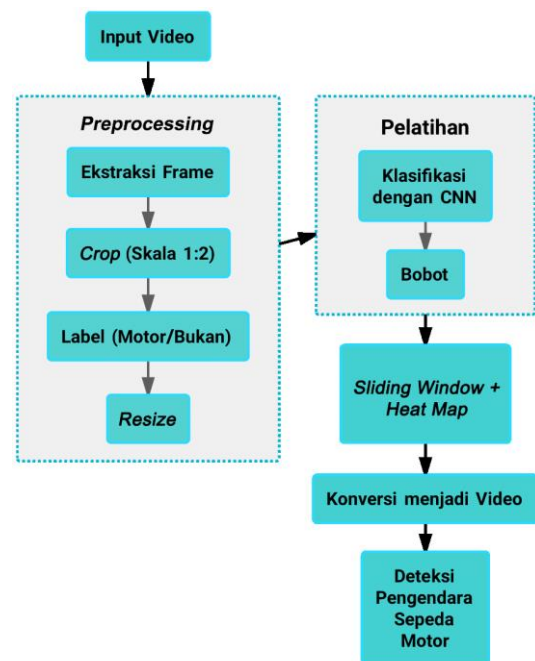
Proses deteksi objek memerlukan teknik lokalisasi untuk menemukan area objek. Salah satu cara yang umum

digunakan adalah *sliding window* [16]. *Sliding window* adalah *window* yang bergeser untuk memprediksi dan mencari area objek yang benar [17]. Dalam sebuah tesis, hasil pergeseran *sliding window* yang terbaik adalah 25% [9]. Namun, penggunaan *sliding window* akan memunculkan masalah lain ketika satu objek yang sama diprediksi lebih dari satu kali sehingga banyak kotak negatif yang muncul [17]. Dalam penelitian sebelumnya, *non-maximum suppression* (NMS) digunakan untuk mengurangi kotak negatif dengan nilai *threshold* tertentu [18], [19]. Selain NMS, *heat map* juga dapat digunakan untuk membuang kotak negatif dengan nilai *threshold* tertentu [20], [21].

Uraian di atas melatarbelakangi pembuatan aplikasi menggunakan metode CNN dan *sliding window* untuk deteksi pengendara sepeda motor.

III. METODOLOGI

Bahasa pemrograman yang digunakan dalam metode ini adalah Python untuk versi Windows. Proses pelatihan dilakukan menggunakan *library* Keras [22].



Gbr. 1 Metode deteksi pengendara sepeda motor menggunakan CNN.

Diagram blok metode deteksi pengendara sepeda motor diperlihatkan pada Gbr. 1. Metode deteksi pengendara sepeda motor dibagi menjadi tiga bagian utama, yaitu *preprocessing*, pelatihan, dan *sliding window*. Detail proses dari metode tersebut adalah sebagai berikut.

1. Video diekstrak menjadi banyak citra.
2. *Cropping* dilakukan terhadap bagian citra pengendara sepeda motor dan bukan pengendara sepeda motor dengan skala 1:2, lalu *colorspace* citra diubah menjadi RGB.
3. Potongan citra diberi label/kelas (pengendara sepeda motor dan bukan pengendara sepeda motor).
4. Hasil potongan citra yang sudah diberi label diubah ukurannya menjadi 200x400 piksel.

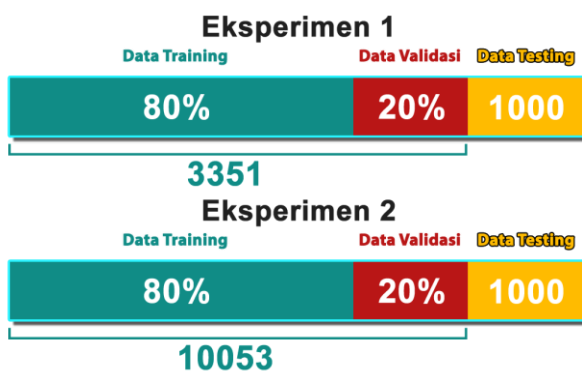
5. Data citra yang sudah terkumpul dilatih dengan CNN menghasilkan bobot. Proses pelatihan menggunakan CPU.
6. Bobot digunakan untuk mendeteksi pengendara sepeda motor dari video pengujian menggunakan *sliding window* dan *heat map*.
7. Semua citra hasil *sliding window* dan *heat map* dikonversi menjadi video dengan format *.mp4*.

A. Data set

Pada Tabel I, dua eksperimen dilakukan dalam pembagian *data set*. Pada eksperimen pertama, *data set* dengan ukuran 200x400 piksel disalin menjadi dua *data set* dan masing-masing dijadikan ukuran 100x200 piksel dan 150x300 piksel. Setiap *data set* dengan ukuran yang berbeda dilatih sehingga menghasilkan tiga bobot. Pada eksperimen kedua, *data set* dari eksperimen pertama disalin lalu dicampur menjadi satu *data set* dan semua data diubah ukurannya menjadi 200x400 piksel sehingga menghasilkan beberapa data yang *blur* (100x200 piksel dan 150x300 piksel menjadi 200x400 piksel) dan yang tidak *blur* (200x400 piksel menjadi 200x400 piksel). Eksperimen kedua dilatih menghasilkan satu bobot dan harus menggunakan memori RAM di atas 8 Gigabyte karena proses pelatihan berhenti ketika memori tidak cukup untuk melatih data dengan jumlah di atas 5.000.

TABEL I
PEMBAGIAN DATA SET

Eksperimen	Ukuran (piksel)	Jumlah Data		
		Pelatihan (80%)	Validasi (20%)	Pengujian
Eksperimen 1	100x200	2.680	671	1.000
	150x300	2.680	671	1.000
	200x400	2.680	671	1.000
Eksperimen 2	200x400	8.042	2.011	1.000



Gbr. 2 Pembagian *data set* deteksi pengendara sepeda motor terhadap eksperimen 1 dan eksperimen 2.

Pada Gbr. 2, *data set* terbagi menjadi tiga bagian, yaitu data pelatihan/*training*, data validasi, dan data pengujian/*testing*. Jumlah data pelatihan dan data validasi pada eksperimen pertama adalah 3.351 (1.517 data berlabel pengendara sepeda motor dan 1.834 data berlabel bukan pengendara sepeda motor). Di sisi lain, jumlah data pelatihan dan data validasi pada eksperimen kedua adalah 10.053 (4.551 data berlabel pengendara sepeda motor dan 5.502 data berlabel bukan pengendara sepeda motor). Untuk masing-masing eksperimen,

80% data digunakan sebagai data pelatihan dan 20% digunakan sebagai data validasi, serta 1.000 data untuk data pengujian. Data pengujian berisi 500 citra pengendara sepeda motor dan 500 citra bukan pengendara sepeda motor. Sampel data citra yang berlabel pengendara sepeda motor dan bukan pengendara sepeda motor ditunjukkan pada Gbr. 3. *Data set* dapat diunduh di situs web <http://www.datasets.stts.edu/imcd>.



Gbr. 3 Sampel data citra berlabel pengendara sepeda motor dan bukan pengendara sepeda motor.

CNN hanya dapat mengekstrak fitur yang ukurannya sama [23]. Oleh karena itu, *data set* eksperimen 1 dan eksperimen 2 memiliki banyak variasi ukuran dengan tujuan untuk mendeteksi pengendara sepeda motor pada ukuran yang bervariasi.

B. Parameter

Pada makalah ini, *Stochastic Gradient Descent* (SGD) digunakan untuk proses pelatihan yang bertujuan untuk meminimalkan *error* dan memperbarui bobot dengan nilai parameter *learning rate* sebesar 0,01, *decay* 5×10^{-4} , dan momentum 0,9. Parameter tersebut digunakan untuk mengurangi *error* pelatihan. Sebagian besar nilai parameter dan metode pelatihan mengikuti artikel AlexNet [6].

$$v_{i+1} := 0,9.v_i - 0.0005.e.w_i - e.\left\langle \frac{\partial L}{\partial w} \right\rangle D_i \quad (1)$$

$$w_{i+1} := w_i + v_{i+1}$$

Pada (1), *i* adalah indeks, *v* adalah momentum, *e* adalah *learning rate*, dan $\left\langle \frac{\partial L}{\partial w} \right\rangle D_i$ adalah turunan yang berfungsi untuk memperbarui bobot *w*.

C. Arsitektur CNN

Arsitektur CNN pada eksperimen 1 dan eksperimen 2 mengikuti arsitektur CNN-MR karena memiliki hasil yang baik dalam mengklasifikasikan objek [15]. Proses pelatihan menggunakan 40 *epoch* karena pada saat melakukan percobaan menggunakan 100 *epoch*, bobot dengan *f1-score* terhadap proses validasi terbaik berhenti di *epoch* ke-26. Ukuran *batch* yang digunakan untuk eksperimen 1 adalah 32, sedangkan ukuran *batch* pada eksperimen 2 adalah 64 supaya waktu pelatihan lebih singkat dan efisien.



Gbr. 4 Contoh arsitektur CNN terhadap data 200x400.

Pada Tabel II, fungsi aktivasi Relu ditambahkan untuk mempercepat proses belajar pada proses pelatihan [6]. *Dropout* juga ditambahkan dengan nilai 0,5 untuk mencegah *overfitting* [6], [13]. Arsitektur CNN terdiri atas empat lapisan konvolusional yang ditambahkan dengan relu dan empat lapisan *max pooling*. Dua lapisan *fully connected* ditambahkan dengan *sigmoid* dan *dropout* serta *softmax* di bagian akhir. Visualisasi arsitektur CNN terhadap data 200x400 ditunjukkan pada Gbr. 4.

TABEL II
ARSITEKTUR CNN UNTUK DETEKSI PENGENDARA SEPEDA MOTOR

Arsitektur CNN	Eksperimen 1 & Eksperimen 2
Lapisan konvolusional 1 + relu	@16 filter 5 x 5
Lapisan <i>max pooling</i> 1	2 x 2
Lapisan konvolusional 2 + relu	@32 filter 5 x 5
Lapisan <i>max pooling</i> 2	2 x 2
Lapisan konvolusional 3 + relu	@64 filter 6 x 6
Lapisan <i>max pooling</i> 3	2 x 2
Lapisan konvolusional 4 + relu	@128 filter 8 x 8
Lapisan <i>max pooling</i> 4	2 x 2
Lapisan <i>fully connected</i> 1	64 parameter
<i>Sigmoid</i>	-
<i>Dropout</i>	0,5
Lapisan <i>fully connected</i> 2	2 parameter
<i>Softmax</i>	-

D. Evaluasi Model

Dalam klasifikasi, bobot yang sudah dilatih perlu diuji kinerjanya. Kinerja bobot dapat diuji dengan cara menghitung *f1-score* [24]. *F1-score* merupakan alat ukur yang mengombinasikan nilai *precision* dan *recall*. *Precision* merupakan alat ukur untuk mengukur presisi atau ketepatan sasaran, sedangkan *recall* merupakan alat ukur yang mengukur jangkauan.

$$precision = \left(\frac{TP}{TP + FP} \right) \quad (2)$$

$$recall = \left(\frac{TP}{TP + FN} \right) \quad (3)$$

$$F1 - score = 2 \times \left(\frac{precision \times recall}{precision + recall} \right) \quad (4)$$

Penjelasan lebih detail terhadap (2) sampai (4) dalam deteksi pengendara sepeda motor ini adalah sebagai berikut.

- TP: jumlah pengendara sepeda motor yang diprediksi sebagai pengendara sepeda motor.
- FP: jumlah pengendara sepeda motor yang diprediksi sebagai bukan pengendara sepeda motor.
- FN: jumlah bukan pengendara sepeda motor yang diprediksi sebagai pengendara sepeda motor.

E. Sliding Window dan Heat Map

Pada eksperimen pertama, *sliding window* memprediksi objek menggunakan tiga bobot yang berukuran masing-masing 100x200 piksel, 150x300 piksel, dan 200x400 piksel. Jarak pergeseran *window* dibuat 25% dari ukuran lebar *window*, menjadi masing-masing 25, 38, dan 50 [9]. Nilai *threshold heat map* pada eksperimen ini bernilai 1 sehingga sebuah lokasi perlu diprediksi dua kali di tempat yang sama untuk menghasilkan *bounding box*. *Bounding box* tersebut merupakan lokasi final pengendara sepeda motor. *Sliding window* dengan ukuran 150x300 piksel diperlihatkan pada Gbr. 5.

Eksperimen kedua memiliki kesamaan dengan eksperimen pertama. Perbedaannya adalah bobot yang digunakan hanya satu dengan ukuran *window* 200x400 piksel seperti yang sudah dibahas pada Tabel I. Dengan satu bobot, objek diprediksi dengan *window* yang berukuran 100x200 piksel, 150x300 piksel, dan 200x400 piksel. Sebagai contoh, area objek dalam *window* yang ukuran 100x200 piksel dijadikan ukuran 200x400 piksel karena menyesuaikan dengan ukuran bobot 200x400 piksel pada saat prediksi. Namun, area objek yang dibesarkan akan menjadi *blur*. Oleh karena itu, data yang

blur (data dari ukuran 100x200 piksel dan 150x300 piksel menjadi 200x400 piksel) perlu dilatih agar data lebih variatif seperti yang sudah dibahas pada subbagian *data set*.

Pada ukuran *window* 200x400 piksel, *sliding window* bergeser dari tinggi 550 piksel sampai 680 piksel. Pada ukuran *window* 150x300 piksel, *sliding window* bergeser dari tinggi 650 piksel sampai 726 piksel. Pada ukuran *window* 100x200 piksel, *sliding window* bergeser dari tinggi 620 piksel sampai 695 piksel.



Gbr. 5 *Sliding window* dengan ukuran 150x300 piksel.

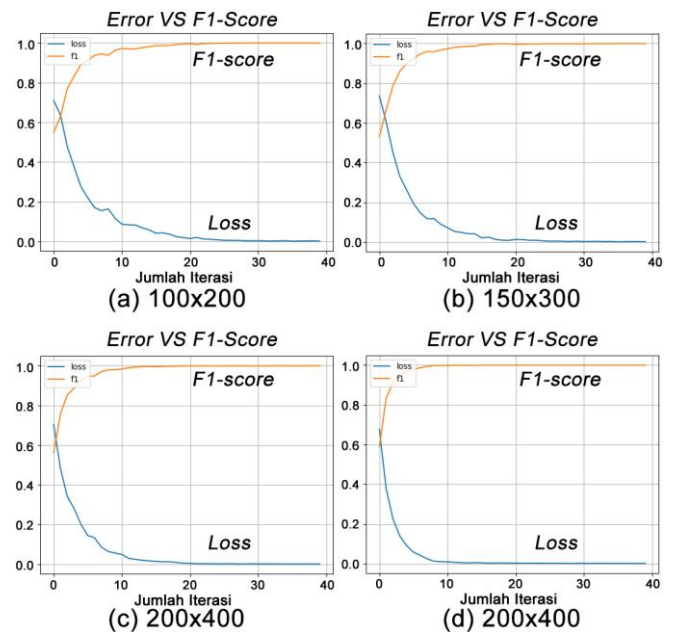
Pada Gbr. 6, salah satu batasan dalam makalah ini adalah pendeteksian pengendara sepeda motor bekerja ketika roda atau bagian dari pengendara sepeda motor menyentuh area berwarna hijau atau berada di area berwarna hijau. Area berwarna hijau terletak 200 piksel dari bawah *frame*.



Gbr. 6 Batasan deteksi pengendara sepeda motor.

IV. HASIL EKSPERIMEN DAN PEMBAHASAN

Hasil pelatihan terhadap eksperimen 1 dan eksperimen 2 ditampilkan melalui Gbr. 7. Pada masa pelatihan, bobot setiap *epoch* disimpan dan dipilih *f1-score* validasi yang tertinggi dan *error* yang terendah. Setelah itu, bobot yang sudah dipilih digunakan untuk memprediksi data pengujian. Pada Gbr. 7, penggunaan fungsi aktivasi relu mempercepat proses belajar karena kenaikan *f1-score* validasi dan penurunan *error* yang tajam di sebelum *epoch* ke-10 sesuai dengan penelitian sebelumnya [6]. Gbr. 7(d) menunjukkan kinerja yang lebih cepat dibandingkan dengan Gbr. 7(a), Gbr. 7(b), dan Gbr. 7(c) karena memiliki data pelatihan yang lebih banyak dan lebih variatif (mengandung data citra *blur*).



Gbr. 7 Grafik *error* dan *f1-score* terhadap ukuran, (a) 100x200 piksel pada eksperimen 1, (b) 150x300 piksel pada eksperimen 1, (c) 200x400 piksel pada eksperimen 1, (d) 200x400 piksel pada eksperimen 2.

Hasil kinerja bobot terhadap data validasi disajikan pada Tabel III. Tampak bahwa waktu pelatihan terlama dihasilkan oleh eksperimen 2 dengan waktu 51.422,9 detik karena waktu pelatihan dipengaruhi oleh besarnya ukuran data dan banyaknya jumlah data. Setelah mengetahui kinerja bobot terhadap data validasi, data pengujian diprediksi dengan bobot yang sudah ada. Pengujian data tersebut menghasilkan *F1-score* yang merupakan ukuran kinerja bobot untuk memprediksi data baru/pengujian. Rata-rata *F1-score* pengujian pada Tabel IV adalah 0,985. Nilai tersebut lebih baik dibandingkan dengan *F1-score* CNN-MR dengan *F1-score* 0,94 [15]. Kinerja bobot terhadap data pengujian ditunjukkan pada Tabel IV.

Satu video yang berdurasi 10 detik dengan kecepatan 10 fps digunakan untuk menguji coba bobot yang sudah terlatih menggunakan *sliding window* dan *heat map*. *Sliding window* bekerja dengan cara memprediksi daerah yang merupakan pengendara sepeda motor.

Proses deteksi pengendara sepeda motor terhadap eksperimen 1 dapat dilihat pada Gbr. 8. Pada eksperimen 1, total waktu yang dibutuhkan untuk menjalankan *sliding window* adalah 4.330 detik. Daerah yang terdeteksi sepeda motor disimpan dalam *folder* A. Dalam *folder* A, 805 citra terdeteksi sebagai pengendara sepeda motor. Pengecekan manual dilakukan dan ditemukan bahwa 55 dari 805 citra (6,83%) bukan merupakan citra pengendara sepeda motor.

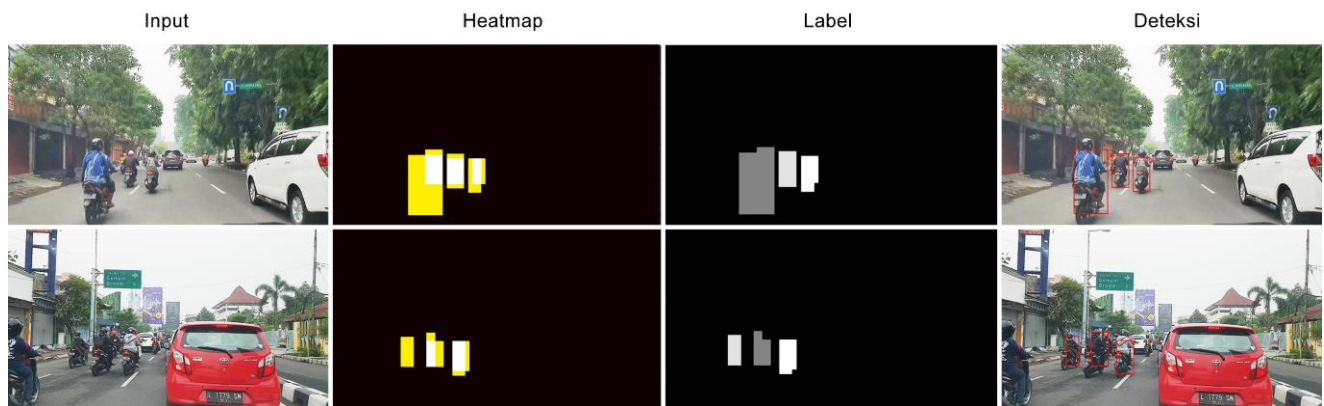
Proses deteksi pengendara sepeda motor terhadap eksperimen 2 dapat dilihat pada Gbr. 9. Pada eksperimen 2, *sliding window* bekerja selama 6.343 detik. Daerah yang terdeteksi sepeda motor disimpan dalam *folder* B. Dalam *folder* B, 724 citra terdeteksi sebagai pengendara sepeda motor. Setelah itu, pengecekan manual dilakukan dan ditemukan bahwa 31 dari 724 citra (4,28%) bukan merupakan citra pengendara sepeda motor.

TABEL III
KINERJA BOBOT TERHADAP DATA VALIDASI

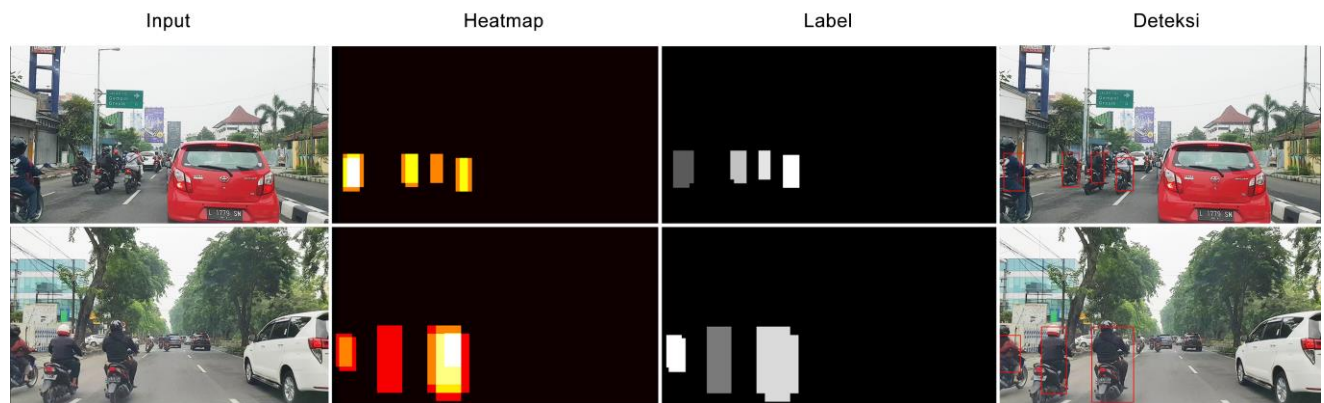
Eksperimen	Ukuran (piksel)	Bobot Terbaik (epoch)	Loss	F1-Score Validasi	Waktu Pelatihan (detik)
1	100x200	36	0,0017	1,0	4.288,16
	150x300	36	0,0015	1,0	9.603,86
	200x400	40	0,0010	1,0	16.667
2	200x400	29	0,0016	0,999	51.422,9

TABEL IV
KINERJA BOBOT TERHADAP DATA PENGUJIAN

Eksperimen	Ukuran (piksel)	TP	FP	FN	TN	Precision	Recall	F1-Score Pengujian
1	100x200	481	19	4	496	0,962	0,992	0,977
	150x300	491	9	3	497	0,982	0,994	0,988
	200x400	494	6	4	496	0,988	0,992	0,989
2	200x400	492	8	6	494	0,984	0,988	0,986



Gbr. 8 Proses deteksi pengendara sepeda motor pada eksperimen 1.



Gbr. 9 Proses deteksi pengendara sepeda motor pada eksperimen 2.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil eksperimen, deteksi pengendara sepeda motor dengan menggunakan CNN mampu menangani fitur yang kompleks dan variatif. Proses pelatihan sebagian besar menggunakan arsitektur dan parameter dari penelitian sebelumnya. Pelatihan bobot terhadap data validasi pada eksperimen 1 dan eksperimen 2 menghasilkan *f1-score*

masing-masing 1, 1, 1, dan 0,999. Waktu pelatihan bobot terhadap data dengan ukuran yang besar jauh lebih lama dibandingkan dengan ukuran data yang lebih kecil. Selain itu, fungsi aktivasi relu mampu mempercepat pelatihan dari *epoch* pertama sampai *epoch* ke-10. Kinerja bobot terhadap data pengujian pada eksperimen 1 dan eksperimen 2 menghasilkan *f1-score* masing-masing 0,977, 0,988, 0,989, dan 0,986.

Sliding window dan *heat map* mampu mendeteksi lokasi pengendara sepeda motor. Kecepatan *sliding window* pada

eksperimen 1 jauh lebih cepat dibandingkan dengan kecepatan *sliding window* pada eksperimen 2. Namun, eksperimen 2 memiliki tingkat *error* yang lebih rendah untuk memprediksi pengendara sepeda motor dibandingkan dengan eksperimen 1. Hal ini dikarenakan data pelatihan yang lebih banyak dan mengandung citra *blur* serta citra tidak *blur*.

B. Saran

Pengembangan terhadap metode deteksi pengendara sepeda motor ini dapat dilakukan dengan menambahkan jumlah sampel data yang lebih variatif (mengandung citra yang *blur* dan citra yang tidak *blur*) dan menggunakan *colorspace* selain RGB.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada seluruh dosen Program Studi Teknologi Informasi Sekolah Tinggi Teknik Surabaya, keluarga, dan penulis kedua serta ketiga yang selalu memberi motivasi dan dukungan selama penelitian berlangsung.

REFERENSI

- [1] A. Soin dan M. Chahande, "Moving Vehicle Detection Using Deep Neural Network," *2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT)*, 2017, hal. 1–5.
- [2] H. Tayara, K. G. Soo, dan K. T. Chong, "Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network," *IEEE Access*, Vol. 6, hal. 2220–2230, 2018.
- [3] (2017) "2016, Jumlah Sepeda Motor Indonesia Tembus Seratus Juta," [Online], <https://databoks.katadata.co.id/datapublish/2017/12/20/2016-jumlah-sepeda-motor-indonesia-tembus-100-juta>, tanggal akses: 12 Januari 2018.
- [4] D. Liu, Y. Xiong, K. Pulli, dan L. Shapiro, "Estimating Image Segmentation Difficulty," *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, 2011, Vol. 6871, hal. 484–495.
- [5] R. T. Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. P. Papadopoulos, dan V. Ferrari, "How hard can it be? Estimating the difficulty of visual search in an image," *Computer Vision and Pattern Recognition Conference (CVPR)*2, 2016, hal. 2157–2166.
- [6] A. Krizhevsky, I. Sutskever, dan G. E. Hinton, "Imagenet Classification With Deep Convolutional Neural Networks," *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 2012, Vol. 1, hal. 1097–1105.
- [7] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, Vol. 61, hal. 85–117, 2015.
- [8] Y. Lecun, Y. Bengio, dan G. Hinton, "Deep learning," *Nature*, Vol. 521, No. 7553, hal. 436–444, 2015.
- [9] D. Molin, "Pedestrian Detection Using Convolutional Neural Networks," Thesis, Linköping University, 2015.
- [10] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, Vol. 1, No. 4, hal. 541–551, 1989.
- [11] S. Faghih-Roohi, S. Hajizadeh, A. Nunez, R. Babuska, dan B. De Schutter, "Deep Convolutional Neural Networks for Detection of Rail Surface Defects," *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, hal. 2584–2589.
- [12] A. Giusti, D. C. Cire, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Fast Image Scanning with Deep Max-Pooling Convolutional Neural Networks," *International Conference on Image Processing (ICIP)*, 2013, hal. 4034–4038.
- [13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, dan R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, Vol. 15, hal. 1929–1958, 2014.
- [14] S. E. Limantoro, Y. Kristian, dan D. D. Purwanto, "Deteksi Pengendara Sepeda Motor Menggunakan Deep Convolutional Neural Networks," *Seminar Nasional Teknologi Informasi dan Komunikasi*, 2017, hal. 79–86.
- [15] C. Bentes dan D. Velotto, "Ship Classification in TerraSAR-X Images With Convolutional Neural Networks," *IEEE Journal of Oceanic Engineering*, Vol. 43, No. 1, hal. 258–266, 2018.
- [16] C. H. Lampert, M. B. Blaschko, dan T. Hofmann, "Beyond Sliding Windows : Object Localization by Efficient Subwindow Search," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, hal. 1–8.
- [17] H. Xie, Q. Wu, B. Chen, Y. Chen, dan S. Hong, "Vehicle Detection in Open Parks Using a Convolutional Neural Network," *2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA)*, 2015, hal. 927–930.
- [18] F. Zhu, Y. Lu, N. Ying, dan G. Giakos, "Fast Vehicle Detection Based on Evolving Convolutional Neural Network," *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, 2017, hal. 1–4.
- [19] P. Dollar, C. Wojek, B. Schiele, dan P. Perona, "Pedestrian Detection : A Benchmark," *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, 2009, hal. 304–311.
- [20] N. Hortovanyi, (2017) "Vehicle Detection and Tracking," [Online]. <https://towardsdatascience.com/vehicle-detection-and-tracking-6665d6e1089b>, tanggal akses: 10 Februari 2018.
- [21] J. Tompson, A. Jain, Y. Lecun, dan C. Bregler, "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation," *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, Vol. 1, hal. 1799–1807.
- [22] F. Chollet, (2015) "Keras," [Online], <https://github.com/keras-team/keras>, tanggal akses: 21 Oktober 2017.
- [23] X. Y. Chen, S. M. Xiang, C. L. Liu, dan C. H. Pan, "Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks," *IEEE Geoscience and Remote Sensing Letters*, Vol. 11, No. 10, hal. 1797–1801, 2014.
- [24] M. Sokolova dan G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," *Information Processing & Management*, Vol. 45, No. 4, hal. 427–437, 2009.