

Pendeteksian Lubang Pada Jalanan Menggunakan Metode *SSD-MobileNet*

Ivan Besando Pakpahan^{*1}, Ika Candra Dewi²

¹Prodi Elektronika dan Instrumentasi, DIKE, FMIPA UGM, Yogyakarta, Indonesia

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: ^{*1}ivanbesando@gmail.com, ²ika.candradewi@ugm.ac.id

Abstrak

Pesatnya kemajuan teknologi dan banyaknya lubang pada jalanan yang perlu diinspeksi membawa manusia untuk melakukan inspeksi lubang dengan menggunakan sistem pendeteksian lubang. Pemrosesan citra digital merupakan metode yang digunakan oleh beberapa orang dalam mendeteksi lubang dengan menggunakan warna sebagai fitur utama yang diekstrak, kemudian pendetektakan machine learning dan deep learning banyak dipelajari dan dikembangkan dalam hal pendeteksian, salah satunya ialah metode *ssd-mobilenet*. Pada penelitian ini digunakan tiga jenis dataset yang didapatkan secara sekunder dari berbagai sumber dan dinamakan dataset normal, dataset dashboard dan dataset closeup, kemudian ketiga dataset ini juga akan digabung dan divariasikan jumlah data latihnya dengan kenaikan variasi sebanyak 500 data sehingga didapatkan hasil model yang bervariasi. Hasil yang didapat dari penelitian ini merupakan bounding box pendeteksian dan juga skor confusion matrix masing-masing model dataset dimana dataset normal mendapatkan skor akurasi 56%, dataset dashboard mendapatkan akurasi 50% dan dataset closeup mendapatkan akurasi 76%.

Kata kunci—pemrosesan citra digital, deep learning, *ssd-mobilenet*, confusion matrix

Abstract

The rapid advancement of technology following the number of potholes on the streets that need to be inspected have led people to develop technology that can inspect pothole using a detection system. Digital image processing is a method used by some people to detect potholes by using its colour as the main extracted feature, after that the field of machine learning and deep learning approaches have been studied and developed in terms of detection, one of which is the *ssd-mobilenet*. In this study three types of dataset were used, they were obtained secondarily from various sources, namely the normal dataset, the dashboard dataset, and the closeup dataset. These three datasets will also be combined and varied in the amount of the training data with an increment of 500 data train so that various model results are obtained. The results obtained are the detection bounding boxes and also the confusion matrix score of each model dataset, where the normal dataset gets an accuracy score of 56%, the dashboard dataset gets 50% and the closeup dataset gets 76%.

Keywords—digital image processing, deep learning, *ssd-mobilenet*, confusion matrix

1. PENDAHULUAN

Jalanan berlubang merupakan masalah yang selalu dihadapi dan tidak akan pernah berakhir, banyaknya aktivitas pada jalanan seiring waktu akan membuat jalanan itu berlubang, hal ini berbahaya bagi pengguna jalan sehingga perlu dilakukan pengecekan berkala. Tahap paling

penting dalam perbaikan jalan ialah inspeksi, secara konvensional manusia akan datang ke lapangan dan melakukan inspeksi visual kemudian melakukan asesmen [1], namun perkembangan teknologi memungkinkan pendeteksian lubang secara otomatis untuk melakukan inspeksi.

Pemrosesan citra digital merupakan metode yang melakukan beberapa operasi pada gambar bertujuan untuk mengekstrak informasi-informasi berguna dari gambar tersebut [2] Pada penelitian yang menggunakan pemrosesan citra digital konvensional warna hitam pada citra lubang mengindikasikan area sekitar hitam itu merupakan objek lubang, daerah jalanan yang tidak berlubang akan kontras dengan daerah jalanan berlubang [2] Informasi warna ini menjadi fitur penting yang diekstrak ketika mengembangkan pendeteksian lubang jalanan, namun akurasi yang didapatkan kurang tinggi karena adanya keterbatasan jarak dan variasi sudut yang mempengaruhi sehingga para cendekiawan melakukan upaya pengembangan dengan metode *machine learning*.

Ranah *machine learning* merupakan ranah ilmu yang bertujuan mengajarkan komputer cara belajar dan bertindak tanpa perlu diprogram secara eksplisit oleh manusia [3] Cara ini membutuhkan fitur seperti halnya pemrosesan citra digital konvensional agar komputer dapat belajar dengan benar, namun beragamnya kondisi lubang seperti lubang kering atau lubang basah yang terisi air mengurangi akurasi pendeteksian hingga 20% dari performa maksimal. Oleh karena itu diperlukan suatu metode yang dapat mendeteksi lubang jalanan dengan bermacam-macam variasi tanpa mengurangi akurasi dari hasil pelatihan. Penelitian ini menggunakan teknik pelatihan *deep learning* dengan menggunakan model arsitektur *SSD-MobileNet*.

2. METODE PENELITIAN

2.1 Pengambilan data dan labelling

Pada penelitian ini, citra lubang didapatkan secara sekunder dari berbagai sumber yaitu *kaggle* yang kemudian akan disebut sebagai *dataset normal*, kemudian citra lubang yang didapat dari penelitian Nienaber *et al.* (2015) [2] yang kemudian akan disebut *dataset dashboard* dan citra lubang yang didapat dari penelitian Alzoubi (2018) [4] yang kemudian akan disebut *dataset closeup*. *Dataset normal* memiliki data sebanyak 874 gambar, *dataset dashboard* memiliki data sebanyak 662 gambar dan *dataset closeup* memiliki data sebanyak 1609 gambar. Total data adalah 3145 gambar yang dibagi menjadi pelatihan, validasi dan pengujian yang dapat dilihat pada Tabel 1.

Tabel 1. Persebaran data citra lubang

<i>Dataset</i>	Data Latih	Data Validasi	Data Uji	Total Data
normal	619	80	175	874
<i>dashboard</i>	477	53	132	662
<i>closeup</i>	1160	127	322	1609

Data citra yang telah didapatkan kemudian dilakukan anotasi dan *labelling* yang bertujuan untuk menandakan lubang pada citra dengan cara membuat *bounding box* pada tepi lubang secara manual dan memberikannya label [1] Informasi yang tersimpan pada proses anotasi dan *labelling* adalah *filename*, *width*, *height*, *class*, *x_min*, *y_min*, *x_max*, *y_max*.

2.2 Analisis sistem

Pada penelitian ini digunakan model arsitektur *SSD-MobileNet*, dimana model ini terdiri dari dua jenis jaringan yaitu jaringan *mobilenet* yang berfungsi sebagai ekstraksi ciri dan jaringan *SSD* sebagai *object detector*. *MobileNet* dan variannya diperkenalkan sebagai alternatif yang dioptimalkan terutama untuk hal kecepatan [7] daripada desain *deep learning* konvensional yang secara umum hanya menambahkan kedalaman jaringan [5][6]. Proses konvolusi yang dilakukan oleh jaringan *mobiletnet* menggunakan *depthwise separate convolutions* yang membagi *kernel* menjadi dua bagian yaitu *depthwise convolution* dan *pointwise convolution* [8]. Kedua *kernel*

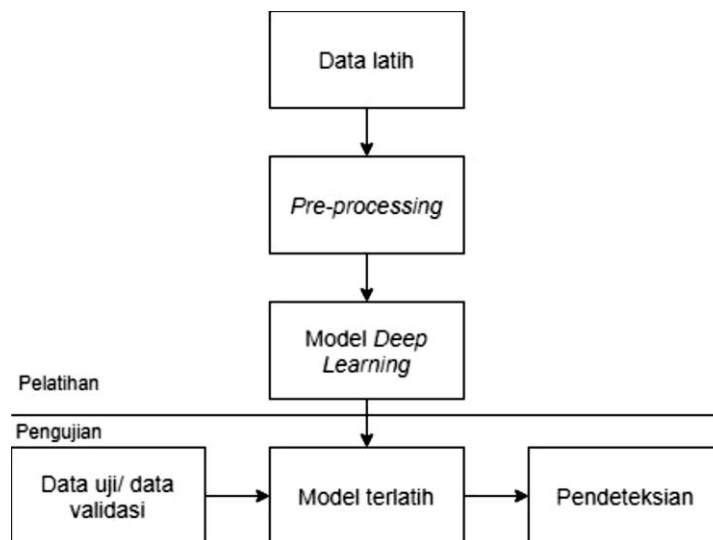
tersebut melakukan operasi konvolusi dengan *layer* sebelumnya dan membentuk *output* berupa *layer* baru. Pada proses ini, *input* *layer* melakukan operasi konvolusi dengan *kernel* 3x3x1 dan diiterasi sebanyak jumlah *channel input layer*. Hasil konvolusi tersebut mengeluarkan *output* berupa *layer* dengan *depth* yang sama dengan *input*-nya, *output* *layer* tersebut juga kemudian dijadikan *input* dan dikonvolusikan dengan *kernel* 1x1 yang memiliki *depth* yang sama, proses kedua ini yang disebut *pointwise convolution*. Penelitian ini menggunakan *depthwise separable convolutions* diikuti dengan normalisasi *batch* dan aktivasi ReLU pada tiap konvolusinya

Single shot detector (SSD) digunakan pada penelitian ini sebagai pendeteksi objek. SSD merupakan objek detector *one-stage* yang berarti model arsitektur ini melakukan klasifikasi dan lokalisasi pada jaringan yang sama (*single network*) [5], hal ini membuat proses SSD menjadi lebih sederhana dibandingkan detektor *two-stage*. *End-to-end learning* dan kesederhanaan model adalah salah satu keuntungan *one-stage detector* [9] Model arsitektur SSD memerlukan jaringan tambahan untuk mengekstrak fitur yang diperlukan, yang pada penelitian ini digunakan *mobilenet*, sehingga fitur-fitur hasil dari ekstraksi *mobilenet* akan digunakan sebagai bahan pelatihan SSD.

Model *SSD-Mobilenet* yang akan digunakan pada penelitian ini akan melakukan pelatihan dan pengujian pada tiga jenis *dataset* yaitu *dataset* normal, *dataset dashboard* dan *dataset closeup*, lalu kemudian juga akan melakukan pelatihan dan pengujian pada gabungan ketiga *dataset* tersebut yang data latihnya divariasikan agar didapatkan banyak hasil model yang variatif.

2.3 Rancangan sistem

Pada penelitian ini akan dibuat sistem pendeteksian lubang pada jalanan menggunakan model arsitektur *ssdmobilenet*. Secara umum, sistem untuk membuat pendeteksian lubang dapat dilihat pada Gambar 1.



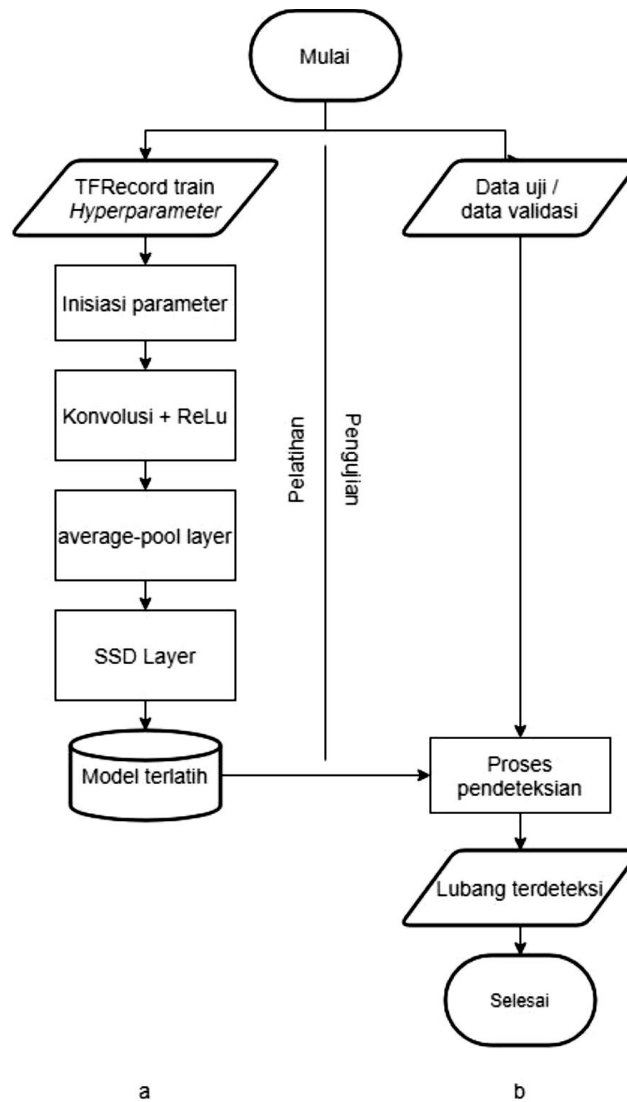
Gambar 1. Diagram blok perancangan sistem

Untuk mendapatkan model terbaik dilakukan pelatihan pada beberapa variasi, yaitu variasi model *dataset* dan variasi jumlah data latih pada *dataset* gabungan. Variasi model *dataset* terdiri dari pelatihan dan pengujian pada model *dataset* normal, *dataset dashboard* dan *dataset closeup*, kemudian variasi jumlah data latih pada *dataset* gabungan normal dan closeup dengan total data 2483 serta variasi jumlah data latih pada gabungan ketiga *dataset* dengan total data sejumlah 3145 data. Sebelum pelatihan dengan variasi dilakukan, dilakukan pencarian *hyperparameter* untuk kemudian dipakai sebagai parameter tetap setiap pelatihan model variasi. Variasi *hyperparameter* yang akan diuji dapat dilihat pada Tabel 2.

Tabel 2. Variasi *hyperparameter*

Variasi	Learning rate	Steps
1	0.00004	2000
		20000
		200000
2	0.0004	2000
		20000
		200000
3	0.004	2000
		20000
		200000

Variasi *hyperparameter* akan dilakukan pada dataset gabungan yang memiliki jumlah data latih sebanyak 2516 data dan data uji sebanyak 629 data yang kemudian hasil terbaik akan dipakai parameternya untuk melatih model data variasi. Diagram alir proses pelatihan dan pengujian ditunjukkan pada Gambar 2.



Gambar 2. Diagram alir proses (a) pelatihan dan (b) pengujian

2.4 Model evaluasi sistem

Proses pengujian sistem dilakukan untuk mengetahui performa kinerja sistem dalam mendeteksi lubang. Metode evaluasi yang dipakai untuk menguji sistem ialah *confusion matrix* yang akan membagi hasil model menjadi empat kategori yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Performa dari sistem dapat dihitung berdasarkan nilai keempat kemungkinan tersebut [10]. Menurut PASCAL VOC *metric*, tiap objek terdeteksi yang mendapat nilai IOU $\geq 50\%$ dianggap sebagai *true positive* [11]. Performa dari sistem ditunjukkan oleh beberapa parameter berikut:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$F1\text{-score} = 2 \times \frac{\text{presisi} \times \text{recall}}{\text{presisi} + \text{recall}} \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1 Hasil pengujian variasi hyperparameter

Pengujian variasi *hyperparameter* dilakukan untuk mengetahui variasi *hyperparameter* mana yang terbaik. Pengujian menggunakan ketiga dataset yang digabung dengan total data latih sebanyak 2516 data dan data uji sebanyak 629 data. Hasil dari pengujian variasi *hyperparameter* kemudian akan dipilih yang memiliki performa terbaik dan *hyperparameter*nya digunakan sebagai *hyperparameter* pelatihan dan pengujian variasi model *dataset* dan variasi model data latih. Tabel berikut menunjukkan hasil pengujian variasi *hyperparameter*.

Tabel 3. Hasil pengujian variasi *hyperparameter*

Steps	mAP		
	0.004 LR	0.0004 LR	0.00004 LR
2000	0.043	0.041	0.074
20000	0.53	0.67	0.56
200000	0.58	0.67	0.69

Berdasarkan Tabel 3 diketahui pengujian dengan variasi *learning rate* 0.00004 dan *steps* 200000 memiliki performa terbaik dibandingkan variasi lainnya, sehingga variasi ini lah yang akan dipakai pada pelatihan dan pengujian berikutnya.

3.2 Hasil pengujian pada dataset normal

Pengujian pada *dataset* normal dilakukan pada 619 data latih, 80 data validasi dan 175 data uji. *Dataset* ini memiliki gambar lubang yang diambil dari berbagai sudut pandang bervariasi sehingga citra lubang ada yang besar dan kecil sehingga cenderung *balanced*. Sebanyak 175 data diuji menggunakan data uji dan didapatkan skor yang ditunjukkan pada Tabel 4.

Tabel 4. Hasil pengujian *dataset* normal

Dataset	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)
Normal	56	89	60	72

Pada Tabel 4 menunjukkan bahwa *dataset* normal memiliki 56% akurasi, 89% presisi dan 60% *recall*. Contoh perbandingan pendeteksian dengan *ground truth* pada model *dataset normal* dapat dilihat pada tabel berikut

Tabel 5. Perbandingan pendeteksian dengan *ground truth* pada *dataset normal*



3.3. Hasil pengujian *dataset dashboard*

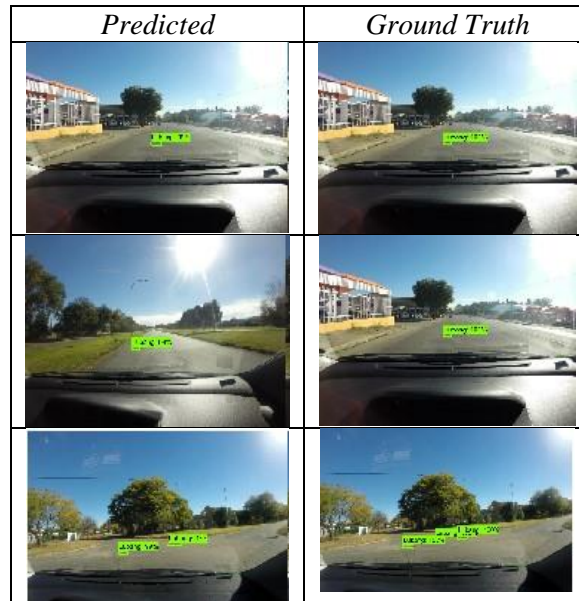
Pengujian pada model *dataset dashboard* dilakukan dengan menggunakan 477 data latih, 53 data validasi dan 132 data uji. Dataset ini memiliki gambar lubang yang diambil dari sudut pandang *dashboard* mobil sehingga citra lubang terlihat lebih kecil jika dibandingkan dengan *dataset* lainnya. Hasil pengujian didapatkan nilai TP 89, FP 9, FN 78 dan TN 0 dengan nilai skor seperti pada Tabel 6.

Tabel 6. Hasil pengujian *dataset dashboard*

<i>Dataset</i>	Akurasi (%)	Presisi (%)	<i>Recall</i> (%)	<i>F1-score</i> (%)
Normal	50	90	53	66

Hasil perbandingan pendeteksian dengan *ground truth* pada *dataset dashboard* dapat dilihat pada Tabel 7.

Tabel 7. Perbandingan pendeteksian dengan *ground truth* pada *dataset dashboard*



3.4 Hasil pengujian *dataset closeup*

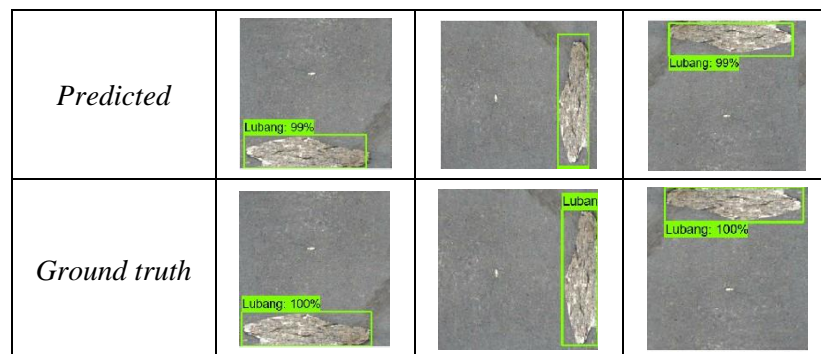
Pengujian pada model *dataset closeup* dilakukan dengan data latih sejumlah 1160 data, 127 data validasi dan 322 data uji. Dataset ini memiliki gambar lubang yang diambil dari jarak dekat sehingga citra lubang tampak besar. Hasil pengujian dilakukan dengan menggunakan evaluasi *confusion matrix* dan didapat nilai TN 279, FP 1, FN 83 dan TN 0. Performa skor metrik dapat dilihat pada Tabel 8.

Tabel 8. Hasil pengujian *dataset closeup*

<i>Dataset</i>	Akurasi (%)	Presisi (%)	<i>Recall</i> (%)	<i>F1-score</i> (%)
Normal	76	99	77	86

Hasil perbandingan pendeteksian dengan *ground truth* pada *dataset closeup* dapat dilihat pada Tabel 9.

Tabel 9. Perbandingan pendeteksian dengan *ground truth* pada *dataset closeup*



3.5 Hasil pengujian variasi jumlah data latih dataset gabungan *closeup* dan normal







Pada tahap ini dilakukan variasi jumlah data latih terhadap gabungan *dataset closeup* dan normal yang selanjutnya disebut *Dataset CN*. *Dataset* ini akan divariasikan jumlah data latihnya dan dilakukan pelatihan serta pengujian pada masing-masing variasinya. Variasi dimulai dari 500 data latih, kemudian 1000 data latih dan 1500 data latih, dan diuji dengan menggunakan data uji yang sama sebanyak 497 data. Performa skor metrik masing-masing variasi dapat dilihat pada Tabel 10.

Tabel 10. Perbandingan pengujian variasi *dataset CN*

Data latih	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)
500	44	95	47	61
1000	51	92	53	66
1500	57	94	58	71

Berdasarkan hasil pengujian secara keseluruhan variasi dengan data latih 1500 menunjukkan performa yang lebih baik dibandingkan dengan variasi data latih lainnya, walaupun pada skor presisi variasi data latih 500 memiliki skor lebih besar, namun skor *recall*nya jauh lebih kecil dibandingkan variasi data latih 1500. Contoh hasil pendeteksian dengan *ground truth* pada model CN variasi data latih 1500 dapat dilihat pada Tabel 11.

Tabel 11. Perbandingan pendeteksian dengan *ground truth* pada dataset CN variasi 1500

<i>Predicted</i>			
<i>Ground Truth</i>			

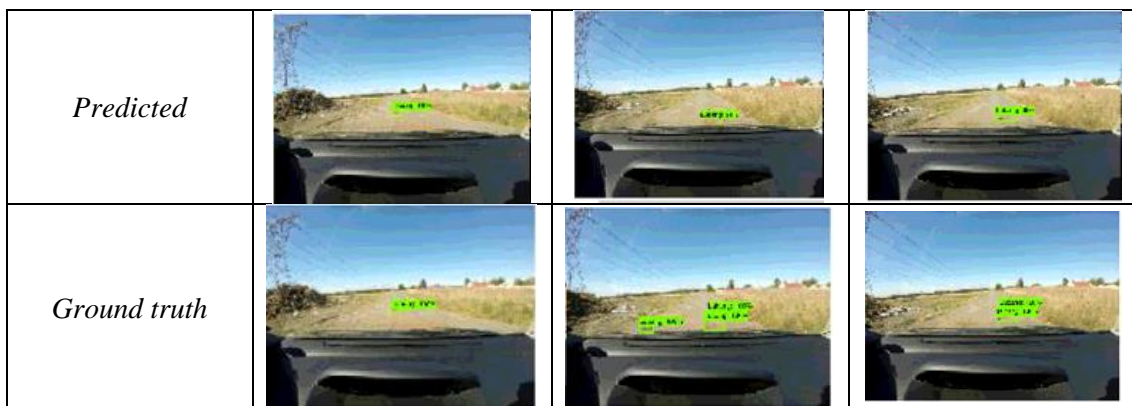
3.6 Hasil pengujian variasi jumlah data latih dataset *closeup*, normal dan *dashboard*

Pada tahap ini dilakukan pengujian pada variasi jumlah data latih *dataset* gabungan *closeup*, normal dan *dashboard* yang selanjutnya disebut *Dataset CND*. *Dataset* ini akan divariasikan jumlah data latihnya dan dilakukan pelatihan serta pengujian pada masing-masing variasinya. Variasi dimulai dari 500 data latih, kemudian 1000 data latih, 1500 data latih, 2000 data latih dan 2500 data latih dan diuji dengan menggunakan data uji yang sama sebanyak 629 data untuk semua model variasi. Performa skor metrik masing-masing variasi dapat dilihat pada Tabel 12.

Tabel 12. Perbandingan pengujian variasi *dataset* CND

Data latih	Akurasi (%)	Presisi (%)	<i>Recall</i> (%)	<i>F1-score</i> (%)
500	44	94	46	61
1000	50	93	51	65
1500	53	93	56	70
2000	55	96	57	72
2500	53	95	55	69

Berdasarkan hasil pengujian secara keseluruhan variasi dengan data latih 2000 menunjukkan performa yang lebih baik dibandingkan variasi lainnya, variasi data latih 2500 yang merupakan variasi data latih terbanyak justru kalah, hal ini dikarenakan terlalu sedikitnya jumlah data validasi yang digunakan arsitektur untuk melatih model. Contoh perbandingan prediksi dengan *ground truth* pada model CND variasi data latih 2000 dapat dilihat pada Tabel 13.

Tabel 13. Perbandingan pendeteksian dengan *ground truth* pada *dataset* CND variasi 2000

4. KESIMPULAN

Berdasarkan hasil penelitian ini dapat diambil kesimpulan bahwa penentuan *hyperparameter* berupa *learning rate* dan *steps* mempengaruhi hasil dari pelatihan arsitektur dalam membangun model pendeteksian lubang. Pada jenis model *dataset*, model data kategori *closeup* menunjukkan performa terbaik dalam mendeteksi lubang dibandingkan model kategori normal dan *dashboard* dengan nilai akurasi sebesar 76%. Pada *dataset* CN, variasi yang menunjukkan performa terbaik dalam penelitian ini berturut-turut yaitu variasi data latih 1500, data latih 100 dan data latih 500. Pada *dataset* CND, variasi yang menunjukkan performa terbaik dalam penelitian ini berturut-turut ialah data latih 2000, data latih 1500, data latih 2500. Data latih 1000 dan data latih 500. Pada penelitian ini *dataset dashboard* memiliki tingkat kesulitan yang lebih dibandingkan *dataset* lainnya, hal ini terlihat dari hasil pengujian dimana *dataset dashboard* memiliki skor terendah dibandingkan kategori *dataset* lainnya, hal ini turut menjadikan *dataset* gabungan CND mendapat performa lebih kecil dibandingkan *dataset* CN. Kesulitan arsitektur dalam melatih model *dataset dashboard* dikarenakan kecilnya objek yang dianotasi pada *ground truth*, sehingga fitur yang dianotasi tidak terekstrak dengan baik.

5. SARAN

Saran untuk mengembangkan penelitian ini antara lain mencoba variasi *hyperparameter* lain agar ditemukan model lebih baik kedepannya, mencoba model arsitektur selain SSD-MobileNet untuk kemudian dibandingkan hasilnya dan menambah pengujian berupa seberapa besar pengaruh intensitas cahaya pada pendeteksian.

DAFTAR PUSTAKA

- [1] Ernin N.U., Eko M.Y., Yoyon K.S., 2019, "Asphalt Pavement Pothole Detection Using Deep Learning method based on YOLO Neural Network", *IEEE*, hal. 35-40
- [2] Nienaber S., Booyesen M.J., Kroon R.S., 2015, "Detecting Potholes Using Simple Image Processing Techniques And Real-World Footage", *SATC*, Pretoria, South Africa
- [3] Karmel A., Adhithiyam M., Senthil K.P., "Machine Learning Based Approach For Pothole Detection", *IJCET*, hal. 882-888
- [4] Alzoubi, Alaa, 2018, "PotDataSet", *Cranfield Online Research Data (CORD)*, Dataset, <https://doi.org/10.17862/cranfield.rd.5999699.v1>, diakses tanggal 1 juni 2020
- [5] Arabi S., Haghighat A., Sharma A., 2019, "A deep learning based solution for construction equipment detection: from development to deployment", *Iowa State University*
- [6] He K., Zhang X., Ren S., Sun J., 2015, "Deep Residual Learning for Image Recognition", arXiv:1512.03385v1, diakses tanggal 10 juni 2020
- [7] Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H., 2017, "MobileNets: Efficient Convolution Neural Networks for Mobile Vision Applications", GoogleInc
- [8] Sifre L, 2014, "Rigid-Motion Scattering For Image Classification", *CMAP*
- [9] Liu W., Anguelov D., Erhan D., Szegedy C., 2015, "SSD: Single Shot Multibox Detector", arXiv:1512.02325v5, diakses tanggal 3 juni 2020
- [10] Visa S., Ramsay B., Ralescu A., Knaap E., 2011, "Confusion Matrix-based Feature Selection", <https://www.researchgate.net/publication/220833270>, diakses tanggal 10 juni 2020
- [11] Everingham M., Van Gool L., Williams C.K.I., Winn J., Zisserman A., "The PASCAL Visual Object Classes (VOC) Challenge", *International Journal of Computer Vision*.