



JEPIN

(Jurnal Edukasi dan Penelitian Informatika)

ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 7
No. 1
April 2021

Sistem Klasifikasi Jenis dan Warna Kendaraan Secara *Real-time* Menggunakan Metode *k-Nearest Neighbor* dan *Framework YOLACT*

Laurensius Adi Kurniawan^{#1}, I Putu Agung Bayupati^{#2}, Kadek Suar Wibawa^{#3}, I Made Sukarsa^{#4}, I Kadek Gunawan^{#5}

[#]Teknologi Informasi, Universitas Udayana

Jl. Raya Kampus Udayana, Jimbaran, Kuta Selatan, Badung, Bali 80361

¹ adikurniawan@student.unud.ac.id

² bayupati@unud.ac.id

³ suar_wibawa@unud.ac.id

⁴ sukarsa@unud.ac.id

⁵ gunawankadek@student.unud.ac.id

Abstrak— Peningkatan jumlah dan variasi jenis kendaraan terus berkembang seiring dengan meningkatnya permintaan pasar. Permasalahan baru timbul akibat meningkatnya jumlah dan variasi jenis kendaraan seperti meningkatnya pelanggaran lalu lintas dan kriminalitas. Dengan kondisi ini, pengawasan pelanggar lalu lintas dan kriminalitas secara manual oleh pihak berwajib akan lebih sulit dilakukan terutama di daerah perkotaan. Sistem pengenalan jenis dan warna kendaraan atau *Vehicle Color, Make and Model Recognition* (VCMR) adalah komponen penting dalam pengembangan sistem pengawasan keamanan di era otomatisasi. Dengan memanfaatkan CCTV, sistem ini dapat diaplikasikan pada sistem gerbang otomatis, pengawasan kendaraan otomatis, dan pemantauan lalu lintas. Sistem VCMR yang bekerja secara *real-time* dapat meningkatkan keamanan dengan menghasilkan data kendaraan lengkap berupa warna, merek dan model kendaraan selain menggunakan pengenalan plat nomor kendaraan. Penelitian ini menggunakan metode *k-Nearest Neighbor* (KNN) untuk mengklasifikasikan warna kendaraan dan *framework YOLACT* dengan arsitektur ResNet-50 yang telah dilatih untuk mengenali merek dan model kendaraan. *Dataset* dalam penelitian ini terdiri dari 10 jenis kendaraan dengan 40 citra data latih tiap kelas dan 10 warna dengan 25 citra data latih tiap kelas. Pengujian dilakukan menggunakan enam model YOLACT dengan *epoch* berbeda dan tiga variasi *frame sampling* untuk mengurangi waktu komputasi. Hasil pengujian pada video data uji empat kendaraan menunjukkan bahwa *frame sampling* 250 milidetik menghasilkan performa terbaik dengan waktu komputasi rata-rata 16,08 detik. Model YOLACT dengan *epoch* yang lebih besar mampu mengenali kendaraan yang berada jauh dari kamera (objek kecil) dengan akurasi yang diperoleh yaitu 91,67% pada *epoch* 254, 344, dan 517.

Kata kunci— KNN, *Framework YOLACT*, Pengenalan Jenis dan Warna Kendaraan, *Real-time*, ResNet-50

I. PENDAHULUAN

Penggunaan mobil berkembang pesat dewasa ini, terutama di kota-kota dengan aktivitas tinggi. Produsen mobil selalu melakukan inovasi jenis mobil baru untuk menyesuaikan segmen pasar dan memperbaharui tampilan kendaraan yang ada untuk menarik konsumen agar membeli dan mengikuti perkembangan zaman. Saat ini, perkiraan jumlah kendaraan secara global lebih dari 1,2 miliar dan berdasarkan penelitian, jumlah ini diperkirakan meningkat dua kali lipat menjadi 2,5 miliar pada tahun 2050 berdasarkan situs Green Car Reports.

Dikarenakan bertambahnya jumlah dan model kendaraan secara signifikan, menimbulkan beberapa permasalahan seperti meningkatnya pelanggaran lalu lintas, kriminalitas dan kemacetan. Dalam melakukan pengawasan pelanggar lalu lintas dan kriminalitas, pihak kepolisian akan sulit mengawasi kendaraan secara manual terutama di daerah perkotaan.

Teknologi maju dengan *Artificial Intelligence* (AI) dapat membantu menyelesaikan permasalahan yang ada dengan melakukan pengawasan terhadap kendaraan setiap saat secara otomatis dan *real-time*. Berbeda dengan manusia, teknologi AI dapat mengamati, mengenali dan mencatat kendaraan dalam waktu yang lama tanpa mengurangi tingkat ketelitian.

Salah satu teknologi AI yang dapat mengenali objek secara *real-time* adalah YOLACT. YOLACT merupakan salah satu *framework instance segmentation* yang dapat di aplikasikan pada sebuah citra atau video dengan kecepatan *real-time* yaitu 33,5 FPS dengan tidak mengurangi tingkat akurasi secara signifikan [1]. Pada penelitian sebelumnya *framework YOLACT* dimanfaatkan untuk melakukan *preprocessing* berupa segmentasi objek pada film untuk selanjutnya dilakukan

klasifikasi jenis bidikan (*shot type classification*) film [2]. Pada pengujian yang dilakukan dalam penelitian tersebut akurasi tertinggi dihasilkan saat menggunakan YOLACT untuk melakukan segmentasi objek.

Penelitian ini menggunakan *framework* YOLACT dengan arsitektur ResNet-50 untuk melakukan pengenalan jenis kendaraan dan Algoritma *k-Nearest Neighbor* (KNN) untuk melakukan pengenalan warna kendaraan. Model YOLACT dilatih untuk mengenali 10 jenis kendaraan yang beredar di Indonesia dan juga 10 warna kendaraan. Metode *frame sampling* digunakan dalam penelitian ini untuk mengurangi sumber daya komputasi yang dibutuhkan sehingga *delay* dapat diminimalkan pada pemrosesan video.

Penelitian ini dapat dimanfaatkan untuk membantu mengawasi lalu lintas secara otomatis, menambah informasi kendaraan dalam sistem e-tilang, meningkatkan keamanan dalam *auto gate system* dan dapat digunakan untuk melakukan riset lainnya seperti analisa tren kendaraan. Dalam penelitian ini skenario pengujian dilakukan seperti pada *gate system* yaitu melakukan pengawasan, pengenalan dan pencatatan jenis serta warna kendaraan secara otomatis dengan *input* video dari kamera pengawas. Petugas pengawas akan terbantu apabila ingin mencari data kendaraan pada suatu waktu tertentu karena data jenis dan warna kendaraan yang terdeteksi sudah tercatat.

II. TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Deteksi dan pengenalan kendaraan menjadi topik yang sedang giat dikembangkan beberapa tahun terakhir untuk mewujudkan *intelligent transportation systems* (ITS) dengan memanfaatkan *artificial intelligence* (AI). Pada umumnya klasifikasi kendaraan dapat dilakukan berdasarkan kelas kendaraan, pabrikan kendaraan, warna kendaraan atau jenis kendaraan. Sejumlah penelitian pendukung tentang pengenalan kendaraan diambil dari berbagai jurnal untuk mendukung penelitian ini.

Penelitian terkait pengenalan jenis kendaraan pernah dilakukan oleh Burak Satar dan Ahmet Emir Dirik [3] dengan judul "*Deep Learning Based Vehicle Make-Model Classification*" menggabungkan metode SSD (*Single Shot Multibox Detector*) dan CNN (*Convolutional Neural Network*). Penelitian berfokus pada pengenalan jenis kendaraan pada enam kelas kendaraan yang populer di Negara Turki. Akurasi paling baik didapatkan pada eksperimen II dengan akurasi sebesar 95.10%, eksperimen II melakukan *fine-tuning* pada model SSD untuk mendeteksi kendaraan dan melatih model ResNet untuk mengenali jenis kendaraan.

Penelitian dengan judul "*Transfer Learning-based Vehicle Classification*" oleh Jo et al. [4] menggunakan *Haar-like features* untuk melakukan deteksi kendaraan dan GoogleNet *transfer learning* untuk mengenali jenis kendaraan. Penelitian mengenali sepuluh jenis kendaraan menggunakan CNN tanpa *transfer learning* dan dengan

transfer learning. Akurasi yang didapatkan menggunakan *transfer learning* menghasilkan akurasi yang lebih baik yaitu 98,3% dan 65,7% tanpa menggunakan *transfer learning*.

Studi untuk mengenali warna kendaraan pernah dilakukan oleh Kim et al. [5] menggunakan CNN dengan judul "*Vehicle Color Recognition via Representative Color Region Extraction and Convolutional Neural Network*". Penelitian ini mengenali plat nomor dari kendaraan lalu menggunakan panjang dan lebar dari plat nomor sebagai informasi untuk melakukan ekstraksi bagian depan kendaraan. Beberapa piksel dipilih secara acak dari hasil ekstraksi lalu piksel yang terpilih digabungkan kembali menjadi sebuah citra yang digunakan untuk melakukan pengenalan warna. SGD (*Stochastic Gradient Descent*) *optimizer* digunakan untuk melakukan *training* pada tujuh kelas warna. Akurasi yang diperoleh sebesar 96,1% menggunakan 5.941 citra.

B. Framework YOLACT

Sejak *Instance Segmentation* banyak dikembangkan untuk meningkatkan akurasinya, muncul beberapa penelitian seperti Mask R-CNN oleh He et al. [6] dan FCIS oleh Li et al. [7] yang menggunakan metode *two-stage instance segmentation* yang memerlukan *re-pooling features* untuk tiap *Region of Interest* (ROI) dan memprosesnya menggunakan *subsequent computations* dengan fokus utama pada performa daripada kecepatan. YOLACT merupakan *real-time instance segmentation framework* dengan tujuan mengisi gap yang ada menggunakan metode *one-stage instance segmentation model* yang cepat tanpa mengurangi tingkat akurasi secara signifikan.

YOLACT mampu mencapai 29,8 mAP (*Mean Average Precision*) dengan kecepatan *real-time* yaitu 33,5 FPS dengan melakukan *training* pada *dataset* MS COCO menggunakan GPU Titan Xp sebanyak 800.000 iterasi. *Feature Pyramid Network* (FPN) [8] digunakan YOLACT karena lapisan fitur terbesarnya adalah yang paling dalam, kemudian *layer* tersebut ditingkatkan resolusinya hingga seperempat dimensi gambar *input* untuk meningkatkan kinerja pada objek kecil.

C. k-Nearest Neighbor

Algoritma *k-Nearest Neighbor* (KNN) adalah algoritma *supervised learning* yang mengklasifikasikan obyek baru berdasarkan mayoritas dari kategori *k*-tetangga terdekat pada data *training*. Prinsip kerja dari algoritma *k-Nearest Neighbor* yaitu mencari jarak terdekat antara data yang akan dievaluasi dengan *K* tetangga (*Neighbor*).

Algoritma *k-Nearest Neighbor* dapat dihitung dengan menghitung nilai *Euclidean distance* dari masing-masing tetangga. *Euclidean distance* dapat direpresentasikan seperti yang ditunjukkan pada persamaan di bawah ini:

$$D(q, p) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (1)$$

Dimana q adalah nilai dari data *training* dan p adalah nilai dari data uji atau bisa disebut *Euclidean vectors*. Setelah nilai *Euclidean distance* dari masing-masing tetangga didapatkan selanjutnya nilai diurutkan dari yang paling kecil, lalu mayoritas dari kategori k -tetangga terdekat dipilih sesuai nilai k yang ditentukan [9].

D. Color Feature Extraction

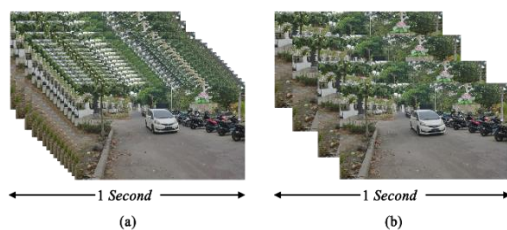
Ekstraksi fitur merupakan proses pengindeksan suatu basis data citra dengan isi dari citra, komponen fitur dapat digunakan untuk membandingkan citra satu dengan citra yang lainnya [10]. Data hasil ekstraksi fitur berupa sekumpulan nilai yang akan digunakan untuk melakukan pencocokan nilai antara citra latih dengan citra uji [11].

Warna adalah fitur penting dan paling mudah yang dirasakan manusia saat melihat gambar sehingga warna menjadi kandidat pertama yang digunakan untuk ekstraksi fitur [12]. *Color feature extraction* merupakan proses ekstraksi fitur warna pada citra berupa nilai warna tiap piksel. Fitur warna merupakan fitur paling umum yang ada pada citra. Citra berwarna memiliki standar ruang warna RGB (*Red, Green, Blue*). Fitur warna memiliki banyak keunggulan yaitu *robustness*, efektif, kemudahan implementasi, kesederhanaan komputasi, dan keperluan penyimpanan yang rendah [13].

Color histogram mewakili citra dari perspektif yang berbeda, merepresentasikan distribusi frekuensi dari nilai warna dalam suatu citra [14]. Histogram warna dapat digunakan untuk menampilkan intensitas kemunculan nilai warna tiap *channel* warna pada citra dan nilai warna puncak dengan intensitas kemunculan tertinggi.

E. Frame Sampling

Antar *frame* yang berdekatan dalam video berisi informasi yang sama dan waktu untuk mengekstraksi tiap *frame* merupakan penghambat dalam klasifikasi video maka perlu dilakukan *subsample frame* video atau *frame sampling* untuk mempercepat proses ekstraksi [15].



Gambar. 1 (a) Jumlah *frame* asli 30 FPS (b) *Frame sampling* setiap 250ms menghasilkan 4 FPS untuk diproses

Frame sampling dilakukan dengan memproses beberapa sampel *frame* tiap detiknya sesuai nilai *frame sampling* yang ditentukan, seperti diilustrasikan dalam Gambar. 1 (b) yang melakukan pengambilan sampel setiap 250ms sehingga hanya 4 *frame* yang diproses dalam 1 detik. *Frame* yang diproses akan lebih sedikit jika dibandingkan dengan tanpa menggunakan *frame sampling* sehingga waktu yang diperlukan untuk melakukan

komputasi lebih sedikit karena hanya beberapa *frame* diproses tiap detik.

III. METODOLOGI PENELITIAN

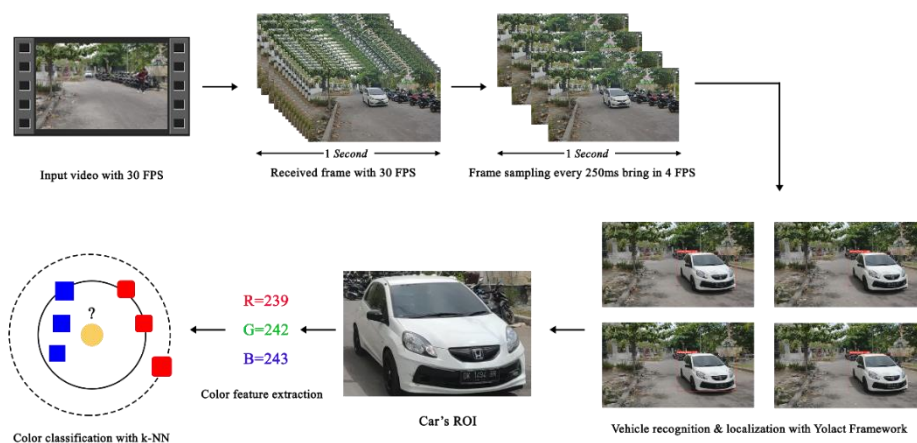
Automated Vehicle Color, Make and Model Recognition (VCMMR) system memainkan peran penting dalam sistem berbasis pengenalan kendaraan secara otomatis, VCMMR dapat dimanfaatkan untuk melengkapi data dari *Automated License Plate Recognition system (ALPR)* untuk lebih meningkatkan keamanan. Penelitian yang berhubungan dengan pengenalan kendaraan biasanya hanya berfokus pada satu pengenalan saja seperti berfokus pada pengenalan jenis, warna, ataupun tipe kendaraan. Penelitian ini menggabungkan pengenalan kendaraan berdasarkan warna dan jenis kendaraan secara bersamaan melalui *input* video.

Dataset jenis kendaraan yang terdiri dari 10 kelas dengan 40 citra tiap kelasnya bersumber dari berbagai *website* penjualan mobil bekas di Indonesia dan *dataset* warna yang terdiri dari 10 kelas dengan 25 citra tiap kelasnya bersumber dari *website* ColorHexa. Sebelum *dataset* jenis kendaraan digunakan untuk melakukan *training* model YOLACT, dilakukan proses *labelling* secara manual pada tiap citra dalam *dataset* jenis kendaraan dikarenakan YOLACT menggunakan format data COCO (*Common Objects in Context*) yang memiliki *file* anotasi dengan informasi dari semua gambar dalam *dataset* [16].

YOLACT menggunakan 3 *backbone* dalam penelitian yaitu Resnet-50, ResNet-101 dan DarkNet-53. ResNet-50 dengan ukuran *input* citra 550×550 lebih direkomendasikan apabila kecepatan proses lebih diutamakan, maka arsitektur ResNet-50 dipilih sebagai *backbone* dalam penelitian ini.

Proses *training* model YOLACT dilakukan dengan menggunakan 6 *epoch* yang berbeda pada *dataset* jenis kendaraan yang sudah melalui proses *labelling*. Proses *training* model pengenalan warna dilakukan dengan mengekstrak nilai warna dari tiap citra dalam *dataset* warna, nilai warna yang paling sering muncul dari tiap *channel* pada citra warna disimpan pada model pengenalan warna.

Pengenalan dan klasifikasi jenis dan warna kendaraan dilakukan pada *input* video dengan kecepatan 30 FPS. Seperti ditampilkan pada Gambar. 2, *frame* dengan kecepatan 30 FPS diterima sistem VCMMR, kemudian sistem melakukan proses *frame sampling* pada setiap *frame* yang diterima untuk meningkatkan kecepatan proses ekstraksi. *Frame* yang dipilih sebagai sampel lalu diproses menggunakan model YOLACT yang sudah dilatih untuk mengenali dan melokalisasi jenis kendaraan. Garis virtual digunakan sebagai titik untuk mencatat jenis kendaraan yang terdeteksi serta mengambil citra *Region of Interest (ROI)* kendaraan. Seperti ditampilkan pada Gambar. 3 garis virtual terdiri dari 3 garis dengan warna dan fungsi berbeda yaitu garis biru muda (*cyan*) untuk memulai *tracking* kendaraan agar kendaraan tidak terdeteksi ganda, garis merah untuk mencatat jenis



Gambar. 2 Gambaran umum sistem klasifikasi jenis dan warna kendaraan



Gambar. 3 Garis virtual

yang terdeteksi dan garis biru untuk mengakhiri *tracking* kendaraan.

Pengenalan warna dilakukan dengan melakukan *color feature extraction* pada ROI mobil yang terdeteksi, sehingga didapatkan nilai warna puncak dari setiap *channel* warna. Nilai warna puncak yang didapatkan lalu digunakan untuk melakukan klasifikasi warna menggunakan algoritma *k-Nearest Neighbor*.

IV. HASIL DAN PEMBAHASAN

Pengujian dilakukan pada empat jenis kendaraan dengan data uji sebanyak empat video dengan detail seperti ditampilkan pada Tabel 1. Data uji video direkam menggunakan kamera ponsel pada resolusi 720p, *frame rate* 30 FPS dengan format *file* mp4. Posisi kamera saat proses perekaman diskenariokan seperti posisi kamera keamanan pada *gate system*.

Pengujian menggunakan 6 model YOLACT yang dilatih dengan *epoch* berbeda. Seperti yang dapat dilihat pada Tabel 2, semakin besar *epoch* yang digunakan saat melakukan pelatihan menghasilkan nilai *box* dan *mask* mAP yang semakin tinggi pula.

Terdapat 3 nilai *frame sampling* yang digunakan yaitu, tanpa *frame sampling*, 250ms, dan 500ms. *Frame sampling* bekerja saat YOLACT tidak mendeteksi adanya mobil pada sampel *frame* yang diproses, apabila YOLACT mendeteksi mobil pada sampel *frame* maka *frame sampling* dihentikan dan sistem memproses seluruh

frame secara normal. Hasil uji coba diukur berdasarkan waktu komputasi dan akurasi dengan persyaratan jenis dan warna kendaraan berhasil dikenali. Semakin cepat waktu komputasi maka performa sistem semakin baik begitu pula sebaliknya, hal ini dikarenakan untuk mencapai pemrosesan video secara *real-time* diperlukan waktu komputasi yang cepat dengan akurasi yang baik.

TABEL I
DETAIL VIDEO DATA UJI

Data Uji	Jenis Kendaraan	Warna	Durasi
Video 1	Honda Brio	Putih	24 Detik
Video 2	Honda CRV	Putih	25 Detik
Video 3	Honda Jazz	Putih	8 Detik
Video 4	Suzuki Swift	Putih	25 Detik

TABEL II
MODEL YOLACT

Model YOLACT	Epoch	Box mAP	Mask mAP
Model 1	84	80,87	94,30
Model 2	169	90,85	98,47
Model 3	172	89,85	98,28
Model 4	254	93,76	98,56
Model 5	344	95,09	98,79
Model 6	517	97,00	99,06

Pengujian dilakukan sebanyak 18 kali pada tiap data uji video dengan mengkombinasikan 6 model YOLACT dan 3 nilai *frame sampling*. Hasil pengujian dikelompokkan berdasarkan model YOLACT dan berdasarkan *frame sampling* yang digunakan dengan menghitung rata-rata akurasi dan rata-rata waktu komputasi seperti ditampilkan pada Tabel 3.

Tabel 4 menampilkan rata-rata waktu komputasi tiap data uji video dikelompokkan berdasarkan *frame sampling*. Penggunaan *frame sampling* berkontribusi pada berkurangnya waktu komputasi yang diperlukan, *frame sampling* dengan nilai 500 ms menghasilkan waktu

komputasi tercepat hal ini dikarenakan sistem memproses *frame* lebih sedikit yaitu 2 *frame* tiap detiknya.

TABEL III
RATA-RATA HASIL PENGUJIAN

Model	Frame Sampling (ms)	Rata-rata	
		Waktu Komputasi (s)	Akurasi
Epoch 84	-	72,52	75 %
Epoch 84	250	15,97	0 %
Epoch 84	500	8,89	25 %
Epoch 169	-	71,66	100 %
Epoch 169	250	16,82	100 %
Epoch 169	500	8,79	50 %
Epoch 172	-	71,99	100 %
Epoch 172	250	15,92	100 %
Epoch 172	500	8,73	50 %
Epoch 254	-	72,79	100 %
Epoch 254	250	16,12	100 %
Epoch 254	500	8,89	75 %
Epoch 344	-	72,75	100 %
Epoch 344	250	16,09	100 %
Epoch 344	500	8,90	75 %
Epoch 517	-	73,44	100 %
Epoch 517	250	15,56	100 %
Epoch 517	500	8,93	75 %

TABEL IV
WAKTU KOMPUTASI BERDASARKAN FRAME SAMPLING

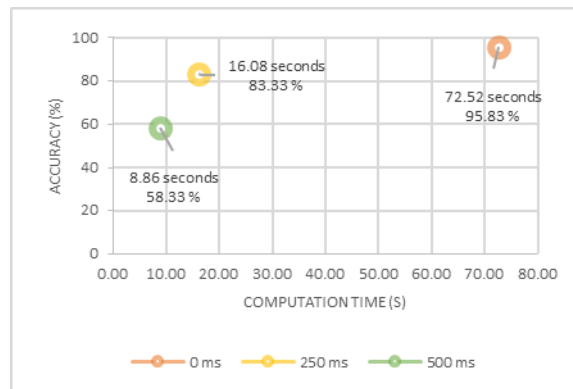
Frame Sampling	Waktu Komputasi (s)				
	Brio	CRV	Jazz	Swift	Rata-rata
-	87,46	86,68	28,70	87,26	72,52
250 ms	20,24	19,54	7,80	16,73	16,08
500 ms	12,10	11,86	3,56	7,90	8,86

Walaupun *frame sampling* dengan nilai 500 ms menghasilkan waktu komputasi tercepat tidak menandakan *frame sampling* tersebut merupakan *frame sampling* terbaik. Hal ini dikarenakan akurasi yang dihasilkan pada *frame sampling* dengan nilai 500 ms menghasilkan akurasi paling rendah seperti dapat dilihat pada Gambar. 4.

Seperti ditampilkan pada Gambar. 5(a), *frame sampling* dengan nilai 500 ms memungkinkan ROI dari kendaraan tidak melalui garis virtual merah yang bertugas mencatat jenis dan warna kendaraan karena perpindahan objek pada sampel *frame* yang diproses terlalu jauh.

Akurasi tertinggi dihasilkan saat tidak menggunakan *frame sampling*. Hal ini dikarenakan sistem memproses seluruh *frame* yang diterima maka ROI kendaraan melewati garis virtual dan berpengaruh pada meningkatnya waktu komputasi yang diperlukan. *Frame sampling* 250ms menghasilkan akurasi dan waktu

komputasi paling optimal diantara 2 nilai *frame sampling* lainnya.



Gambar. 4 Performa *frame sampling*



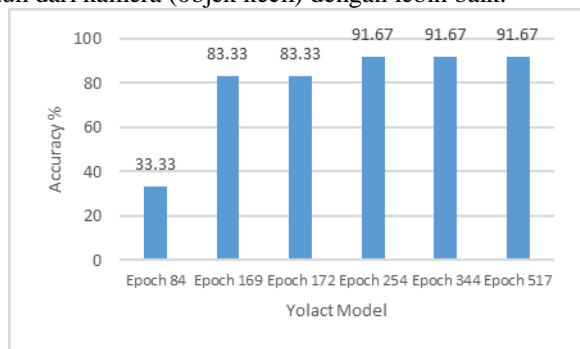
Gambar. 5 (a) Video diproses menggunakan *frame sampling* 500ms (b) Video diproses menggunakan *frame sampling* 250ms

Gambar. 5(a) menggambarkan kondisi saat jenis dan warna kendaraan tidak tercatat dalam sistem dikarenakan ROI mobil tidak melalui garis catat virtual (garis merah) melainkan melompati garis. Hal ini terjadi karena *frame* yang diproses dalam 1 detik terlalu sedikit sehingga perpindahan mobil dalam *frame* yang diproses terlalu jauh. Pada Gambar. 5(b) menggambarkan kondisi saat mobil tercatat dalam sistem, ROI mobil melalui garis catat virtual karena tidak banyak *frame* yang dilewatkan tiap detik sehingga perpindahan mobil dalam *frame* yang diproses tidak terlalu jauh.

Pencatatan jenis kendaraan dan pengenalan warna kendaraan dilakukan saat ROI mobil melalui garis catat virtual, maka akurasi pengenalan warna kendaraan menjadi satu kesatuan dengan akurasi pengenalan jenis kendaraan. Berdasarkan pengujian yang dilakukan pada 4 video data uji, sistem dapat mengenali warna kendaraan pada 4 video data uji dengan benar. Akurasi pengenalan warna kendaraan juga dipengaruhi oleh tingkat kecerahan lingkungan dan distraksi lingkungan seperti cahaya terhalang pepohonan maupun kendaraan dimodifikasi sehingga kendaraan memiliki warna gradasi.

Gambar. 6 menampilkan model YOLACT dengan epoch 254, 344, dan 517 menghasilkan akurasi paling baik pada pengujian yang dilakukan menggunakan 4 video kendaraan. Berdasarkan hasil pengamatan, semakin besar epoch yang digunakan dalam proses *training* model

YOLOACT maka sistem mengenali kendaraan yang berada jauh dari kamera (objek kecil) dengan lebih baik.



Gambar. 6 Grafik akurasi YOLOACT

V. KESIMPULAN

Vehicle Color, Make and Model Recognition (VCMMR) mengambil peran penting dalam bidang keamanan di era otomatisasi. Sistem VCMMR dapat menghasilkan data kendaraan berupa warna, merek dan model kendaraan yang dapat digunakan untuk melengkapi data dari sistem *Automatic License Plate Recognition (ALPR)* yang digunakan dalam sistem pengawasan kendaraan otomatis. Penelitian ini menghadirkan pengenalan jenis dan warna kendaraan pada *input* video menggunakan *framework* YOLOACT dengan arsitektur ResNet-50 dan metode KNN. Model dilatih untuk mengklasifikasikan 10 jenis kendaraan dengan data latih sebanyak 40 citra tiap kelas dan 10 warna dengan 25 citra data latih tiap kelas. Metode *frame sampling* digunakan untuk mengurangi waktu komputasi yang diperlukan saat pengujian. Pengujian dilakukan menggunakan enam model YOLOACT dengan *epoch* berbeda dan tiga variasi *frame sampling*. Berdasarkan hasil pengujian pada empat video data uji, penggunaan *frame sampling* dengan nilai 250 ms mampu mengurangi waktu komputasi lebih optimal dengan waktu komputasi rata-rata 16,08 detik. *Frame sampling* dengan nilai 500 ms tidak menjadi *frame sampling* terbaik walaupun menghasilkan waktu komputasi tercepat karena akurasi yang dihasilkan lebih rendah dari *frame sampling* dengan nilai 250 ms. Model YOLOACT dengan *epoch* lebih besar mampu mengenali kendaraan yang berada jauh dari kamera (objek kecil) dengan akurasi yang diperoleh yaitu 91,67% pada *epoch* 254, 344, dan 517. Berdasarkan pengamatan yang dilakukan saat pengujian, model *epoch* 517 mampu mengenali kendaraan yang berada jauh dari kamera dengan lebih baik daripada model lain yang diuji.

UCAPAN TERIMA KASIH / ACKNOWLEDGMENT

Studi ini didukung oleh Prodi Teknologi Informasi, Fakultas Teknik, Universitas Udayana, Bali, Indonesia.

Terima kasih kepada Bolya dkk. yang telah bekerja keras dalam membangun *framework* YOLOACT.

REFERENSI

- [1] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLOACT: {Real-time} Instance Segmentation," *ICCV*, p. 11, 2019.
- [2] H. Y. Bak and S. B. Park, "Comparative study of movie shot classification based on semantic segmentation," *Appl. Sci.*, vol. 10, no. 10, pp. 1–10, 2020.
- [3] B. Satar and A. E. Dirik, "Deep learning based vehicle make-model classification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11141 LNCS, pp. 544–553, 2018.
- [4] S. Y. Jo, N. Ahn, Y. Lee, and S. Kang, "Transfer Learning-based Vehicle Classification," *2018 Int. SoC Des. Conf.*, pp. 127–128, 2018.
- [5] K. J. Kim *et al.*, "Vehicle Color Recognition via Representative Color Region Extraction and Convolutional Neural Network," *Int. Conf. Ubiquitous Futur. Networks, ICONF*, vol. 2018-July, pp. 89–94, 2018.
- [6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Mar. 2017.
- [7] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 4438–4446, 2017.
- [8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 2019-Novem, pp. 1702–1707, Dec. 2016.
- [9] A. Kataria and M. D. Singh, "A Review of Data Classification Using K-Nearest Neighbour Algorithm," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 6, pp. 354–360, 2013.
- [10] N. L. W. S. R. Ginantra, "Deteksi Batik Parang Menggunakan Fitur Co-Occurrence Matrix Dan Geometric Moment Invariant Dengan Klasifikasi KNN," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 7, no. 1, p. 40, 2016.
- [11] I. W. A. Suryawibawa, I. K. G. Darma Putra, and N. K. A. Wirdiani, "Herbs Recognition Based on Android using OpenCV," *Int. J. Image, Graph. Signal Process.*, vol. 7, no. 2, pp. 1–7, 2015.
- [12] D. Zhang, "Color Feature Extraction," in *Perspectives on Content-Based Multimedia Systems*, Boston: Kluwer Academic Publishers, 2002, pp. 49–67.
- [13] R. Revathi and M. Hemalatha, "Efficient method for feature extraction on video preprocessing," *ACM Int. Conf. Proceeding Ser.*, pp. 539–543, 2012.
- [14] D. Srivastava, R. Wadhvani, and M. Gyanchandani, "A Review: Color Feature Extraction Methods for Content Based Image Retrieval," *Int. J. Comput. Eng. Manag.*, vol. 18, no. 3, pp. 9–13, 2015.
- [15] J. R. R. Uijlings, I. C. Duta, N. Rostamzadeh, and N. Sebe, "Realtime video classification using dense HOF/HOG," *ICMR 2014 - Proc. ACM Int. Conf. Multimed. Retr. 2014*, pp. 145–152, 2014.
- [16] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014.