PENGEMBANGAN APLIKASI DETEKSI OBJEK UNTUK PENYANDANG DISABILITAS TUNANETRA MENGGUNAKAN *YOLO*

1st Bella Cantika Dewi Nurfatikhah
Teknik Informatika
Universitas Telkom Purwokerto
Indonesia
cntkabella@student.telkomuniversity.ac.id

2nd Aditya Dwi Putro W, S.Kom., M.Kom
Teknik Informatika
Universitas Telkom Purwokerto
Indonesia
adityaw@telkomuniversity.ac.id

3rd Atika Ratna Dewi, S.Si., M.Sc
Teknik Informatika
Universitas Telkom Purwokerto
Indonesia
atikad@telkomuniversity.ac.id

Abstrak* — Berdasarkan dengan data menurut Kementrian Kesehatan RI, hampir mencapai 1,5% penduduk di Indonesia telah mengalami disabilitas tunanetra yang dimana tunanetra itu sendiri adalah suatu kondisi fisik yang terbatas karena mengalami kekurangan dalam penglihatannya. Sehingga, para tunanetra pada umumnya menggunakan tongkat untuk alat bantu dalam kesehariannya. Dengan perkembangan era teknologi yang canggih dari tahun ke tahun, banyak program aplikasi yang dibuat khusus untuk membantu para penyandang disabilitas tunanetra seperti contoh aplikasi alat untuk membantu tunanetra berjalan dengan metode yang berbeda beda untuk membantu para tunanetra. Diangkat dari permasalahan tersebut, penulis bertujuan ingin membuat suatu program aplikasi yang dapat digunakan untuk membantu para disabilitas tunanetra untuk melakukan aktifitas hariannya. Dimana program aplikasi tersebut dapat mendeteksi suatu objek dan halangan yang berada di depan kamera menggunakan metode YOLO, dan dapat mengenali wajah seseorang yang sudah melewati proses training menggunakan CNN. Serta output aplikasi yang berupa suara. Hasil akurasi yan diharapkan oleh penulis pada penerapan prediksi detection ini yaitu sebesar >=75%.

Kata kunci*— *Convolutional Neural Network, GTTS, Tunanetra, Confution matrix, YOLO

I. PENDAHULUAN

Berdasarkan data oleh Kementrian Kesehatan RI pada tahun 2017, penyandang disabilitas tunanetra di Indonesia mencapai 1,5% dari seluruh penduduk Indonesa, ini bukanlah jumlah yang kecil. Jika saat ini penduduk Indonesia berjumlah 270 juta, berarti sekurang-kurangnya saat ini ada 4,050,000 penyandang tunanetra [1]. Tunanetra sendiri bukan hanya mencakup untuk penyandang buta total saja tetapi juga mencakup kepada penyandang lemah penglihatan juga seperti katarak.

Penyandang disabilitas tunanetra menurut kutipan dari Dr. Asep Supena, M.Psi bahwa tunanetra (Visual Imprairment) adalah individu yang mengalami gangguan penglihatan cukup berat dan membutuhkan layanan Pendidikan atau pembelajaran khusus. Dari kebanyakan penyandang disabilitas tunanetra, mereka lebih mengandalkan indra peraba dan pendengaran untuk melakukan kegiatan. Meskipun seperti itu tentu saja penyandang tunanetra tidak dapat bekerja secara maksimal karena tidak seperti halnya indra penglihatan yang mendapatkan informasi lebih spesifik misalnya jenis, warna, dan bentuk [2].

Penderita tunanetra menggunakan peraba, pendengar dan penciuman untuk mengetahui benda-benda yang berada di sekitarnya. Akan tetapi mereka tidak bisa mendeteksi jenis dan bendanya, sehingga para penyandang tunanetra kesulitan dalam menemukan benda yang mereka cari. Dengan kemajuan teknologi informasi dan komunikasi yang sangat pesat dari tahun ke tahun, tidak terkecuali banyak pula teknologi yang berkembang untuk membantu para tunanetra melakukan kegiatan dengan mudah. Karena perkembangan tersebut, banyak teknologi aplikasi yang berkembang untuk membantu para penyandang disabilitas tunanetra. Antara lain aplikasi seperti sensor deteksi ultrasonic menggunakan media sabuk, sensor yang berada ditongkat untuk mendeteksi, dan juga mendeteksi benda disekitar menggunakan algoritma random forest [3][4].

Pada Penelitian sebelumnya, pembuatan aplikasi untuk membantu tunanetra mendeteksi ekspresi wajah dengan deep learning menggunakan model CNN (Convolutional Neural Network) dengan akurasi rata-rata dengan akurasi rata-rata mencapai 80%[5]. Selain itu, pembuatan aplikasi untuk mendeteksi objek dengan menerapkan machine learning berupa teknologi pengenalan deteksi objek dengan menggunakan metode YOLO (You Only Look Once) [6]. Adapun berbagai metode lain yang dapat digunakan dalam melakukan pendeteksian seperti Mask RCNN, R-CNN, faster R-CNN, dan masih banyak lagi yang lainnya.

Dari permasalahan yang terjadi sebelumnya, penulis memilih membuat suatu program aplikasi kamera deteksi objek dan pengenalan wajah dengan menggunakan 2 fungsi yaitu YOLO (You Only Look Once) karena dapat melakukan pengenalan objek secara realtime yang tegolong cepat karena arsitekturnya dan CNN (ResNet50) untuk klarifikasi gambar wajah dan output yang akan dihasilkan berupa suara. Adapun keuntungan yang didapat menggunakan sistem ini antara lain para penyandang disabilitas tunanetra dapat mendeteksi

objek dan mengenali orang disekitar agar mengantisipasi kejahatan oleh orang yang tidak dikenal.

II. KAJIAN TEORI

A. Deteksi *Objek*

Deteksi objek merupakan suatu pendeteksian terhadap bentuk atau visual baik berupa gambar maupun video. Deteksi objek digunakan untuk melakukan verifikasi terhadap objek ataupun menganalisa objek yang didapatkan dan bisa sesuai ataupun tidak sesuai tergantung dengan dataset yang optimal.

B. *Pengenalan Wajah*

Pengenalan wajah adalah suatu proses yang dilakukan dalam mengidentifikasi maupun memverifikasi citra data wajah dengan membuat sebuah algoritma komputasi dan disimpan pada database dimana hasil identifikasi wajah tersebut dilakukan perbandingan dengan *eigenface* dari data wajah yang disimpan di database.

C. *You Only Look Once (YOLO)*

YOLO (*You Only Look Once*) merupakan suatu pendekatan menggunakan jaringan saraf tiruan untuk mendeteksi objek secara *real time*. Alasan disebutnya *YOLO* yaitu karena cara kerjanya yang hanya sekali mendeteksi. *YOLO* merupakan bagian dari *CNN* dengan membagi citra menjadi beberapa ruang dan memprediksi setiap kotak pembatas dan *probabilitas* untuk setiap ruang[24][25].

D. *You Only Look Once version 8 (YOLOv8)*

YOLOv8 adalah algoritma *real-time* untuk deteksi objek menggunakan *CNN* (Convolutional Neural Network). Ini adalah versi terbaru dari keluarga *YOLO*, dirilis pada tahun 2023. Tujuan utama *YOLOv8* sendiri adalah untuk mempermudah deteksi objek dengan akurat.

E. *Convolutional Neural Network (CNN)*

Convolutional Neural Network atau yang disingkat dengan *CNN* merupakan suatu metode *machine learning* yang dirancang khusus untuk pengenalan dan klasifikasi gambar. *CNN* memiliki kemampuan mengolah citra dikarenakan *CNN* berusaha memiliki kemampuan untuk meniru sistem pengenalan citra pada *visual cortex* manusia.

F. *Convolutional Neural Network CSPDarknet53*

Fungsi dari *CSPDarknet53* adalah backbone pada *YOLO* yang bertugas mengekstraksi fitur penting dari gambar. Pengembangan *CSPDarknet53* merupakan penyempurnaan dari *DarkNet53* (*YOLOv3*) dengan penambahan *Cross Stage Partial Connections (CSP)* untuk integrasi fitur lebih efektif.

G. Confusion *Matrix*

Confusion matrix merupakan suatu evaluasi performa pada model berbentuk tabel yang berisi kelas prediksi dan kelas kebenaran untuk menyatakan seberapa jumlah data uji yang benar[30][31].

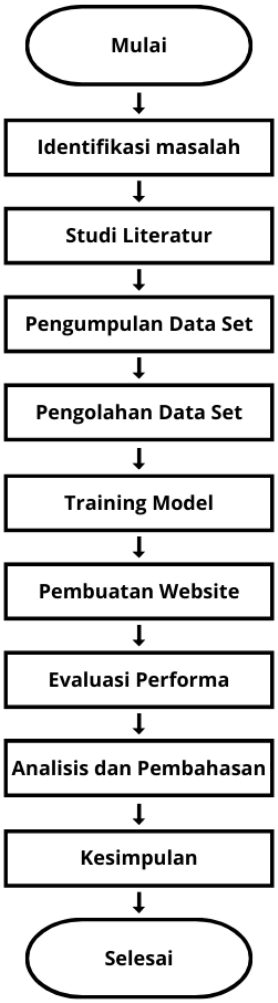
H. GTTS *(Google Text To Speech)*

GTTS merupakan suatu library yang berfungsi untuk mengubah tulisan menjadi audio yang disimpan dalam format audio seperti *mp3* dan digunakan untuk berkomunikasi dengan lawan bicara. *GTTS* berinteraksi dengan database Google API yang telah di Translate ke dalam Bahasa Indonesia.

Artikel ditulis dalam ukuran kertas A4, maksimal 5000 kata dan ditulis menggunakan spasi 1

III. METODE

Memberikan gambaran rancangan penelitian yang meliputi prosedur atau langkah-langkah penelitian, waktu penelitian, sumber data, cara perolehan data dan menjelaskan metode yang akan digunakan dalam penelitian [10 pts].
Diagram alir sangat dibutuhkan dalam proses penelitian ini. Diagram alir di bawah ini merupakan langkah yang diambil oleh penulis untuk mendukung proses penelitian yang akan dibuat. Tujuannya adalah agar penelitian dapat berjalan lebih terarah dan sistematis.



GAMBAR 1

Diagram Alir Penelitian

A. Identifikasi *Masalah*

Penulis menentukan masalah atau tujuan yang ingin dipecahkan yaitu membuat aplikasi deteksi objek.

B. Studi Literatur

Pada studi literatur ini penulis melakukan pendekatan mengenai topik yang diambil dengan cara mencari referensi yang relevan dengan landasan teori. Pada studi literatur ini, sudah dilakukan pada kajian Pustaka dan landasan teori di bab II di atas.

C. Pengumpulan Dataset

Pada pengumpulan dataset ini, dibagi menjadi 3(dua kategori) yaitu :

1. Dataset Objek adalah dataset yang kita perlukan untuk mendeteksi suatu objek yang berada di depan kamera. Pada dataset objek ini, dataset yang pertama digunakan adalah COCO dataset. COCO dataset sendiri merupakan singkatan dari Common Objects in Context, dikarenakan kumpulan data gambar yang dibuat dengan tujuan memajukan pengenalan gambar.

2. Dataset objek yang kedua lebih cenderung menggunakan gambar atau foto yang berhubungan dengan jalanan yaitu antara lain pagar dengan 146 data gambar pagar, tembok dengan 78 data gambar tembok , tiang dengan 64 data gambar tiang, dan juga selokan dengan 101 data gambar selokan. Berikut kumpulan sampel swafoto data beserta label yang sudah disertakan. Dataset objek jalanan dibagi menjadi tiga bagian untuk keperluan pelatihan, validasi, dan pengujian model. Proporsi pembagiannya adalah 88% untuk data latih, 8% untuk data validasi, dan 4% untuk data uji

3. Dataset Wajah adalah dataset yang kita perlukan untuk mengenali wajah dari orang yang dikenali. Pada dataset ini memerlukan train dataset foto pada wajah lalu dilabeli sesuai nama pada foto.

IV. HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan hasil penelitian yang berupa dokumentasi dari pelatihan dan pengujian model yang dilatih dan model *Pre-trained*, dokumentasi pelatihan akan berupa label dan prediksi data latih kemudian metrik evaluasi dari pelatihan, kemudian pada tahap pengujian akan berupa hasil uji dari model yang sudah diimplementasikan dengan Webcam menggunakan streamlit.

A. Pelatihan Data Objek Jalanan

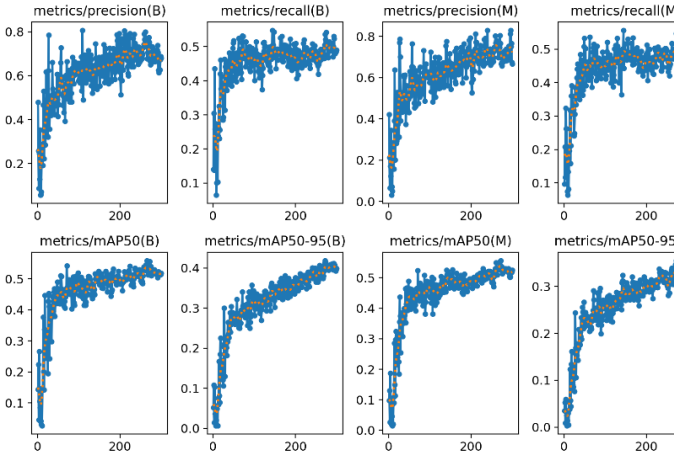
GAMBAR 1
(A)Label Validation Data Objek Jalanan

Gambar 1(A) merupakan data *validation* yang digunakan untuk pengujian pada data latih. Dari gambar, terlihat beberapa objek yang dilabel, antara lain tembok, pagar, selokan, dan tiang.

GAMBAR 2

(A)Hasil Prediksi Validation Data Objek Jalanan

Gambar 2(A) merupakan hasil prediksi dari model deteksi objek jalanan yang ada pada gambar 2(A). Terdapat hasil deteksi dengan *confidence score* 1.0, objek yang terdeteksi adalah pagar, dan tembok, kemudian hasil deteksi dengan *confidence score* 0.1 – 0.9 objek yang terdeteksi adalah selokan, pagar, dan tembok, terdapat juga beberapa gambar yang tidak dapat terdeteksi oleh model, gambar tersebut memiliki objek tiang, pagar, dan tembok.



GAMBAR 3

(A) Grafik Hasil Pelatihan Data Objek Jalanan

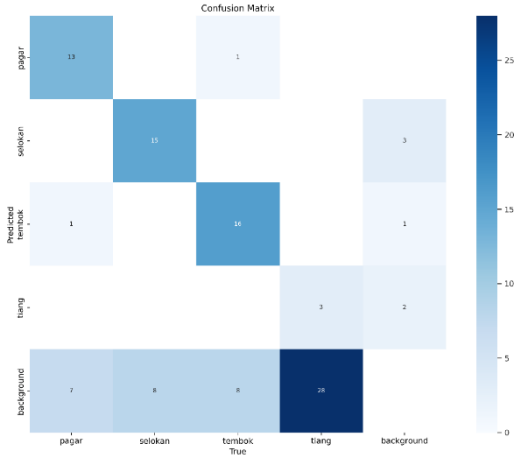
Gambar 3(A) menunjukkan hasil dari pelatihan data objek jalanan, nilai dari metrik evaluasi grafik tersebut terdapat pada tabel gambar 3.

TABEL 1

(A) Metrik Hasil Pelatihan Data Objek Jalanan

Epoch	*Precision*	*Recall*	*F1-score*
300	0.66	0.49	0.56

Berdasarkan tabel 1(A) model yang dievaluasi memiliki *precision* yang cukup baik (0.66), yang berarti prediksi positifnya cukup akurat. Namun, *recall-nya* relatif rendah (0.49), yang berarti model kurang baik dalam mendeteksi semua objek yang ada. *F1-score* sebesar 0.56 menunjukkan bahwa ada ruang untuk perbaikan dalam menyeimbangkan *precision* dan *recall.*



GAMBAR 4

(A) Confusion Matrix Pelatihan Data Objek Jalanan

Gambar 4(A) menampilkan *confusion matrix* dari pelatihan data objek jalanan yang mana :

- Baris "pagar", Kolom "pagar": Angka 13 menunjukkan bahwa ada 13 gambar yang sebenarnya merupakan pagar, dan model memprediksi dengan benar sebagai pagar. Ini adalah *True Positive* untuk kelas "pagar".
- Baris "tembok", Kolom "pagar": Angka 1 menunjukkan bahwa ada 1 gambar yang sebenarnya merupakan pagar, tetapi model salah memprediksi sebagai tembok. Ini juga kesalahan klasifikasi.
- Baris "*background*", Kolom "tiang": Angka 20 menunjukkan bahwa ada 20 gambar yang sebenarnya merupakan tiang, tetapi model salah memprediksi sebagai *background*. Ini juga kesalahan klasifikasi.

TABEL 2

(A)*Confution matrix* data objek jalanan

		Predicted Class				
		Pagar	Selokan	Tembok	Tiang	Background
Actual Class	Pagar	13	1	0	0	1
	Selokan	0	15	0	3	2
	Tembok	1	0	18	1	0
	Tiang	0	3	1	2	0
	Background	7	8	8	26	0

Tabel 2(A) menunjukkan *Confution matrix* data objek jalanan dari data test dengan perhitungan nilai precission berkisar antara 0.063 – 0.66, nilai recall berkisar antara 0.333 – 0.9 dan nilai F1 score berkisar 0.105 – 0.766.

Pada kelas pagar, selokan, dan tembok menunjukkan nilai precision, recall, dan F1-score yang relatif baik, dengan nilai di atas 0.5. Namun, kelas tiang menunjukkan nilai yang lebih rendah, yaitu precision sebesar 0.063, recall sebesar 0.333, dan F1-score sebesar 0.105. Hal ini disebabkan oleh pengambilan gambar tiang yang kurang ideal contohnya dalam banyak kasus, satu gambar mengandung lebih dari satu

tiang, sehingga model kesulitan untuk mengidentifikasi setiap tiang secara individual.

Rendahnya nilai precision pada kelas tiang menunjukkan bahwa model sering salah mengklasifikasikan objek lain sebagai tiang. Sementara itu, rendahnya nilai recall mengindikasikan bahwa model seringkali gagal mendeteksi tiang yang sebenarnya.

B. Pelatihan *Data Wajah*

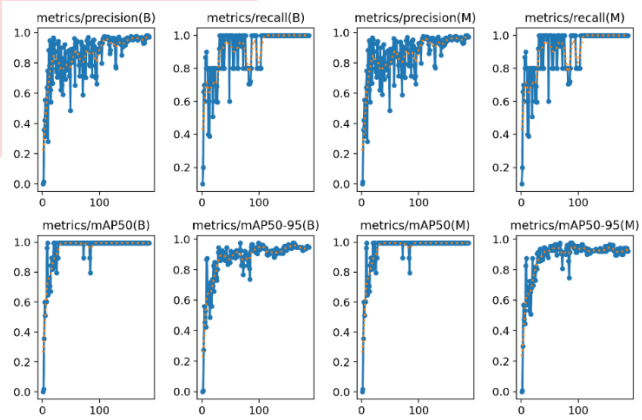
GAMBAR 1
(B) Label Validation Data Wajah

Gambar 1 merupakan data *validation* yang digunakan untuk pengujian pada data latih model deteksi wajah. Dari gambar, terlihat beberapa wajah yang dilabel, antara lain merupakan wajah dari Farah, Kelvin, Afifah, Ary, dan Latifah.

GAMBAR 2

(B)Hasil Prediksi Validation Data Wajah

Gambar 2(B) merupakan hasil prediksi dari model deteksi wajah yang ada pada gambar 1(B) Terdapat hasil deteksi dengan *confidence score* 1.0, wajah yang terdeteksi adalah wajah dari Latifah dan Kelvin, kemudian hasil deteksi dengan *confidence score* 0.1 – 0.9 wajah yang terdeteksi merupakan wajah dari Farah dan Ary, lalu terdapat juga kesalahan deteksi pada wajah dari Afifah terdeteksi Farah, pada gambar terdapat dua warna *Bounding Box* yang berarti model mendeteksi wajah tersebut dengan dua class.



Gambar 3

(C) Grafik Hasil Pelatihan Data Wajah

Gambar 3(B) menunjukkan hasil dari pelatihan data wajah, nilai dari metrik evaluasi grafik tersebut terdapat pada tabel 4.4.

TABEL 1

(A) Metrik Hasil Pelatihan Data Wajah

Epoch	*Precision*	*Recall*	*F1-Score*
187	0.97	1	0.98

Berdasarkan tabel 1(A), precision yang sangat tinggi (0.97) menunjukkan bahwa model sangat akurat dalam memprediksi label positif. Model jarang melakukan kesalahan dengan memprediksi sesuatu sebagai positif padahal sebenarnya negatif. recall yang sempurna (1) menunjukkan bahwa model berhasil mendeteksi semua contoh positif yang ada dalam dataset. Model tidak melewatkan contoh positif sama sekali. F1-score yang sangat tinggi (0.98)

mengkonfirmasi bahwa model memiliki keseimbangan yang sangat baik antara precision dan recall.

C. Pengujian

Pengujian dilakukan menggunakan web localhost yang dibangun dengan library streamlit tampilan dari web terdapat pada gambar 1(A)

GAMBAR 1

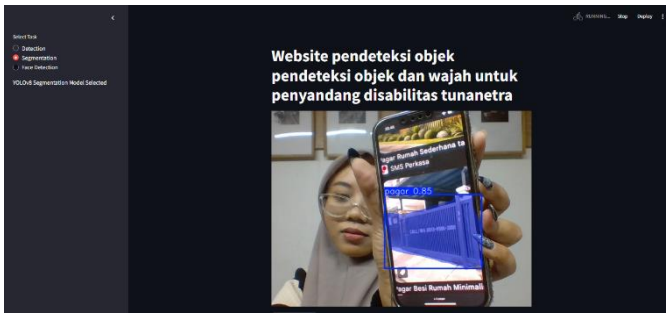
(C) Tampilan web localhost menggunakan streamlit

GAMBAR 2

(C) Tampilan web localhost menggunakan streamlit menu *Detection*

GAMBAR 3

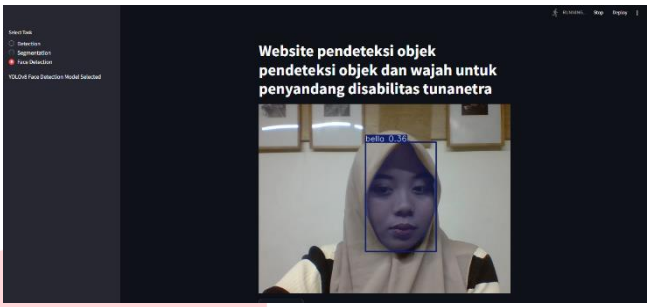
(C) Tampilan web localhost menggunakan streamlit



menu *Segmentation*

GAMBAR 4

(D) Tampilan web localhost menggunakan streamlit menu *Face Detection*



Proses dimulai dengan pemilihan model deteksi, yang dapat berupa deteksi objek umum, *segmentasi* (data objek jalanan), dan deteksi wajah. Setelah model dipilih, webcam diaktifkan dengan menekan tombol untuk menangkap frame video secara berurutan. Untuk menu *Detection* menampilkan deteksi *data coco* seperti pada Gambar 4.10. Lalu pada gambar 4.11. terdapat tampilan menu *Segmentation* yang dimana menampilkan data yang sudah dilatih seperti pagar, tembok, tiang, dan selokan. Terakhir ada tampilan *Face Detection* yang menampilkan wajah yang sudah di latih yang tertera pada gambar 4.11.

Setiap frame yang ditangkap kemudian diubah ukurannya menjadi 640 x 480 piksel untuk efisiensi pemrosesan dan ditampilkan. Sistem kemudian melakukan deteksi objek pada frame tersebut. Jika tidak ada objek yang terdeteksi, proses kembali ke pengambilan frame berikutnya. Jika objek terdeteksi, sistem akan menampilkan *Bounding Box* (kotak yang mengelilingi objek), label (nama objek), dan *confidence score* (tingkat kepercayaan deteksi).

Selanjutnya, sistem memeriksa apakah *confidence score* lebih dari 50%. Jika tidak, proses kembali ke pengambilan frame berikutnya. Jika *confidence score* memenuhi syarat, fitur *text-to-speech* diaktifkan untuk memberikan *Output* suara yang berupa kelas dari objek yang terdeteksi. Setelah proses selesai atau pengguna menghentikan aplikasi, webcam dimatikan.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa deteksi objek menggunakan YOLOv8 memiliki kinerja yang bervariasi. Model deteksi objek jalanan memiliki precision baik (0.66), namun recall rendah (0.49), sementara model deteksi wajah menunjukkan hasil sempurna dengan precision 0.97 dan recall 1. Namun, model kesulitan mendeteksi objek tiang dan wajah tertentu. Perbedaan kinerja ini dipengaruhi oleh variasi data dan keterbatasan data uji.

Saran untuk penelitian selanjutnya adalah meningkatkan kualitas data dengan memperluas variasi kondisi, mengembangkan aplikasi berbasis Android, mengeksplorasi arsitektur model lebih canggih, serta melakukan optimasi hyperparameter untuk meningkatkan kinerja dan efisiensi pelatihan.

REFERENSI

Direkomendasikan menggunakan *reference management tools* (*mendeley*), format style menggunakan IEEE. Contoh penulisan referensi IEEE Style:

Print References

- Book
  - Book Author(s). Book title. Location: Publishing company, year, pp.

  Example:

  [1] W.K. Chen. *Linear Networks and Systems*. Belmont, CA: Wadsworth, 1993, pp. 123-35.
- Book Chapters
  - Author(s). "Chapter title" in Book title, edition, volume. Editors name, Ed. Publishing location: Publishing company, year, pp.

  Example:

  [1] J.E. Bourne. "*Synthetic structure of industrial plastics*," in Plastics, 2nd ed., vol. 3. J. Peters, Ed. New York: McGraw-Hill, 1964, pp.15-67.
- Article in a Journal
  - Author(s). "Article title". Journal title, vol., pp, date.

  Example:

  [1] G. Pevere. "*Infrared Nation." The International Journal of Infrared Design*, vol. 33, pp. 56-99, Jan. 1979.

Electronic References

- Books
  - Author. (year, Month day). Book title. (edition). [Type of medium]. Vol. (issue). Available: site/path/file [date accessed].

  Example:

  [1] S. Calmer. (1999, June 1). *Engineering and Art. (2nd edition*). [*On-line*]. 27(3). *Available*: www.enggart.com/examples/students.html [May 21, 2003].
- Journal
  - Author. (year, month). "Article title." Journal title. [Type of medium]. Vol. (issue), pages. Available: site/path/file [date accessed].

  Example:

  [1] A. Paul. (1987, Oct.). "*Electrical properties of flying machines." Flying Machines. [Online]*. 38(1), pp. 778-998. *Available*: www.flyingmachjourn/properties/fly.edu [Dec. 1, 2003].
- World Wide Web

  Author(s)*. "Title." Internet: complete URL, date updated* [date accessed].

  Example:

  [1] M. Duncan. "*Engineering Concepts on Ice"*. Internet: www.iceengg.edu/staff.html, Oct. 25, 2000 [Nov. 29, 2003].