Pengenalan Objek Dengan Model Pra-Terlatih SSD MobileNet Pada Aplikasi Kasir

Nazil Ilham Burhanudin[1], Arif Dwi Laksito[2]*, Acihmah Sidauruk[3],
Muhammad Resa Arif Yudianto[4], Alfie Nur Rahmi[5]

Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta[1], [2], [3], [5]
Jl. Padjajaran, Ring Road Utara, Condong Catur, Sleman
Fakultas Teknik, Universitas Muhammadiyah Magelang[4]
Jl. Mayjen Bambang Soegeng, Glagak, Sumberrejo, Mertoyudan, Magelang

nazil.1799@students.amikom.ac.id[1], arif.laksito@amikom.ac.id[2]*, acihmah@amikom.ac.id[3],
resamuhammad96@unimma.ac.id[4], alfienurrahmi@amikom.ac.id[5]

***Abstract*—Object recognition is a type of image processing technique that is frequently employed in current applications such as facial identification, vehicle detection, and automated cashiers. One issue with barcode and RFID cashier apps is that they cannot scan several products at the same time. The cashier application employing object identification using picture images is believed to be able to distinguish more than one object in order to speed up the transaction process. The usage of SSD pre-trained models with MobileNet architecture to detect items in automatic cashier applications is discussed in this paper. This study put the model to the test on three types of soft drink objects: coca-cola, floridina, and good day. A smartphone camera was used to collect the data, which totaled 203 images. The findings indicated that the product object identification method was 82.9% accurate, 97.5% precise, and 84.7% recall. The object recognition process takes between 365 and 827 milliseconds, with an average time of 695 milliseconds (0.69 seconds).**

Keywords— Object detection, MobileNet, Tensorflow

***Abstrak*—Pengenalan objek adalah teknologi pengolahan citra yang banyak digunakan di berbagai aplikasi modern seperti pengenalan wajah, deteksi kendaraan, dan aplikasi kasir otomatis. Salah satu masalah pada aplikasi kasir *barcode* dan RFID adakah tidak bisa melakukan *scanning* pada *multi item* secara bersamaan. Aplikasi kasir dengan menggunakan deteksi objek menggunakan citra gambar di klaim terbukti dapat mengenali lebih dari 1 objek sehinggal mempercepat proses transaksi. Penelitian ini membahas penggunaan model pra-terlatih SSD dengan arsitektur MobileNet untuk mengenali objek pada aplikasi kasir otomatis. Penelitian ini menguji kinerja model pada tiga kelas objek softdrink yaitu coca-cola, floridina dan good day. Data diambil menggunakan kamera smartphone dengan jumlah data keselurahn adalah 203. Hasil penelitian menunjukan proses pengenalan objek produk dengan akurasi 82,9%, presisi 97,5%, dan recall 84,7%. Proses pengenalan produk membutuhkan waktu antara 365 ms hingga 827 ms dengan rata-rata waktu yang dibutuhkan yaitu 695 ms (0,69 detik).**

Kata Kunci— Deteksi objek, MobileNet, Tensorflow

I. PENDAHULUAN

Industri 4.0 membawa dampak positif yang besar terhadap semua sektor bidang, salah satunya yaitu bidang ekonomi. Pemanfaatan teknologi informasi terbukti mampu meningkatkan kapabilitas suatu organisasi bisnis yang secara tidak langsung juga dapat meningkatkan pendapatan. Hal tersebut yang menjadikan faktor penggunaan teknologi yang tepat guna dalam mendukung proses bisnis menjadi sebuah tolok ukur persaingan antar organisasi bisnis [1]. Beragam dampak positif telah banyak dirasakan oleh berbagai stakeholder yang memanfaatkan teknologi informasi dalam membantu proses bisnis mereka. Berbagai dampak positif tersebut diantaranya adalah proses transaksi jual beli yang menjadi lebih cepat dan mudah [2].

Kemudahan dan kecepatan proses transaksi tersebut salah satunya disebabkan karena penggunaan aplikasi kasir. Melalui aplikasi tersebut penjual dapat melakukan perekapan jenis dan jumlah item yang dibeli serta total yang harus dibayarkan lebih cepat melalui *scanning barcode* pada setiap produk. Perkembangan teknologi yang begitu cepat berpengaruh terhadap keberlangsungan kegiatan bisnis yang telah menerapkan *Information and Communications Technology* (ICT) dalam membantu proses bisnis mereka [3]. Oleh sebab itu seorang stakeholder atau pelaku usaha harus mengikuti perkembangan teknologi yang ada saat ini. Pada faktanya memang teknologi *scan barcode* terhadap produk dahulu banyak digunakan sejak tahun 1990an dan sempat menjadi populer.

Akan tetapi seiring dengan perkembangan teknologi yang semakin canggih, mengakibatkan teknologi *barcode* menjadi kurang efektif akibat kelemahan-kelemahan yang dimilikinya. Berbagai kelemahan yang ada pada teknologi tersebut diantaranya yaitu proses pemindaian terhadap produk harus dilakukan secara manual. Selain itu jika kemasan produk mengalami kerusakan yang mengakibatkan *barcode* tidak dapat dapat terbaca dengan jelas, hal tersebut juga menjadi kelemahan lain dari teknologi tersebut [4]. Semakin beragamnya varian produk baru membuat teknologi barkode dan RFID menjadi

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i2.1659, Copyright ©2023
Submitted : March 28, 2023, Revised : May 6, 2023, Accepted : May 12, 2023, Published : July 1, 2023

kurang efektif untuk menyimpan data yang baru. Posisi *barcode* yang kurang pas ketika dilakukan *scanning* mengakibatkan proses transaksi yang tidak lagi cepat dan efisien [5]. Oleh sebab itu diperlukan mekanisme pengenalan produk yang lebih cepat dan lebih akurat. Peralihan menjadi industri 4.0 terbukti telah memberikan dampak positif yang signifikan bagi keberlanjutan dan peningkatan kinerja bisnis Usaha Mikro Kecil dan Menengah (UMKM). Melalui ciri khas dan keunggulan teknologi berbasis otomatisasi mampu memberikan kontribusi yang baik bagi para pelaku bisnis [6].

Salah satu contoh penerapan teknologi otomatisasi yaitu pendeteksian dan klasifikasi penyakit daun kopi berdasarkan tingkat keparahannya melalui pengenalan citra daun kopi [7]. Klasifikasi penyakit daun kopi secara otomatis tersebut menggunakan metode Yolov7 terhadap 1664 data yang terbagi menjadi 4 kelas yaitu daun dengan kriteria bagus, penyakit Miner, Phoma dan Rust. Hasil yang diperoleh yaitu model mampu membedakan penyakit daun kopi ke dalam 4 kelas melalui pengenalan citra daun dengan tingkat presisi sebesar 92%, sehingga dapat membantu sektor agrikultur dalam mengantisipasi penyakit daun kopi secara lebih dini. Penelitian lain yang serupa tentang klasifikasi rambu lalu lintas yang terdiri dari 2050 data yang terbagi ke dalam 10 kelas. Sejumlah 1750 data digunakan untuk melatih model dan sisanya digunakan sebagai data testing yang diolah menggunakan algoritma *Convolutional Neural Network* (CNN). Hasil penelitian memperoleh akurasi sebesar 99% [8]. Melalui penelitian tersebut mampu membantu para pengendara dalam memahami setiap rambu lalu lintas yang ada.

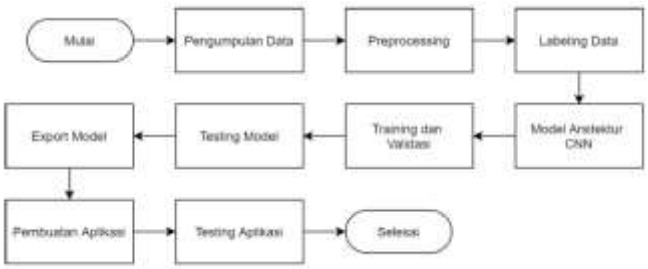
Dari beberapa penelitian serupa tentang pengenalan citra [7], [8], diperoleh hasil bahwa kinerja *deep learning* menggunakan algoritma CNN lebih baik dibandingkan algoritma *machine learning* [9]. CNN sendiri memiliki beberapa varian yang dibedakan dari jenis arsitekturnya, di mana antara 1 arsitektur dengan arsitektur yang lain memiliki kinerja model yang berbeda-beda [10]. 21 arsitektur CNN dibandingkan dalam menentukan paru-paru yang terkena covid-19 melalui citra X-Ray dan didapatkan 5 arsitektur CNN terbaik [11]. Kelima arsitektur tersebut yaitu MobileNet, EfficientNetB2, DenseNet169, InceptionResNetV2 dan InceptionV3. Penelitian lain mendapatkan hasil bahwa arsitektur MobileNet memiliki kinerja yang cukup efektif dengan kebutuhan sumber daya yang kecil dibandingkan arsitektur CNN yang lain. Melalui keunggulan tersebut arsitektur MobileNet dapat berjalan di hampir semua environment [12].

Arsitektur CNN ini juga dapat digunakan dalam menghitung jumlah orang yang melakukan antrian di kasir serta dapat menganalisa lamanya waktu yang dibutuhkan setiap pelanggan dalam melakukan pembayaran [13]. Akurasi yang diperoleh melalui arsitektur model R-CNN sebesar 95%. Salah satu model arsitektur MobileNet yaitu SSD MobileNetV2 melalui TensorFlow Object Detection API sebagai model pra-terlatih mampu melakukan pengenalan objek perangkat keras yaitu kamera, handphone, headphone, mouse dan laptop. Melalui skenario pembagian 500 dataset ke dalam 70% sebagai data latih, 20% sebagai data validasi dan sisanya sebagai data uji. Model ini mampu memperoleh kinerja yang cukup baik dengan nilai akurasi sebesar 93% serta dapat mengenali objek lebih dari 1 [14]. TensorFlow Object Detection API juga mampu mengenali objek yang terbagi ke dalam kelas yang banyak seperti penelitian sebelumnya mengenai pengenalan bahasa isyarat yang terbagi ke dalam 25 alphabet [15]. Meskipun hanya terdiri dari sedikit data citra yaitu sejumlah 650 gambar yang terbagi ke dalam 25 kelas, model pendeteksi dengan TensorFlow Object Detection API mampu memperoleh nilai kepercayaan sebesar 85,45%.

Berdasarkan beberapa penelitian terdahulu, salah satu masalah pada aplikasi kasir barcode dan RFID adakah tidak bisa melakukan scanning pada multi item secara bersamaan. Aplikasi kasir dengan menggunakan deteksi objek menggunakan citra gambar di klaim terbukti dapat mengenali lebih dari 1 objek sehinggal mempercepat proses transaksi. Oleh karena itu di penelitian ini kami akan membangun prototipe aplikasi kasir menggunakan arsitektur pra-terlatih MobileNet dalam mengenali produk di kasir karena model ini memiliki kinerja yang baik serta mampu mengenali objek lebih dari satu. Selain itu penggunaan TensorFlow Object Detection API dengan model pra-terlatih MobileNet tidak membutuhkan *resource* yang besar, sehingga fleksibel diterapkan pada semua platform baik komputer yang menggunakan GPU maupun hanya CPU. Sehingga diharapkan mampu meningkatkan kinerja dan pelayanan kasir dalam melakukan transaksi yang lebih cepat dan akurat.

II. METODE PENELITIAN

Alur penelitian ini diawali dari pengumpulan data, melakukan pre-processing dan labeling, pelatihan machine learning model dan implementasi di aplikasi. Diagram alur penelitian tersaji di Gambar 1 dibawah ini.

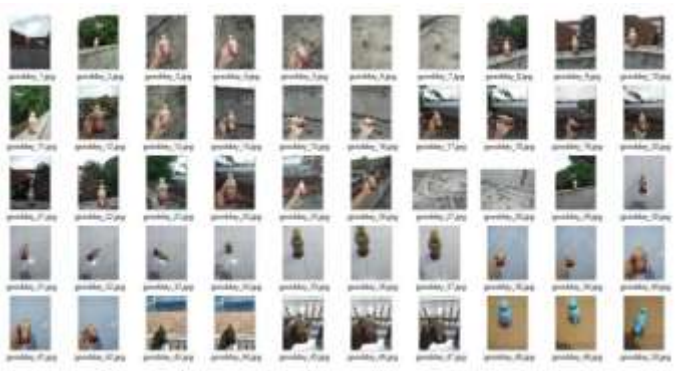


Gambar 1. Diagram alur urutan penelitian

A. *Pengumpulan Data*

Pada tahap ini dilakukan proses pengambilan gambar produk yang akan digunakan untuk penelitian ini. Data yang digunakan merupakan gambar produk yang diambil secara langsung dengan menggunakan kamera smartphone yang sama. Pengambilan data ini disimpan dalam bentuk .jpg dan diambil dengan sudut yang berbeda. Gambar 2 menunjukan beberapa contoh data yang telah dikumpulkan.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i2.1659, Copyright ©2023
Submitted : March 28, 2023, Revised : May 6, 2023, Accepted : May 12, 2023, Published : July 1, 2023



Gambar 2. Contoh kumpulan data objek

Dataset yang digunakan dikelompokan menjadi 2 jenis yaitu data train dan test dimana data train terdapat 186 data yang akan digunakan untuk proses training dan data test digunakan untuk proses testing. Total seluruh dataset yang dipakai yaitu sebanyak 203 gambar seperti pada informasi di Tabel 1 berikut.

TABLE I. POPULASI DATA

Jenis Data	**Label**	**Jumlah Data**
Train	Coca-cola	58 gambar
Train	Floridina	64 gambar
Train	Good day	64 gambar
Test	Data campuran	17 gambar
Total data		203 gambar

Dari keseluruhan data yang akan digunakan untuk proses *training*, data tersebut akan dipisahkan menjadi 2 yaitu data *train* dan data validasi. Data *train* merupakan data yang akan digunakan untuk proses *training*, sedangkan data validasi digunakan untuk proses evaluasi model. Pembagian data tersebut dilakukan dengan perbandingan 80% data *train* dan 20% data validasi.

B. Pre-processing *dan* Labeling Data

Pada tahap *pre-processing* dilakukan proses resize pada semua gambar. Proses *resize* gambar diperlukan untuk memperkecil ukuran file serta membuat proses training model menjadi lebih cepat dan ringan. Ukuran asli pada gambar yaitu 3120 x 4160 pixel, kemudian ukuran gambar diturunkan menjadi 780 x 1040 pixel.

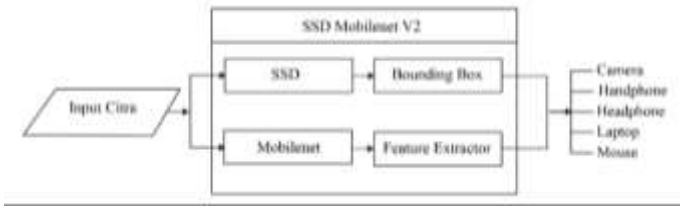
Selanjutnya, data yang telah melalui proses *resizing* kemudian dapat dilakukan proses *labeling* untuk memberikan label dan *bounding box* pada setiap produk pada gambar. Proses labeling ini dilakukan dengan menggunakan software *labelimg* yang tersedia secara open source. Label untuk setiap gambar tersebut kemudian disimpan dalam bentuk file .xml.

C. Model Arsitektur CNN

Pada tahap ini dilakukan pembuatan model arsitektur CNN, dimana akan menggunakan arsitektur pra-terlatih SSD MobileNetV2. Untuk menggunakan model pra-terlatih diperlukan file checkpoint dari model pra-terlatih, kemudian akan dilanjutkan dengan menggunakan dataset objek yang ingin di deteksi. File *checkpoint* dari model pra-terlatih dapat di unduh dari halaman *github repository tensorflow* model. Selain file *checkpoint* dari model, dibutuhkan juga file konfigurasi yang akan digunakan untuk mengubah konfigurasi dari *hyperparameter* yang perlu disesuaikan dengan dataset yang digunakan untuk proses training. *Hyperparameter* tersebut meliputi :

- Banyak class yaitu banyaknya class dalam dataset yang akan digunakan untuk proses training.
- Ukuran batch yaitu banyaknya pengelompokan data dalam sekali proses. Semakin besar nilai batch maka akan membuat proses menjadi lebih cepat tetapi akan meningkatkan penggunaan memori, sehingga perlu disesuaikan dengan ukuran dataset dan spesifikasi *device* yang digunakan.
- Banyak step untuk proses training dan evaluasi merupakan banyaknya langkah atau iterasi yang diperlukan untuk proses training atau evaluasi.
- Lokasi *file* yang diperlukan untuk proses *training* dan evaluasi, *file-file* tersebut meliputi *file* dataset untuk training, dataset untuk evaluasi, *file checkpoint* model pra-terlatih, *file* konfigurasi model pra-terlatih, *file* label, dan lokasi penyimpanan *checkpoint* saat proses training.

Arsitektur dari model pra-terlatih SSD MobileNetV2 tersaji seperti pada Gambar 3. SSD MobileNet terdiri dari base model SSD dan MobileNet sebagai Network Model. MobileNet bekerja utuk mengklasifikasi objek yang telah terdeteksi oleh SSD dengan *Bounding Box*. Selanjutnya, penggabungan SSD dan MobileNet akan meningkatkan proses pembuatan aplikasi deteksi objek. Pada penelitian Aningtiyas tersebut objek klasifikasi yang digunakan adalah camera, handphone, headphone, laptop dan mouse. Sedangkan pada penelitian ini menggunakan produk minuman sebagai dengan 3 kategori yaitu: Coca-cola, Floridina dan Good day.



Gambar 3. Arsitektur model SSD MobileNetV2 [14]

III. HASIL DAN PEMBAHASAN

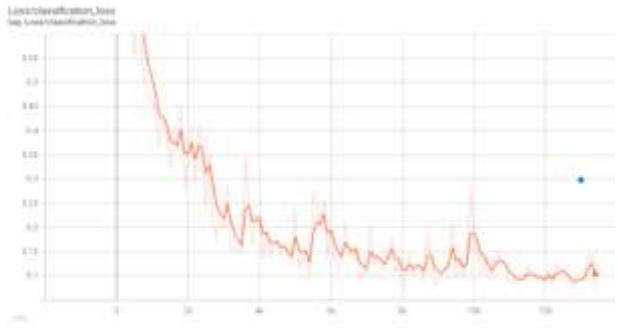
Pada bab ini dibahas terkait implementasi aplikasi kasir dan pengujian dari model *machine learning* yang digunakan sebagai pendeteksi objek.

A. Pelatihan Model

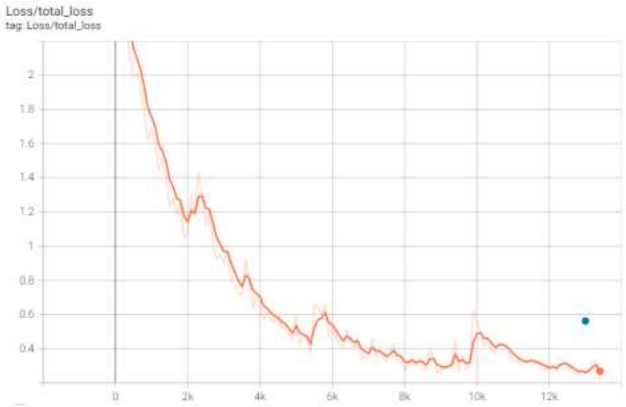
Hasil dari proses pelatihan yang telah dilakukan diperoleh nilai *classification loss* sebesar 0,090902075 dan *total loss* 0,25104278. Untuk proses pelatihan tersebut digunakan nilai

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i2.1659, Copyright ©2023
Submitted : March 28, 2023, Revised : May 6, 2023, Accepted : May 12, 2023, Published : July 1, 2023

train steps sebesar 13000, *val steps* 1000 dan *batch size* 32. Proses pelatihan membutuhkan waktu 1 jam 35 menit 55 detik, dimana proses training dilakukan pada Google Collaboratory dengan menggunakan *runtime* GPU akselerator. Adapun grafik nilai *classification loss* dam *total loss* dari proses pelatihan terdapat pada gambar 4 dan 5 di bawah ini. Pada grafik tersebut menunjukan di *epoch* awal *loss* masih cukup tinggi, kemudian berangsur menurun pada *epoch* ke 4000. Dapat dikatakan bahwa model ini bekerja baik pada proses pengenalan objek dari data *training*.



Gambar 4. Grafik *clasification loss*



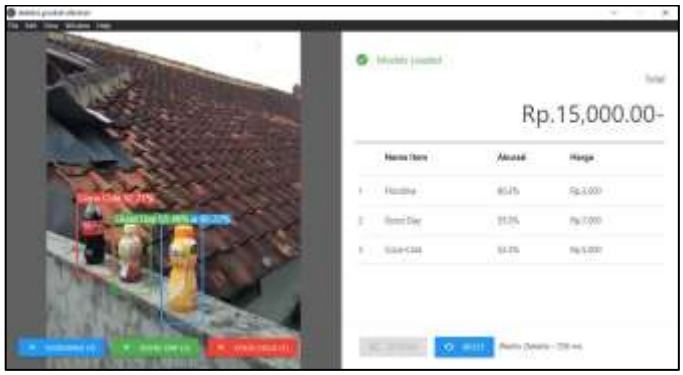
Gambar 5. Grafik *total loss*

Selanjutnya apabila model telah selesai dibuat, maka model tersebut akan di *export* menjadi *file* model bertipe *Tensorflow* JS. Hasil dari proses *export* kemudian akan menghasilkan file model Binary (.bin) dan JavaScript Object Notation (.json) yang dapat diimplementasikan pada aplikasi deteksi objek.

Setelah model yang dibutuhkan telah selesai dibuat dan dapat bekerja sebagaimana mestinya, kemudian dilakukan proses pembuatan aplikasi hingga selesai. Pada penelitian ini *framework* yang digunakan untuk pembuatan aplikasi yaitu Electron JS. Aplikasi prototipe ini memuat beberapa fitur saja seperti, deteksi produk, menampilkan *bounding box*, dan menampilkan harga dan total belanja.

B. Implementasi aplikasi

Implementasi aplikasi ini sebatas pada prototipe dengan minimum fungsionalitas dapat mendeteksi gambar sebagai input. *Form* identifikasi digunakan untuk melakukan identifikasi citra yang dimasukkan melalui tombol *input* gambar. Kemudian seluruh informasi prediksi akan ditambahkan beserta *bounding box* setelah menekan tombol deteksi. Apabila objek dapat terdeteksi akan menampilkan *bounding box* area objek yang terdeteksi kemudian muncul harga dan banyak item yang terdeteksi beserta dengan total belanja. Secara lengkap tampilan *form* tersebut pada gambar 6 berikut.



Gambar 6. Form identifikasi objek

C. Pengujian kecepatan deteksi

Pengujian kecepatan deteksi dilakukan agar dapat mengetahui apakah deteksi objek produk dari gambar dapat dimanfaatkan dalam implementasi di kasir swalayan. Data gambar yang digunakan yaitu data test yang merupakan gambar dengan banyak objek produk. Pengujian dilakukan sebanyak 3 kali menggunakan *consumer PC* dengan spesifikasi *processor* intel-i5 2,4 Ghz dan 4GB RAM, kemudian diambil rata-rata dari ketiga pengujian tersebut untuk mendapatkan nilai yang akurat.

Berdasarkan hasil pengujian diperoleh waktu yang diperlukan untuk menjalankan deteksi yaitu 365 sampai dengan 827 ms dan rata-rata waktu yang dibutuhkan yaitu 695 ms. Dari hasil pengujian dapat dilihat bahwa beberapa kecepatan waktu pendeteksian menunjukkan nilai tidak konsisten, hal tersebut dapat dipengaruhi oleh banyak faktor seperti kualitas gambar *input*, spesifikasi *hardaware* dan lain sebagainya.

D. Pengujian akurasi deteksi

Proses pengujian akurasi dilakukan dengan 17 gambar yang terdiri dari 3 produk. Adapun hasil pengujian gambar produk dapat dilihat pada Tabel 2 berikut ini.

TABLE II. HASIL PENGUJIAN AKURASI DETEKSI

Gambar	Jumlah Aktual	Objek terdeteksi	TP	FP	PN
test_1.jpg	2	2	2		
test_2.jpg	2	2	2		
test_3.jpg	2	2	2		
test_4.jpg	2	1	1		1
test_5.jpg	3	3	3		

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i2.1659, Copyright ©2023
Submitted : March 28, 2023, Revised : May 6, 2023, Accepted : May 12, 2023, Published : July 1, 2023

Gambar	Jumlah Aktual	Objek terdeteksi	TP	FP	PN
test_6.jpg	3	1	1		2
test_7.jpg	3	2	2		1
test_8.jpg	3	3	3		
test_9.jpg	3	3	2	1	
test_10.jpg	3	2	2		1
test_11.jpg	3	2	2		1
test_12.jpg	3	2	2		1
test_13.jpg	3	3	3		
test_14.jpg	3	3	3		
test_15.jpg	3	3	3		
test_16.jpg	3	3	3		
test_17.jpg	3	3	3		
Total			39	1	7

Berdasarkan hasil pengujian didapatkan hasil *confusion matrix* yang disajikan pada tabel 2 di bawah ini.

TABLE III. HASIL PERHITUNGAN CONFUSION MATRIX

Matrik		Kondisi Aktual	
		Ada	Tidak Ada
Hasil Deteksi	Terdeteksi	39	1
	Tidak Terdeteksi	7	0

Sedangkan untuk perhitungan akurasi, presisi, dan recall berdasarkan formula (x,y,z) diuraikan pada tabel 3 berikut.

TABLE IV. HASIL PERHITUNGAN NILAI AKURASI, PRESISI DAN RECALL

Aspek	Rumus	Hasil
Akurasi	$39/(39+7+1) * 100\%$	82,9%
Presisi	$39/(39+1) * 100\%$	97,5%
Recall	$39/(39+7) * 100\%$	84,7%

Dari tabel diatas diperoleh hasil identifikasi menghasilkan nilai akurasi 82,9%, presisi 97,5% dan nilai *recall* sebesar 84,7%.

IV. KESIMPULAN

Dari hasil penelitian yang telah kami lakukan menunjukkan bahwa aplikasi kasir deteksi objek SSD MobileNet dan Tensorflow Object Detection API dapat mendeteksi citra produk dengan baik dan dapat mengenali lebih dari 1 objek bersamaan. Waktu yang dibutuhkan aplikasi untuk mendeteksi objek berkisar antara 365 hingga 827 ms dengan rata-rata 695 ms. Hasil pengujian deteksi berdasarkan 203 data yang terdiri dari 186 data untuk proses training, sedangkan 17 data lainnya untuk proses test dan mendapatkan nilai akurasi sebesar 82,9%, presisi 97,5%, dan nilai *recall* sebesar 84,7%.

Penelitian ke depan diharapkan aplikasi tersebut dapat diimplementasikan secara detail dengan menggunakan *input* langsung dari *camera* atau *device* lain. Penambahan kategori produk juga perlu dilakukan sehingga model dapat dengan baik mengenali beberapa jenis produk dalam aplikasi kasir. Selain itu, penelitian selanjutnya juga perlu mempertimbangkan model pra-terlaltih yang lain sebagai perbandingan.

REFERENSI

[1] A. Đordevic, Y. Klochkov, S. Arsovski, N. Stefanovic, L. Shamina, and A. Pavlovic, "The impact of ict support and the efqm criteria on sustainable business excellence in higher education institutions," *Sustainability (Switzerland)*, vol. 13, no. 14, Jul. 2021, doi: 10.3390/su13147523.

[2] A. Qosasi, E. Maulina, M. Purnomo, A. Muftiadi, E. Permana, and F. Febrian, "The impact of Information and Communication Technology capability on the competitive advantage of small businesses," *International Journal of Technology*, vol. 10, no. 1, pp. 167–177, 2019, doi: 10.14716/ijtech.v10i1.2332.

[3] T. Gajewska, D. Zimon, G. Kaczor, and P. Madzík, "The impact of the level of customer satisfaction on the quality of e-commerce services," *International Journal of Productivity and Performance Management*, vol. 69, no. 4, pp. 666–684, Apr. 2020, doi: 10.1108/IJPPM-01-2019-0018.

[4] J. Davies and Y. Wang, "Physically Unclonable Functions (PUFs): A New Frontier in Supply Chain Product and Asset Tracking," *IEEE Engineering Management Review*, vol. 49, no. 2, pp. 116–125, Apr. 2021, doi: 10.1109/EMR.2021.3069366.

[5] X. Zhu, M. Liu, Y. Zhao, L. Dong, M. Hui, and L. Kong, "Product detection based on CNN and transfer learning," SPIE-Intl Soc Optical Eng, Sep. 2019, p. 69. doi: 10.1117/12.2526236.

[6] M. Haseeb, H. I. Hussain, B. Ślusarczyk, and K. Jermsittiparsert, "Industry 4.0: A solution towards technology challenges of sustainable business performance," *Soc Sci*, vol. 8, no. 5, May 2019, doi: 10.3390/socsci8050154.

[7] N. Fitrianingsih Hasan *et al.*, "Deteksi dan Klasifikasi Penyakit Pada Daun Kopi Menggunakan Yolov7," *Sistem Informasi dan Komputer)*, vol. 12, pp. 30–35, doi: 10.32736/sisfokom.v12i1.1545.

[8] M. Akbar, A. S. Purnomo, and S. Supatman, "Multi-Scale Convolutional Networks untuk Pengenalan Rambu Lalu Lintas di Indonesia," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 3, pp. 310–315, Dec. 2022, doi: 10.32736/sisfokom.v11i3.1452.

[9] R. Sujatha, J. M. Chatterjee, N. Z. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocess Microsyst*, vol. 80, Feb. 2021, doi: 10.1016/j.micpro.2020.103615.

[10] D. Bhatt *et al.*, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electronics (Switzerland)*, vol. 10, no. 20. MDPI, Oct. 01, 2021. doi: 10.3390/electronics10202470.

[11] F. A. Breve, "COVID-19 detection on Chest X-ray images: A comparison of CNN architectures and ensembles[Formula presented]," *Expert Syst Appl*, vol. 204, Oct. 2022, doi: 10.1016/j.eswa.2022.117549.

[12] J. Huang *et al.*, "BM-Net: CNN-Based MobileNet-V3 and Bilinear Structure for Breast Cancer Detection in Whole Slide Images," *Bioengineering*, vol. 9, no. 6, Jun. 2022, doi: 10.3390/bioengineering9060261.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i2.1659, Copyright ©2023
Submitted : March 28, 2023, Revised : May 6, 2023, Accepted : May 12, 2023, Published : July 1, 2023

[13] B. B. M. Wantania, S. R. U. A. Sompie, and F. D. Kambey, "Penerapan Pendeteksian Manusia Dan Objek Dalam Keranjang Belanja Pada Antrian Di Kasir," *Jurnal Teknik Informatika*, vol. 15, no. 2, pp. 101–108, Jul. 2020, doi: 10.35793/JTI.15.2.2020.29004.

[14] P. R. Aningtiyas, A. Sumin, and S. Wirawan, "Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih: Array," *Jurnal Ilmiah Komputasi*, vol. 19, no. 3, pp. 421–430, Sep. 2020, doi: 10.32409/JIKSTIK.19.3.68.

[15] S. Srivastava, A. Gangwar, R. Mishra, and S. Singh, "Sign Language Recognition System Using TensorFlow Object Detection API," in *Communications in Computer and Information Science*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 634–646. doi: 10.1007/978-3-030-96040-7_48.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i2.1659, Copyright ©2023
Submitted : March 28, 2023, Revised : May 6, 2023, Accepted : May 12, 2023, Published : July 1, 2023