

# sensor flow 1

*by* Tes 1

---

**Submission date:** 27-Mar-2023 04:37PM (UTC+0700)

**Submission ID:** 2047894912

**File name:** sensor\_flow\_no\_format.docx (658.66K)

**Word count:** 2310

**Character count:** 14387

**Abstract**— One of the business sectors of trade that has grown in importance to economic development is the supermarket. Although employing object detection with camera input can speed up transactions because it enables them to recognize several objects, not many supermarkets do so. In this study, a pre-trained SSD MobileNet model is used to create a cashier application utilizing the Tensorflow Object detection API. According to the study's findings, the process of product object recognition has an accuracy rate of 82.9%, a precision rate of 97.5%, and a recall rate of 84.7%. The time it takes to recognize a product ranges from 365 ms to 827 ms, on average (0.69 seconds).

**Keywords**— Object detection, MobileNet, Tensorflow

**Abstrak**— Dalam perkembangannya supermarket merupakan salah satu komersial dalam bidang perdagangan yang memegang peranan penting dalam perkembangan perekonomian. Namun, belum banyak supermarket yang menggunakan deteksi objek menggunakan citra gambar dengan input camera, dimana hal ini dapat mempercepat proses transaksi karena memungkinkan dapat mengenali objek lebih dari 1. Penelitian ini bertujuan mengimplementasikan aplikasi kasir menggunakan Tensorflow Object detection API dengan menggunakan model pra-terlatih SSD MobileNet. Hasil penelitian menunjukkan 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 factor 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 terbaca dengan jelas, hal tersebut juga menjadi kelemahan lain dari teknologi tersebut [4]. Hal tersebut tentu akan mengakibatkan proses transaksi yang tidak lagi cepat dan efisien. Peralihan menjadi industry 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 [5].

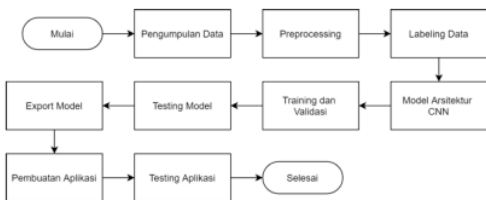
Salah satu contoh penerapan teknologi otomatisasi yaitu pendeteksian dan klasifikasi penyakit daun kopi berdasarkan tingkat keparahannya melalui pengenalan citra daun kopi [6]. Klasifikasi penyakit daun kopi secara otomatis tersebut menggunakan metode Yolov7 terhadap 1664 data yang terbagi menjadi 4 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 yang sisanya digunakan sebagai data testing yang diolah menggunakan algoritma *Convolutional Neural Network* (CNN). Hasil penelitian memperoleh akurasi sebesar 99% [7]. Melalui penelitian tersebut mampu membantu para pengendara dalam memahami setiap rambu lalu lintas yang ada.

Dari beberapa penelitian serupa tentang pengenalan citra [6], [7], diperoleh hasil bahwa kinerja *deep learning* menggunakan algoritma CNN lebih baik dibandingkan algoritma *machine learning*. CNN sendiri memiliki beberapa varian yang dibedakan dari jenis arsitekturnya, di mana antara 1 arsitektur dengan yang lain memiliki kinerja model yang berbeda-beda. 5 arsitektur CNN terbaik terplih melalui perbandingan terhadap 21 arsitektur CNN dalam menentukan paru-paru yang terkena covid-19 melalui citra X-Ray [8]. 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 [9].

Arsitektur CNN ini juga dapat digunakan dalam menghitung jumlah orang yang melakukan antrian di kasir serta dapat menganalisa lamanya waktu yang dibutuhkan tiap pelanggan dalam melakukan pembayaran [10]. 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-trlatih 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 [11]. Berdasarkan beberapa penelitian terdahulu, penulis memilih arsitektur pra-trlatih MobileNet dalam mengenali produk di kasir karena model ini memiliki kinerja yang baik serta mampu mengenali objek lebih dari 1. Selain itu penggunaan TensorFlow Object Detection API dengan model pra-trlatih 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 implementasi di aplikasi. Diagram alur penelitian tersaji di gambar 1 dibawah ini.

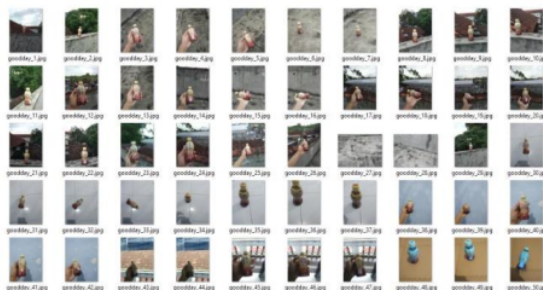


Gambar 1. Diagram alur urutan penelitian

22

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 menunjukkan beberapa contoh data yang telah dikumpulkan.



Gambar 2. Contoh kumpulan data objek

Dataset yang digunakan dikelompokkan menjadi 2, 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 labeling yang tersedia secara open source. Label untuk setiap gambar tersebut kemudian disimpan dalam bentuk file .xml.

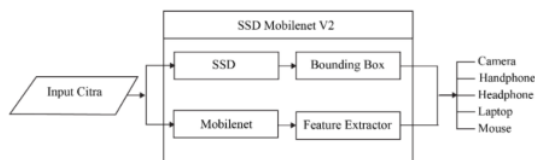
C. Model Arsitektur CNN

Pada tahap ini dilakukan pembuatan model arsitektur CNN, dimana akan menggunakan arsitektur pra-trlatih SSD MobileNetV2. Untuk menggunakan model pra-trlatih diperlukan file checkpoint dari model pra-trlatih, 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 untuk mengklasifikasi objek yang telah terdeteksi oleh SSD dengan Bounding Box. Selanjutnya, penggabungan SSD dan MobileNet akan meningkatkan proses pembuatan aplikasi deteksi objek. Pada penelitian Aningtyas 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 [11]

17

### 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 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* dan *total loss* dari proses pelatihan terdapat pada gambar 5 dan 6 dibawah ini



Gambar 5. Grafik *classification loss*



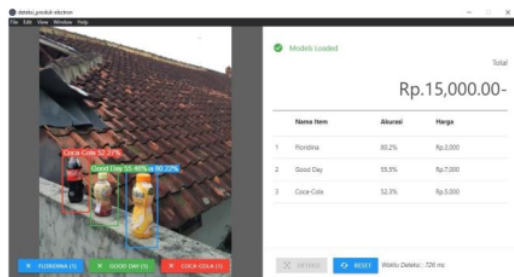
Gambar 6. Grafik *total loss*

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 sebatas pada prototipe, dimana fungsionalitas dari deteksi input gambar berjalan sesuai. 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 4 berikut.



Gambar 4. 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 pegujian 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 kece<sup>15</sup>an waktu pendeteksian menunjukkan nilai tidak konsisten, hal tersebut dapat dipengaruhi oleh banyak faktor seperti kualitas gambar input, spesifikasi hardware dan lain sebagainya.

D. Pengujian akurasi deteksi

Proses pengujian akurasi dilakukan dengan 17 gambar yang <sup>20</sup>liri 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		
test_6.jpg	3	1	4		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 <sup>21</sup>didapatkan hasil *confusion matrix* yang disajikan pada tabel 2 dibawah 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. <sup>11</sup>KESIMPULAN

Dari hasil penelitian yang telah kami lakukan menunjukan bahwa SSDS MobileNet dan Tensorflow Object Detection API dapat mendeteksi citra produk dengan baik. 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 kedepan diharapkan dapat mengimplementasikan aplikasi 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 praterlaltih yang lain sebagai perbandingan.

# sensor flow 1

## ORIGINALITY REPORT

11%

SIMILARITY INDEX

8%

INTERNET SOURCES

5%

PUBLICATIONS

4%

STUDENT PAPERS

## PRIMARY SOURCES

1	Submitted to UIN Sultan Syarif Kasim Riau Student Paper	2%
2	Submitted to Universitas Brawijaya Student Paper	1%
3	<a href="http://ojs.amikom.ac.id">ojs.amikom.ac.id</a> Internet Source	1%
4	Submitted to Informatics Education Limited Student Paper	1%
5	Submitted to Universitas Islam Indonesia Student Paper	1%
6	<a href="http://jurnal.unimed.ac.id">jurnal.unimed.ac.id</a> Internet Source	1%
7	Firlan Firlan. "Penggunaan Algoritma TOPSIS Dalam Pengukuran Modal Manusia", Jurnal Sisfokom (Sistem Informasi dan Komputer), 2022 Publication	<1%
8	MASAKI MURATA, MAKOTO NAGAO. "An Estimate of Referent of Nouns in Japanese	<1%

# Sentences with Referential Property of Nouns", Journal of Natural Language Processing, 1996

Publication

9

Sultoni Sultoni, Rudy Hariyanto. "Pengenalan Wajah Secara Real Time Menggunakan Metode Camshift dan Operator Erosi Berdasarkan Citra Wajah", JICTE (Journal of Information and Computer Technology Education), 2017

Publication

<1 %

10

[beritasatumediald.bz](http://beritasatumediald.bz)

Internet Source

<1 %

11

[ratuhidayanthi.blogspot.com](http://ratuhidayanthi.blogspot.com)

Internet Source

<1 %

12

[repository.unpas.ac.id](http://repository.unpas.ac.id)

Internet Source

<1 %

13

[slotqq1x2.net](http://slotqq1x2.net)

Internet Source

<1 %

14

[doaj.org](http://doaj.org)

Internet Source

<1 %

15

[jurnalpenyakitdalam.ui.ac.id](http://jurnalpenyakitdalam.ui.ac.id)

Internet Source

<1 %

16

[repository.its.ac.id](http://repository.its.ac.id)

Internet Source

<1 %

[seminar.uny.ac.id](http://seminar.uny.ac.id)

17	Internet Source	<1 %
18	<a href="http://www.neliti.com">www.neliti.com</a> Internet Source	<1 %
19	<a href="http://www.scilit.net">www.scilit.net</a> Internet Source	<1 %
20	<a href="http://www.slideshare.net">www.slideshare.net</a> Internet Source	<1 %
21	Toni Arifin, Daniel Ariesta. "PREDIKSI PENYAKIT GINJAL KRONIS MENGGUNAKAN ALGORITMA NAIVE BAYES CLASSIFIER BERBASIS PARTICLE SWARM OPTIMIZATION", Jurnal Tekno Insentif, 2019 Publication	<1 %
22	<a href="http://doku.pub">doku.pub</a> Internet Source	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off