



## Rancang Bangun Sistem Pendeteksi Aneka Ragam Buah Menggunakan MobileNetv2

Ario Prima<sup>1✉</sup>

<sup>1</sup>Universitas Singaperbangsa Karawang

[ario.prima19051@student.unsika.ac.id](mailto:ario.prima19051@student.unsika.ac.id)

### Abstrak

Teknologi saat ini berkembang dengan sangat cepat, tidak terkecuali kan dalam bidang Artificial Intelligence (AI) atau yang lebih kita kenal dengan sebutan kecerdasan bawaan. Salah satu metode yang digunakan dalam pembuatan AI ini adalah Deep Learning. Deep learning menggunakan pendekatan dalam pembelajaran mesin menggunakan jaringan saraf tiruan dalam mempelajari pola-pola yang kompleks, keunggulan dari deep learning adalah kemampuannya dalam mempelajari dan memproses data yang sangat besar dan kompleks. Dalam penelitian kali ini algoritma deep learning yang akan digunakan adalah Convolutional Neural Network (CNN) yang mana telah menjadi fondasi untuk berbagai aplikasi berbasis AI yang sukses. CNN sendiri saat ini memiliki beberapa arsitektur, pada penelitian kali ini arsitektur yang digunakan adalah SSD MobileNetv2. Pada sistem kali ini akan menggunakan library untuk melatih model nya menggunakan TensorFlow, sistem ini dibangun guna membantu para penyandang tunanetra yang memiliki keterbatasan kemampuan dalam penglihatannya untuk mampu mengenali aneka ragam buah-buahan.

**Kata kunci:** Artificial Intelligence, Deep Learning, Convolutional Neural Network, SSD MobileNetv2, TensorFlow.

*JSISFOTEK is licensed under a Creative Commons 4.0 International License.*



### 1. Pendahuluan

Perkembangan teknologi yang semakin pesat akhir-akhir ini telah memberikan dampak yang cukup signifikan terdapat berbagai bidang, salah satunya yaitu bidang teknologi kecerdasan buatan atau yang biasa dikenal dengan sebutan Artificial Intelligence (AI). Salah satu metode yang digunakan dalam pembuatan sistem berbasis AI ini adalah Deep Learning, deep learning menggunakan pendekatan jaringan syaraf tiruan dalam mempelajari pola-pola yang kompleks. Salah satu algoritma dalam metode ini adalah Convolutional Neural Network (CNN). CNN sendiri memiliki kemampuan yang sangat baik dalam mengekstraksi fitur-fitur gambar. CNN sendiri terdiri dari beberapa arsitektur yang salah satunya yaitu SSD *MobileNetv2* yang mana merupakan salah satu arsitektur CNN yang memiliki kemampuan yang sangat baik dalam mendeteksi objek pada gambar secara akurat dan cepat. Arsitektur ini dapat bekerja dengan menggunakan perangkat yang memiliki sumber daya yang ringan. Dalam pembuatan sistem deteksi aneka ragam buah-buahan ini, akan menggunakan salah satu library untuk melatih model nya yang mana menggunakan TensorFlow, hal ini karena TensorFlow memiliki kompatibilitas yang luas dan juga banyaknya user yang menggunakan library ini sehingga dapat membantu apabila terjadi error maka akan lebih mudah untuk menemukan solusi nya.

Sistem pendeteksi aneka ragam buah-buahan ini akan dibuat secara real-time dengan keluaran atau output berupa suara. Sistem pendeteksi ini akan dibangun pada perangkat Raspberry Pi 3. Diharapkan dengan adanya sistem ini akan membantu para penyandang tunanetra yang memiliki kekurangan dalam kemampuan penglihatan dapat mengenali berbagai aneka ragam jenis buah-buahan.

Terdapat beberapa penelitian yang terkait mengenai implementasi sistem pendeteksi objek menggunakan arsitektur MobileNet ini, di antaranya sebagai berikut:

1. Penelitian oleh Qian Xiang, Xiaodan Wang, Rui Li, Guoling Zhang, Jie Lai, dan Qingshuang Hu pada tahun 2019 mengenai klasifikasi gambar buah untuk sistem pemetik robotik menggunakan MobileNetv2 dengan tingkat akurasi rata-rata mencapai 85,12% [1].
2. Penelitian yang dilakukan oleh Ejaz Kha, Muhammad Zia Ur Rehman, Fawed Ahmed, dan Muhammad Attique Khan pada tahun 2021 mengenai Deteksi Penyakit pada buah jeruk yang terbagi menjadi enam kelas dengan menggunakan metode MobileNetv2 SqueezeNet yang menghasilkan tingkat akurasi mencapai 96% [2].
3. Penelitian yang dilakukan oleh Zulfahmi Syahputra pada tahun 2023 mengenai klasifikasi jenis-jenis apel dengan menggunakan metode SSD MobileNetV2 yang mana penelitian ini menghasilkan tingkat akurasi sebesar 95,8% [3].

*Artificial Intelligence* atau biasa disebut dengan Sistem Kecerdasan Buatan adalah suatu cabang ilmu dari bidang ilmu komputer yang berusaha untuk menanamkan kecerdasan layaknya seperti manusia kepada mesin[4]. Kecerdasan buatan juga merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan suatu hal, dalam pandangan manusia adalah cerdas[5].

*Machine learning* adalah salah satu pendekatan dalam bidang *artificial intelligence* yang berfokus pada kemampuan mesin untuk belajar dan meningkatkan kinerjanya dari pengalaman tanpa secara eksplisit di program secara khusus. Definisi pengalaman disini ialah informasi sebelumnya yang telah tersedia dan bisa dijadikan data pembelajaran[6].

*Deep learning* merupakan suatu metode yang digunakan dalam pengembangan kecerdasan buatan. Metode ini berdasarkan pada pendekatan jaringan syaraf tiruan dalam mempelajari pola-pola data yang kompleks. *Deep learning* disebut sebagai *Deep* (dalam) karena struktur dan jumlah jaringan saraf pada algoritmanya sangat banyak bisa mencapai lapisan hingga ratusan lapisan[7]. Teknologi *deep learning* ini salah satu teknologi yang paling populer untuk mengenali suatu keinginan atau objek yang memiliki tingkat keakuratan lebih tinggi dibanding dengan metode mesin sebelumnya[8]. Saat ini *deep learning* telah membawa terobosan dalam berbagai bidang *artificial intelligence*, *deep learning* dapat menentukan struktur yang begitu rumit di dalam sebuah dataset yang sangat besar dengan menggunakan algoritma *backpropagation*[9]. *Deep learning* memiliki beberapa algoritma, akan tetapi yang terkenal adalah algoritma *Convolutional Neural Network* (CNN) yang mana pada umumnya terkenal akan kemampuan deteksi dan klasifikasi objeknya.

*Convolutional Neural Network* atau yang biasa disingkat dengan CNN ini merupakan salah satu jenis algoritma *deep learning* yang memiliki kemampuan menerima *input* berupa gambar atau citra untuk dipelajari dan bisa menentukan aspek atau objek apa saja yang ada di dalam sebuah gambar dan dapat digunakan atau diolah oleh mesin untuk “belajar” mengenali gambar dan dapat membedakan suatu gambar dengan gambar lainnya[10]. CNN sendiri memiliki beberapa arsitektur seperti ResNet, AlexNet, MobileNet, dan lain-lain. Penerapan CNN pada umumnya menggunakan TensorFlow[11].

*Single Shot MultiBox Detector* atau SSD adalah salah satu algoritma yang populer dalam deteksi objek. SSD adalah pendeteksi yang cukup cepat dan dapat digunakan pada *real-time* video. Deteksi dijelaskan oleh kotak pembatas atau *bounding box* dan untuk setiap *bounding box* model juga memprediksi kelas[12]. Cara kerja pada algoritma SSD didasarkan pada *feed-forward convolutional network*, yaitu pada suatu objek akan ditentukan *bounding box* yang berukuran tetap dan pada area *bounding box* tersebut akan menampilkan nilai skor masing-masing pada setiap kelas objek[13]. Bagian lapisan fitur pada *convolutional* ditambahkan ke ujung *base network* oleh SSD sebanyak enam layer konvolusional ekstra yang berfungsi memprediksi dengan aspek rasio berbeda[14].

MobileNet merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan akan sumber komputasi berlebih[15]. Perbedaan arsitektur MobileNet dengan arsitektur CNN adalah penggunaan pada layer konvolusi yang dimana ketebalan filternya sesuai dengan ketebalan dari *input* gambar[16]. Versi kedua dari MobileNet di rilis pada april 2017 dan diberi nama MobileNetv2[17]. Pada versi ini terdapat dua fitur tambahan yang terdiri dari *Linier Bottleneck* dan *Shortcut Connections* antar *Bottlenecks*[18].

TensorFlow merupakan salah satu *framework deep learning* dan juga salah satu *library* untuk data *science* yang bersifat *open-source* yang dikembangkan oleh peneliti dari tim google[19]. TensorFlow dikembangkan dengan tujuan untuk melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam. TensorFlow menggabungkan aljabar komputasi, teknik pengoptimalan kompilasi untuk mempermudah penghitungan ekspresi matematis, sehingga dapat mengurangi masalah waktu yang dibutuhkan untuk melakukan perhitungan[20].

## **2. Metode Penelitian**

Metode penelitian pada sistem pendeteksi aneka ragam buah-buahan ini adalah sebagai berikut:

### **1. Pengumpulan Data**

Pada penelitian ini dataset yang digunakan untuk melakukan proses pelatihan bersifat umum dan dapat digunakan oleh siapapun. Dataset ini didapatkan dari sebuah website yang bernama robotflow, dengan nama *user* pemiliknya adalah zakirkx12

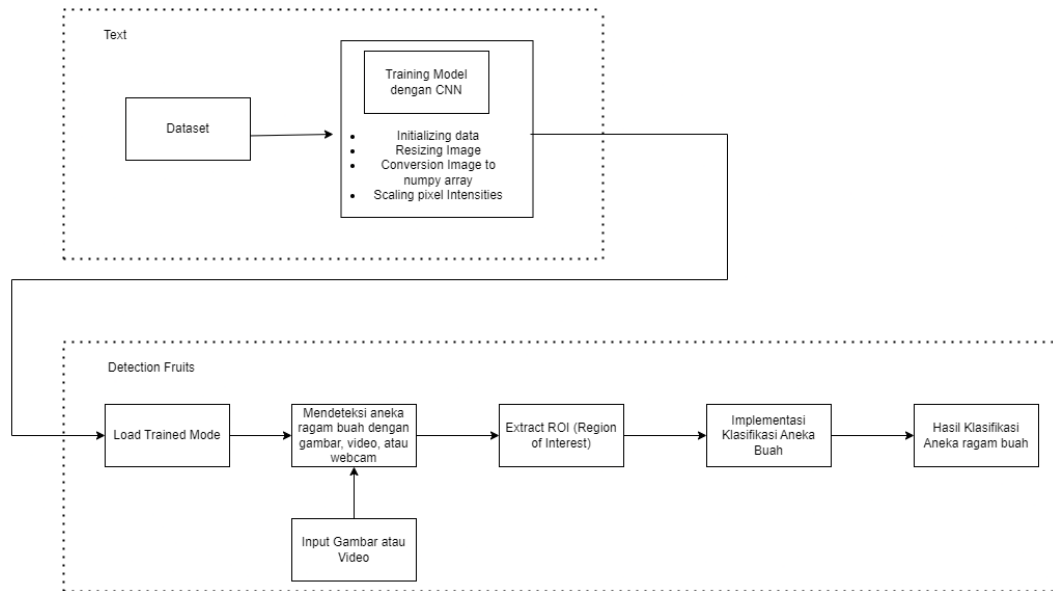
### **2. Pengolahan data awal**

Pada bagian ini tahapan *preprocessing* digunakan untuk inialisasi data dan daftar tabel, melakukan *resize gambar*, mengkonversi ke format *array*, dan melakukan *scaling* pada *pixel intensities*, selanjutnya melakukan *pre-processed image* dan label terkait pada data dan list tabel masing-masing, serta memastikan proses training data dalam format NumPy.

### **3. Model yang diusulkan**

Pada penelitian ini menggunakan metode *scikit-learns convenience* untuk membagi dataset menjadi tiga bagian, yaitu 80% *training*, 30% *validation*, dan 30% *test*. Metode yang digunakan adalah *deep learning* dengan algoritma *Convolutional Neural Network* (CNN) menggunakan arsitektur SSD *MobileNetv2*, yang mana *library* yang digunakan dalam proses pelatihannya yaitu TensorFlow dengan menggunakan Google

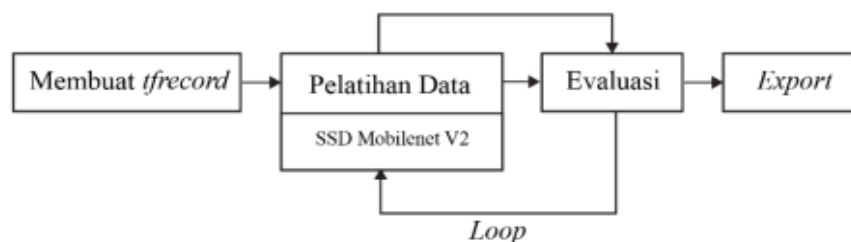
Collaboratory atau biasa disebut dengan Colab sebagai lingkungan untuk pengembangannya yang berbasis *cloud*.



Gambar 1. Model yang diusulkan

#### 4. Pengembangan model arsitektur

Pengembangan model arsitektur SSD MobileNetv2 menggunakan library TensorFlow Object Detection API. Berikut ini adalah bagan, untuk ini berikut ini adalah skenario pengembangan arsitektur menggunakan TensorFlow Object Detection API yang dapat dilihat seperti pada gambar dibawah ini:



Gambar 2. Skenario Pengembangan Arsitektur

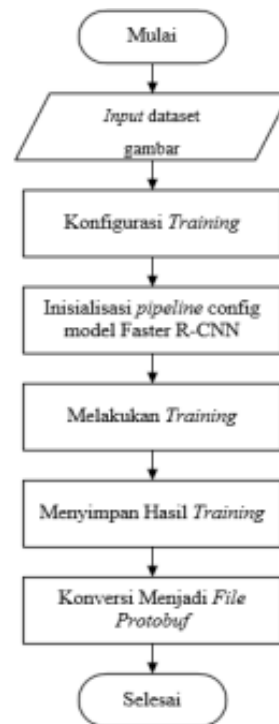
#### 5. Membuat TFRecord

Pembuatan TFRecord dapat mengurangi penggunaan penyimpanan dan mempercepat waktu pembacaan data, karena menggunakan format serialisasi biner yang lebih kecil dibandingkan dengan format CSV atau TS. Ada dua tahap dalam pembuatan TFRecord, yaitu konversi data mentah CSV atau TSV menjadi format protokol TensorFlow (protobuf), kemudian menyimpannya dalam file TFRecord.

#### 6. Pelatihan Data

Setelah menyiapkan dataset dan membuat TFRecord, tahapan selanjutnya adalah konfigurasi model. Model bisa didapatkan dari GitHub TensorFlow Object Detection Model Zoo, pada sistem ini menggunakan SSD MobileNetv2 yang sudah terlatih dengan dataset COCO. Konfigurasi model akan mencakup detail seperti jumlah kelas yang ingin dideteksi, ukuran gambar input, dan parameter lain yang dibutuhkan.

Setelah konfigurasi model ditentukan, langkah selanjutnya adalah konfigurasi pipeline pelatihan yang mana biasanya telah tersedia dengan nama "pipeline.config". Dimana file ini mencakup tentang file *path* ke TFRecord, file ke labelmap, serta parameter pelatihan seperti jumlah epoch, *learning rate*, dan algoritma pengoptimalan yang digunakan selama pelatihan. Pada proses pelatihan akan memanfaatkan Google Collaboratory atau biasa disebut dengan Colab yang mana menyediakan fasilitas GPU untuk mempercepat proses pelatihan model SSD MobileNetv2. Berikut adalah *flowchart* alur proses pelatihan yang dapat dilihat pada gambar dibawah ini.



Gambar 3. Alur proses pelatihan

### 3. Hasil dan Pembahasan

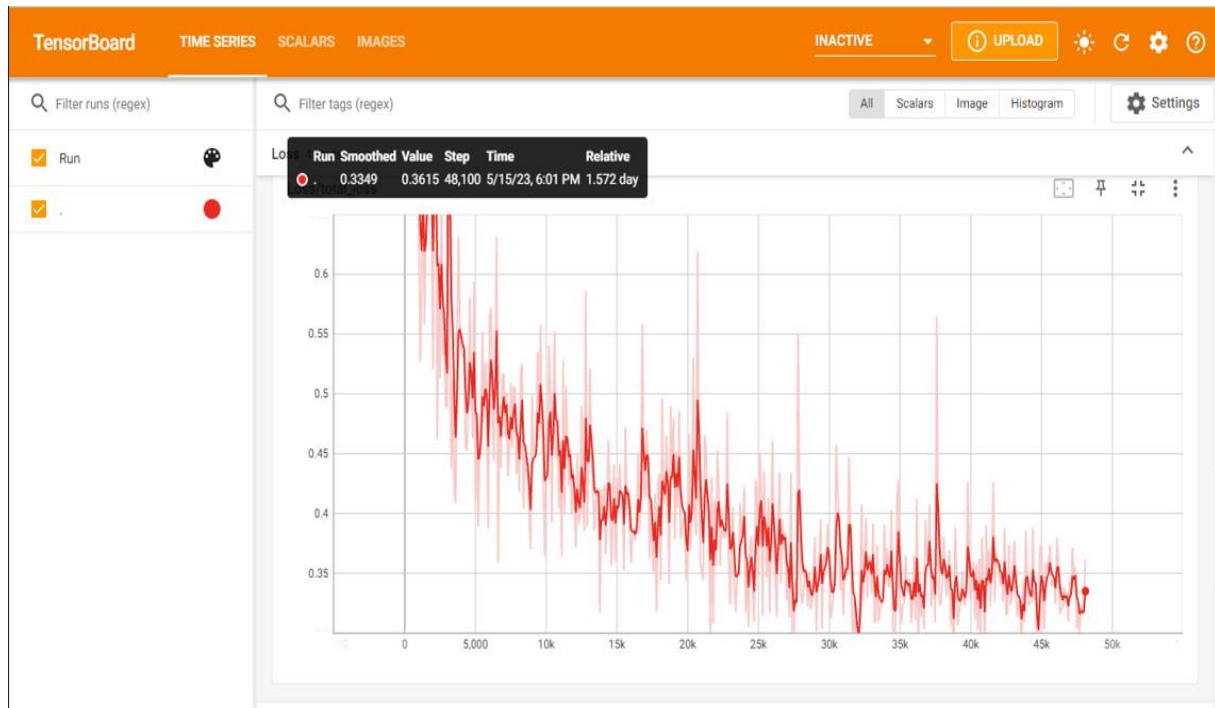
Setelah dilakukan proses pelatihan menggunakan arsitektur SSD MobileNetv2 akan menghasilkan folder *training* yang mana akan di generate, sehingga model dapat digunakan untuk melakukan proses klasifikasi dan deteksi. Saat proses pelatihan model ada beberapa tahap yang dilalui seperti *initializing data*, *resizing image*, dan konversi gambar ke NumPy array. Dari hasil pelatihan akan didapatkan hasil *loss* dan *learning rate* yang dapat dilihat pada Gambar 4 dan Gambar 5

```

INFO:tensorflow:Step 1100 per-step time 0.236s
I0530 05:32:46.783202 139936307005248 model_lib_v2.py:705] Step 1100 per-step time 0.236s
INFO:tensorflow:{'Loss/classification_loss': 0.16795506,
...
'Loss/localization_loss': 0.05584059,
'Loss/regularization_loss': 0.08570312,
'Loss/total_loss': 0.3094988,
'learning_rate': 0.0559997}
I0530 05:32:46.783516 139936307005248 model_lib_v2.py:708] {'Loss/classification_loss': 0.16795506,
'Loss/localization_loss': 0.05584059,
'Loss/regularization_loss': 0.08570312,
'Loss/total_loss': 0.3094988,
'learning_rate': 0.0559997}
INFO:tensorflow:Step 1200 per-step time 0.231s
I0530 05:33:09.837364 139936307005248 model_lib_v2.py:705] Step 1200 per-step time 0.231s
INFO:tensorflow:{'Loss/classification_loss': 0.14283186,
'Loss/localization_loss': 0.057760283,
'Loss/regularization_loss': 0.08566751,
'Loss/total_loss': 0.28625965,
'learning_rate': 0.0586664}
I0530 05:33:09.837684 139936307005248 model_lib_v2.py:708] {'Loss/classification_loss': 0.14283186,
'Loss/localization_loss': 0.057760283,
'Loss/regularization_loss': 0.08566751,
'Loss/total_loss': 0.28625965,
'learning_rate': 0.0586664}

```

Gambar 4. Training Model

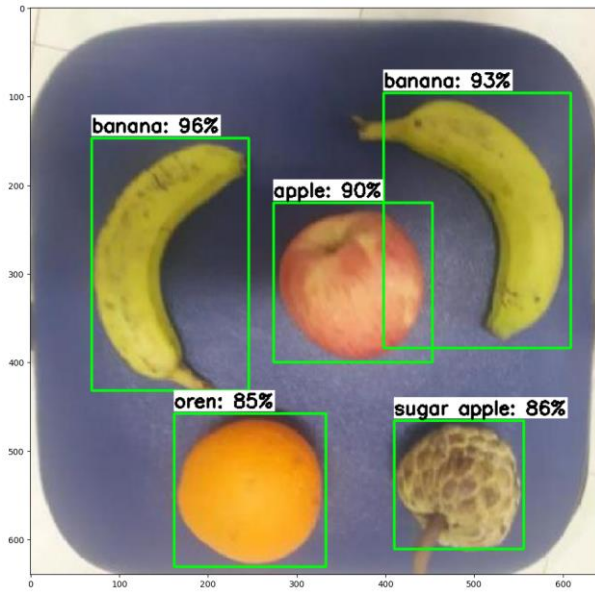
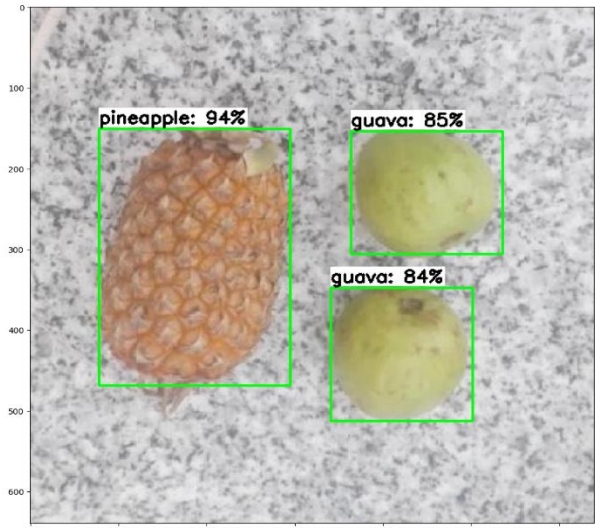


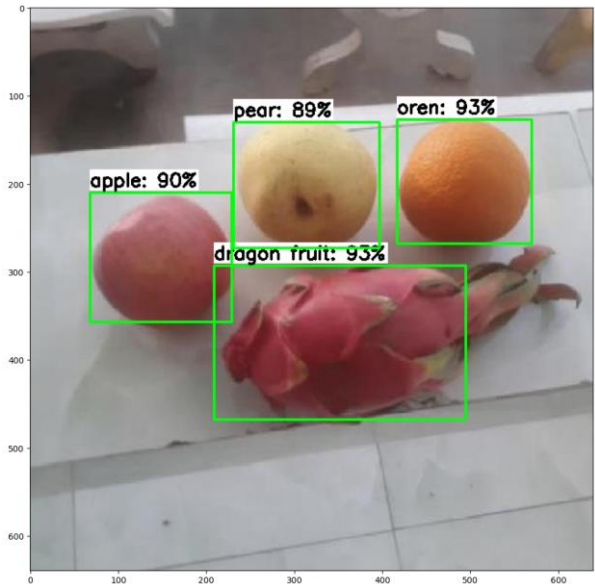
Gambar 5. Hasil Grafik Total loss

Selanjutnya hasil proses pengujian model, proses inferensi akan menghasilkan kotak pembatas yang menunjukkan posisi dan ukuran objek yang terdeteksi dalam gambar, berikut adalah tampilan gambar yang dapat dilihat pada tabel 1.

Tabel 1. Hasil Pengujian

No	Pengujian	Gambar
1	<p><b>Deteksi Uji:</b>  <i>Apple, Banana, Sugar Apple, Dragon Fruit</i></p> <p><b>Akurasi:</b>  <i>Apple: 92% dan 88%</i>  <i>Banana: 91% dan 89%</i>  <i>Sugar Apple: 85%</i>  <i>Dragon fruit: 85%</i></p>	<p>The image shows a collection of fruits on a light-colored surface. Green bounding boxes are drawn around each fruit, and labels with accuracy percentages are placed next to them. The labels are: 'apple: 92%' (top center), 'banana: 91%' (top left), 'dragon fruit: 90%' (top right), 'sugar apple: 85%' (center), 'apple: 88%' (bottom left), and 'banana: 89%' (bottom right).</p>

2	<p><b>Deteksi Uji:</b> <i>Banana, Apple, Sugar Apple, Oren</i></p> <p><b>Akurasi:</b> <i>Banana: 96% dan 93%</i> <i>Sugar Apple: 86%</i> <i>Apple: 90%</i> <i>Oren: 85%</i></p>	 <p>The image shows a top-down view of four fruits on a blue background. Two bananas are positioned at the top, an apple is in the center, and an orange is at the bottom. A green bounding box is drawn around each fruit. The labels and their corresponding accuracies are: banana: 96% (left banana), banana: 93% (right banana), apple: 90% (center apple), oren: 85% (bottom orange), and sugar apple: 86% (bottom right, which appears to be a sugar apple).</p>
3	<p><b>Deteksi Uji:</b> <i>Pineapple, guava</i></p> <p><b>Akurasi:</b> <i>Pineapple: 94%</i> <i>Guava: 85% dan 84%</i></p>	 <p>The image shows a top-down view of three fruits on a speckled surface. A pineapple is on the left, and two guavas are on the right. Green bounding boxes are drawn around each fruit. The labels and their corresponding accuracies are: pineapple: 94% (left pineapple), guava: 85% (top right guava), and guava: 84% (bottom right guava).</p>

4	<p><b>Deteksi Uji:</b>  <i>Dragon Fruit, apple, pear, oren</i></p> <p><b>Akurasi:</b>  <i>Dragon fruit:</i> 93%  <i>Apple:</i> 90%  <i>Pear:</i> 89%  <i>Oren:</i> 93%</p>	
---	--	--

#### 4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dengan menggunakan model *Convolutional Neural Network* (CNN) dengan arsitektur *MobileNetV2*, program telah berhasil mendeteksi objek pada delapan kelas yaitu *Guava, Pear, Apple, Sugar Apple, Banana, Oren, Pineapple*, dan *Dragon fruits* yang menghasilkan tingkat akurasi yang baik. Model yang dihasilkan menampilkan hasil yang baik saat melakukan *training* yang mana tidak mengalami *overfitting* hal ini karena tingkat total *loss* yang rendah.

#### Daftar Rujukan

- [1] Q. Xiang, X. Wang, R. Li, G. Zhang, J. Lai, and Q. Hu, "Fruit image classification based on Mobilenetv2 with transfer learning technique," in *Proceedings of the 3rd international conference on computer science and application engineering*, 2019, pp. 1–7.
- [2] E. Khan, M. Z. U. Rehman, F. Ahmed, and M. A. Khan, "Classification of diseases in citrus fruits using SqueezeNet," in *2021 International Conference on Applied and Engineering Mathematics (ICAEM)*, IEEE, 2021, pp. 67–72.
- [3] Z. Syahputra, "PENERAPAN SSD-MOBILENET DALAM IDENTIFIKASI JENIS BUAH APEL," *Indonesian Journal of Education And Computer Science*, vol. 1, no. 1, pp. 1–6, 2023.
- [4] W. M. Baihaqi, F. Sulistiyana, and A. Fadholi, "PENGENALAN ARTIFICIAL INTELLIGENCE UNTUK SISWA DALAM MENGHADAPI DUNIA KERJA DI ERA REVOLUSI INDUSTRI 4.0," *RESWARA: Jurnal Pengabdian Kepada Masyarakat*, vol. 2, no. 1, pp. 79–88, 2021.
- [5] K. Umi, "Pengenalan Kecerdasan Buatan (Artificial Intelligence) Kepada Para Remaja," *Universitas Bina Dharma*, 2022.
- [6] J. Pujoseno, "Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Alat Tulis," 2018.
- [7] W. Dadang, "Memahami Kecerdasan Buatan berupa Deep Learning," *Mach Learn*, 2018.
- [8] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and trends® in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
- [9] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," *Large-scale kernel machines*, vol. 34, no. 5, pp. 1–41, 2007.
- [10] P. Arfienda, "Materi Pendamping Memahami Convolutional Neural Networks Dengan Tensorflow." 2019.
- [11] Nurfita, Royani Darma, and S. Gunawan Ariyanto, "Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari," 2018.
- [12] C. G. W. Pramana, D. C. Khrisne, and N. P. Sastra, "Rancang Bangun Object Detection Pada Robot Soccer Menggunakan Metode Single Shot Multibox Detector (SSD MobileNetV2)," *Jurnal SPEKTRUM Vol*, vol. 8, no. 2, 2021.



- [13] S. Fuady, N. Nehru, and G. Anggraeni, "Deteksi Objek Menggunakan Metode Single Shot Multibox Detector Pada Alat Bantu Tongkat Tunanetra Berbasis Kamera," *Journal of Electrical Power Control and Automation (JEPCA)*, vol. 3, no. 2, pp. 39–43, 2020.
- [14] I. M. G. Sunarya *et al.*, "Deteksi arteri karotis pada citra ultrasound b-mode berbasis convolution neural network single shot multibox detector," *Jurnal Teknologi dan Sistem Komputer*, vol. 7, no. 2, pp. 56–63, 2019.
- [15] S. Apendi, C. Setianingsih, and M. W. Paryasto, "Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode Single Shot Multibox Detector," *eProceedings of Engineering*, vol. 10, no. 1, 2023.
- [16] V. A. Utama, S. A. Wibowo, and R. Rahmania, "Investigasi Pengaruh Step Training pada Metode Single Shot Multibox Detector untuk Marker dalam Teknologi Augmented Reality," *Jurnal Ilmiah Fifo*, vol. 12, no. 1, pp. 1–11, 2020.
- [17] I. Mudzakir and T. Arifin, "Klasifikasi Penggunaan Masker dengan Convolutional Neural Network Menggunakan Arsitektur MobileNetv2," *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, vol. 12, no. 1, pp. 76–81, 2022.
- [18] R. O. Ekoputris, "MobileNet: Deteksi Objek pada Platform Mobile," *Medium, May*, vol. 9, 2018.
- [19] D. Manajang, S. R. U. A. Sompie, and A. Jacobus, "Implementasi Framework Tensorflow Object Detection API Dalam Mengklasifikasi Jenis Kendaraan Bermotor," *Jurnal Teknik Informatika*, vol. 15, no. 3, pp. 171–178, 2021.
- [20] S. R. Dewi, "Deep Learning Object Detection Pada video menggunakan tensorflow dan convolutional neural network," 2018.