

IMPLEMENTASI METODE DEEP LEARNING UNTUK MENENTUKAN TINGKAT KEMATANGAN BUAH PEPAYA CALIFORNIA

Muhammad Ihsan Muttaqin, Arnisa Stefanie, Lela Nurpulaela

Program Studi Teknik Elektro S1, Fakultas Teknik

Universitas Singaperbangsa Karawang, Jalan HS. Ronggo Waluyo Karawang, Indonesia

ihسانmuttaqin1502@gmail.com

ABSTRAK

Perkembangan teknologi kecerdasan buatan pada saat ini sangat cepat dan digunakan pada berbagai industri. Pemanfaatan teknologi kecerdasan buatan ini harus dapat dioptimalkan pada seluruh bidang kehidupan manusia agar pekerjaan manusia jadi lebih mudah, salah satunya bidang pada bidang pertanian. Saat ini penggunaan kecerdasan buatan pada bidang pertanian masih belum maksimal, seperti pada buah pepaya, Buah pepaya memiliki kandungan gizi dan vitamin yang bagus membuatnya banyak dikonsumsi oleh masyarakat Indonesia. Namun, para petani saat ini masih menentukan tingkat kematangan buah secara manual yang hasilnya kurang optimal karena faktor manusia. Peneliti membuat sistem untuk mengidentifikasi tingkat kematangan buah pepaya California dengan metode pembelajaran mendalam yaitu MobileNet-SSD V2, suatu algoritma object detection yang merupakan pengembangan dari metode CNN yang mampu bekerja pada perangkat dengan sumber daya terbatas dengan num steps sebanyak 20000 dan dataset sebanyak 300 gambar.

Kata kunci: Kecerdasan Buatan, Pepaya California, Pembelajaran Mendalam, MobileNet-SSD V2

1. PENDAHULUAN

Buah pepaya dalam bahasa latin disebut dengan (*Carica papaya L.*) kaya akan manfaat serta kaya kandungan gizi dan vitaminnya yang bagus [1]. Pada buah pepaya terdapat kandungan vitamin C dan serat yang bagus untuk pencernaan. selain vitamin C, vitamin yang terkandung lainnya adalah vitamin A, Vitamin B, Vitamin E dan Vitamin K, serta antioksidan seperti karoten, zeaxanthin dan flavonoid. Bahkan beberapa mineral penting seperti, kalium, kalsium, magnesium, zat besi dapat ditemukan dalam pepaya [2]. Dengan banyaknya manfaat yang dimiliki oleh buah papaya membuat masyarakat tertarik untuk mengkonsumsinya, ditambah lagi pohon buah papaya tersebut mampu untuk beradaptasi pada iklim tropis seperti di Indonesia. Selain itu perawatan buah papaya juga murah dan mudah untuk dilakukan.

Banyak permasalahan yang dapat dijumpai ketika mengidentifikasi tingkat kematangan buah secara tradisional. Hal tersebut dapat juga disebabkan oleh beberapa hal antara lain sifat manusia yang memiliki kelemahan seperti lelah, ketidakakuratan dikarenakan keterbatasan fisik, dan kelainan pada panca indra manusia yang pada akhirnya menyebabkan hasil identifikasi kematangan tersebut menjadi tidak akurat [3]. Saat ini para pedagang dan petani buah pepaya masih memantau kematangan pepaya secara manual. Sistem manual yang dimaksud adalah dengan memantau dengan mata manusia dan menerka-nerka dan hanya dengan ditekan-tekan serta dari aromanya saja. Hal tersebut membuat para pedagang dan petani kesulitan untuk mengetahui tingkat kematangan buah pepaya tersebut [4].

Pada buah pepaya terdapat tiga tingkat kematangan yaitu belum matang, setengah matang dan matang. Namun terkadang manusia memiliki

kelemahan dalam menentukan buah yang matang atau belum. Buah pepaya sendiri kematangan nya diukur dengan beberapa parameter, salah satunya adalah warna dari buah nya [5]. Cara analisa melalui warna kulit ini, terdapat permasalahan seperti adanya subjektivitas dalam menentukan kematangan karena tidak adanya tolak ukur yang baku dalam penentuannya. Pemantauan yang tidak efisien tersebut mengakibatkan hasil panen pepaya menjadi terkendala dan tidak maksimal.

Berdasarkan permasalahan yang telah dijelaskan, pemanfaatan metode Deep Learning dapat dilakukan untuk mendeteksi buah papaya secara lebih efisien. Dalam mendeteksi tingkat kematangan buah papaya, sistem ini menggunakan salah satu algoritma object detection MobileNet-SSD V2 yang merupakan teknik arsitektur dalam model deep learning yang paling umum digunakan proses object detection karena mampu bekerja secara realtime dan terbukti efisien dengan akurasi yang baik dan dapat dijalankan pada perangkat komputasi yang ringan [6].

Metode Deep Learning dibutuhkan dalam mendeteksi objek berupa buah papaya kemudian menganalisis warna buah papaya untuk menentukan tingkat kematangannya. Dengan menggunakan MobileNet-SSD V2 membuat petani akan mendapatkan informasi mengenai tingkat kematangan buah papaya secara lebih cepat dan akurat sehingga dapat menghemat tenaga serta waktu dalam mengidentifikasi tingkat kematangan buah papaya tersebut.

2. TINJAUAN PUSTAKA

Dalam merancang penelitian ini, terdapat beberapa tinjauan pustaka yang dilakukan antara lain, yaitu penelitian terdahulu yang telah dilakukan,

kemudian penjelasan secara singkat mengenai Deep Learning, CNN, MobileNet-SSD V2, dan lainnya yang menunjang penelitian ini.

2.1. Penelitian Terdahulu

Penelitian pertama yaitu Krista Bella, dkk. dengan judul penelitian Implementasi Sistem Pendeteksi Tingkat Kematangan Buah Pepaya Menggunakan Metode RGB dimana peneliti merasa perbedaan tingkat kematangan buah menjadi sebuah masalah karena akan menentukan buah pepaya mana yang layak untuk dikonsumsi. Menurut peneliti saat ini untuk mengetahui kematangan buah pepaya hanya dengan menerka-nerka dan ditekan-tekan, peneliti merancang sebuah sistem yang dapat membantu masyarakat yang kurang paham tingkatan kematangan buah pepaya. Sistem yang digunakan menggunakan RGB yang merupakan sebuah hasil campuran warna-warna primer. Hasil yang dicapai mendapat nilai akurasi sebesar 50%. peneliti memberikan saran agar saat proses pengambilan gambar diharapkan memperhatikan posisi dan pencahayaan [7].

Penelitian kedua yang dilakukan oleh Ellif, Sampe Hotlan Sitorus, Rahmi Hidayati dengan judul Klasifikasi Kematangan Pepaya Menggunakan Ruang Warna HSV dan Metode *Naive Bayes Classifier*. Pada penelitian tersebut, peneliti membangun sistem pengklasifikasian tingkat kematangan buah pepaya memanfaatkan ruang warna HSV (*Hue, Saturation, and value*) dan metode *naive bayes* yaitu dengan mencari nilai probabilitas pada gambar yang diuji berdasarkan tingkat kemiripannya dengan citra latih yang tersedia. Peneliti menggunakan 50 buah pepaya yang dibagi menjadi 30 data latih dan 20 data uji mendapatkan hasil percobaan yang sesuai [8].

2.2. Deep Learning

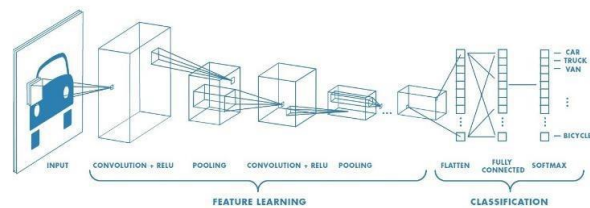
Deep Learning merupakan bagian dari cabang ilmu kecerdasan buatan atau biasa disebut dengan Artificial Intelligence (AI) yang menggunakan jaringan syaraf tiruan dalam mempelajari dan menganalisis data [9]. Deep learning memungkinkan model komputasi dari beberapa lapisan pemrosesan untuk mempelajari dan mempresentasikan data dengan meniru bagaimana cara kerja otak manusia dalam memahami informasi dan mengambil keputusan.

Metode ini mampu bekerja dengan jumlah data yang besar dan beradaptasi untuk menyelesaikan permasalahan yang sulit diselesaikan oleh algoritma kecerdasan buatan lainnya. Deep learning dapat diaplikasikan ke berbagai macam hal untuk membuat pekerjaan manusia menjadi lebih efisien, cepat, dan akurat. Deep learning memiliki beberapa jenis algoritma, antara lain seperti *Convolutional Neural Network* (CNN), *Recurrent Neural Network* (RNN), *Long Short Term Memory Network* (LSTM), dan *Self Organizing Maps* (SOM).

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah satu metode dari *Deep Learning* yang dikhususkan untuk pengolahan citra digital sehingga dapat digunakan dalam pendeteksian objek [10]. CNN merupakan salah satu teknik paling sukses dalam *computer vision* dalam memahami dan mengerti arti dari gambar yang dimasukkan. Hal tersebut karena metode ini mengimplementasikan sistem pengenalan citra pada visual cortex manusia [11]. CNN banyak digunakan dalam mengklasifikasikan data yang memiliki label menggunakan metode supervised learning, yaitu target yang diharapkan adalah berupa input yang diterima oleh jaringan telah diketahui sebelumnya.

Pada umumnya CNN terdiri dari banyak layer untuk melakukan proses dan ekstraksi fitur dari data yang didapatkan, dan biasanya terdiri dari 3 jenis utama *neural network* seperti, *convolutional layer*, *pooling layer*, dan *fully connected layer*. *Convolutional layer* merupakan layer yang melakukan operasi konvolusi pada output yang dihasilkan dari layer sebelumnya dan menjadi dasar. Konvolusi yaitu istilah matematis dalam pengaplikasian fungsi pada output fungsi lain secara berulang [12].



Gambar 1. Arsitektur *Convolutional Neural Network*

2.4. Layer

Dalam Deep Learning (DL), layer atau lapisan adalah bagian yang penting dalam arsitektur Artificial Neural Network (ANN). Setiap lapisan yang berada dalam arsitektur ANN mempunyai peran khusus dalam pemrosesan data agar kemampuan jaringan tersebut dapat mempelajari suatu model representasi yang semakin kompleks [13].

Input Layer adalah lapisan pertama dalam jaringan yang menerima data masukan yang diberikan dan fitur-fitur dari dataset yang akan diproses. Layer ini bertugas untuk membawa data tersebut kedalam sistem untuk diproses pada layer berikutnya.

Hidden Layer atau Lapisan tersembunyi yaitu sebutan untuk lapisan yang terdapat diantara *Input layer* dan *Output layer*. *Hidden Layer* terdiri dari beberapa lapisan tergantung pada kompleksitas jaringan.

Pooling Layer merupakan lapisan yang digunakan untuk mengambil suatu gambar dengan ukuran dimensi yang besar kemudian mengecilkannya, akan tetapi tetap mempertahankan informasi yang penting yang berada di gambar tersebut, proses tersebut disebut dengan reduksi

sampel (*downsampling*)[14]. *Pooling Layer* berfungsi mengurangi jumlah parameter di dalam jaringan untuk menghasilkan representasi model yang lebih ringkas untuk menghindari *overfitting*.

2.5. MobileNet-SSD V2

MobileNet-SSD merupakan model *object detection* yang menggabungkan arsitektur *MobileNet* dengan metode ojek deteksi *Single Shot Multibox Detector* (SSD). Mobilenet adalah model yang mengandalkan kecepatan tetapi terdapat kelemahannya yaitu akurasi, SSD terbukti membantu meningkatkan akurasi dan mempertahankan kecepatan pada model ini. Model ini dirancang khusus pada perangkat dengan sumber daya terbatas namun akurasi ketepatan yang dihasilkan cukup tinggi.

MobileNet-SSD V2 adalah peningkatan dari model sebelumnya yaitu MobileNet V1 dengan modularitas yang jauh lebih baik untuk memperbaiki kelemahan yang terdapat pada MobileNet-SSD V1 dengan meningkatkan performa dan efisiensi model dengan mengurangi kompleksitas serta meningkatkan efisiensi pada komputasi.

Dalam arsitektur MobileNet-SSD V2 menggunakan MobileNet V2 sebagai *backbone* atau tulang punggung detector SSD yang menggabungkan beberapa teknik seperti ekspansi-linear, *bottleneck layer*, dan *inverted residual blocks* untuk meningkatkan kinerja dan efisiensi [15]. Arsitektur ini membantu menghasilkan representasi-fitur yang lebih baik dan meningkatkan performa deteksi objek.



Gambar. 2. Arsitektur MobileNet-SSD V2

Pada arsitektur MobileNet-SSD V2 terdapat *convolutional layer* dan 17 modul *residual invers* seperti yang ditunjukkan pada gambar 2 diatas. Setiap modul residual invers memiliki lapisan konvolusional 1x1, lapisan konvolusional yang dapat dipisahkan berdasarkan kedalaman (*Dwise*) 3x3, *batch normalisasi* (bn), dan fungsi eksitasi Relu6. Peta fitur input juga digabungkan dengan peta fitur output tanpa mengubah ukurannya. Memanfaatkan modul residual invers memiliki keuntungan dalam menangani masalah penghilangan gradien untuk mentransfer informasi gradien dengan benar ke lapisan jaringan yang lebih dalam untuk membentuk pelatihan yang baik selama *backpropagation*.

Lapisan konvolusi berskala 1/16 menyampling peta fitur Mobilenet-SSD tradisional. Pengaturan ini akan menyebabkan deteksi kecil yang buruk dan tidak stabil. Untuk mengatasi masalah ini, dalam arsitektur jaringan kami, kami menggunakan peta fitur pada

skala 1/16 dan 1/32 sebagai gantinya. Selain itu, setelah jaringan tulang punggung, kami menambahkan empat residu terbalik untuk mengekstrak peta fitur pada skala 1/64, 1/128, 1/256, dan 1/512. Terakhir, modul FPN meningkatkan peta fitur skala yang berbeda ini.

2.6. TensorFlow

Tensorflow merupakan library yang menggunakan grafik aliran data (*data flow graph*) untuk mempresentasikan komputasi yang melibatkan operasi matematika yang kompleks yang digunakan pada pengembangan dan pelatihan model kecerdasan buatan. Tensorflow pertama kali diumumkan oleh tim Google Brain pada tahun 2015. Tensorflow sangat memudahkan dan mempercepat penelitian dan penerapan model neural network [16].

TensorFlow telah mengalami beberapa rilis versi sejak awal peluncurannya. Setiap versi memiliki fitur, perbaikan, dan peningkatan kinerja. Perubahan versi Tensorflow 1.x menuju 2.x mengalami perubahan yang signifikan dalam hal antarmuka pemrograman dan penyederhaan. Tensorflow memiliki berbagai fungsi dan library yang membantu berbagai macam tujuan, sebagaimana yang ditampilkan pada tabel berikut.

Tabel 1. Library TensorFlow

Library	Fungsi
TensorFlow Keras	High-level API yang menyederhanakan proses pembuatan, pelatihan, dan evaluasi model kecerdasan buatan.
TensorFlow Datasets	Library untuk mengelola dan memuat dataset untuk digunakan dalam pelatihan model.
TensorFlow Serving	Framework untuk menyajikan model yang dilatih ke dalam lingkungan produksi.
TensorFlow Lite	Platform yang dioptimasi untuk menjalankan model kecerdasan buatan pada perangkat mobile dan perangkat terbatas lainnya.
TensorFlow Core	Library inti yang menyediakan fungsi-fungsi dasar untuk definisi, operasi, dan pelatihan model.

2.7. Python

Python adalah bahasa pemrograman tingkat tinggi yang dinamis, berorientasi objek, dan digunakan pada pengembangan perangkat lunak. Python pertama kali dikembangkan oleh seorang ilmuwan asal Belanda bernama Guido Van Rossum pada tahun 1991. Python dirancang dengan penekanan pada keterbacaan kode sehingga membuatnya menjadi bahasa pemrograman dengan syntax sederhana yang ramah bagi para pemula dan mudah dipelajari. Hal tersebut juga yang membuat Python menjadi pilihan dalam pengembangan prototipe, aplikasi web, analisis data, kecerdasan buatan dan lainnya [17].

Salah satu keunggulan Python adalah banyaknya jumlah pustaka dan kerangka kerja (*framework*) yang ada sehingga memungkinkan pengembang untuk

memperluas kegunaan Python. Keberagaman pustaka tersebut menjadikan Python dapat digunakan secara luas dalam berbagai industri dan aplikasi dengan mudah.

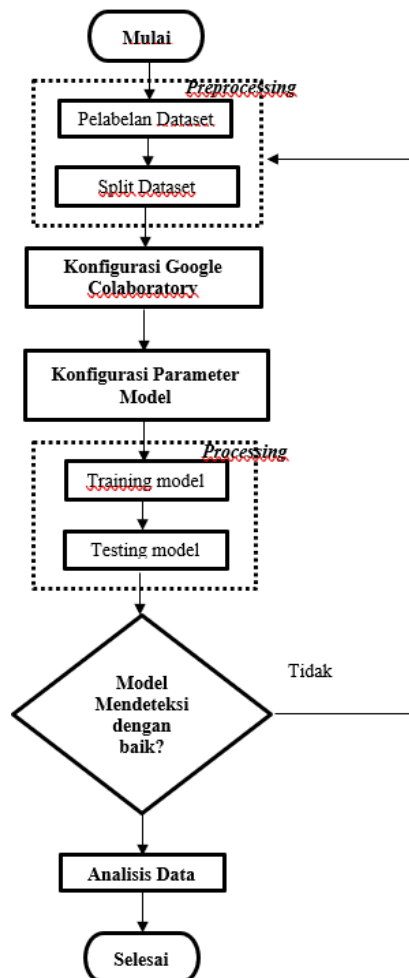
Tabel 2. Library Python

Pustaka/Library	Kegunaan
NumPy, Pandas, dan Matplotlib	Melakukan analisis data dan visualisasi
Django dan Flask	<i>Framework</i> atau kerangka kerja untuk membangun aplikasi web
TensorFlow dan PyTorch	Pengembangan <i>Artificial Intelligence</i> (AI) atau kecerdasan buatan
Pygame	Pengembangan game
Tkinter atau PyQt	Pengembangan aplikasi desktop
SciPy dan scikit-learn	Mendukung analisis permodelan ilmiah

3. METODE PENELITIAN

Penelitian yang dilakukan ini bertujuan untuk menguji metode MobileNet-SSD V2 dalam mendeteksi buah pepaya dan menentukan tingkat kematangannya berdasarkan warna.

3.1. Rancangan Sistem



Gambar. 3. Flowchart Penelitian

Pada penelitian ini peneliti menggunakan bahasa pemrograman Python karena memiliki banyak pustaka yang mendukung proses komputasi dalam bidang kecerdasan buatan. Dalam proses perancangan ini, seperti pada gambar 3 dibawah dapat dijelaskan bahwa proses kerja sistem terdapat beberapa tahap.

3.2. Pengumpulan Data

Pada metode pengumpulan data, gambar pepaya California didapatkan secara langsung memfoto pada perkebunan buah pepaya yang dibagi menjadi 2 kelas, yaitu mentah dan setengah matang. Karena pada saat kami melakukan survey ke kebun pepaya, kami mendapat informasi bahwa petani melakukan panen saat kondisi setengah matang dan menyimpan di tempat lain agar terhindar dari hama. Dengan total dataset sebanyak 300 gambar dan pembagian data dapat dilihat pada tabel 1 sebagai berikut.

Tabel 3. Pembagian Dataset

Jumlah Dataset	Split Dataset		
	Training Data	Validation Data	Testing Data
300	240	30	30

Data tersebut kemudian masuk pada tahap anotasi dan labelling dengan tujuan menandakan buah pepaya pada gambar dengan membuat bounding box dan memberikannya label sesuai dengan kelas objek tersebut.

Training Data yang dimaksud pada tabel 3 merupakan data yang akan dilatih oleh model dalam menentukan ketepatan model mendeteksi objek berdasarkan kelas yang telah ditentukan pada proses labelling. Validation Data adalah data yang digunakan untuk memvalidasi model selama proses pelatihan. Testing Data adalah data yang digunakan dalam menguji model saat proses pelatihan sudah selesai dilakukan.

3.3. Preprocessing Data

Tahap pertama pada sistem ini adalah *Preprocessing* data, pada tahap ini terjadi 2 proses yaitu *labelling* dan *split dataset*. *Labelling* adalah mengidentifikasi kelas objek dan *split dataset* adalah metode pemisahan data menjadi 2 (dua) bagian atau lebih yang membentuk subhimpunan data.

3.4. Konfigurasi

Tahap selanjutnya yaitu, Konfigurasi Google Colaboratory, sistem yang akan dibuat menggunakan platform google colaboratory dan harus mengatur beberapa konfigurasi seperti GPU. Setelah itu, Konfigurasi Parameter Model, tahap ini dilakukan untuk menyesuaikan dengan pelatihan model yang dibuat.

Tabel 4. Konfigurasi Parameter Model

Parameter	Nilai
Num steps	20000
Batch size	4

Parameter	Nilai
Image size	320x320
Num Classes	3
Activation	ReLU
Num epoch	1

3.5. Processing

Tahap Processing yang terdiri dari 2 proses training model dan testing model.

- *Training model* adalah proses dimana model jaringan akan mempelajari data yang diberikan untuk menghasilkan prediksi yang telah ditargetkan. Pada proses pelatihan data dibagi menjadi *training* 80%, *validation* 10%, dan *testing* 10%
- *Testing model* adalah pengujian terhadap model yang sudah dilatih dengan memasukkan gambar untuk menguji apakah model sudah dapat bekerja sesuai yang diharapkan

3.6. Analisis Data

Metode analisis yang digunakan pada penelitian ini adalah analisis deskriptif. Hal ini karena analisis deskriptif merupakan metode yang berfungsi mendeskripsikan atau memberikan gambaran secara sistematis data yang telah diperoleh pada penelitian ini.

4. HASIL DAN PEMBAHASAN

4.1. Pengujian Sistem

Model yang telah dibuat kemudian masuk kedalam tahap pengujian sistem dilakukan dengan memasukkan foto kedalam model untuk mengetahui apakah model sudah dapat mendeteksi buah pepaya dan mengidentifikasi tingkat kematangannya.



Gambar. 4. Pengujian Sistem

Pada gambar 4. Menampilkan sistem telah mampu membaca buah pepaya dan menentukan kematangannya. Terdapat 7 buah pepaya yang terdeteksi dengan tingkat kematangan mentah

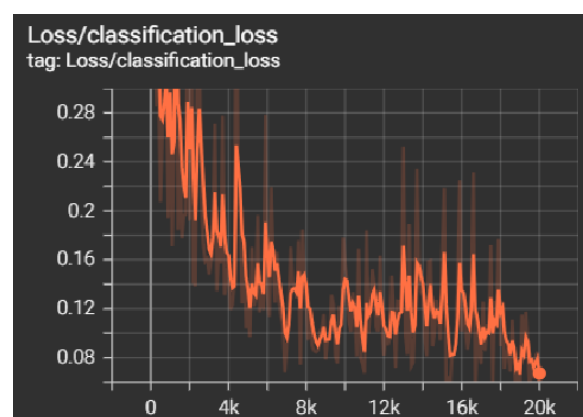
4.2. Loss Function

Loss Function atau fungsi kerugian merupakan rumus yang diminimalkan oleh algoritma pembelajaran mesin selama pelatihan model. Cara ini digunakan untuk mengetahui pengaturan parameter terbaik untuk model agar hasil yang dicapai optimal.

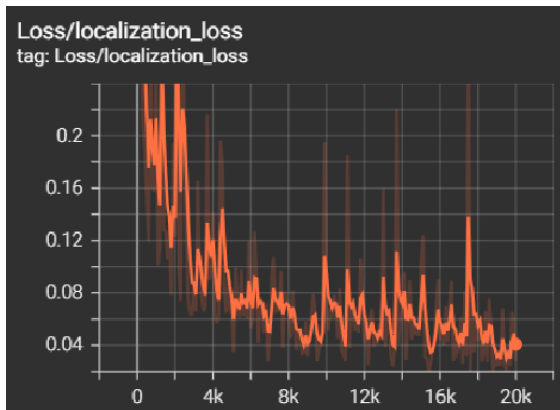
Tabel 5. Nilai Loss Function

Num Steps	Loss Function				Learning Rate
	Classification loss	Localization loss	Regularization loss	Total _loss	
1000	0.3510	0.2662	0.1530	0.7704	0.0800
2000	0.1916	0.1225	0.1474	0.4615	0.0799
3000	0.1478	0.0862	0.1424	0.3765	0.0796
4000	0.1605	0.1411	0.1370	0.4387	0.7926
5000	0.0630	0.1320	0.3163	0.3163	0.0786
6000	0.2784	0.1196	0.1280	0.3123	0.0779
7000	0.0913	0.5014	0.1232	0.2647	0.0770
8000	0.1259	0.0674	0.1187	0.3120	0.0760
9000	0.0784	0.3166	0.1144	0.2245	0.0748
10000	0.1559	0.0754	0.1107	0.3421	0.0735
11000	0.0673	0.0279	0.1071	0.2024	0.0720
12000	0.0818	0.0415	0.1035	0.2269	0.0704
13000	0.2523	0.1599	0.1003	0.5126	0.0687
14000	0.1293	0.0669	0.0970	0.2933	0.0668
15000	0.1666	0.1051	0.0946	0.3665	0.0649
16000	0.1012	0.0446	0.0910	0.2380	0.0628
17000	0.0748	0.0372	0.0894	0.2015	0.0673
18000	0.0894	0.0430	0.0874	0.2199	0.0585
19000	0.0541	0.0238	0.0855	0.1639	0.0561
20000	0.0490	0.0282	0.0832	0.1605	0.0538

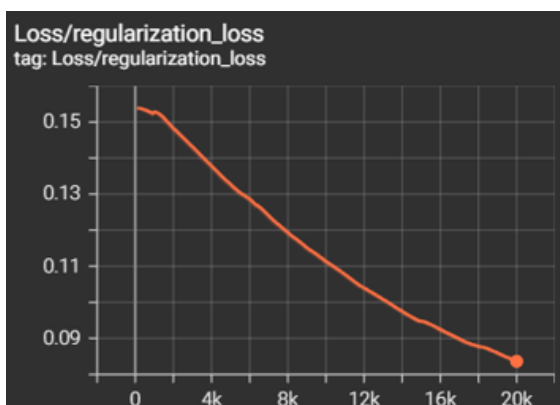
Classification loss merupakan fungsi yang melatih klasifikasi untuk menentukan jenis objek target. *Localization loss* adalah fungsi untuk melatih *bounding box* yang di prediksi dengan nilai sebenarnya. *Regularization loss* adalah fungsi yang digunakan untuk mencegah terjadinya *overfitting* dan meningkatkan generalisasi jaringan syaraf. *Total loss* yaitu rata-rata dari ketiga loss yang diukur. Sedangkan *Learning Rate* merupakan hyperparameter yang digunakan untuk mengatur seberapa besar perubahan parameter model yang terjadi selama proses pelatihan. Berikut ini adalah grafik berdasarkan data tabel 5 diatas.



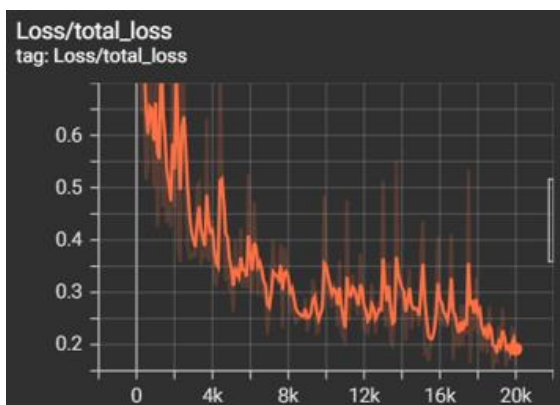
Gambar. 5. Classification_loss



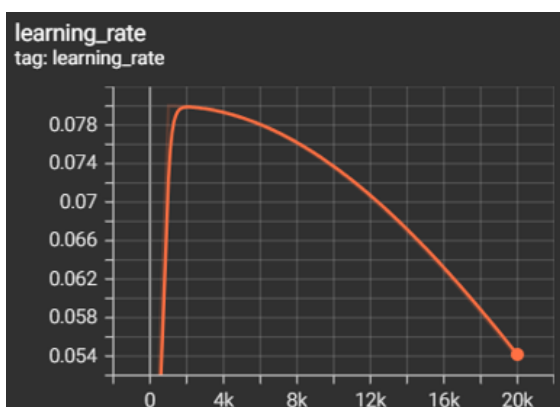
Gambar. 6. Localization_loss



Gambar. 7. Regularization loss



Gambar. 8. Total loss



Gambar. 9. Learning Rate

Pada Gambar 5. Menampilkan grafik classification loss pada pelatihan sebanyak 20.000 num steps dan mendapatkan hasil 0.0490. Pada Gambar 6. Menampilkan grafik localization loss pada pelatihan sebanyak 20.000 num steps dan mendapatkan hasil 0.0282. Pada Gambar 7. Menampilkan grafik Regularization loss pada pelatihan sebanyak 20.000 num steps dan mendapatkan hasil 0.0832. Pada Gambar 8. Menampilkan grafik Total loss pada pelatihan sebanyak 20.000 num steps dan mendapatkan hasil 0.1605. Pada Gambar 9. Menampilkan grafik Learning Rate pada pelatihan sebanyak 20.000 num steps dan mendapatkan hasil 0.0538.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan untuk mengidentifikasi tingkat kematangan buah pepaya California menggunakan algoritma *deep learning object detection* yaitu MobileNet-SSD V2 telah diperoleh kesimpulan bahwa sistem yang dibuat ini dapat berfungsi dengan baik dan mampu mengidentifikasi kematangan pepaya California dengan baik. Seluruh percobaan dan pengujian yang telah dilakukan menunjukkan hasil bahwa sistem bisa digunakan.

Akan tetapi terdapat juga kelemahan dari sistem ini seperti dataset yang dilatih perlu lebih banyak agar akurasi model lebih baik, dibutuhkannya suatu *database* yang menyimpan hasil deteksi, algoritma yang lebih tinggi dalam tingkat komputasinya agar hasil yang didapat lebih maksimal, dan meningkatkan resolusi kamera yang lebih baik agar hasil yang didapatkan lebih baik.

DAFTAR PUSTAKA

- [1] Prahara H.W, Sela E.I. 2016. *Tingkat Kematangan Buah Pepaya Dengan Jaringan Syaraf LVQ*. Seminar Riset Teknologi Informasi (SRITI) tahun 2016.
- [2] Mirani .T, Sinduningrum .E, Dzikrillah A.R. 2018. *Rancang Bangun Sistem Pengenalan Citra untuk Tingkat Kematangan Buah Pepaya California Berdasarkan Warna Berbasis*. ISSN No. 2502-8782 Vol. 3, 2018.
- [3] Arrofiqoh .N.E, dan Harintaka. 2018. *Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi*. Geomatika Volume 24 No.2 November 2018: 61-68.
- [4] M. dan E. Nurraharjo. 2019. *Sistem Deteksi Kematangan Buah Alpukat Menggunakan Metode Pengolah Citra*. Dinamika Informatika, pp. 2086-9398; 2019
- [5] S. Jatmika dan D. Purnamasari. 2014. *Rancang Bangun Alat Pendeteksi Kematangan Buah Apel dengan Menggunakan Metode Image Processing Berdasarkan Komposisi Warna*. Jurnal Ilmiah Teknologi dan Informasi ASIA, 2014.

- [6] Azeddine Elhassouny, Florentin Smarandache. 2019. *Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks*. IEEE/ICCSRE2019, 22- 24 July, 2019, Agadir, Morocco.
- [7] Widayarsi .K.B, Rosiana U.D, Pramudhita A.N. 2021. *Implementasi Sistem Pendeteksi Tingkat Kematangan Buah Pepaya Menggunakan Metode RGB*. SMATIKA Jurnal Volume 11 Nomor 01, Juni Tahun 2021 ISSN: 2087-0256, e-ISSN: 2580-6939.
- [8] Audy, Zaini, 2022. Analisis Kualitas Jagung Berbasis IoT *dengan* Penerapan Model SSD Mobilenet dan Histogram. Jurnal Nasional Teknik Elektro dan Teknologi Informasi | Vol. 11, No. 2, Mei 2022.
- [9] Janiesch. C, Zschech. P, Heinrich. K. 2021. *Machine Learning and Deep Learning*. Electronic Markets (2021) 31:685–695.
- [10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. 2016. *SSD: Single shot multibox detector*. European conference on computer vision (pp. 21-37).
- [11] LeCun, Y., Bengio, Y., & Hinton, G. 2015. *Deep learning*. Nature, 521(7553), 436–444.
- [12] Sharma. N, Jain. V, Mishra. A. 2018. *An Analysis Of Convolutional Neural Networks For Image Classification*. Procedia Computer Science Vol 132. Page 377-384.
- [13] Desai. M, Shah. M. 2021. *An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN)*. Clinical eHealth Volume 4, 2021, Pages 1-11.
- [14] Suyanto, Ramadhani, K. N., & Mandala, S. 2019. *Deep Learning Modernisasi Machine Learning untuk Big Data*. (1st ed.). Informatika.
- [15] Chiu. C., Y. Tsai. C., Y. Ruan. D., M. Shen, Y., G. Lee, T., T. 2020. *Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems*. International Conference on System Science and Engineering (ICSSE).
- [16] Bo Pang, Nijkamp. E, Wu. N.Y. 2019. *Deep Learning With TensorFlow: A Review*. Journal of Educational and Behavioral Statistics Vol. XX, No. X, pp. 1–22.
- [17] K.R. Srinath. 2017. *Python – The Fastest Growing Programming Language*. International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 12 | Dec-2017. e-ISSN: 2395-0056 p-ISSN: 2395-0072.