

## PEMANFAATAN YOLO UNTUK PENGENALAN KESEGARAN BUAH MANGGA

Muhammad Syifa'un Nuha<sup>1</sup>, Reddy Alexandro H.<sup>2</sup>  
*Magister Teknologi Informasi, S2 Teknologi Informasi, iSTTS*  
*Jalan Ngagel Jaya Tengah 73 - 77, Surabaya, Indonesia*  
*+62 31 502 7920, +62 31 504 1509.*  
*E-mail: f44unt@gmail.com*

### ABSTRAKS

Buah merupakan salah satu kebutuhan selain makanan pokok, tidak hanya dikalangan tertentu saja, tapi disemua kalangan. Indonesia menjadi penghasil mangga terbesar keempat di dunia. Sehingga diperlukan suatu sistem yang dapat secara otomatis mengidentifikasi kebusukan dan kesegaran dari 3 jenis buah Mangga menggunakan pengolahan gambar, memperbaiki teknik penyortiran dan penilaian yang tidak ilmiah yang dilakukan secara manual, sehingga bisa meningkatkan kualitas jual mangga dengan menggunakan algoritma YOLO. Penelitian ini menggunakan 3 jenis mangga yang terdiri dari mangga golek, gedong, dan manalagi, dan melakukan uji coba dengan beberapa skenario yaitu semua gambar mangga segar, semua gambar mangga busuk, dan semua gambar mangga segar dan busuk. Hasil dari penelitian menunjukkan beberapa uji coba yang dilakukan, maka didapatkan nilai rata-rata precision, recall, dan f1- score Skenario pertama yaitu mangga segar semua didapatkan tingkat akurasi 80%, precision 82%, dan recall 87%, didapatkan F1-score 84%. Kemudian skenario yang kedua yaitu mangga busuk semua didapatkan tingkat akurasi 76%, precision 76%, dan recall 87%, didapatkan F1-score 81%. Dan yang ketiga yaitu mangga segar dan busuk, didapatkan tingkat akurasi 73%, precision 66%, dan recall 81%, didapatkan F1-score 73%. dapat disimpulkan bahwa hasil penelitian ini masih tergolong underfitting. Hal ini dikarenakan masih butuh banyak dataset yang lebih banyak dan variannya yang mempunyai ciri-ciri yang ada kemiripan masing-masing kelasnya.

*Kata Kunci: mangga, identifikasi, yolo*

### ABSTRACT

*Fruit is one of the necessities besides staple food, not only among certain people, but in all circles. Indonesia is the fourth largest mango producer in the world. So we need a system that can automatically identify rot and freshness of 3 types of mango using image processing, improve sorting techniques and unscientific assessments that are done manually, so as to improve the quality of selling mangoes using the YOLO algorithm. This study used 3 types of mango consisting of mango golek, gedong, and manalagi, and conducted an experiment with several scenarios, namely all pictures of fresh mangoes, all pictures of rotten mangoes, and all pictures of fresh and rotten mangoes. The results of the study show that several trials were carried out, so that the average value of precision, recall, and f1-score was obtained. The first scenario, namely fresh mangoes, all obtained an accuracy rate of 80%, precision 82%, and recall 87%, obtained an F1-score of 84%. Then the second scenario, namely rotten mangoes, all obtained an accuracy rate of 76%, precision 76%, and recall 87%, obtaining an F1-score of 81%. And the third, fresh and rotten mangoes, obtained an accuracy rate of 73%, precision 66%, and recall 81%, obtaining an F1-score of 73%. it can be concluded that the results of this study are still classified as underfitting. This is because more datasets and variants are still needed which have characteristics that are similar to each class*

*Keywords: mango, identification, yolo.*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Buah merupakan salah satu kebutuhan selain makanan pokok, tidak hanya dikalangan tertentu saja, tapi disemua kalangan. Technical Manager PT Rainbow Agrosiences Adi Wihardi mengatakan, dengan produksi 2,4 juta ton mangga per tahun, Indonesia menjadi penghasil mangga terbesar keempat di dunia. tetapi meskipun merupakan produsen mangga terbesar ke-empat, indonesia tidak mampu memaksimalkan ekspor mangga ke negara lain, hanya sekitar 0,1% dari total mangga. Secara

umum, warna buah magga bisa menunjukkan kesegaran buah dan adanya kebusukan.

Dalam penelitian ini, sebuah algoritma yang diusulkan untuk secara otomatis mengidentifikasi kebusukan dan kesegaran dari 3 jenis buah Mangga menggunakan pengolahan gambar. Kerangka kerja ini dapat diterapkan di berbagai bidang seperti perusahaan manufaktur, supermarket, minimarket, pasar tradisional, dan toko buah – buahan. Tujuan dari penelitian ini adalah untuk menerapkan algoritma pengolahan gambar yang dapat membantu dalam mengotomatisasi proses identifikasi kesegaran mangga dan juga kebusukan Mangga. Alasan utama

dilakukan penelitian ini adalah memperbaiki teknik penyortiran dan penilaian yang tidak ilmiah yang dilakukan secara manual, sehingga bisa meningkatkan kualitas jual mangga dengan menggunakan algoritma YOLO.

Tujuan utama dari penelitian ini adalah untuk merancang sebuah algoritma YOLO yang dapat mengidentifikasi kesegaran dan juga kebusukan mangga, deteksi buah Mangga berdasarkan bentuk fitur oleh analisis citra digital. Penelitian yang diusulkan ini merupakan upaya untuk mengimplementasikan proyek yang dirancang secara ekstensif berdasarkan topik yang dibahas.

## 1.2 Referensi

### 1.2.1. CNN (Convolutional Neural Network)

Ada beberapa jenis jaringan syaraf pada deep learning seperti Recurrent Neural Networks (RNN), Artificial Neural Networks (ANN) dan Convolutional Neural Networks (CNN). Untuk setiap jenis metode ini, mereka memiliki spesialisasi, kelebihan dan kekurangan masing-masing.

Untuk memahami CNN, kita dapat mengamatinnya sebagai dua bagian yang berbeda. Dalam input, gambar disajikan sebagai matriks piksel yang memiliki 2 dimensi untuk gambar grayscale. Warna diwakili oleh dimensi ketiga, dengan kedalaman 3 untuk mewakili warna dasar (Red, Green, Blue). (Girshick et al., 2014)

Bagian yang pertama dari CNN adalah bagian yang berbelit-belit. Berfungsi untuk ekstraktor fitur gambar. Pada bagian ini, gambar melewati serangkaian filter, atau kernel konvolusi, membuat gambar baru yang disebut convolution maps. (Jmour et al., 2018)

### 1.2.2. Fully-Connected Layer (FC Layer)

Feature map yang dihasilkan dari feature extraction layer masih berbentuk multidimensional array, sehingga kita harus melakukan "flatten" atau reshape feature map menjadi sebuah vector agar bisa kita gunakan sebagai input dari fully-connected layer.

Pada Fully Connected Layer setiap node terhubung langsung ke setiap node di layer sebelumnya dan di layer sesudahnya. Disini masing-masing node di frame terakhir di pooling layer dihubungkan sebagai vektor ke layer pertama dari fully connected layer. Kelemahan utama dari fully connected layer adalah fully connected layer mencakup banyak parameter yang memerlukan komputasi kompleks dalam contoh pelatihan. Karena itu node dan koneksi yang dihapus bisa dilakukan dengan menggunakan teknik dropout. Misalnya LeNet dan AlexNet mendesain jaringan yang dalam dan luas sembari menjaga perhitungan yang kompleks tetap konstan. (Albawi et al., 2017)

### 1.2.3. Convolution layer

Proses konvolusi memanfaatkan apa yang disebut sebagai filter. Seperti Iayaknya gambar, filter memiliki ukuran tinggi, lebar, dan tebal tertentu. Filter ini diinisialisasi dengan nilai tertentu (random atau menggunakan teknik tertentu seperti Glorot), dan nilai dari filter inilah yang menjadi parameter yang akan di-update dalam proses learning. (Dharmadi & medium.com, 2018)

Konvolusi (*convolution layer*) merupakan cara untuk mengkombinasikan dua buah deret angka yang menghasilkan deret angka yang ketiga. Lapisan konvolusi dibentuk dengan menjalankan filter di atasnya. Filter merupakan blok lain atau kubus dengan tinggi dan lebar yang lebih kecil namun kedalaman yang sama yang tersapu di atas gambar dasar atau gambar asli. Filter digunakan untuk menentukan pola apa yang akan dideteksi yang selanjutnya dikonvolusi atau dikalikan dengan nilai pada matriks input, nilai pada masing-masing kolom dan baris pada matriks sangat bergantung pada jenis pola yang akan dideteksi.

### 1.2.4. Algoritma YOLO (You Only Look Once)

Algoritma YOLO merupakan algoritma deep learning untuk deteksi objek yang menggunakan pendekatan berbeda dari algoritma lain, yaitu menerapkan sebuah jaringan syaraf tunggal pada keseluruhan citra. YOLO mendeteksi sebuah objek dalam beberapa tahap yaitu :

pertama membagi citra dalam region/grid berukuran  $s \times s$ . Grid-grid tersebut bertanggung jawab untuk mendeteksi objek. Pada tiap grid juga akan diprediksi bounding box beserta nilai confidence. Nilai confidence ini menunjukkan seberapa yakin bounding box tersebut berisi objek dan seberapa akurat prediksinya. (Redmon & Farhadi, 2017)

Tahap kedua yaitu Tiap bounding box memiliki 5 nilai informasi yaitu  $x, y, w, h$  dan  $c$ . Nilai  $x$  dan  $y$  adalah koordinat titik tengah bounding box yang terprediksi, nilai  $w$  dan  $h$  adalah rasio ukuran lebar dan tinggi relatif terhadap grid, dan  $c$  adalah nilai confidence bounding box tersebut. (Lan et al., 2018)

Tahap ketiga pada algoritma YOLO, tiap grid akan memprediksi nilai class probabilitas jika diprediksi terdapat objek di dalamnya.

### 1.2.5. Prediksi Bounding Box

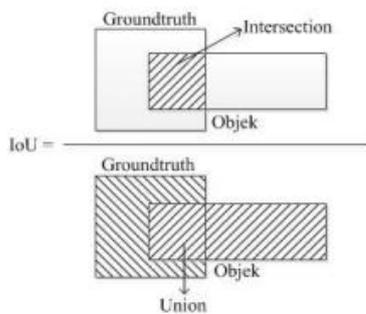
Untuk memprediksi bounding box, YOLOv2 akan memprediksi koordinat titik tengah bounding box dengan relatif terhadap lokasi grid, dengan menggunakan k-means clustering dengan menggunakan nilai  $k = 5$  yang memberikan perbandingan nilai recall dan kompleksitas model yang baik. Sedangkan untuk menghitung jarak euclidean YOLOv2 menggunakan pendekatan IOU. (Redmon & Farhadi, 2017)

### 1.2.6. Intersection over Union (IoU)

Ketika mendeteksi letak objek, algoritma deteksi objek harus dipertimbangkan. Beberapa algoritma pendeteksian mungkin memerlukan pendeteksian letak objek dengan akurasi tinggi, sementara yang lain lebih toleran terhadap kesalahan dalam penempatan kotak pembatas. Keakuratan kotak biasanya diukur dengan menggunakan Intersection over Union (IoU). IoU menghitung area pertemuan antara kotak prediksi objek dan kotak kebenaran dasar (ground truth) dan membaginya dengan area persatuan mereka. Perumusan IoU ditunjukkan pada rumus 1 sedangkan ilustrasi dari perumusan tersebut disajikan pada Gambar 1.1.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (1)$$

Saat mengevaluasi algoritma pendeteksian objek, ambang IoU sebesar 0.5 biasanya digunakan untuk menentukan apakah deteksi benar. (Everingham et al., 2010) Namun nilai IoU = 0.5 mempunyai area yang cukup longgar. Sehingga umumnya diinginkan nilai IoU yang lebih besar dari 0.5. (Zitnick & Dollár, 2014)



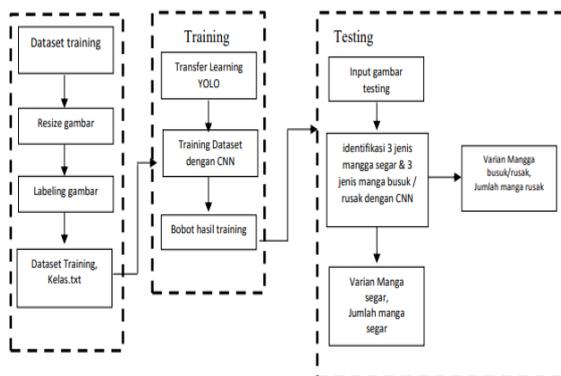
Gambar 1. Ilustrasi perhitungan IOU

## 2. PEMBAHASAN

### 2.1 Perancangan

Pada pokok bahasan ini akan dibahas mengenai desain arsitektur dari sistem, alternative algoritma yang akan digunakan.

#### 2.1.1. Perancangan Arsitektur Sistem



Gambar 2. Blok diagram identifikasi kesegaran buah mangga.

#### a. Preprocessing

Tahapan preprocessing yaitu dataset training dirubah ukurannya menjadi 448 x 448 pixel, selanjutnya dibagi menjadi 6 label kelas yaitu kelas Mangga Manalagi Segar, Mangga Golek segar, Mangga Gedong Gincu segar, Mangga Manalagi busuk/cacat, Mangga Golek busuk/cacat, dan Mangga Gedong Gincu busuk/cacat, serta membuat anotasi tiap citra yaitu memberi bounding box secara manual menggunakan labeling sehingga akan diketahui Bx (titik pusat objek di x), By (titik pusat objek di y), Bw (lebar objek), Bh (tinggi objek) tiap gambar dalam bentuk file .txt untuk masing-masing gambar, serta sebuah file txt berisi nama kelas (kelas Mangga Manalagi Segar, Mangga Golek segar, Mangga Gedong Gincu segar, Mangga Manalagi busuk/cacat, Mangga Golek busuk/cacat, dan Mangga Gedong Gincu busuk/cacat).

#### b. Training

Tahapan training yaitu transfer learning YOLO, selanjutnya dataset yang sudah disiapkan ditraining dengan menggunakan CNN kemudian hasil dari training adalah bobot pelatihan untuk dataset buah Mangga Manalagi Segar, Mangga Golek segar, Mangga Gedong Gincu segar, Mangga Manalagi busuk/cacat, Mangga Golek busuk/cacat, dan Mangga Gedong Gincu busuk/cacat, selanjutnya disimpan sebagai bobot yang akan digunakan pada tahap testing.

#### c. Testing

Tahap terakhir yaitu testing, Pada tahap ini, input berupa gambar selanjutnya dilakukan proses deteksi oleh sistem dengan CNN, mana Mangga Manalagi Segar, Mangga Golek segar, Mangga Gedong Gincu segar, Mangga Manalagi busuk/cacat, Mangga Golek busuk/cacat, dan Mangga Gedong Gincu busuk/cacat. Citra dibagi menjadi 7 grid dimana setiap grid akan memprediksi bounding box dan setiap bounding box memprediksi Bx, By, Bw, Bh, confidence score. kemudian mangga diidentifikasi mana yang Mangga segar dan mana yang mangga cacat. Hasil deteksi pada citra output terdapat bounding box dan nama kelas, dari hasil identifikasi tersebut akan diketahui jenis mangga yang segar dan jenis mangga yang busuk/rusak.

#### 2.1.2. Pengumpulan data

Pada penelitian ini data berupa gambar mangga. data diambil sendiri melalui hp vivo y12 dan iphone 6s, juga melalui google search image, dan instagram melalui aplikasi android (Saver). jumlah data yang didapat sebanyak 1500, yang terdiri dari 1200 data training dan 300 data testing yang dikumpulkan berupa gambar yang nantinya akan dilakukan pelabelan manual.



Gambar 3. Contoh Mangga Manalagi segar



Gambar 4. Contoh Mangga Manalagi Busuk

## 2.2 Transfer Learning Yolo

Pada bab ini diberikan gambaran dan penjabaran mengenai langkah-langkah algoritma yolo untuk identifikasi 3 jenis mangga segar dan busuk. berikut gambaran umum dari algoritma yolo untuk mendeteksi objek.

### 2.2.1. Object Localization

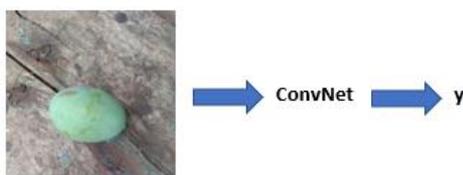
Untuk membangun deteksi objek, langkah awal yaitu belajar tentang pelokalan objek. Langkah-langkah untuk melakukan pelokalan objek yaitu *Image Classification, Classification with Localization, Defining the target label Y.*

### 2.2.2. Object Detection

Tujuan dari pelatihan adalah menentukan gambar berupa kelas mangga manalagi segar atau yang lainnya. Artinya, gambar Z hanyalah mangga manalagi segar. Jadi kita bisa mengambil gambar yang didalamnya terdapat mangga manalagi segar. Dengan label dataset training, kita kemudian dapat melatih ConvNet dengan memasukkan gambar, seperti salah satu gambar yang terdapat mangga manalagi ini. Dan kemudian tugas dari ConvNet adalah mengeluarkan y, 0 atau 1, apakah ada mangga manalagi atau tidak.

Tabel 1. Dataset Training

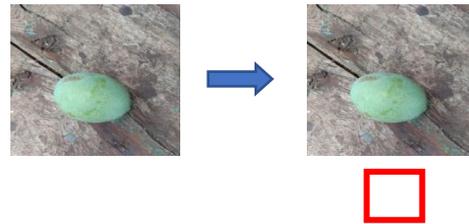
Z				
y	1	1	1	0



Gambar 5. Tugas ConvNet mengeluarkan nilai y

### 2.2.3. Sliding Windows Detection

Langkah awal yaitu memilih ukuran jendela tertentu, yang ditunjukkan pada gambar 6



Gambar 6. Gambar input dengan kotak pembatas persegi

Dan kemudian kotak pembatas persegi akan dimasukkan ke dalam ConvNet. Dan mungkin untuk wilayah kecil di kotak merah itu, akan tertulis TIDAK, kotak merah kecil itu tidak berisi mangga manalagi segar. Dalam Algoritma *Sliding Windows Detection*, yang kita lakukan adalah meneruskan sebagai masukan, gambar kedua yang sekarang dibatasi oleh kotak merah ini bergeser sedikit dan memasukkannya ke ConvNet. Jadi kami memberi porsi hanya wilayah gambar di kotak merah convNet dan menjalankan convNet lagi. Dan untuk gambar ketiga dan seterusnya.



Gambar 7 Langkah-langkah *Sliding Windows Detection*

### 2.2.4. Bounding Box Prediction

Pada bagian 3 (*Sliding Windows*), kami menggunakan implementasi konvolusional *Sliding Windows* dan yang lebih efisien secara komputasi tetapi penyimpanan memiliki masalah karena tidak menghasilkan *bounding box* yang paling akurat. Pada bagian ini, bagaimana kita bisa mendapatkan prediksi kotak pembatas kita agar lebih akurat.



Gambar 8. Merubah ukuran jendela geser



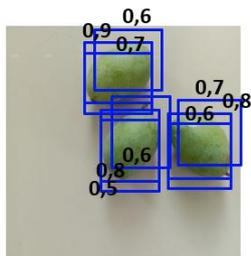
Gambar 9. Membandingkan prediksi bounding box



Gambar 10. Menghasilkan bounding box yang lebih akurat

### 2.2.5. Non-Max Suppression

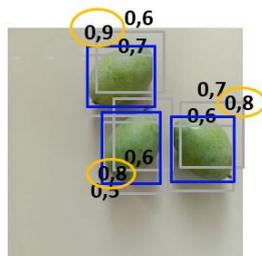
Non-Max Suppression merupakan salah satu teknik yang dapat meningkatkan output YOLO secara signifikan. Salah satu masalah dari Deteksi Objek adalah algoritma mungkin menemukan beberapa deteksi dari objek yang sama. Daripada mendeteksi objek hanya sekali, itu mungkin mendeteksinya beberapa kali. Perhatikan gambar di bawah ini:



Gambar 11. Algoritma YOLO mendeteksi objek beberapa kali

Inti dari Non-Max Suppression yaitu mengambil kotak dengan probabilitas maksimum dan menekan kotak terdekat dengan probabilitas non-maks. Ada beberapa poin tentang algoritma Non-Max Suppression yaitu:

1. Buang semua kotak yang memiliki probabilitas kurang dari atau sama dengan threshold yang ditentukan sebelumnya (katakanlah, 0,5)
2. Untuk kotak yang tersisa:
  - Pilih kotak dengan probabilitas tertinggi dan anggap itu sebagai prediksi keluaran
  - Buang kotak lain yang memiliki IoU lebih besar dari ambang batas dengan kotak keluaran dari langkah di atas
3. Ulangi langkah 2 hingga semua kotak diambil sebagai prediksi keluaran atau dibuang.



Gambar 12 . Menampilkan probabilitas yang nilainya besar

## 2.3 Implementasi dan Pengujian

### 2.3.1. Dataset training

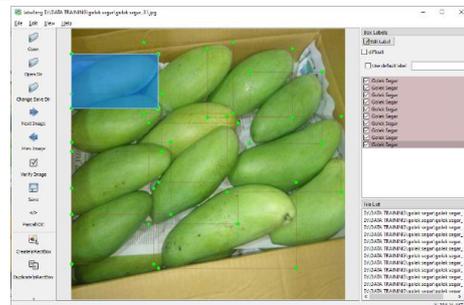
Berikut ini adalah dataset untuk mangga golek segar, mangga golek busuk, mangga manalagi segar, mangga manalagi busuk, mangga gedong segar, dan mangga gedong busuk. Setiap kelas dataset terdiri dari masing-masing 200 data gambar. 200 Mangga golek segar, 200 mangga golek busuk, 200 mangga manalagi segar, 200 mangga manalagi busuk, 200 mangga gedong segar, dan 200 mangga gedong busuk



Gambar 13. Dataset training mangga golek segar

### 2.3.2. Proses Pelabelan

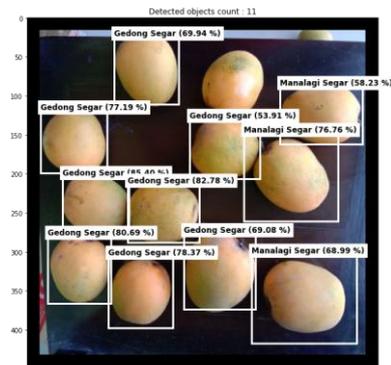
Untuk proses pelabelan, disini menggunakan tools labelling. LabelImg adalah alat anotasi gambar grafis.



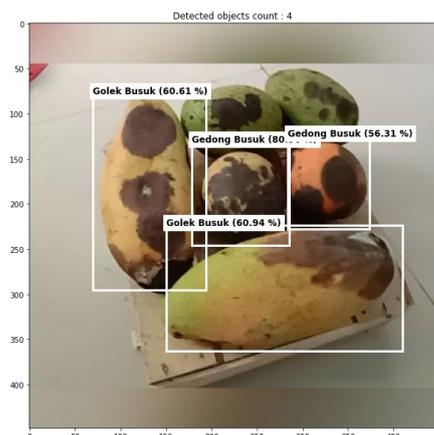
Gambar 14. Proses pelabelan dataset training dengan tool labelling.py

### 2.3.3. Pengujian

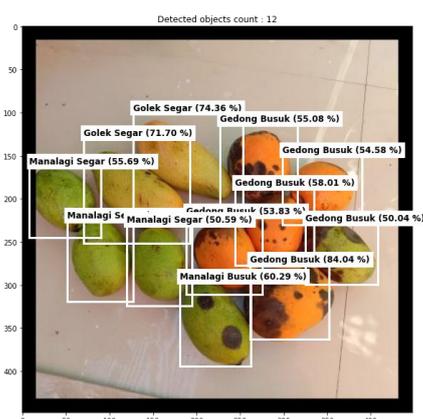
Pengujian dilakukan dengan tiga scenario, yang pertama semuanya mangga segar, kemudian semuanya mangga busuk, dan yang terakhir semua mangga segar dan busuk. Berikut contoh 3 hasil pengujian dengan melakukan tiga scenario



Gambar 15. Hasil deteksi mangga dari sekenario pengujian mangga segar



Gambar 16. Hasil deteksi mangga dari sekenario pengujian mangga busuk



Gambar 17. Hasil deteksi mangga dari sekenario pengujian mangga segar dan busuk

Hasil dari pengujian data gambar tersebut adalah identifikasi mana mangga yang segar dan mana mangga busuk. Untuk menghitung tingkat akurasi menggunakan 3 confusion matrix. Confusion matrix yang pertama, mendeteksi berapa banyak objek mangga yang terdeteksi dan yang tidak, kemudian confusion matrix yang kedua mendeteksi berapa banyak objek yang terdeteksi busuk dan yang segar, dan confusion matrix yang ketiga yaitu mendeteksi objek dari tiap-tiap kelas, sehingga dari ketiga confusion matrix tersebut bisa dihitung berapa tingkat akurasinya baik precision, recall, maupun f1-score

### 3. KESIMPULAN

Dari ujicoba dengan menggunakan skenario didapatkan berapa akurasi dari tiap-tiap ujicoba skenario tersebut. Skenario pertama yaitu mangga segar semua didapatkan tingkat akurasi 80%, precision 82%, dan recall 87%, didapatkan F1-score 84%. Kemudian skenario yang kedua yaitu mangga busuk semua didapatkan tingkat akurasi 76%, precision 76%, dan recall 87%, didapatkan F1-score 81%. Dan yang ketiga yaitu mangga segar dan busuk, didapatkan tingkat akurasi 73%, precision 66%, dan recall 81%, didapatkan F1-score 73%.

dapat disimpulkan bahwa hasil penelitian ini masih tergolong underfitting. Hal ini dikarenakan masih butuh banyak dataset yang lebih banyak dan variannya yang mempunyai ciri-ciri yang ada kemiripan masing-masing kelasnya.

### PUSTAKA

- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1–6.
- Dharmadi, R., & medium.com. (2018). *Convolutional Layer*. <https://Medium.Com/.https://medium.com/nodeflux/mengenal-convolutional-layer-dan-pooling-layer-3c6f5c393ab2>
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2), 303–338.
- Grishick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.
- Jmour, N., Zayen, S., & Abdelkrim, A. (2018). Convolutional neural networks for image classification. *2018 International Conference on Advanced Systems and Electric Technologies (IC\ASET)*, 397–402.
- Lan, W., Dang, J., Wang, Y., & Wang, S. (2018). Pedestrian detection based on YOLO network model. *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, 1547–1551.
- Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7263–7271.
- Zitnick, C. L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. *European Conference on Computer Vision*, 391–405.