

# Application of Convolution Neural Network Algorithm for Rice Type Detection Using Yolo v3

<sup>1</sup>Kiki Ahmad Baihaqi, <sup>2</sup>Yana Cahyana

<sup>1,2</sup>Universitas Buana Perjuangan Karawang  
Email: kikiahmad@ubpkarawang.ac.id

## Abstract

*Rice is a staple food that contains a lot of energy for human life. There are several types of rice that are often sold in rice shops in general, namely IR42 rice, Pera rice, glutinous rice and Pandan fragrant rice. For now, there are still many people who do not recognize the 4 types of rice, especially millennials, for this reason, research is carried out on the introduction of rice types. The purpose of this study is to make it easier for buyers to identify the type of rice that is in the rice shop so as to minimize fraud by rice traders. The method used in this study is the YOLO (You Only Look Once) v3 method for detecting rice types. The implementation of the image detection process using YOLO (You Only Look Once) v3 has been tested for 12 samples. Based on the results of testing 12 detection experiments on digital image objects, it was obtained 100% where in the picture there were 4 types of rice, 4 grains of rice and 3 types of rice shapes.*

**Keywords:** Convolution Neural Network, YOLO(You Only Look Once) Method, Rice

## 1. INTRODUCTION

Pertanian merupakan sejenis proses produksi yang khas yang didasarkan atas proses pertumbuhan tanaman dan hewan. Kegiatan-kegiatan produksi didalam setiap usaha tani merupakan suatu bagian usaha, dimana biaya dan penerimaan penting (A.T Mosher, 1968:19). Karawang merupakan daerah penghasil padi kedua terbesar di Tingkat Nasional. Pada saat ini, perkembangan ilmu pengetahuan dan teknologi pada saat ini yang begitu pesatnya, banyak sekali alat-alat yang dapat mempermudah manusia dalam menyelesaikan pekerjaannya secara otomatis.

Computer vision adalah salah satu bidang yang mendukung pekerjaan manusia sebagai contohnya di sektor pertanian. Penelitian yang terdahulu membahas tentang mendeteksi buah menggunakan teknologi computer vision. Seperti mendeteksi tanaman tebu dengan Algoritma yang digunakan untuk mendeteksi tanaman tebu tersebut adalah algoritma CNN berbasis YOLO (You Only Look Once). Aplikasi ini berhasil untuk mendeteksi tebu sebesar 95% dengan nilai threshold 0.1, menghasilkan skor precision sebesar 1.00, skor recall sebesar 0.95 dan skor accuracy sebesar 0.95 pada tebu.

You Only Look Once (YOLO) adalah sebuah algoritme yang dikembangkan untuk mendeteksi sebuah objek secara real-time. YOLOv3 merupakan metode mesin teknologi deteksi jenis padi yang sering digunakan karena 3 x lebih cepat yang beroperasi dalam 22 ms pada Map (mean Average Precision) dan Fps pada yolo dasar 45 frame. Dengan digunakannya penelitian ini bertujuan untuk

mengembangkan deteksi jenis padi secara akurat terhadap berbagai citra. Bahasa python digunakan pada google colab karena mendukung GPU (Graphics Processing Unit) tesla K80 hingga 12GB, konfigurasi dengan adanya GPU dengan data lebih dari 2.000 bisa menghemat waktu dan mendapatkan akurasi yang tinggi.

## 2. METHODS

### 2.1 Analisa dan Pengumpulan Data

Pada tahap analisis data yang digunakan adalah menggunakan metode algoritma YOLO (You Only Look Once) versi 3. Jenis penelitian ini adalah melakukan eksperimen dengan tujuan untuk menciptakan pengetahuan baru dari teori-teori yang sudah ada. Penerapan pada penelitian ini adalah dimana dataset yang digunakan untuk pengujian bersumber dari pengumpulan gambar jenis Beras yang bertempat di Pasar Beras Karawang, kemudian dilakukan pengujian dataset pada metode YOLOv3 untuk melihat hasil akurasi deteksi objek jenis padi Beras. Dari hasil foto tersebut di proses ke Labeling kemudian diberi anotasi pada setiap gambar beras untuk mendapat koordinat objek yang akan dilatih.

### 2.2 YOLO (You Only Look Once)

You Only Look Once (YOLO) adalah sebuah algoritme yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Sistem pendeteksian yang dilakukan adalah dengan menggunakan repurpose classifier atau localizer untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala. Daerah dengan citra yang diberi score paling tinggi akan dianggap sebagai sebuah pendeteksian (Unsky, 2017). Pada tahun 2015 Joseph Redmon pertama kalinya memperkenalkan YOLO (You Only Look Once). YOLO mempunyai arsitektur algoritma CNN (Convolutional Neural Network). Tahun 2017, Joseph Redmon dan Ali Farhadi merilis YOLO v2 dengan meningkatkan akurasi dan kecepatan algoritma. Tahun 2018, Joseph Redmon dan Ali Farhadi merilis YOLO v3 yang memiliki performance semakin bagus pada deteksi objek.

#### 2.2.1 Ide Dasar

YOLO membagi gambar atau video yang diinput menjadi  $S \times S$  grid. Jika titik tengah koordinat pada GT (Ground Truth) suatu objek jatuh ke dalam grid, maka grid tersebut bertanggung jawab untuk mendeteksi suatu objek. Grid yang terbagi dalam beberapa bagian tersebut akan di prediksi bounding box yang terdapat objek di dalamnya lalu YOLO akan menyelesaikan semua masalah dalam satu proses.

#### 2.2.2 Struktur Jaringan

YOLO menggunakan  $1 \times 1$  convolutional layer ( untuk integrasi antar channel) +  $3 \times 3$  convolutional layer sebagai pengganti permulaan modul. Jaringan pada YOLO v1 mengandung 24 convolutional layers dan 2 full connection layers.

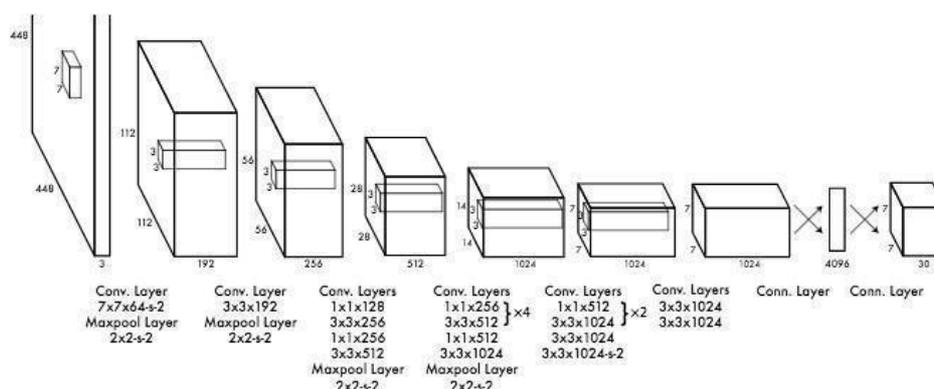


Figure 1 YOLO v1 Network Architecture

### 2.2.3 Proses Tahapan

Membagi citra dalam region  $S \times S$ . Tiap grid akan diprediksi bounding box beserta nilai confidence. Tiap bounding box memiliki 5 nilai informasi  $x, y, w, h$  dan  $c$ . Nilai  $x$  dan  $y$  adalah kordinat titik tengah bounding box yang terprediksi, nilai  $w$  dan  $h$  adalah rasio ukuran lebar tinggi relative terhadap grid, dan  $c$  adalah nilai confidence bounding box tersebut. Tiap grid akan memprediksi nilai class probabilitas jika diprediksi terdapat objek di dalamnya. Dalam proses pengujiannya nilai class probability di kalikan dengan nilai confidence dari bounding box.

Dengan keterbatasan dataset YOLO yang hanya menyimpan class 80 objek dibantu dengan arsitektur algoritma CNN untuk memenuhi kekurangan tersebut . YOLO merupakan algoritma yang berdasarkan regession Dimana dalam sekali proses running tersebut akan menghasilkan output prediksi dan bounding box untuk setiap objek.

### 2.3 Training Dataset

Training dataset sendiri menggunakan Framework Darknet. Darknet yang digunakan dalam penelitian ini adalah Darknet53,conv.74. Untuk mendapatkan hasil dataset dengan darknet ada dua cara yaitu menggunakan program google colab dan komputasi GPU/CPU. Proses training dataset pada gambar yang telah diberi label. Dari proses label tersebut yang dilatih akan mendapatkan pengenalan model pada program, seperti pada gambar dibawah ini:

```

82 Avg (IOU: 0.940923), count: 4, class_loss = 0.000000, iou_loss = 0.013815, total_loss = 0.013816
94 Avg (IOU: 0.929777), count: 12, class_loss = 0.410338, iou_loss = 0.123717, total_loss = 0.534055
106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000

82 Avg (IOU: 0.944390), count: 9, class_loss = 0.012379, iou_loss = 0.020959, total_loss = 0.033338
94 Avg (IOU: 0.969156), count: 6, class_loss = 0.000011, iou_loss = 0.019445, total_loss = 0.019456
106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000007, iou_loss = 0.000000, total_loss = 0.000007

82 Avg (IOU: 0.946060), count: 8, class_loss = 0.000102, iou_loss = 0.011552, total_loss = 0.011654
94 Avg (IOU: 0.965646), count: 8, class_loss = 0.000002, iou_loss = 0.011220, total_loss = 0.011222
106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000

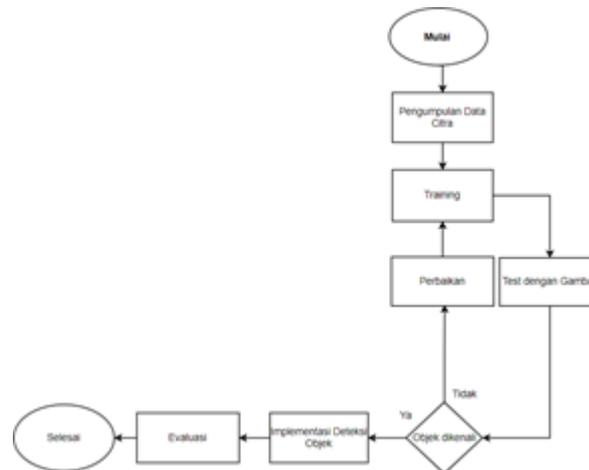
82 Avg (IOU: 0.929976), count: 4, class_loss = 0.000000, iou_loss = 0.012535, total_loss = 0.012535
94 Avg (IOU: 0.943882), count: 12, class_loss = 0.000000, iou_loss = 0.023454, total_loss = 0.023454
106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000

82 Avg (IOU: 0.941519), count: 8, class_loss = 0.000000, iou_loss = 0.015514, total_loss = 0.015514
94 Avg (IOU: 0.946021), count: 8, class_loss = 0.000000, iou_loss = 0.016201, total_loss = 0.016202
106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000

82 Avg (IOU: 0.954967), count: 5, class_loss = 0.000012, iou_loss = 0.007373, total_loss = 0.007385
94 Avg (IOU: 0.935940), count: 11, class_loss = 0.000040, iou_loss = 0.031577, total_loss = 0.031617
    
```

Figure 2 YOLO v1 Output Pelatihan

## 2.4 Implementasi



**Figure 3 Flowchart System**

Pada gambar 3.2 blok pertama adalah pengumpulan data citra pada setiap jenis Beras yang akan di deteksi lalu pemberian anotasi pada setiap gambar yang akan dilatih. Blok kedua adalah proses training yang menimplementasikan Algoritma YOLO (You Only Look Once). YOLO v3 adalah Sistem pendeteksian yang dilakukan adalah dengan menggunakan repurpose classifer atau localizer untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala. Daerah dengan citra yang diberi score paling tinggi akan dianggap sebagai sebuah pendeteksian. Blok ke tiga adalah pengujian dilakukan untuk mengetahui apakah penerapan metode YOLO mendapatkan nilai akurasi yang baik untuk pengenalan jenis Beras. Blok ke empat adalah proses perbaikan apabila hasil dari pengujian mendapatkan kesalahan yang menyebabkan proses pendeteksian gagal. Hal ini biasanya disebabkan karena proses training yang belum optimal. Blok ke lima adalah implemenstasi pendeteksian langsung ke objek Beras yang telah disiapkan berupa gambar / foto jenis Beras . Blok ke enam adalah Evaluasi hasil dilakukan agar mengetahui akurasi dari model dan juga hasil dari pengenalan gambar. Hal ini dilakukan agar mengetahui tingkat keberhasilan dari model dalam mengenali gambar jenis Beras.

## 2.4 Pengujian

Proses pengujian yang di terapkan adalah dengan mengambil gambar dari bentuk dari masing-masing jenis beras. Hal ini dilakukan agar mengetahui tingkat keberhasilan dari model dalam mengenali gambar jenis Beras. Cara mendapatkan nilai tingkat akurasi adalah dengan rumus.

$$Akurasi = \frac{Jumlah\ Data\ Benar}{Jumlah\ Seluruh\ Data} \times 100\% \quad (1)$$

### 3. RESULTS AND DISCUSSION

Pada penelitian ini, akan dipaparkan hasil dari pengujian dan Analisis di dalam urutan system dan proses implementasi yang telah dibuat. Dalam penelitian ini dataset yang digunakan adalah dataset hasil dari pengambilan gambar yang dilakukan di pasar Beras Karawang. Dari hasil foto tersebut di proses ke Labelimg kemudian diberi anotasi pada setiap gambar beras tersebut untuk mendapatkan koordinat objek yang akan dilatih. Pengambilan gambar dibagi menjadi 3 jenis yaitu gambar beras utuh, gambar beras patahan dan gambar beras menir. Selanjutnya setiap gambar akan diberikan label dengan hasil anotasi Format <Object Class Id> <center- x> <center-y> <Width><Height> yang mendefinisikan kelas dan koordinat objek yang akan dilatih setiap barisnya. Label dibuat yang pada gambar menggunakan aplikasi Labelimg yang disimpan dalam bentuk file (\*.txt) yang ditempatkan didalam file gambar di folder “dataset beras”.

Berbagai macam proses training dan konfigurasi yang ada pada YOLO mengharuskan untuk terlebih dahulu memperhatikan proses pengenalan jenis objek. Nama file konfigurasi pelatihan pada Yolo adalah file cfg. file obj.data berisikan classes serta lokasi dari file train, validasi/test, obj.names dan juga lokasi untuk membackup data file bobot hasil dari pelatihan yang selanjutnya akan disimpan. Obj.names adalah file yang berisikan label jenis-jenis beras. Umumnya pada dataset COCO yang dilatih adalah 80 classes. Karena yang digunakan disini hanya 4 classes, yaitu beras Ir42, beras Pera, beras Ketan dan beras Pandan Wangi. Selanjutnya akan dipilih secara acak untuk proses training dengan proses 80% data training dan 20% data testing. Training dataset sendiri menggunakan Framework Darknet.

Hasil pelatihan selama 33 jam mendapatkan avg loss 0,0500 pada iterasi 6000 batch. Semakin kecil avg loss yang didapatkan maka hasil deteksi yang didapatkan akan lebih akurat. Jumlah iterasi sangat berpengaruh pada hasil avg loss, maka dari itu penentuan jumlah iterasi sangat baik untuk proses deteksi. Ditunjukkan pada gambar di bawah ini.

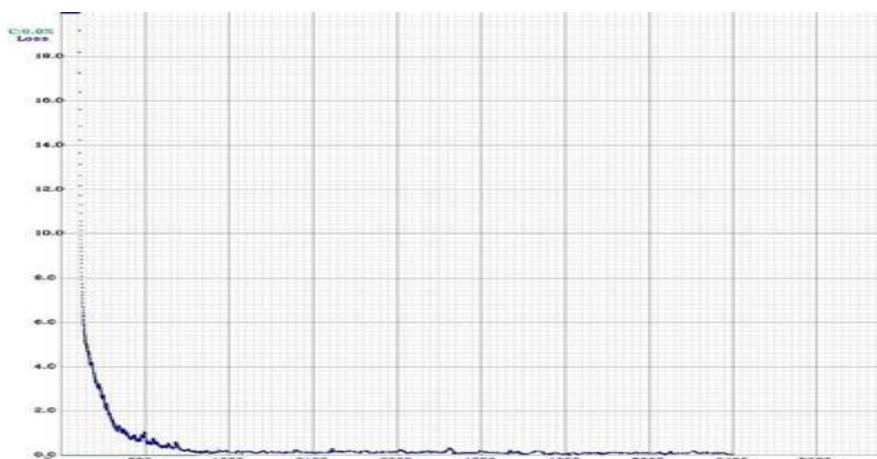
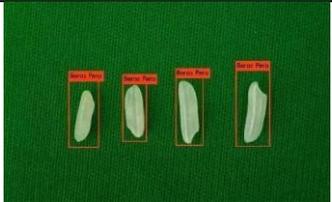
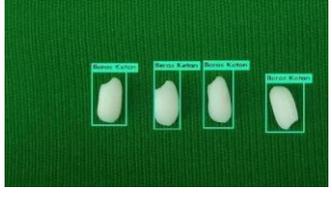
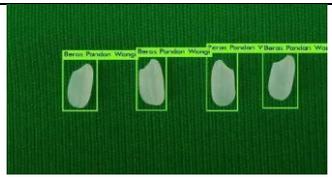
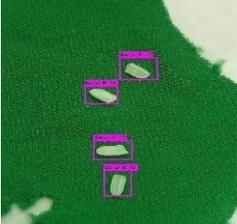
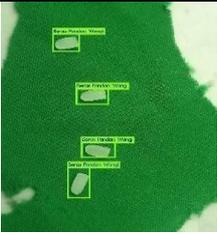
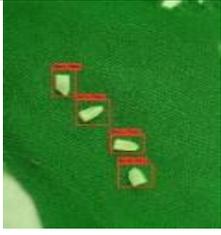
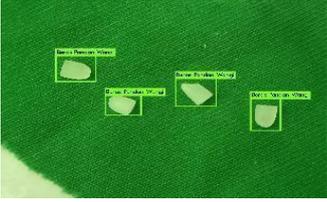


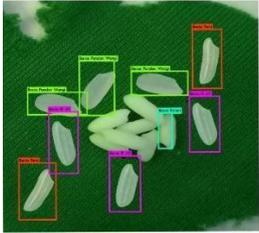
Figure 4 Avg Loss

Hasil pengujian dari 13 foto beras dalam 4 bentuk beras dan 4 jenis beras akan dipaparkan dengan tabel sebagai berikut.

Table 1 Hasil Pengujian

| No. | Bentuk Jenis Beras      | Gambar Hasil Objek  | Akurasi                                |
|-----|-------------------------|---|--|
| 1   | Beras Utuh IR42         |    | Terdeteksi ( <i>Confidence</i> : 100%) |
| 2   | Beras Utuh Pera         |    | Terdeteksi ( <i>Confidence</i> : 100%) |
| 3   | Beras Utuh Ketan        |   | Terdeteksi ( <i>Confidence</i> : 100%) |
| 4   | Beras Utuh Pandan Wangi |  | Terdeteksi ( <i>Confidence</i> : 100%) |
| 5   | Beras Patahan IR42      |  | Terdeteksi ( <i>Confidence</i> : 100%) |
| 6   | Beras Patahan Pera      |  | Terdeteksi ( <i>Confidence</i> : 100%) |

|    |                           |   |                                       |
|----|---------------------------|---|---------------------------------------|
| 7  | Beras PatahanKetan        |    | Terdeteksi ( <i>Confidence:</i> 100%) |
| 8  | Beras PatahanPandan Wangi |    | Terdeteksi ( <i>Confidence:</i> 100%) |
| 9  | Beras Menir IR42          |   | Terdeteksi ( <i>Confidence:</i> 100%) |
| 10 | Beras Menir Pera          |  | Terdeteksi ( <i>Confidence:</i> 100%) |
| 11 | Beras Menir Ketan         |  | Terdeteksi ( <i>Confidence:</i> 100%) |
| 12 | Beras Menir Pandan Wangi  |  | Terdeteksi ( <i>Confidence:</i> 100%) |

|    |                |   |   |
|----|----------------|---|---|
| 13 | Beras Campuran |  | <p>Terdeteksi (<i>Confidence</i>)</p> <ul style="list-style-type: none"> <li>• Pandan wangi :100%</li> <li>• Pera : 100%</li> <li>• Ketan : 55%</li> <li>• IR42 : 99%</li> </ul> <p>Total dari 15 jumlah beras di foto beras hanya 60% yang terdeteksi)</p> |
|----|----------------|---|---|

Gambar yang dijadikan uji dalam deteksi ini adalah 12 sample dan semua gambar tersebut terdeteksi dengan benar. Sehingga hasil akurasinya adalah:

$$\text{Akurasi} = \frac{12}{12} \times 100\%$$

Dengan hasil 100 % terdeteksi maka pengenalan jenis beras tersebut dinyatakan berhasil. Proses selanjutnya adalah mengecek tiap iterasi pada file backup yang ada di google drive. Namun, dalam pendeteksian beras yang di tumpuk hanya 60% saja yang terdeteksi.

#### 4. KESIMPULAN

Berdasarkan proses tahapan penelitian dalam Deteksi 4 jenis beras yaitu beras Ir42, beras Pera, beras Ketan dan beras Pandan Wangi dengan metode Algoritma YOLOv3 dapat disimpulkan bahwa pada penerapan algoritma YOLO (You Only Look Once)v3 dalam mendeteksi ke empat jenis Beras pada citra digital didalam pemograman python berhasil diimplementasikan. Berdasarkan hasil pengujian Ketika dilakukan 12 kali pendeteksian ke objek citra digital pada posisi gambar beras berjajar atau tidak tertumpuk hasilnya akan didapatkan nilai akurasi 100%. Namun, jika posisi gambar beras bertumpuk maka nilai akurasi yang di dapat hanya 60%. Jenis beras dapat ditambahkan dengan varietas lain seperti pertama, varietas Padi Hibrida contohnya beberapa jenis beras Sembada B3, Intani 1, Adirasi 64 dan yang lainnya. Yang kedua, varietas Padi Unggul contohnya jenis beras IR64, Cisadane, Ciherang dan masih banyak yang lainnya. Yang ketiga, varietas Padi Lokal contohnya jenis beras Indramayu, Srimulih, Gropak dan masih banyak lagi lainnya. Pengambilan citra dan label dilakukan dari berbagai arah dan posisi beras seperti meletakkan beras dengan posisi menyamping, tegak dan ditumpuk lebih banyak lagi. Menggunakan metode lain untuk proses deteksi dan membandingkan tingkat akurasi dari setiap metode.

#### REFERENCES

- [1] Abi Rachman Wasril, D. (2020). Pembuatan Pendekteksi Obyek Dengan Metode You Only Look Once (YOLO) untuk Automated Teller Machine (ATM). Majalah Ilmiah UNIKOM.

- [2] Alexey AB, "AlexeyAB/darknet: Windows and Linux version of Darknet Yolo v3 v2 Neural Networks for object detection (Tensor Cores are used)," GitHub. 2019.
- [3] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3," 2019 1st Int. Conf. Unmanned Veh. Syst. UVS 2019, pp. 1–6, 2019
- [4] Dedy Agung Prabowo, D. (2018). Deteksi dan Perhitungan Objek Berdasarkan Warna menggunakan COlor Objek Tracking. Jurnal Pseudocode, Volume V nomor 2.
- [5] Hr.Wibi Bagas, D. (2020). Deteksi Buah untuk Klasifikasi Berdasarkan Jenis Dengan Algoritma Berbasis YOLOv3. Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi), Vol.4 No.3.
- [6] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.
- [7] Muhammad Alfin Jimly Asshiddiqie, D. (2020). Deteksi Tanaman Tebu Pada Lahan Pertanian Menggunakan Metode Convolutional Neural Network. Jurnal Informatika dan Sistem Informasi Vol.1 No.1.
- [8] M. Ju, H. Luo, Z. Wang, B. Hui, and Z. Chang, "The application of improved YOLO V3 in multi-scale target detection," Appl. Sci., vol. 9, no. 18, 2019.
- [9] Pandu Satrio, "Pengaruh Drone Terhadap Citra Yang Dihasilkan Pada Pemantauan Tanaman Padi" Sunita Nayak, "Training YOLOv3 : Deep Learning based Custom Object Detector | Learn OpenCV," Learn OpenCV, 2019.