

IMPLEMENTASI KLASIFIKASI JENIS TEMBAKAU CACAH DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CV MALIKUS

Muhammad Pachlevy Arirul, Amalia Eka Rakhmania, Sri Wahyuni Dali

Teknik Elektro, Politeknik Negeri Malang
Jl. Soekarno Hatta 9 Malang Jawa Timur, Indonesia
pahleviarirul69@gmail.com

ABSTRAK

CV. Malikus adalah sebuah pabrik rokok yang berlokasi di Kabupaten Blitar, Jawa Timur. Dalam produksi rokok CV Malikus membeli Tembakau cacah yang sudah jadi dan dibeli dari petani tembakau dari berbagai provinsi. CV Malikus juga memiliki peran dalam mendorong ekonomi negara Indonesia dalam produksi rokok kretek untuk masyarakat kalangan menengah bawah dengan menjual rokok dengan harga yang terjangkau dan merekrut tenaga kerja dengan batas skill yang rendah. Pembuatan Implementasi Klasifikasi Jenis Tembakau Cacah Dengan Metode *Convolutional Neural Network* (CNN) Pada CV. Malikus ini didasari oleh kebutuhan mitra yaitu CV. Malikus yang memiliki tujuan dalam produksi rokok untuk meningkatkan efisiensi dalam produksi dan membantu kinerja karyawan dengan keterbatasan indera mata serta meningkatkan Sumber Daya Manusia (SDM) yang mampu bersaing di perkembangan teknologi deep learning, alat ini digunakan sebagai pembantu efisiensi dalam produksi dengan menggunakan mikrokontroler. Komponen yang digunakan seperti Logitech C270 HD dan Raspberry Pi 5. Komponen-komponen ini digunakan sebagai sistem utama dan untuk Bahasa pemrograman menggunakan Python, Google Colab dan Roboflow untuk pembuatan Dataset dan Coding. Komponen-komponen tersebut akan digunakan untuk membuat alat Klasifikasi Tembakau Cacah dengan menggunakan metode CNN (*Convolutional Neural Network*).

Kata kunci : Tembakau, YOLO, Raspberry Pi 5, *Convolutional Neural Network*, Dataset

1. PENDAHULUAN

CV. Malikus merupakan Perusahaan di bidang produksi rokok untuk memproduksi rokok kretek di area Kabupaten Malang dan sekitarnya. Dalam produksi rokok CV. Malikus membeli Tembakau cacah yang sudah jadi dan dibeli dari petani tembakau dari berbagai provinsi. CV. Malikus juga memiliki peran dalam mendorong ekonomi negara Indonesia dalam produksi rokok kretek untuk masyarakat kalangan menengah bawah dengan menjual rokok dengan harga yang terjangkau dan merekrut tenaga kerja dengan batas skill yang rendah.

Terdapat permasalahan khusus yang dialami pegawai baru yang belum bisa mengenali jenis tembakau yang diproduksi oleh CV. Malikus. Oleh karena itu, untuk membantu pegawai baru dan pegawai yang berkebutuhan khusus seperti buta warna dibutuhkan suatu sistem yang dapat meringankan pekerjaan mereka dalam proses produksi rokok. Dalam proses produksinya, CV. Malikus masih menggunakan cara manual dengan indera manusia untuk memilah jenis tembakau cacah dalam produksi rokok mereka. Tujuan dibuatnya sistem ini untuk melakukan peningkatan fasilitas di perusahaan CV. Malikus sebagai alat bantu dalam produksi rokok. Sistem ini menggunakan Raspberry Pi 5 dan Webcam Logitech C270 HD yang berbasis bahasa pemrograman python. Hal ini bertujuan agar pekerja yang dilatih mampu memilah jenis tembakau cacah pada teknologi deep learning. dan mampu turut serta dalam peningkatan produktifitas perusahaan dibidang produksi.

Maka dari itu perangkat yang digunakan berupa microcomputer Raspberry PI 5 dengan efisiensi daya

yang rendah dan memiliki portabilitas agar tidak memakan banyak tempat [1]. Model deep learning yang digunakan yaitu model YOLO yang berbasis dari *Convolutional Neural Network* (CNN), Model ini dipilih untuk klasifikasi jenis tembakau, karena model YOLO dikenal dengan kecepatan dalam deteksi secara real time dan cukup ringan digunakan pada Microcomputer atau perangkat yang membutuhkan daya kecil [2]. Oleh karena itu, hasil akhir yang diinginkan adalah investasi pada peningkatan produktifitas pegawai untuk memahami dan menerapkan teknologi deep Learning yang difasilitasi oleh CV. Malikus melalui “Implementasi Klasifikasi Jenis Tembakau Cacah Dengan Metode *Convolutional Neural Network* (CNN) Pada CV. Malikus”.

2. TINJAUAN PUSTAKA

2.1. *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf tiruan yang dikhususkan untuk untuk melakukan berbagai tugas, seperti klasifikasi citra, deteksi objek, dan segmenasi. *Convolutional Neural Network* merupakan pengembangan dari *Multi Layer Perceptron* (MLP) dan merupakan salah satu algoritma dari *deep learning*. CNN pertama kali diperkenalkan oleh Yann LeCun, Léon Bottou, Yoshua Bengio, dan Patrick Haffner pada tahun 1998. Ada beberapa konsep penting yang terkait dengan CNN yaitu *Convolutional Layer* Dan *Pooling Layer*. Metode CNN memiliki hasil paling signifikan dalam pengenalan citra, hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada visual

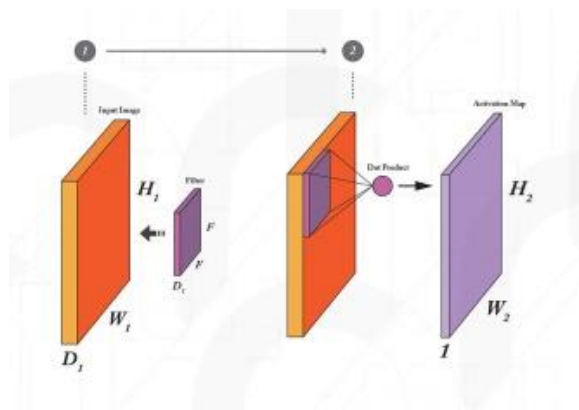
cortex manusia, sehingga memiliki kemampuan mengolah informasi citra [3].

2.2. Deep Learning

Deep learning (DL) yang merupakan sebuah teknik berbasis jaringan saraf tiruan telah banyak digunakan dalam beberapa tahun terakhir sebagai salah satu metode implementasi *Machine Learning* (ML) model *deep learning* dapat mempelajari komputasinya sendiri dengan menggunakan otaknya sendiri. *Deep learning* dirancang untuk terus menganalisa data seperti pada otak manusia dalam mengambil keputusan. Pada *deep learning* lapisan pembelajaran dipresentasikan melalui model yang disebut *neural networks*, dalam harfiah lapisan ini ditumpuk di atas satu sama lain. *Deep learning* memiliki suatu pandangan baru tentang representasi pembelajaran dari data yang menekankan pada lapisan pembelajaran (*layers*), semakin berturut-turut lapisan yang ada maka hasilnya akan lebih merepresentasikan objek yang ada. Menurut LeCun et al, *deep learning* menemukan struktur rumit dalam kumpulan data besar dengan menggunakan algoritma backpropagation untuk menunjukkan bagaimana mesin harus mengubah parameter internal yang digunakan untuk menghitung representasi di setiap lapisan dari representasi di lapisan sebelumnya [4].

2.3. Convolutional Layer

Proses konvolusi memanfaatkan apa yang disebut sebagai filter. Seperti layaknya 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. Gambar input CNN selalu berbentuk kotak. Proses untuk gambar non-rectangular masih belum diketahui. Filter pun mengikuti karakteristik kotak tersebut [5].

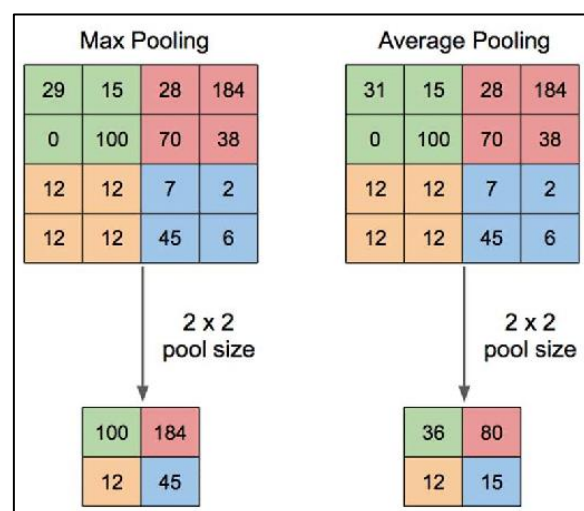


Gambar 1. Proses *convolutional layer*

2.4. Pooling Layer

Pooling layer pada *convolutional neural network* (CNN) adalah lapisan yang digunakan untuk mereduksi dimensi spasial dari peta fitur yang dihasilkan oleh lapisan konvolusi. Hal ini dilakukan

dengan menjalankan operasi pengelompokan (*pooling*) seperti maksimum atau rata-rata pada wilayah-wilayah tertentu dari peta fitur. *Pooling layer* membantu mengurangi jumlah parameter yang perlu dipelajari dan jumlah komputasi yang dilakukan dalam jaringan, serta membuat model lebih tahan terhadap variasi posisi fitur dalam gambar yang telah di-imput. Terdapat dua tipe utama dari *pooling layer*, yaitu *max pooling* dan *average pooling*. *Max pooling* memilih elemen maksimum dari setiap wilayah, sementara *average pooling* menghitung rata-rata elemen-elemen dalam wilayah tersebut. *Pooling layer* khususnya berguna untuk meningkatkan efisiensi komputasi dalam jaringan CNN [5]. Gambar *Max Pooling* dan *Average Pooling* ditunjukkan pada Gambar 2.



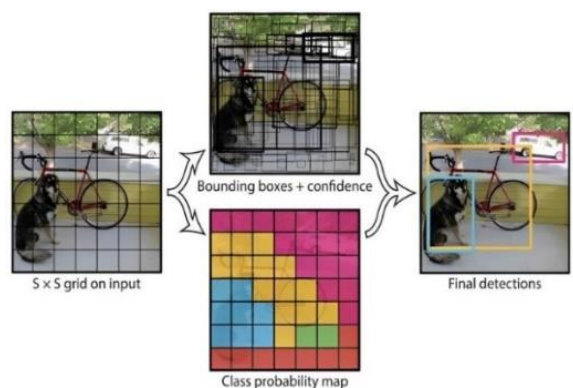
Gambar 2. *Max pooling* dan *Average pooling* [6]

2.5. YOLO (You Only Look Once)

YOLO merupakan algoritma yang dapat digunakan mendeteksi suatu objek secara *real-time*. Prinsip kerja YOLO adalah membagi gambar input menjadi beberapa kotak dan memprediksi seberapa luas setiap kotak dan seberapa mungkin setiap kotak akan ada. Setiap sel kotak memprediksi *bounding box* dan *confidence score*-nya jika sebuah objek pusatnya jatuh di salah satu sel kotak. *Confidence score* adalah besarnya persentase keyakinan dan akurasi model bahwa objek pada kotak tersebut terletak. x, y, w, h , dan *confidence* adalah lima prediksi di *bounding box*. Koordinat (x,y) mewakili pusat dari kotak relatif ke batas sel grid. Koordinat (w,h) atau lebar dan tinggi mewakili pusat dari kotak relatif ke gambar. Serta *confidence* mewakili *Intersection over Union* antara kotak prediksi dan kotak *ground truth*. *Ground truth* adalah data acuan yang menjadi sumber kebenaran untuk membandingkan hasil prediksi dari suatu model atau sistem dengan data yang sebenarnya. Pada YOLO, terdapat parameter epoch yaitu mengacu pada satu kali melatih model dengan satu kali iterasi melalui seluruh dataset pelatihan [7].

Proses pelatihan model YOLO dengan cara yaitu dengan mengambil citra secara manual atau

mengambil dataset yang sudah ada. Selanjutnya citra yang sudah ada dilakukan Anotasi atau Pelabelan pada objek citra. Selanjutnya setelah melakukan pelabelan, objek yang sudah di label akan dilakukan pembagian menjadi tiga bagian yaitu *Train*, *Val* dan *Test*. Pada bagian *Train* adalah proses dimana model dilatih untuk mengenal objek atau citra yang sudah di anotasi atau dilabel sebelumnya. Pada bagian *val* adalah citra yang sudah dilatih sebelumnya akan dilakukan validasi untuk mengevaluasi kinerja model selama proses pelatihan. Selanjutnya pada bagian *Test* adalah langkah terakhir dimana model yang sudah dibuat sebelumnya akan dilakukan percobaan untuk menguji fungsionalitas dan performanya [8].



Gambar 3. YOLO (You Only Look Once)

2.6. Python

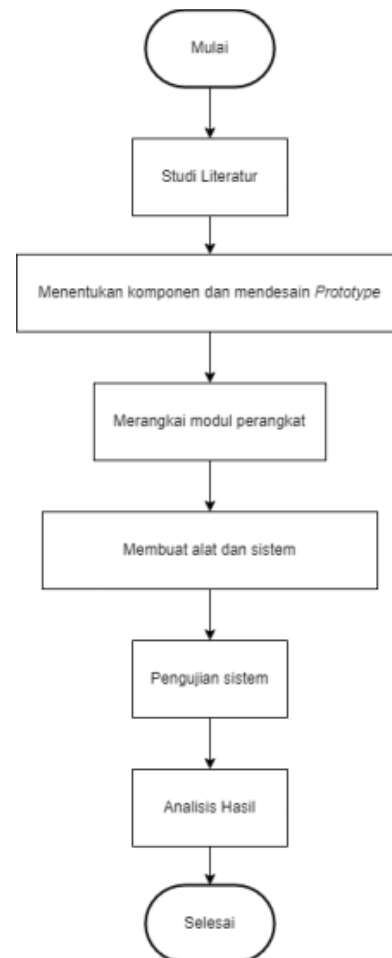
Python merupakan bahasa pemrograman yang dapat melakukan sejumlah eksekusi intruksi secara *interpretatif*. Python merupakan salah satu bahasa pemrograman yang paling mudah dipahami dan digunakan, oleh sebab itu, banyak pengembang dan bahkan masyarakat menggunakan bahasa ini untuk mengembangkan proyek nya ataupun sebagai bahan pembelajaran. Python dibuat oleh programmer asal Belanda bernama Guido Van Rossum pada tahun 1990. Sampai saat ini, versi dari python telah mencapai versi 3.10.13 [9].

2.7. Google Collab

Google Colab atau Google Colaboratory, adalah sebuah executable document yang dapat digunakan untuk menyimpan, menulis, serta membagikan program yang telah ditulis melalui Google Drive. Software ini pada dasarnya serupa dengan Jupyter Notebook berbentuk cloud yang dijalankan menggunakan browser, seperti Mozilla Firefox dan Google Chrome dan memungkinkan penggunaanya untuk menjalankan kode Python tanpa perlu melakukan proses instalasi dan setup lainnya, semua keperluan setting dan adjustment akan diserahkan ke cloud [10].

3. METODE PENELITIAN

Perancangan “Implementasi Klasifikasi Jenis Tembakau Cacah Dengan Metode *Convolutional Neural Network* (CNN) Pada CV. Malikus” ini termasuk ke dalam jenis penelitian Pembuatan dan Pengembangan. Jenis penelitian ini dilakukan untuk mendapatkan hasil dari objek penelitian yang berfokus pada implementasi teknologi deep learning untuk alat bantu produktifitas karyawan perusahaan. Penelitian ini didasari oleh masifnya perkembangan teknologi di bidang industri, namun masih minim keunggulan sumber daya manusia yang ada.



Gambar 4. Tahapan penelitian

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pembuatan Alat

Komponen yang digunakan adalah Raspberry PI 5 sebagai *microcomputer* sekaligus tempat proses klasifikasi citra tembakau, *Webcam Logitech C270* HD yang sudah terpasang di dalam wadah plastik sebagai sensor klasifikasi citra tembakau secara real time. Untuk wadah terbuat dari *corrugated plastic* dengan ukuran tinggi 32 cm dan lebar 32 cm. Hasil implementasi ditunjukkan pada Gambar 3 berikut:



Gambar 3. Pembuatan alat tampak luar



Gambar 4. Pembuatan alat tampak dalam

Hasil Penelitian ini adalah setelah pembuatan dataset di roboflow, tahap selanjutnya adalah training dataset. Citra Tembakau di training dengan bantuan Google Colab.

1. Training

Pada tahap ini gambar yang sudah dilabel selanjutnya dilakukan proses *training* yaitu model YOLO akan melakukan pelatihan pola dalam data mengenali foto-foto yang sudah dilabel sebelumnya. Dengan menggunakan *epoch* 100 dan *batch* 16.

```
# Train YOLOv8n
!yolo train model=yolov8n.pt data=/content/data.yaml epochs=100 imgsz=640 batch=16

Downloading https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8n.pt to 'yolov8n.pt'...
100% 6.25M/6.25M [00:00<00:00, 118MB/s]
Ultralytics YOLOv8 2.58 w/ Python-3.10.12 torch-2.1.2cu121 CUDA-0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8n.pt, data=/content/data.yaml, epochs=100, time=0
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100% 755k/755k [00:00<00:00, 15.7MB/s]
Overriding model.yaml nc=80 with nc=2

from n      params  module                                arguments
0  -1  1      464    ultralytics.nn.modules.conv.Conv      [3, 16, 3, 2]
1  -1  1      4672   ultralytics.nn.modules.conv.Conv      [16, 32, 3, 2]
2  -1  1      7368   ultralytics.nn.modules.block.C2f      [32, 32, 1, True]
3  -1  1     18568   ultralytics.nn.modules.conv.Conv      [32, 64, 3, 2]
4  -1  2     49664   ultralytics.nn.modules.block.C2f      [64, 64, 2, True]
5  -1  1     73984   ultralytics.nn.modules.conv.Conv      [64, 128, 3, 2]
6  -1  2    197632   ultralytics.nn.modules.block.C2f      [128, 128, 2, True]
7  -1  1     395424   ultralytics.nn.modules.conv.Conv      [128, 256, 3, 2]
8  -1  1     468288   ultralytics.nn.modules.block.C2f      [256, 256, 1, True]
9  -1  1    164488   ultralytics.nn.modules.block.SPPF     [256, 256, 5]
10 -1  1         0   torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
11 [-1, 6] 1         0   ultralytics.nn.modules.conv.Conv      [1]
```

Gambar 5. Proses *training* pada Google Colab

2. Validation

Pada tahap ini model YOLO yang sudah dilatih selanjutnya melakukan proses validasi disetiap beberapa *epoch* (satu siklus lengkap melalui seluruh data *training*), evaluasi ini mengukur rata – rata akurasi deteksi untuk semua kelas objek tembakau.

```
# Validate YOLOv8n
!yolo val model=/content/runs/detect/train/weights/best.pt data=/content/data.yaml

Ultralytics YOLOv8 2.58 w/ Python-3.10.12 torch-2.1.2cu121 CUDA-0 (Tesla T4, 15102MiB)
Model summary (fused): 188 layers, 3,486,488 parameters, 0 gradients, 0.1 GFLOPs
val: Scanning /content/valid/labels_c2f... 148 images, 0 backgrounds, 0 corrupt, 100% 148/148 [00:00<00:00, 711/s]
Class      Images  Instances  Box(r)    mAP50  mAP50-95  100% 10/10 [00:04<00:00, 2.11it/s]
  all         148         324      0.975     0.92     0.799
Bojonegoro    124         248      0.997     0.95     0.91
Madura         24         48      0.954     0.829   0.846
Results saved to runs/detect/val
Learn more at https://docs.ultralytics.com/modes/val
```

Gambar 6. Proses *validation* pada Google Colab

3. Predict

Pada tahap ini model YOLO yang sudah divalidasi selanjutnya akan dilakukan tahap *predict*, selanjutnya model YOLO akan dievaluasi menggunakan citra foto dari hasil *validation* yang akan digunakan. Proses Validation pada Google Colab ditunjukkan pada Gambar 6.

```
# Run inference on an image with YOLOv8n
!yolo predict model=/content/runs/detect/train/weights/best.pt source=/content/test/images conf=0.5

image 26/80 /content/test/images/image_067.jpg.rf.a5f73c73373814a62c1fb43908684cc.jpg: 640x640 2 Bojone
image 27/80 /content/test/images/image_067.jpg.rf.a5f73c73373814a62c1fb43908684cc.jpg: 640x640 2 Bojone
image 28/80 /content/test/images/image_067.jpg.rf.ee853853e047d47a8c908c0c72bbaa4c.jpg: 640x640 2 Bojone
image 29/80 /content/test/images/image_073.jpg.rf.8197f2c00809f7123460b0496055f.jpg: 640x640 2 Bojone
image 30/80 /content/test/images/image_073.jpg.rf.b57a02a58c97767d9e82884f43ac06b.jpg: 640x640 2 Bojone
image 31/80 /content/test/images/image_073.jpg.rf.eca8690027ed081cd718af9322126b.jpg: 640x640 2 Bojone
image 32/80 /content/test/images/image_073.jpg.rf.f080aaaf72708de4ea2307bc640a76.jpg: 640x640 2 Bojone
image 33/80 /content/test/images/image_081.jpg.rf.2c38657326682c5e1f97354213266.jpg: 640x640 2 Bojone
image 34/80 /content/test/images/image_081.jpg.rf.5a55048faa9fa0a18fd01f6a17fa8b.jpg: 640x640 2 Bojone
image 35/80 /content/test/images/image_081.jpg.rf.a5e0cf62a7c27a656c7878cbe81cbbd.jpg: 640x640 2 Bojone
image 36/80 /content/test/images/image_081.jpg.rf.c6f01b378f2f1eb954d149745114c20.jpg: 640x640 2 Bojone
image 37/80 /content/test/images/image_082.jpg.rf.88c668472fcb1f5832d359b6dc83.jpg: 640x640 2 Bojone
image 38/80 /content/test/images/image_082.jpg.rf.371873253d8e067b0dbdcab848deb831.jpg: 640x640 2 Bojone
image 39/80 /content/test/images/image_082.jpg.rf.52fa3d1254703e086f0f020652f5a1a.jpg: 640x640 2 Bojone
image 40/80 /content/test/images/image_082.jpg.rf.6509b576db29180dc6976d3f33ad884a.jpg: 640x640 2 Bojone
image 41/80 /content/test/images/image_092.jpg.rf.191f771a17895f9e9a0ca93dc077.jpg: 640x640 2 Bojone
image 42/80 /content/test/images/image_092.jpg.rf.a802685760265c8b04003368229fead7.jpg: 640x640 2 Bojone
image 43/80 /content/test/images/image_092.jpg.rf.b9a14c829e1356f746658314d015feab6.jpg: 640x640 2 Bojone
image 44/80 /content/test/images/image_092.jpg.rf.c4a6d8f736ff956fab1fafc1cbeafdb5.jpg: 640x640 2 Bojone
image 45/80 /content/test/images/image_094.jpg.rf.7c3b58a6d4c33f0571dc0b9e99a9a.jpg: 640x640 2 Bojone
image 46/80 /content/test/images/image_096.jpg.rf.daed72affc3680a24db583226268f61.jpg: 640x640 2 Bojone
```

Gambar 7. Proses validatin pada Google Colab

4.2. Hasil Pengujian Jarak Webcam ke Objek Tembakau

Pengujian ini dilakukan saat menguji model YOLO untuk klasifikasi tembakau Bojonegoro dan Madura dengan pencahayaan sebesar 1000 Lux. Hasil dari pengujian jarak webcam ke objek tembakau ditunjukkan pada Tabel 1 dan Tabel 2

Tabel 1. Hasil pengujian jarak webcam ke objek tembakau bojonegoro

No	Gambar Deteksi	Jarak Webcam (cm)	Keberhasilan Klasifikasi
1.		5	X
2.		10	X
3.		15	✓
4.		20	✓
5.		25	✓

No	Gambar Deteksi	Jarak Webcam (cm)	Keberhasilan Klasifikasi
6.		30	✓

Tabel 1 merupakan hasil pengujian jarak untuk Tembakau Bojonegoro pengujian ini dilakukan dengan mengubah jarak objek dari 5cm, 10 cm, 15 cm, 20 cm, 25 cm, dan 30 cm.

Table 2. Hasil pengujian jarak webcam ke objek tembakau madura

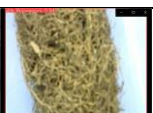
No	Gambar Deteksi	Jarak Webcam (cm)	Keberhasilan Klasifikasi
1.		5	✓
2.		10	✓
3.		15	✓
4.		20	✓
5.		25	X
6.		30	X

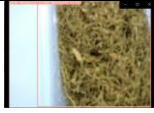
Tabel 2 merupakan hasil pengujian jarak untuk Tembakau Madura pengujian ini dilakukan dengan mengubah jarak objek dari 5cm, 10cm, 15 cm, 20 cm, 25 cm, dan 30 cm.

4.3. Hasil Pengujian Akurasi Model YOLO

Pengujian ini dilakukan untuk menguji akurasi dari model YOLO dengan jarak optimal 15 cm dengan pencahayaan sebesar 1000 Lux.

Tabel 3. Pengujian akurasi model YOLO

No	Gambar	Hasil Deteksi	Keberhasilan Klasifikasi
1.		Bojonegoro	✓

No	Gambar	Hasil Deteksi	Keberhasilan Klasifikasi
2.		Bojonegoro	✓
3.		Bojonegoro	✓
4.		Madura	X
5.		Madura	X
6.		Madura	✓
7.		Madura	✓
8.		Bojonegoro	X
9.		Madura	✓
10.		Madura	✓

Berdasarkan hasil dari tabel 3 model YOLO berhasil mendeteksi 7 tembakau dan gagal mendeteksi sebanyak 3 percobaan.

4.4. Hasil pengujian pencahayaan klasifikasi YOLO

Pengujian ini dilakukan untuk menguji akurasi dari model YOLO dengan mengubah tingkat pencahayaan.

Tabel 4. Hasil pengujian pencahayaan

No	Gambar	Tingkat pencahayaan	Keberhasilan Klasifikasi
1.		5 Lux	X

No	Gambar	Tingkat pencahayaannya	Keberhasilan Klasifikasi
2.		60 Lux	✓
3.		120 Lux	✓
4.		160 Lux	✓
5.		220 Lux	✓
6.		270 Lux	✓
7.		310 Lux	✓
8.		370 Lux	✓
9.		420 Lux	✓
10.		470 Lux	✓
11.		520 Lux	✓

Berdasarkan hasil dari tabel 4 model YOLO berhasil mendeteksi 10 tembakau dan gagal mendeteksi sebanyak 1 percobaan.

4.5. Hasil Pengujian Akurasi Model YOLO Berdasarkan Tingkat Epoch

Pada bagian ini akan disajikan hasil pengujian pengaruh jumlah pelatihan terhadap deteksi model YOLO yang ditunjukkan pada Tabel 5 dan pencahayaan yang digunakan adalah 310 lux

Tabel 5. Hasil pengujian *epoch* pada tembakau Bojonegoro

Epoch	Hasil Deteksi	Keberhasilan Klasifikasi
1		X
50		✓
100		✓
120		✓

Tabel 6. Hasil pengujian *epoch* pada tembakau Madura

Epoch	Hasil Deteksi	Keberhasilan Klasifikasi
1		✓
50		✓
100		✓
120		X

Berdasarkan pada Tabel 5 dan Tabel 6 hasil klasifikasi dengan *confidence* paling tinggi yaitu 100 *epoch*.

4.6. Hasil Pengujian Respon Eksekusi Sistem

Pada bagian ini akan disajikan hasil pengujian kecepatan eksekusi dalam menjalankan program model YOLO di Visual Studio Code pada Raspberry PI 5 dan PC.

Tabel 7. Hasil pengujian kecepatan eksekusi model YOLO

Percobaan	Raspberry PI 5	PC/Laptop
1	11,67 detik	24,32 detik
2	6,74 detik	4,53 detik
3	5,87 detik	6,64 detik
4	5,43 detik	6,74 detik
5	5,72 detik	6,27 detik
6	5,68 detik	6,76 detik
7	6,21 detik	6,41 detik
8	5,43 detik	5,32 detik
9	5,64 detik	5,56 detik
10	5,72 detik	5,89 detik

Berdasarkan hasil dari pengujian pada Tabel 7 kecepatan eksekusi kode pada Raspberry Pi 5 dan PC memiliki kesamaan kecepatan proses eksekusi model YOLO dengan waktu 5 detik.

4.7. Pembahasan

Berdasarkan masalah yang terjadi pada CV. Malikus dibuat alat untuk mengklasifikasi tembakau menggunakan model YOLO dan *input* berupa webcam dengan jarak optimal 15 cm untuk deteksi tembakau. *Epoch* yang digunakan adalah 100 *epoch* dan 16 batch. Karena berdasarkan pada Tabel 5 dan Tabel 6. tingkat deteksi lebih tinggi dan stabil berada di *epoch* 100. Untuk 16 *Batch* digunakan agar saat pelatihan pada Google Colab waktu yang dibutuhkan lebih efektif.

Berdasarkan dari data hasil pengujian jarak klasifikasi tembakau yang ditunjukan pada Tabel 1 dan Tabel 2, untuk klasifikasi Tembakau Bojonegoro pada jarak dibawah 10 – 5 cm model YOLO gagal mengklasifikasi jenis tembakau tersebut. Untuk Tembakau Madura model YOLO gagal mengklasifikasi jenis tembakau di jarak di atas 20 – 30 cm, karena di jarak tersebut terjadi perubahan warna yang menyebabkan gagal klasifikasi. Perubahan warna ini disebabkan oleh *white balance* yang diterima oleh webcam, sehingga warna berubah secara signifikan yang disebabkan oleh warna putih yang diterima oleh webcam. Untuk jarak optimal model YOLO dalam klasifikasi tembakau yaitu dijarak 15 – 20 cm.

Berdasarkan data hasil pengujian akurasi model YOLO pada tabel 4.3 tingkat keberhasilan model YOLO dalam mengklasifikasi jenis tembakau yaitu sebanyak 70% dengan *error* sebanyak 30%. Model YOLO berhasil mendeteksi benar sebanyak 7 percobaan, sedangkan sebanyak 3 percobaan gagal

klasifikasi. Hasil akurasi dan galat yang diperoleh dari persamaan berikut:

$$error = \frac{3}{10} \times 100\% = 30\%$$

$$akurasi = \frac{7}{10} \times 100\% = 70\%$$

Berdasarkan data dari hasil Pengujian *epoch* pada Tabel 5 dan Tabel 6 nilai *confidence* tertinggi di kedua jenis tembakau yaitu di 100 *epoch* yaitu Tembakau Bojonegoro bernilai 0.86 dan Tembakau Madura bernilai 0.94. Hal ini menunjukan model yang digunakan sudah optimal karena nilai *confidence* lebih dari 0,5 dan kedua tembakau tersebut terdeteksi dengan benar.

5. KESIMPULAN DAN SARAN

Penelitian yang telah dilakukan memberikan wawasan penting mengenai klasifikasi jenis tembakau cacah dengan menggunakan sensor webcam Logitech C270 HD. Dari pengujian yang dilakukan, terungkap bahwa sensor ini memiliki kemampuan mendeteksi tembakau pada jarak optimal 15 cm hingga 20 cm. Hal ini menunjukkan potensi sensor webcam dalam aplikasi klasifikasi tembakau. Selain itu, pencahayaan terbukti sebagai faktor krusial dalam proses klasifikasi. Dengan pencahayaan yang tepat, sensor webcam dapat mengklasifikasikan jenis tembakau dengan akurasi yang tinggi, yaitu 80% dari sampel yang diuji. Sistem *white balance* pada sensor memainkan peran penting dalam proses ini, dengan menyesuaikan penerimaan cahaya dan mengatur warna putih yang ditangkap, sehingga memungkinkan klasifikasi yang lebih akurat. Terakhir, penggunaan dataset pada model YOLOv8n menunjukkan bahwa untuk mencapai akurasi di atas 80%, diperlukan citra yang beragam, termasuk variasi dalam kondisi pencahayaan, sudut pengambilan gambar, dan jumlah citra yang lebih banyak. Temuan ini membuka jalan bagi pengembangan lebih lanjut dalam teknologi klasifikasi tembakau, dengan memanfaatkan kemajuan dalam pengolahan citra dan pembelajaran mesin.

Berdasarkan hasil pengujian yang telah dilaksanakan, para peneliti telah memberikan beberapa rekomendasi untuk penelitian mendatang. Salah satu saran utama adalah peningkatan kualitas sensor kamera atau webcam, khususnya dalam hal *white balance* dan *auto focus*. Peningkatan ini diharapkan dapat meningkatkan akurasi dalam klasifikasi tembakau. Selain itu, ditekankan pula pentingnya pengembangan dataset yang lebih beragam. Dengan variasi data yang lebih luas, diharapkan sistem klasifikasi jenis tembakau dapat bekerja dengan lebih tepat. Rekomendasi ini menunjukkan komitmen terhadap peningkatan teknologi yang dapat mendukung penelitian di bidang pertanian, khususnya dalam klasifikasi produk tembakau yang memiliki peran penting dalam industri.

DAFTAR PUSTAKA

- [1] U. S. Permatasari and I. R. Widiyari, "Analisis Routing Protokol Optimized Link State Routing (OLSR) Pada Raspberry Pi," *Aiti*, vol. 16, no. 2, pp. 151–164, 2020, doi: 10.24246/aiti.v16i2.151-164.
- [2] L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once)," 2021.
- [3] P. Adi Nugroho, I. Fenriana, and R. Arijanto, "IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA," *J. ALGOR*, vol. 2, no. 1, 2020, [Online]. Available: <https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- [4] M. Ikbal, R. Adi Saputra, K. Kunci Kendaraan Otonom, and P. Objek, "PENGENALAN RAMBU LALU LINTAS MENGGUNAKAN METODE YOLOV8," 2024.
- [5] R. Dharmadi, "Mengenal Convolutional Layer Dan Pooling Layer," *Nodeflux*, 2018, [Online]. Available: <https://medium.com/nodeflux/mengenal-convolutional-layer-dan-pooling-layer-3c6f5c393ab2>
- [6] M. Yani, B. Irawan, and C. Setiningsih, "Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail," *J. Phys. Conf. Ser.*, vol. 1201, no. 1, 2019, doi: 10.1088/1742-6596/1201/1/012052.
- [7] A. A. Soebroto, "Buku Ajar AI, Machine Learning & Deep Learning," 2019. [Online]. Available: <https://www.researchgate.net/publication/348003841>
- [8] I. Amri, "Implementasi Algoritma Convolutional Neural Network Untuk Menerjemahkan Bahasa Isyarat," *J. Multidisiplin Saintek Vol.*, vol. 2, no. 9, pp. 70–87, 2024, [Online]. Available: <https://ejournal.warunayama.org/kohe>
- [9] S. Rahman et al., *Python: Dasar Dan Pemrograman Berorientasi Objek*. 2023.
- [10] E. Febrywinata, "Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab," *Juli*, vol. 2, no. 4, pp. 185–193, 2024, [Online]. Available: <https://doi.org/10.61132/mercurius.v2i4.162>