**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

# Implementasi Algoritma Convolutional Neural Network dan YOLOV8 Untuk Klasifikasi Ras Kucing

**Abdul Rohim Adinata*, Tatang Rohana, Kiki Ahmad Baihaqi, Sutan Faisal**

Fakultas Ilmu Komputer, Teknik Informatika, Universitas Buana Perjuangan, Karawang, Indonesia
Email: [1,*]if19.abdulrohimadinata@mhs.ubpkarawang.ac.id, [2]tatang.rohana@ubpkarawang.ac.id,
[3]kikiahmad@ubpkarawang.ac.id, [4]sutan.faisal@ubpkarawang.ac.id
Email Penulis Korespondensi: if19.abdulrohimadinata@mhs.ubpkarawang.ac.id
Submitted: **21/09/2024**; Accepted: **13/12/2024**; Published**: 18/12/2024**

**Abstrak**−Kucing dengan nama ilmiah *Felis catus* adalah salah satu hewan peliharaan yang sangat populer dan sering dipelihara di berbagai belahan dunia. Terdapat banyak jenis atau ras kucing yang masing-masing memiliki karakteristik dan ciri khas tersendiri, seperti corak, bentuk tubuh, bulu, dan warna. Namun, karena banyaknya ras dan keunikan dari setiap ras tersebut, seringkali sulit bagi orang awam untuk membedakan jenis ras kucing yang ada. Oleh karena itu, diperlukan teknologi dalam mengidentifikasi dan membedakan ras kucing. Dengan membandingkan metode Convolutional Neural Network (CNN) dan YOLOV8, penelitian ini bertujuan untuk mengembangkan model klasifikasi ras kucing. Penelitian ini menggunakan data dari enam ras kucing berbeda yaitu Bengal, Bombay, Himalaya, Lokal, Persia, dan Sphynx. Semuanya ada 1.200 gambar, dengan 200 gambar untuk setiap ras. Sebelum data digunakan untuk pelatihan dengan metode CNN dan YOLOV8, dilakukan tahap *preprocessing* yang meliputi *resize* dan *rescale* untuk metode CNN, sedangkan untuk YOLOV8 dilakukan proses pelabelan data. Ada dua bagian dataset yaitu 20% data validasi dan 80% data pelatihan. Proses pelatihan dilakukan dengan parameter yang sama untuk setiap model, yaitu *learning rate* sebesar 0,001, *batch size* sebesar 15, dan 100 *epoch*. Dari hasil pengujian dengan *confusion matrix*, model YOLOV8 menunjukkan kinerja terbaik dengan nilai akurasi sebesar 99%, precision 96,1%, recall 98,4%, dan F1-score 97,2%.

**Kata Kunci:** Confusion Matrix; Convolutional Neural Network; Klasifikasi; Ras Kucing; YOLOV8

**Abstract**−The cat with the scientific name Felis catus is a very popular pet and is often kept in various parts of the world. There are many types or breeds of cats, each of which has its own characteristics and characteristics, such as style, body shape, fur and color. However, because of the many breeds and the uniqueness of each breed, it is often difficult for ordinary people to differentiate between the types of cat breeds that exist. Therefore, technology is needed to identify and differentiate cat breeds. By comparing the Convolutional Neural Network (CNN) and YOLOV8 methods, this research aims to develop a cat breed classification model. This research uses data from six different cat breeds, namely Bengal, Bombay, Himalayan, Local, Persian and Sphynx. There are 1,200 images in all, with 200 images for each race. Before the data is used for training with the CNN and YOLOV8 methods, a preprocessing stage is carried out which includes resize and rescale for the CNN method, while for YOLOV8 a data labeling process is carried out. There are two parts to the dataset: 20% validation data and 80% training data. The training process is carried out with the same parameters for each model, namely a learning rate of 0.001, batch size of 15, and 100 epochs. From the test results with the confusion matrix, the YOLOV8 model shows the best performance with an accuracy value of 99%, precision 96.1%, recall 98.4%, and F1-score 97.2%.

**Keywords**: Confusion Matrix; Convolutional Neural Network; Classification; Cat Breeds; YOLOV8

## 1. PENDAHULUAN

Kucing atau dengan nama ilmiah *Felis Catus* ialah salah satu hewan yang banyak disukai dan dipelihara oleh banyak orang. Diketahui bahwa kucing hidup berdampingan dengan manusia dan menyebar ke berbagai wilayah di dunia. Kucing memiliki banyak sekali jenis atau ras, setiap ras kucing memiliki ciri dan karakteristik khususnya sendiri. Di seluruh dunia, terdapat 315 ras kucing [1]. Ciri khas tiap ras biasanya terletak pada corak, bentuk tubuh, bulu, dan warnanya. Namun, karena banyaknya ras kucing dan beragamnya ciri pada setiap jenis ras kucing membuat sulit sekali untuk membedakan jenis rasnya, dan bagi orang awam akan sulit untuk mengetahui jenis ras kucing tersebut. Sehingga berdasarkan permasalahan tersebut terdapat peluang untuk memanfaatkan teknologi *computer vision* dalam mengklasifikasi ras kucing. Oleh karena itu, tujuan dari penelitian ini adalah untuk mengembangkan model sistem yang dapat mengkategorikan ras kucing.

*Computer vision* telah mengalami perkembangan yang signifikan selama beberapa dekade, dengan usaha yang bertujuan untuk memungkinkan komputer memahami informasi visual secara mendalam. Perkembangan ini mencakup berbagai tingkat pemahaman, mulai dari tugas yang sederhana seperti mendeteksi gambar hingga tugas yang lebih kompleks seperti menginterpretasikan keseluruhan konteks dalam sebuah adegan visual [2]. Klasifikasi dapat dilakukan dengan berbagai metode, salah satunya adalah *Deep Learning*, yang efektif dalam memahami informasi visual seperti gambar atau citra. Sebuah metode pembelajaran yang disebut *Deep Learning* memanfaatkan jaringan saraf tiruan berlapis yang terlihat seperti struktur di otak manusia. Di dalam *Deep Learning*, neuron-neuron saling terhubung, membentuk jaringan yang sangat rumit [3]. *Convolutional Neural Network* (CNN) merupakan salah satu metode *Deep Learning* yang hasil klasifikasi atau deteksi gambarnya saat ini paling signifikan. Metode ini dapat mengidentifikasi informasi-informasi penting dari setiap gambar secara otomatis, sehingga meningkatkan akurasi dan efisiensi dalam proses pengenalan citra [4]. Hasilnya, *Convolutional Neural Network* (CNN) semakin populer dan sering digunakan untuk mengklasifikasikan objek.

*You Only Look Once* (YOLO) adalah metode baru untuk deteksi objek. Joseph Redmon dan rekan-rekannya pertama kali mendeskripsikan YOLO dalam makalah tahun 2016 berjudul "*You Only Look Once: Unified, Real-Time*

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

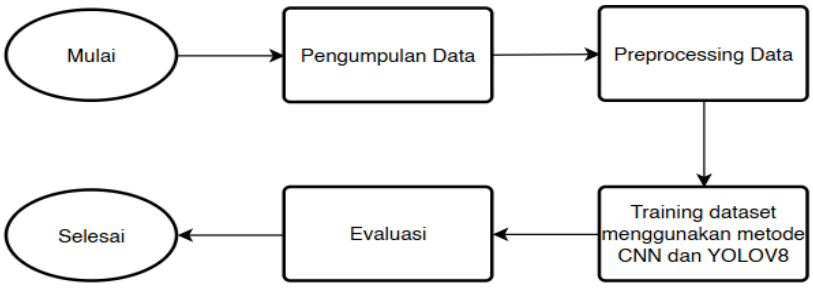
*Object Detection*." YOLO sebagai salah satu metode dalam *deep learning*, biasa dipakai untuk deteksi objek secara *real-time* dengan berbasis CNN. YOLO telah menjadi salah satu metode yang sangat populer dalam mendeteksi objek secara *real-time* [5]. Berbeda dengan metode sebelumnya yang harus melewati beberapa tahapan, YOLO menerapkan metode jaringan konvolusi tunggal pada seluruh gambar dan mengatasi masalah pendeteksian objek secara bersamaan dengan memprediksi kotak pembatas yang menandai posisi objek dan probabilitas kelas objek yang terkait dengan setiap kotak pembatas tersebut [6][7]. YOLO memungkinkan identifikasi objek dalam gambar dengan cepat dan andal, membuatnya menjadi pilihan utama dalam berbagai konteks aplikasi yang memerlukan deteksi objek secara efisien [8][9].

Telah dilakukan beberapa penelitian serupa mengenai klasifikasi atau deteksi ras kucing. Salah satunya deteksi ras kucing menggunakan penggabungan skala model CNN. Dalam penelitian tersebut, data yang digunakan untuk data latih berjumlah 2.160 citra, sedangkan data uji berjumlah 180 citra. Hasil dari pengujian model mencapai tingkat akurasi sebesar 95% dalam deteksi ras kucing [10]. Metode lain yang digunakan untuk klasifikasi ras kucing yaitu YOLOv5. Pada penelitian tersebut, dataset yang digunakan untuk data train berjumlah 1200 data, sedangkan untuk data validation berjumlah 120 data. Dari hasil pengujian menunjukkan bahwa model terbaik didapatkan pada skenario epoch ke-60 dengan batch size 16, dengan Precision sebesar 0.9844, Recall sebesar 1.0, mAP 0.5 sebesar 0.9933, dan mAP 0.5:0.95 sebesar 0.9144 [11]. Penelitian lain menggunakan algoritma *Support Vector Machine* dan *Naive Bayes* pada klasifikasi ras kucing. Dalam penelitian tersebut, dilakukan komparasi antara kedua algoritma tersebut dalam klasifikasi citra ras kucing. Ada 910 gambar dalam dataset pelatihan. Berdasarkan temuan penelitian, algoritma *Naive Bayes* memiliki akurasi sebesar 79,5%, sedangkan algoritma *Support Vector Machine* memiliki akurasi sebesar 88,4% [12]. Beberapa penelitian terdahulu menggunakan metode YOLO adalah penelitian terhadap pendeteksian jenis kendaraan di parkiran menggunakan algoritma YOLOV1. Dalam penelitian ini diperoleh hasil berdasarkan metode YOLOV1 dengan presentase tingkat akurasi sebesar 98.6% [13]. Kemudian penelitian yang membahas tentang mendeteksi tipe beras menggunakan teknologi *computer vision* dengan metode YOLOV3. Pada penelitian tersebut menjelaskan bagaimana penerapan metode YOLOV3 diterapkan dan berdasarkan hasil pengujian 12 kali percobaan pada objek gambar berhasil dalam mendeteksi 4 jenis beras diperoleh hasil akurasi sebesar 100% [14]. Penelitian lain yang dijadikan rujukan adalah penelitian tentang penerapan sistem *traffic counting* dengan algoritma YOLOV4. Hasil eksperimen yang diperoleh dalam penelitian tersebut menunjukkan bahwa hasil deteksinya memiliki tingkat akurasi yang cukup baik dalam memisahkan frame-frame dari data video [15].

Berdasarkan klarifikasi dari penelitian terdahulu, maka algoritma atau metode yang akan digunakan dalam penelitian ini adalah *Convolutional Neural Network* (CNN) dan *You Only Look Once* (YOLO) dengan versi baru khususnya YOLOV8. YOLOV8 dianggap sebagai yang paling canggih karena kecepatan inferensi kumpulan data COCO yang lebih rendah tetapi nilai *Mean Average Precision* (mAP) yang lebih tinggi [16][17]. Penggunaan algoritma CNN dan YOLOV8 pada penelitian ini bertujuan untuk melihat secara empiris perbedaan hasil pengujian model dari kedua algoritma tersebut untuk klasifikasi citra ras kucing. Pada proses *training* dengan kedua metode tersebut akan menggunakan *hyperparameter batch size*, *learning rate*, dan *epoch* yang sama. Dengan dibuatnya model sistem klasifikasi ras kucing ini diharapkan dapat mempermudah dan membantu dalam mengetahui jenis ras kucing.

## 2. METODOLOGI PENELITIAN

Tahapan penelitian merupakan tahapan-tahapan yang dilakukan dalam suatu penelitian secara jelas dan terstruktur agar bisa memperoleh hasil dari tujuan penelitian.



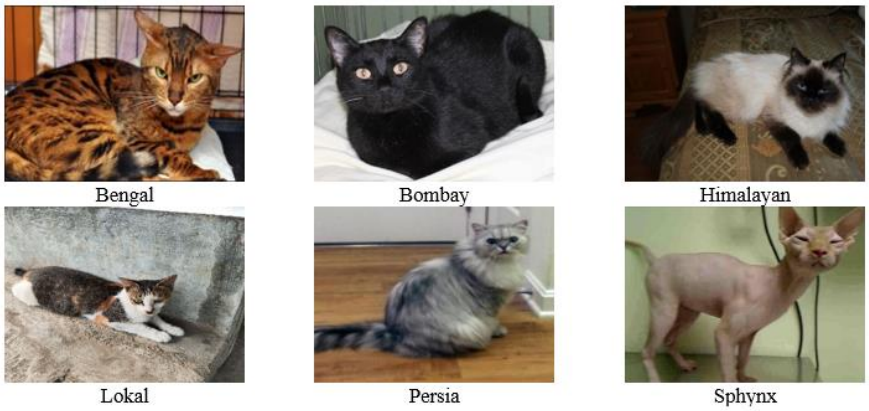

**Gambar 1.** Tahapan penelitian

Pada Gambar 1 di atas ditampilkan diagram alir tahapan penelitian "Penarapan Algoritma Convolutional Neural Network dan YOLOV8 Untuk Klasifikasi Ras Kucing". Tahapan penelitian pada penelitian ini terdiri dari beberapa langkah, yang pertama pengumpulan data, dilanjutkan dengan *preprocessing* data. Setelah itu, dilakukan *training* dataset menggunakan metode CNN dan YOLOV8, dan setelah *training* dataset selesai, dilanjutkan dengan analisis dan evaluasi hasil dari *training* dataset.

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

### 2.1 Pengumpulan Data

Ada dua tahap dalam proses pengumpulan dataset. Pertama, gambar (terbatas pada kucing lokal) diekstraksi dari video yang direkam oleh peneliti. Kedua, gambar diambil dari *website* penyedia dataset *Kaggle* dengan kata kunci pencarian "*cat breeds*". Berikut adalah sampel citra dataset dan tabel rangkuman dataset ras kucing.




**Gambar 2.** Citra Ras Kucing untuk *dataset*

Gambar 2 merupakan sampel cintra ras kucing untuk dataset. Terdapat 6 ras kucing yaitu, Bengal, Bombay, Himalayan, Lokal, Persian, dan Sphynx.

**Tabel 1.** Dataset Ras Kucing

| Dataset | Jumlah |
|---|---|
| Bengal | 200 |
| Bombay | 200 |
| Himalayan | 200 |
| Lokal | 200 |
| Persian | 200 |
| Sphynx | 200 |

Berdasarkan Tabel 1 terdapat 6 jenis dataset yang masing-masing berjumlah 200 dataset, dari pengumpulan dataset tersebut didapatkan data berjumlah 1200 gambar, dalam penelitian ini ras kucing yang dipilih ada 6 ras yaitu, Bengal, Bombay, Himalayan, Lokal, Persian, dan Sphynx.

### 2.2 Preprocessing Data

*Preprocessing* data pada *Convolutional Neural Network* (CNN) merupakan langkah penting untuk memastikan bahwa data yang dimasukkan ke dalam model sesuai dan mendukung pembelajaran model secara efektif. Tahapan preprocessing pada CNN meliputi dua proses utama yaitu *resize* dan *rescale*. *Resize* untuk mengubah ukuran gambar ke dimensi yang sesuai dengan input model, sehingga setiap gambar memiliki ukuran yang konsisten dan dapat diproses oleh CNN. *Rescale* untuk menyesuaikan skala nilai piksel gambar, biasanya dari rentang [0, 255] menjadi [0, 1], dengan cara membagi nilai piksel dengan 255. Ini bertujuan untuk mempersingkat konvergensi selama proses pelatihan model, meningkatkan efisiensi pembelajaran.

Pada YOLOV8 dataset yang digunakan dilakukan anotasi citra dengan menggunakan kotak pembatas dan nama kelas untuk setiap objek gambar, proses anotasi citra melibatkan pembuatan label. *Makesense.ai* digunakan untuk proses pelabelan yang dilakukan secara manual untuk seluruh dataset gambar.

Setelah melakukan *preprocessing* data, tahap selanjutnya membuat struktur dataset. Dataset dibagi menjadi 2 set yaitu *train set* dan *validation set.* Dalam penelitian ini, dataset dibagi dengan perbandingan 80% dan 20%. *Train set* digunakan untuk melatih model, sedangkan *validation set* digunakan untuk mengukur akurasi model atau mencari *hyperparameter* seperti *confusion matrix*, *recall*, *precision*, dan sebagainya pada model.

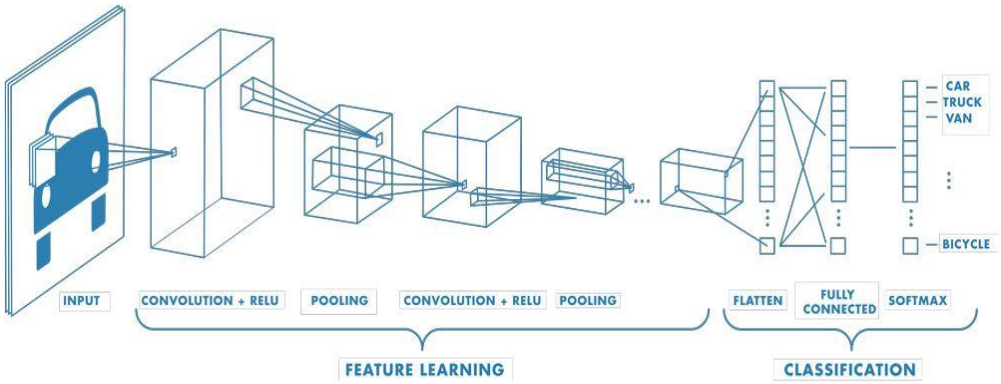
### 2.3 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN), merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang khusus untuk mengolah data dalam bentuk dua dimensi seperti gambar, merupakan salah satu metode *Deep Learning* [3]. *Convolutional Neural Network* (CNN) secara bersamaan akan mengekstraksi fitur dan mengklasifikasikan gambar, berbeda dengan algoritma klasifikasi umum, yang memisahkan proses ekstraksi fitur dan klasifikasi. Seiring dengan proses pembelajaran, ekstraksi fitur juga dilakukan pada algoritma CNN [18].

CNN dapat digunakan dan diimplementasikan untuk pengenalan citra dengan tingkat akurasi yang dapat menyaingi kemampuan manusia. CNN mampu meningkatkan akurasi pengenalan dan karena alasan ini, metode CNN sering digunakan dalam proses klasifikasi objek. Data yang telah diberi label menggunakan *supervised learning*

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

digunakan untuk mengklasifikasikan dengan CNN. Tujuan dari *supervised learning* adalah mengelompokkan data baru ke dalam kategori yang sudah ada berdasarkan pembelajaran dari data latih karena terdapat dataset yang telah dilatih dan variabel yang ditargetkan [19].



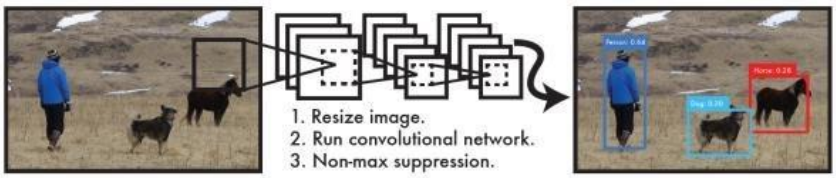

**Gambar 3.** Arsitektur CNN

Arsitektur CNN dapat dilihat pada Gambar 3. CNN memiliki sejumlah lapisan (*layer*) yang berfungsi untuk melakukan *filtering* pada setiap tahapnya. Proses ini disebut sebagai proses *training*. Pada tahap *training*, terdapat tiga lapisan utama, yaitu *Convolutional layer*, *Pooling layer*, dan *Fully Connected layer* [20].

### 2.4 You Only Look Once (YOLO)

*You Only Look Once* (YOLO) adalah metode baru untuk deteksi objek. Joseph Redmon dan rekan-rekannya pertama kali mendeskripsikan YOLO dalam makalah tahun 2016 berjudul "*You Only Look Once: Unified, Real-Time Object Detection*." YOLO sebagai salah satu metode dalam *deep learning*, digunakan untuk melakukan deteksi objek secara *real-time* dengan berbasis *Convolutional Neural Network*. YOLO telah menjadi salah satu metode yang sangat populer dalam mendeteksi objek secara real-time [5][21].

Deteksi objek telah menjadi elemen krusial dalam berbagai aplikasi, termasuk kendaraan *autonomous*, *robotics*, kamera pengawas, dan *augmented reality*. Di antara berbagai algoritma deteksi objek, algoritma YOLO (*You Only Look Once*) menonjol karena berhasil mencapai keseimbangan yang luar biasa antara kecepatan dan akurasi. YOLO memungkinkan identifikasi objek dalam gambar dengan cepat dan andal, membuatnya menjadi pilihan utama dalam berbagai konteks aplikasi yang memerlukan deteksi objek secara efisien [8]. YOLO menggunakan metode jaringan konvolusi tunggal untuk memprediksi kotak pembatas dan label kelas yang terkait dengan setiap kotak pembatas saat melakukan deteksi objek pada gambar [7].




**Gambar 4.** Sistem deteksi YOLO [7]

Langkah pertama adalah mengolah gambar dan mengubah ukurannya menjadi 448 x 448 seperti terlihat pada Gambar 4. Langkah selanjutnya adalah *non-max suppression* yang bertujuan untuk membuat kotak pembatas dan mengidentifikasi label setiap objek yang terdeteksi [22].

YOLOV8, yang dirilis pada bulan Januari 2023 oleh Ultralytics, merupakan kelanjutan dari YOLOV5 yang dikembangkan oleh perusahaan tersebut. Tugas penglihatan seperti deteksi objek, segmentasi, estimasi pose, pelacakan, dan klasifikasi semuanya didukung oleh YOLOV8 [8]. YOLOV8 menonjolkan arsitektur baru, *convolutional layers* yang ditingkatkan, dan sistem deteksi yang lebih canggih, menjadikannya pilihan unggul untuk deteksi objek secara real-time. YOLOV8 juga memberikan dukungan untuk algoritma *computer vision* terbaru, seperti segmentasi dan deteksi multi-objek dalam gambar atau video [23]. Model ini terbukti lebih efektif dibandingkan dengan versi sebelumnya karena memiliki jaringan konvolusional yang ditingkatkan sehingga menghasilkan peningkatan presisi dan kecepatan. YOLOV8 juga mengintegrasikan fitur *pyramid network* untuk mengenali objek dengan ukuran yang berbeda [23].

### 2.5 Training Dataset menggunakan metode CNN

Untuk menghasilkan model dengan akurasi tertinggi, dilakukan pelatihan. Menggunakan Jupyter Notebook sebagai *tools* dan bahasa pemrograman Python, prosedur ini menggunakan algoritma CNN. Akurasi model ditingkatkan dengan menggunakan *optimizer* Adam. [24]. Sebelum *optimizer* Adam dapat digunakan, nilai *learning rate* yang akan diterapkan pada kumpulan data pelatihan harus ditetapkan. Salah satu parameter penting yang digunakan untuk

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

mengontrol pembaruan bobot selama latihan adalah *learning rate*. Jumlah contoh pelatihan yang diproses dalam satu iterasi, atau nilai *batch size*, juga ditentukan. Nilai *learning rate* dan *batch size* yang digunakan dalam proses pelatihan akan ditentukan sesuai kebutuhan penelitian.

### 2.6 Training Dataset menggunakan metode YOLOV8

*Training* data dilakukan menggunakan Google Colab karena menyediakan GPU 12 GB dengan dukungan Nvidia, memungkinkan proses training data berlangsung dengan cepat. Tujuan dari proses *training* adalah untuk melatih komputer mengenali pola atau karakteristik setiap kelas dengan memproses gambar dan anotasi. Komputer kemudian menggunakan informasi ini untuk membuat keputusan atau prediksi. Tahap pertama melibatkan pengubahan file coco128.yaml menjadi custom_dataset.yaml dan penyesuaian jumlah kelas serta nama kelas yang akan digunakan. Setelah itu, dilakukan *cloning repository* YOLOv8 dari GitHub dan mengunggah dataset.

Selanjutnya, mengunggah custom_data.yaml yang telah dibuat ke dalam direktori YOLOv8. Menyesuaikan jumlah *epoch*, *batch size*, dan *learning rate* pada kode train.py sesuai kebutuhan penilitain. Jumlah *batch size* menentukan jumlah gambar yang akan diproses sebelum bobot (*weight*) mengalami pembaharuan. Jumlah *epoch* menentukan seberapa lama komputer belajar atau training. Dan *learning rate* untuk menentukan nilai koreksi bobot selama proses *training*. Kemudian, menjalankan kode train.py untuk memulai training atau pembuatan model. Hasil dari proses training adalah model itu sendiri dan *hyperparameter confusion matrix* dari model yang telah dibuat.

### 2.7 Evaluasi

Evaluasi yang dilakukan pada penelitian ini sebagai tahapan dalam perbandingan tingkat akurasi algoritma yang sedang diuji yaitu *Convolutional Neural Network* dan YOLOV8. Analisa hasil didapatkan dari *confusion matrix* pada masing-masing algoritma. *Confusion matrix* adalah matrix yang digunakan untuk membandingkan hasil prediksi model untuk mengevaluasi kinerja model. Alat ini memberikan informasi tentang seberapa baik model memprediksi kelas ke dalam kategori yang benar atau salah dan membantu mengidentifikasi kesalahan prediksi model. *Confusion matrix* mengukur empat nilai: *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Jumlah data kelas x yang benar yang terdeteksi selama pengujian disebut sebagai TP. Banyaknya data non-kelas x yang terdeteksi dengan benar disebut TN. Banyaknya data bukan kelas x yang teridentifikasi sebagai kelas x adalah FP, dan banyaknya data kelas x yang teridentifikasi bukan kelas x adalah FN [25].

**Tabel 2.** Ilustrasi Confusion Matrix

| | Predicted : Yes | Predicted : No |
|---|---|---|
| Actual : Yes | TP | FN |
| Actual : No | FP | TN |

Selain memberikan gambaran tentang *confusion matrix* itu sendiri, Tabel 2 memberikan ringkasan empat istilah yang digunakan sebagai ukuran dalam *confusion matrix* yang telah dijelaskan sebelumnya. Evaluasi tersebut dilakukan untuk menghitung nilai dari *accuracy*, *precision*, *recall*, dan *F1-Score*. Rumus yang digunakan untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *F1-Score* adalah sebagai berikut :

$$Accuracy\ (\%) = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Precision\ (\%) = \frac{TP}{FP+TP} \quad (2)$$

$$Recall\ (\%) = \frac{TP}{FN+TP} \quad (3)$$

$$F1 - Score\ (\%) = \frac{(2\ x\ recall\ x\ precision)}{(recall+precision)} \quad (4)$$

*Accuracy* dan nilai kinerja model klasifikasi yang benar. Rasio prediksi yang benar terhadap total data pelatihan adalah *accuracy*. Rasio prediksi positif yang benar terhadap total prediksi hasil positif adalah *precision*. Rasio prediksi positif yang benar terhadap jumlah total data yang sebenarnya termasuk dalam kelas tersebut disebut *recall*. *F1-score* adalah rata-rata antara *precision* dan *recall* [26]. Untuk nilai matrix di enam kelas yaitu Bengal, Bombay, Himalaya, Lokal, Persia, dan Sphynx digunakan hasil *macro avg* dari *accuracy*, *precision*, *recall*, dan *f1-score*.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Preprocessing Data

Adapun preprocessing dari CNN yaitu dengan melakukan *resize* dan *rescale* pada data yang akan dijadikan dataset untuk training model.

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

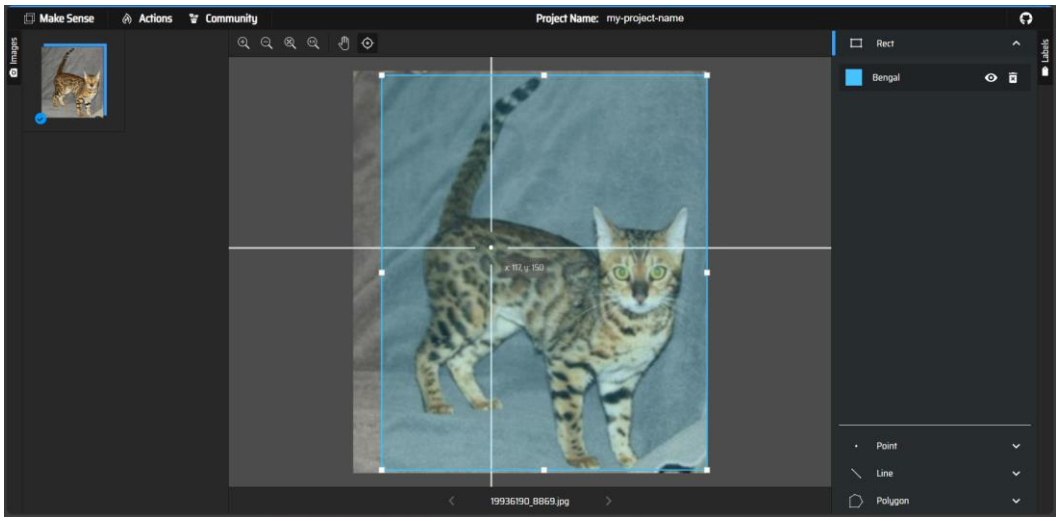

**Gambar 5.** *Resize* citra pada dataset

Gambar 5 merupakan hasil dari proses *resize* pada citra yang akan dijadikan dataset untuk model CNN. Pada pelatihan model CNN dalam penelitian ini, proses *resize* dilakukan untuk memperkecil ukuran citra agar sesuai dengan input model CNN saat *training*. Setiap citra dalam dataset diubah ukurannya menjadi seragam, yaitu 224x224 piksel, untuk memastikan konsistensi dan efisiensi dalam pemrosesan data oleh model.


**Gambar 6.** *Rescale* citra pada dataset

Gambar 6 menunjukkan hasil dari proses *rescale* pada citra yang akan dijadikan dataset pada mode CNN. Pada pelatihan model CNN dalam penelitian ini, proses rescale dilakukan untuk mengubah skala nilai piksel pada setiap gambar dari rentang [0,255] menjadi [0,1] dengan membagi setiap piksel dengan 255. Proses rescale ini bertujuan untuk mempercepat pelatihan model dengan mereduksi rentang nilai piksel, sehingga model dapat belajar lebih efisien.

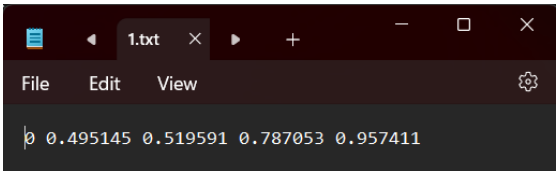
Berikut ini proses anotasi citra untuk dataset YOLOV8 secara manual pada semua dataset yang dilakukan di *website Makesense.ai*.




**Gambar 7.** *Labeling* data menggunakan *Makesense.AI*

Gambar 7 menunjukkan proses anotasi citra yang dilakukan menggunakan *Makesense.ai.* Hasil dari anotasi citra berupa file dengan format (.txt), objek yang telah diberi label akan digunakan untuk melatih model pada YOLOV8. File label ini mencakup enam kelas, yaitu Bengal, Bombay, Himalayan, Lokal, Persian, dan Sphynx, serta parameter *bounding box* yang dibuat selama proses pelabelan data.

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913




**Gambar 8.** Hasil *Labeling* citra

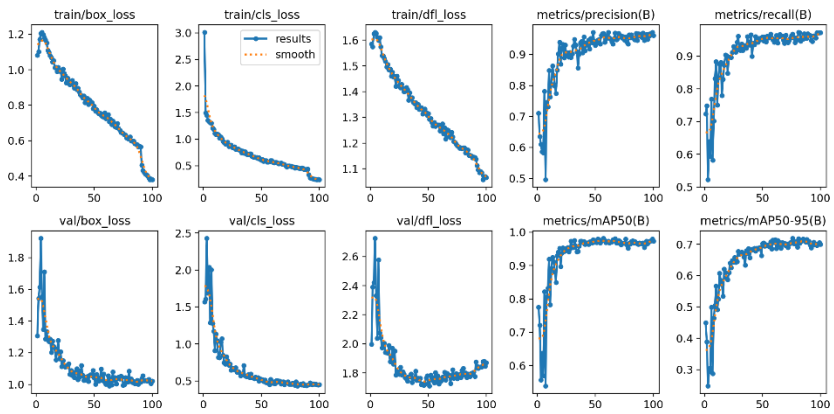
Gambar 8 merupakan hasil dari proses anotasi citra yang dilakukan di website *Makesense.ai*, format file dari anotasi citra yang dilakukan berupa file dengan format (.txt).

### 3.2 Hasil Training

Hasil dari proses *training* data menggunakan metode CNN dan YOLOv8 menghasilkan model dan *hyperparameter*. Proses pelatihan pada CNN dilakukan menggunakan *tool* Jupyter Notebook dengan bahasa pemrograman Python, sedangkan pelatihan YOLOV8 dilakukan menggunakan Google Colab. 1.200 gambar yang digunakan dalam proses pelatihan dibagi menjadi dua set: data pelatihan dan data validasi. Dataset tersebut dibagi menjadi data pelatihan (960 data) dan data validasi (240 data) dengan rasio 80% dan 20% untuk penelitian ini. *Training* pada CNN dilakukan menggunakan *optimizer* Adam untuk mengoptimalkan model dan meningkatkan akurasi. Sementara itu, YOLOV8 menggunakan model dari Ultralytics. Setiap model akan digunakan selama proses pelatihan dengan *batch size* dan nilai *learning rate* yang sama. Semua training model menggunakan jumlah *epoch* yang sama, yaitu 100 *epoch*. Untuk *learning rate* dengan nilai 0,001 dan *batch size* dengan nilai 15.


**Gambar 9.** Grafik *Loss* dan *Accuracy* model CNN

Pada Gambar 9 menunjukkan grafik *loss* dan *accuracy* model menggunakan CNN. *Training Loss* (Garis Merah) menunjukkan seberapa baik model melakukan generalisasi terhadap data pelatihan. Semakin rendah nilai loss, semakin baik performa model pada data pelatihan. *Validation Loss* (Garis Hijau), di sisi lain, menunjukkan seberapa baik model melakukan generalisasi terhadap data validasi yang tidak dilihat selama pelatihan. Pada awalnya, *training loss* menurun dengan cepat, yang menandakan bahwa model dengan cepat belajar dari data pelatihan. Namun, *validation loss* menunjukkan fluktuasi yang cukup besar, yang bisa menjadi indikasi adanya *overfitting*. *Overfitting* terjadi ketika model terlalu mempelajari detail dan pola dari data pelatihan sehingga terlalu spesifik pada data tersebut. Akibatnya, model ini bekerja sangat baik dengan data pelatihan, namun tidak dapat menangani data baru yang belum pernah dilihat atau digeneralisasikan sebelumnya, sehingga model ini kurang baik menangani data baru dengan maksimal.



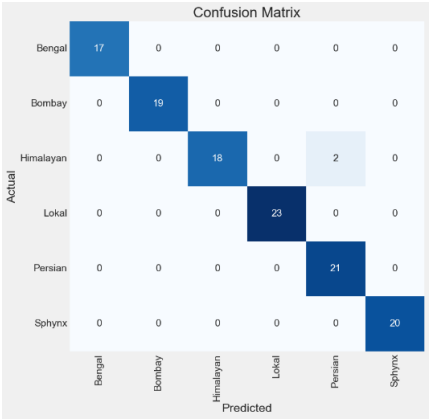

**Gambar 10.** Grafik *Loss* dan *Accuracy* model YOLOV8

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

Gambar 10 merupakan grafik *loss* dan *accuracy* dari model menggunakan YOLOV8. Semua matrix loss pada grafik menunjukkan penurunan yang signifikan, baik untuk data pelatihan maupun validasi, yang menandakan bahwa model belajar dengan baik. Matrix evaluasi seperti *precision*, *recall*, dan mAP juga menunjukkan peningkatan yang stabil, mengindikasikan bahwa model tidak hanya belajar dengan baik tetapi juga mampu melakukan generalisasi dengan baik pada data validasi. Secara umum, hal ini menunjukkan bahwa model memiliki performa yang memuaskan dan tidak menunjukkan overfitting atau underfitting yang signifikan.

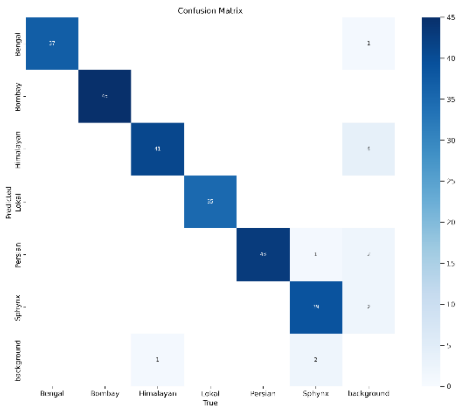
### 3.3 Evaluasi Model

Langkah selanjutnya adalah menguji model untuk menentukan nilai *confusion matrix* setelah menyelesaikan pelatihan model dan mendapatkan model terbaik. Dari tabel *confusion matrix* ini, akan dihitung nilai *accuracy*, *precision*, *recall*, dan *F1-score*. Berikut adalah hasil dari pengujian yang didapatkan berupa tabel *confusion matrix*.




**Gambar 11.** Confusion matrix model CNN

Hasil pengujian model CNN dengan *hyperparameter learning rate* 0,001 dan *batch size* 15 ditunjukkan pada Gambar 11. Berdasarkan hasil pengujian model mampu mencapai *accuracy* 98%, *precision* 98,5%, *recall* 98,3%, dan *f1-score* sebesar 98,3%.




**Gambar 12.** Confusion matrix model YOLOV8

Hasil pengujian model dari *hyperparameter* pada model YOLOV8 dengan *batch size* 15 dan *learning rate* 0.001 ditunjukkan pada Gambar 12. Berdasarkan hasil pengujian tersebut, model berhasil mencapai nilai *accuracy* sebesar 99%, *precision* sebesar 96,1%, *recall* sebesar 98,4%, dan *f1-score* dengan nilai 97,2%.

**Tabel 3.** Hasil pengujian model

| Metode | Learning rate | Batch size | Epoch | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|
| CNN | 0,001 | 15 | 100 | 98% | 98,5% | 98,3% | 98,3% |
| YOLOV8 | 0,001 | 15 | 100 | 99% | 96,1% | 98,4% | 97,2% |

Hasil pengujian model berbasis *confusion matrix* dirangkum pada Tabel 3. Model CNN memiliki nilai *accuracy* sebesar 98%, sedangkan model YOLOV8 memiliki nilai sebesar 99%, berdasarkan model yang telah menjalani pelatihan. Namun, model CNN memiliki performa lebih baik daripada model YOLOV8 dalam hal *precision* dan *F1-score*.

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

## 4. KESIMPULAN

Berdasarkan hasil pengujian dan evaluasi model CNN dan YOLOV8 untuk klasifikasi ras kucing dengan *hyperparameter* yang sama yaitu *learning rate* 0,001, *batch size* 15, dan *epoch* 100 hasilnya dapat disimpulkan bahwa model YOLOV8 dengan nilai *accuracy* 99%, *precision* sebesar 96,1%, *recall* sebesar 98,4%, dan *f1-score* sebesar 97,2% secara nilai *accuracy* lebih unggul dibandingkan dengan model CNN yang mendapatkan hasil nilai *accuracy* 98%, *precision* sebesar 98,5%, *recall* sebesar 98,3%, dan *f1-score* sebesar 98,3%. Namun evaluasi dari kedua model algoritma menggunakan *confusion matrix* dalam mengklasifikasi citra hanya selisih 1%. Yang artinya kedua algoritma tersebut memiliki akurasi yang tinggi dalam mengklasifikasi citra. Dilihat dari grafik *loss* dan *accuracy* model YOLOV8 secara keseluruhan menunjukkan bahwa model memiliki grafik yang baik dan tidak mengalami *overfitting*. Pada grafik *loss* dan *accuracy* model CNN, terlihat bahwa model mengalami *overfitting*, yang ditandai dengan perbedaan yang cukup besar antara *loss* dan *accuracy* pada data *training* dan validasi. Fluktuasi yang signifikan pada *validation loss* dan *accuracy* menunjukkan bahwa model tidak mampu melakukan generalisasi dengan maksimal pada data baru. Sebagai solusi untuk mengatasi *overfitting* pada model CNN, dapat dilakukan augmentasi data terlebih dahulu pada dataset untuk menambah variasi data dan meningkatkan kemampuan generalisasi model. Karena pada penelitian sebelumnya berhasil mendapatkan model CNN yang baik tanpa indikasi *overfitting* setelah melakukan augmentasi data terlebih dahulu, langkah ini dapat dijadikan pertimbangan penting untuk diterapkan dalam penelitian berikutnya. Augmentasi data dapat membantu meningkatkan variasi dalam dataset, sehingga model dapat lebih baik dalam melakukan generalisasi terhadap data baru dan mengurangi risiko *overfitting*.

## REFERENCES

[1] M. Afif, A. Fawwaz, K. N. Ramadhani, and F. Sthevanie, "Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural Network ( CNN )," *Jurnal Tugas Akhir Fakultas Informatika*, vol. 8, no. 1, pp. 715–730, 2021.
[2] A. Esteva *et al.*, "Deep learning-enabled medical computer vision," *npj Digit. Med.*, vol. 4, no. 1, pp. 1–9, 2021, doi: 10.1038/s41746-020-00376-2.
[3] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network ( Cnn ) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12–21, 2020.
[4] A. Antoni, T. Rohana, and A. R. Pratama, "Implementasi Algoritma Convolutional Neural Network Untuk Klasifikasi Citra Kemasan Kardus Defect dan No Defect," *Build. Informatics, Technol. Sci.*, vol. 4, no. 4, pp. 1941–1950, 2023, doi: 10.47065/bits.v4i4.3270.
[5] A. Harun, Mustakim, and O. B. Kharisma, "Implementasi Deep Learning Menggunakan Metode You Only Look Once untuk Mendeteksi Rokok," *J. Media Inform. Budidarma*, vol. 7, no. 1, pp. 107–116, 2023, doi: 10.30865/mib.v7i1.5409.
[6] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *J. Phys. Conf. Ser.*, vol. 1004, no. 1, 2018, doi: 10.1088/1742-6596/1004/1/012029.
[7] T. Al *et al.*, "PEOPLE COUNTING FOR PUBLIC TRANSPORTATIONS USING YOU ONLY LOOK ONCE METHOD," *Jurnal Teknik Informatika (JUTIF)*, vol. 2, no. 1, pp. 57–66, 2021, doi: 10.20884/1.jutif.2021.2.2.77.
[8] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO: From YOLOv1 and Beyond," *arXiv (Cornell University)*, pp. 1–34, 2023, doi: 10.48550/arXiv.2304.00501.
[9] K. Ahmad Baihaqi and C. Zonyfar, "Deteksi Lahan Pertanian Yang Terdampak Hama Tikus Menggunakan Yolo v5," *Syntax J. Inform.*, vol. 11, no. 02, pp. 01–11, 2022, doi: 10.35706/syji.v11i02.7226.
[10] N. Azahro Choirunisa, T. Karlita, and R. Asmara, "Deteksi Ras Kucing Menggunakan Compound Model Scaling Convolutional Neural Network," *Technomedia J.*, vol. 6, no. 2, pp. 236–251, 2021, doi: 10.33050/tmj.v6i2.1704.
[11] M. Ismu Rahayu, "KLASIFIKASI RAS KUCING MENGGUNAKAN METADATA DATASET KAGGLE DENGAN FRAMEWORK YOLO v5," *J. Teknol. Inf. dan Komun.*, vol. 12, no. 1, pp. 14–18, 2023, doi: 10.58761/jurtikstmikbandung.v12i1.1418.
[12] J. Kusuma, A. Jinan, M. Z. Lubis, and R. Rosnelly, "Komparasi Algoritma Support Vector Machine Dan Naive Bayes Pada Klasifikasi Ras Kucing," *J. Generic*, vol. 14, no. 1, pp. 8–12, 2022, doi: 10.18495/generic.v14i1.122.
[13] A. A. B, A. Amin, and M. W. Kasrani, "Penerapan Metode Yolo Object Detection V1 Terhadap Proses Pendeteksian Jenis Kendaraan Di Parkiran," *J. Tek. Elektro Uniba (JTE UNIBA)*, vol. 6, no. 1, pp. 194–199, 2021, doi: 10.36277/jteuniba.v6i1.130.
[14] K. A. Baihaqi and Y. Cahyana, "Application of Convolution Neural Network Algorithm for Rice Type Detection Using Yolo v3," *Systematics*, vol. 3, no. 2, pp. 272–280, 2021, doi: 10.35706/sys.v3i2.5874.
[15] Y. Pratama and E. Rasywir, "Eksperimen Penerapan Sistem Traffic Counting dengan Algoritma YOLO (You Only Look Once) V.4," *J. Media Inform. Budidarma*, vol. 5, no. 4, pp. 1438–1446, 2021, doi: 10.30865/mib.v5i4.3309.
[16] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," *arXiv (Cornell University)*, pp. 1-10, 2023, doi: 10.48550/arXiv.2305.09972.
[17] M. Jamal, S. Faisal, D. S. Kusumaningrum, and T. Rohana, "APPLICATION OF YOLO V8 FOR PRODUCT DEFECT DETECTION IN MANUFACTURING COMPANIES," *Jusikom Prima*, vol. 8, no. 1, pp. 1–11, 2024.
[18] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 104–108, 2020, doi: 10.26740/jinacs.v1n02.p104-108.
[19] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.*, vol. 3, no. 2, pp. 49–56, 2018, doi: 10.32528/justindo.v3i2.2254.
[20] A. Santoso and G. Ariyanto, "Implementasi Deep Learning berbasis Keras untuk Pengenalan Wajah," *Emit. J. Tek. Elektro*, vol. 18, no. 1, pp. 15–21, 2018, doi: 10.23917/emitor.v18i01.6235.
[21] K. Ahmad Baihaqi, C. Zonyfar, and B. Nugraha, "PENGENALAN JENIS CANDI BERDASARKAN BENTUK DAN

**Building of Informatics, Technology and Science (BITS)**
Volume 6, No 3, Desember 2024 Page: 1658−1667
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.5913

MODELNYA MENGGUNAKAN MOTODE CONVOLUTIONAL NEURAL NETWORK (CNN) PADA YOLLO v3," *Syntax J. Inform.*, vol. 10, no. 02, pp. 13–23, 2021, doi: 10.35706/syji.v10i02.5665.

[22] M. D. Anggraini and H. Al Fatta, "SOCIAL DISTANCING DETECTION FINDING OPTIMAL ANGLE WITH YOLO V3 DEEP LEARNING METHOD," *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 5, pp. 1449–1454, 2022, doi: 10.20884/1.jutif.2022.3.5.390.

[23] B. Wang, "Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8," *arXiv (Cornell University)*, pp. 5349–5357, 2023, doi: 10.48550/arXiv.2304.08256.

[24] E. Oktafanda, "Klasifikasi Citra Kualitas Bibit dalam Meningkatkan Produksi Kelapa Sawit Menggunakan Metode Convolutional Neural Network (CNN)," *J. Inform. Ekon. Bisnis*, vol. 4, no. 3, pp. 72–77, 2022, doi: 10.37034/infeb.v4i3.143.

[25] M. A. Rohman and D. Arifianto, "Penerapan Metode Euclidean Probality dan Confusion Matrix dalam Diagnosa Penyakit Koi," *J. Smart Teknol.*, vol. 2, no. 2, pp. 122–130, 2021.

[26] V. Sajwan and R. Ranjan, "Classifying flowers images by using different classifiers in orange," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6 Special Issue 3, pp. 1057–1061, 2019, doi: 10.35940/ijeat.F1334.0986S319.