



ARTICLE

Augmentasi Gambar untuk Identifikasi Labu dengan Menggunakan Teachable Machine

Image Augmentation for Pumpkin Identification Using Teachable Machine

Kurnia Prayogi* dan Zico Pratama Putra

Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Nusa Mandiri, Depok, Jakarta, Indonesia

*Penulis Korespondensi: kurniaaprayogii1@gmail.com

(Disubmit 11-07-24; Diterima 09-08-24; Dipublikasikan online pada 05-02-25)

Abstrak

Penggunaan Google Teachable Machine pada dasarnya melibatkan algoritma Convolutional Neural Network (CNN), yang memungkinkan model untuk belajar mengenali objek, mengekstrak fitur, dan melakukan klasifikasi dengan akurasi tinggi. Implementasi ini diharapkan dapat meningkatkan efisiensi dan akurasi proses identifikasi labu dalam sektor pertanian, mengurangi kesalahan manusia, dan meningkatkan produktivitas. Selain itu, penelitian ini memberikan wawasan praktis tentang bagaimana teknologi pembelajaran mesin yang dapat diterapkan dalam kehidupan sehari – hari, memperluas aksesibilitas teknologi ini kepada lebih banyak individu. Berdasarkan analisis dan pengujian dengan menggunakan berbagai parameter seperti epoch dan dengan Konfigurasi batch size 64, learning rate 0.001, model pelatihan menunjukkan akurasi 96% pada kelas “Pumpkin Blur” dengan epoch 100, sementara F1-Score pengujian terbaik pada kelas “Pumpkin Blur” dan “Pumpkin Contrast” memiliki hasil 96% dan 94%. Diharapkan di masa mendatang, penelitian ini dapat diperluas lebih lanjut dengan melakukan penyesuaian parameter secara lebih teliti untuk mencapai kinerja yang optimal dalam menangani data yang baru atau belum pernah dilihat sebelumnya.

Kata kunci: Google Teachable Machine; Convolutional Neural Network; Labu; Pertanian

Abstract

The use of Google Teachable Machine fundamentally involves Convolutional Neural Network (CNN) algorithms, enabling the model to learn object recognition, feature extraction, and classification with high accuracy. This implementation is expected to enhance the efficiency and accuracy of pumpkin identification processes in the agricultural sector, reduce human error, and improve productivity. Additionally, this research provides practical insights into how machine learning technology can be applied in everyday life, thereby expanding accessibility to this technology for a broader audience. Based on analysis and testing with various parameters such as epochs and a configuration of batch size 64 and learning rate 0.001, the training model achieved 96% accuracy for the “Pumpkin Blur” class at epoch 100. The best testing F1-Scores were 96% for “Pumpkin Blur” and 94% for “Pumpkin Contrast.” It is anticipated that future research will further explore parameter adjustments to achieve optimal performance in handling new or previously unseen data.

KeyWords: Google Teachable Machine; Convolutional Neural Network; Pumpkin; Agriculture

1. Pendahuluan

Identifikasi objek dalam sektor pertanian merupakan aspek penting yang telah mengalami kemajuan besar dalam beberapa tahun terakhir, didukung oleh perkembangan pesat teknologi kecerdasan buatan yang semakin mendekati atau bahkan melampaui kemampuan sensorik manusia[1]. Penerapan teknologi ini

This is an Open Access article - copyright on authors, distributed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY SA) (<http://creativecommons.org/licenses/by-sa/4.0/>)

How to Cite: M. I. Alharits *et al.*, “Augmentasi Gambar untuk Identifikasi Labu dengan Menggunakan Teachable Machine”, *JIKO (JURNAL INFORMATIKA DAN KOMPUTER)*, Volume: 9, No.1, Pages 25–39, Februari 2025, doi: 10.26798/jiko.v9i1.1375.

meliputi berbagai kebutuhan seperti identifikasi tanaman, pengendalian hama, penilaian kualitas tanaman, dan lainnya. Salah satu tantangan di bidang pertanian adalah identifikasi dan klasifikasi tanaman, termasuk labu, yang memiliki berbagai jenis dan karakteristik. Dalam konteks ini, penggunaan pembelajaran mesin terbukti sebagai solusi efektif untuk mengotomatisasi proses identifikasi dan klasifikasi tanaman, di mana teknologi saat ini berperan penting dalam membantu tugas-tugas harian yang esensial[2]. Teachable Machine, sebuah platform yang dikembangkan oleh Google, mempermudah pembuatan model pembelajaran mesin secara cepat dan mudah, sehingga dapat diakses oleh semua orang tanpa memerlukan keahlian teknis yang mendalam[3].

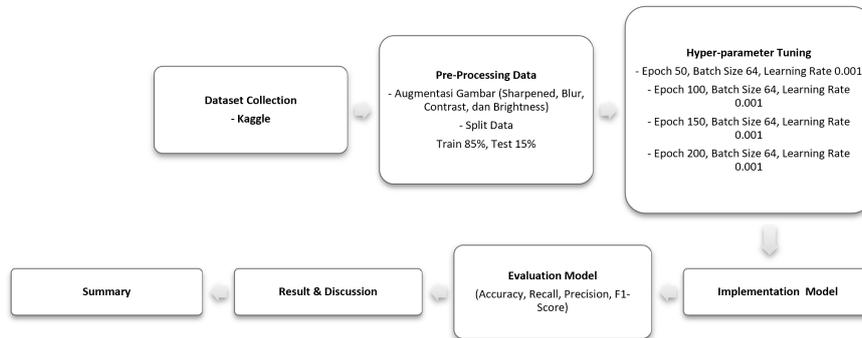
Dengan aplikasi pembelajaran mesin yang luas yang ditemukan pada platform berbasis web, Android, atau iOS, serta penggunaan coding di Jupyter atau Google Colab dengan metode seperti *Artificial Neural Network* (ANN), yang mengharuskan pengguna untuk memahami cara mengoperasikan kode tersebut, studi ini bertujuan untuk fokus pada individu, khususnya petani atau pedagang, yang menerapkan pembelajaran mesin tanpa coding. Hal ini dicapai melalui penggunaan *Google Teachable Machine* (GTM), yang ramah pengguna dan memfasilitasi berbagi proyek melalui penyimpanan Google Cloud. Penelitian ini bertujuan untuk menerapkan pembelajaran mesin menggunakan *Google Teachable Machine* untuk mengidentifikasi labu dari kumpulan data gambar labu yang beragam, melatih model untuk mengenali berbagai jenis labu dengan akurasi tinggi sesuai parameter yang ditentukan. Implementasi ini tidak hanya meningkatkan efisiensi dan akurasi proses identifikasi labu dalam pertanian, tetapi juga memberikan wawasan praktis tentang penerapan teknologi pembelajaran mesin dalam kehidupan sehari-hari. Pendekatan ini diharapkan dapat membantu petani dalam mengidentifikasi jenis labu dengan cepat dan akurat, mengurangi kesalahan manusia, dan meningkatkan produktivitas pertanian. Selain itu, penggunaan Teachable Machine pada dasarnya menerapkan algoritma *Convolutional Neural Network* (CNN), yang memungkinkan pembelajaran mendalam untuk memproses gambar resolusi tinggi menggunakan model distribusi nonparametrik[4]. Model CNN dapat secara mandiri belajar mengenali objek, mengekstrak fitur dari objek, dan melakukan tugas klasifikasi. Oleh karena itu, penggunaan Google Teachable Machine memberikan peluang bagi individu tanpa latar belakang teknis untuk berkontribusi dalam pengembangan model pembelajaran mesin, sehingga memperluas aksesibilitas teknologi ini.

Studi ini juga mengacu pada beberapa penelitian sebelumnya yang memanfaatkan platform *Google Teachable Machine* (GTM). Dalam “Pemanfaatan Aplikasi Teachable Machine Untuk Pengenalan Binatang Menggunakan Konsep *Convolutional Neural Network* (CNN)” oleh Yoga Pratama, Uning Lestari, dan Amir Hamzah, dijelaskan bahwa selama proses pengujian sistem klasifikasi gambar binatang menggunakan dataset yang telah dilatih sebelumnya, gambar-gambar diinput atau diunggah dari dataset binatang dan diproses oleh sistem untuk mencocokkan dataset yang ada. Hasil pengujian menunjukkan akurasi rata-rata 68% untuk berbagai jenis binatang. Akurasi ini bervariasi tergantung pada kemampuan sistem dalam mengenali dan mengklasifikasikan gambar, dengan beberapa binatang seperti anjing, kambing, sapi, dan kuda mencapai akurasi sekitar 60%. Hasil ini menyoroti pentingnya sinkronisasi yang baik antara dataset uji dan dataset pelatihan untuk meningkatkan akurasi keseluruhan sistem klasifikasi gambar binatang. Penelitian lain berjudul “Implementasi Machine Learning Untuk Mengidentifikasi Tanaman Hias Pada Aplikasi Tierra” oleh Dhio Immanuel Salintohe, Hasniati, dan Izmy Alwiah Musdar menggunakan pembelajaran mesin untuk mengidentifikasi tanaman hias melalui aplikasi Tierra, memanfaatkan situs web *Teachable Machine* dengan algoritma convolutional neural network sebagai alat untuk membuat model pembelajaran mesin. Data dikumpulkan melalui observasi dan studi dokumen, diikuti dengan pengujian untuk menghitung persentase akurasi. Proses data pelatihan melibatkan 30 gambar tanaman hias di situs web *Teachable Machine*, menghasilkan tingkat akurasi pembelajaran mesin sebesar 89%. Ini menunjukkan bahwa pembelajaran mesin dapat diandalkan untuk mengidentifikasi tanaman hias dalam aplikasi Tierra.

2. Metode

Metode penelitian kuantitatif merupakan pendekatan yang bergantung pada data numerik, mulai dari pengumpulan hingga interpretasi hasil[5]. Penelitian ini akan menerapkan metode kuantitatif dengan menggunakan kemampuan klasifikasi dari *Google Teachable Machine* untuk mengidentifikasi labu. Tujuan penelitian ini adalah untuk mengevaluasi *Accuracy*, *Recall*, *F1-Score*, dan *Precision*, guna menilai efektivitas teknologi ini dalam mendeteksi berbagai jenis labu. Diharapkan bahwa penerapan teknologi pembelajaran

an mesin ini di masa mendatang dapat membantu petani dengan menyediakan cara yang lebih cepat dan akurat untuk mengidentifikasi jenis labu, mengurangi kesalahan, serta meningkatkan efisiensi di sektor pertanian. Penelitian ini mencakup beberapa tahapan yang dijelaskan secara rinci dalam Gambar 1 di bawah ini.



Gambar 1. Tahapan Penelitian

Penelitian ini terdiri dari beberapa tahapan yang sistematis. Tahap pertama adalah pengumpulan dataset yang dilakukan melalui platform Kaggle. Selanjutnya, dilakukan proses pre-processing data yang mencakup augmentasi gambar dengan modifikasi ketajaman, blur, kontras, dan kecerahan dengan variasi sebesar 15% hingga 25%. Setelah proses augmentasi, data dibagi secara otomatis menjadi 85% untuk data latih (train) dan 15% untuk data uji (test) pada Google Teachable Machine. Tahap berikutnya adalah hyper-parameter tuning, di mana eksperimen dilakukan dengan variasi jumlah epoch (50, 100, 150, 200), ukuran batch sebesar 64, dan learning rate sebesar 0.001. Model kemudian dilatih menggunakan Google Teachable Machine dengan parameter-parameter yang telah ditentukan. Evaluasi model dilakukan dengan mengukur metrik kinerja seperti accuracy, recall, precision, dan F1-Score. Hasil dari pengujian ini dianalisis secara mendalam dan dibahas untuk memberikan pemahaman yang lebih komprehensif mengenai kinerja model. Tahap akhir dari penelitian ini adalah penyusunan kesimpulan berdasarkan hasil evaluasi dan diskusi yang telah dilakukan.

2.1 Pengumpulan Data

Penelitian ini memanfaatkan dataset dari Kaggle yang disusun oleh M Israk Ahmed dan Shahriyar Mahmud Mamun dengan judul “Vegetable Image Dataset”. Dataset ini mencakup 15 jenis sayuran umum yang ditemukan di seluruh dunia, seperti kacang, pare, labu botol, terong, brokoli, kubis, paprika, wortel, kembang kol, mentimun, pepaya, kentang, labu, lobak, dan tomat. Namun, fokus khusus dari penelitian ini adalah pada gambar-gambar labu atau pumpkin yang diklasifikasikan ke dalam 5 kelas: Pumpkin Augmentation, Pumpkin Sharpened, Pumpkin Blur, Pumpkin Normal, dan Pumpkin Contrast. Setiap kelas terdiri dari 170 gambar dengan ukuran 224×224 dalam format *.jpg. Di Google Teachable Machine, sekitar 85% dari total gambar secara acak digunakan untuk pelatihan mesin pembelajaran secara default dan dijadikan data pelatihan, sementara 15% sisanya digunakan sebagai data uji untuk mengevaluasi akurasi hasil pelatihan.

Tabel 1. Jumlah Data Latih dan Data Uji

Objek Gambar	Data Latih	Data Uji
Pumpkin Augmentation	145	25
Pumpkin Sharpened	145	25
Pumpkin Blur	145	25
Pumpkin Normal	145	25
Pumpkin Contrast	145	25

2.2 Pre-Processing

Data yang dikumpulkan secara otomatis dibagi menjadi data pelatihan dan pengujian menggunakan Google Teachable Machine. Sebelum melanjutkan dengan pelatihan model, langkah-langkah pra-pemrosesan

akan dilakukan. Pra-pemrosesan mencakup augmentasi gambar, mempertimbangkan berbagai kondisi gambar baik dari pandangan manusia langsung maupun masalah inheren seperti ketajaman, kecerahan, atau kejelasan.

2.2.1 Gambar Augmentasi

Dari data yang terkumpul, pada tahapan augmentasi gambar itu sendiri memainkan peran krusial dalam meningkatkan kinerja model pembelajaran mendalam pada tugas – tugas visi komputer. Dengan memahami tantangan dan kebutuhan augmentasi gambar, peneliti dapat memilih metode yang tepat dan mengembangkan algoritma baru untuk meningkatkan kemampuan generalisasi model[6]. Tujuan dari Augmentasi data dapat meningkatkan kualitas gambar yang dihasilkan oleh model serta mengurangi masalah overfitting, yang sering terjadi ketika model dilatih pada dataset yang terbatas. Dengan demikian, model dapat lebih baik dalam menggeneralisasi gambar dengan kualitas yang lebih tinggi dan mengurangi ketergantungan pada data asli yang terbatas[7].

Dalam penelitian ini, peneliti mengimplementasikan dan membandingkan beberapa metode augmentasi yang umum digunakan seperti meningkatkan ketajaman, kontras, kombinasi ketajaman dan kontras, lalu menerapkan efek kabur pada gambar. Selain itu dilakukannya evaluasi efektivitasnya dalam klasifikasi menggunakan jaringan mendalam berdasarkan evaluasi kinerja kuantitatif. Hasil evaluasi menunjukkan efektivitas berbagai teknik augmentasi pada berbagai tipe gambar dengan menggunakan beberapa metrik yaitu Accuracy, Sensitivity, Spesifisity, F1-Score, Recall, dan Precision. Pemilihan teknik augmentasi harus dilakukan dengan hati – hati berdasarkan tipe gambar yang diproses, karena efektivitasnya dapat sangat bervariasi tergantung pada jenis gambar, arsitektur jaringan, dan ukuran dataset yang digunakan[8].

Augmentasi gambar ini akan menggunakan beberapa metode dengan nilai kuantitatif untuk memastikan penetapan yang tepat, reproduksi konsisten di berbagai konteks, dan perbandingan yang lebih objektif antara berbagai pengaturan atau metode, seperti yang disebutkan dalam Tabel 2.

Tabel 2. Tingkatan Pada Metode Augmentasi

Metode Augmentasi	Tingkatan
Blur	70%
Sharpened	15%
Contrast	7%

2.2.2 Hyperparameter Tuning

Google Teachable Machine (GTM) memiliki tiga parameter pembelajaran: epoch, ukuran batch, dan laju pembelajaran. Parameter-parameter ini dapat disesuaikan untuk mengoptimalkan penelitian, dimana studi ini memfokuskan pada penyetelan epoch, sementara laju pembelajaran tetap diatur pada 0.001 dan ukuran batch diatur pada 64.

Tabel 3. Hyperparameter Tuning

Parameter	Masukan
Batch	[64]
Learning Rate	[0.001]
Epoch	[50,100,150,200]

2.3 Convolutional Neural Network (CNN)

Algoritma Convolutional Neural Network (CNN) adalah salah satu metode yang menggunakan teknologi pembelajaran mendalam. CNN mampu mengklasifikasikan gambar dan video, serta mendeteksi objek dalam gambar atau wilayah tertentu di dalam gambar. CNN terdiri dari lapisan-lapisan yang memiliki susunan neuron dalam bentuk tiga dimensi, yaitu lebar, tinggi, dan kedalaman. Lebar dan tinggi mencerminkan ukuran lapisan, sedangkan kedalaman menunjukkan jumlah lapisan. Jaringan saraf konvolusional

(Convolutional Neural Networks, CNN) terdengar seperti kombinasi antara matematika dan biologi, namun jaringan ini telah menjadi salah satu inovasi paling berpengaruh dalam bidang visi komputer dan pemrosesan gambar. CNN adalah versi khusus dari multilayer perceptron (MLP). Mereka dikembangkan berdasarkan cara kerja neuron di korteks visual hewan. CNN memiliki beberapa lapisan, dimulai dengan Lapisan Ekstraksi Fitur (Feature Extraction Layer) yang mencakup Lapisan Konvolusional (Convolutional Layer) dan Lapisan Penggabungan (Pooling Layer). Lapisan kedua adalah Lapisan Terhubung Seluruhnya (Fully Connected Layer), yang bertugas mengklasifikasikan fitur-fitur yang diekstraksi dari lapisan sebelumnya[9].

2.4 Google Teachable Machine

Teachable Machine adalah sebuah situs web yang dibuat oleh Google untuk membantu pengguna membangun model pembelajaran mesin tanpa menulis kode. Platform ini menggunakan algoritma *Convolutional Neural Network* (CNN), jenis algoritma pembelajaran mendalam yang dirancang untuk memproses data dalam format dua dimensi seperti gambar dan audio[4]. *Teachable Machine* menggunakan *Convolutional Neural Networks* (CNN) untuk mengenali gambar. Awalnya, gambar-gambar tersebut diubah ukurannya. CNN terdiri dari tiga jenis utama jaringan saraf, yaitu lapisan konvolusi, lapisan pooling, dan lapisan terhubung penuh[10]. CNN kemudian mencari pola seperti tepi dan tekstur, dan hasilnya dimasukkan ke dalam lapisan-lapisan yang menambah kompleksitas. Lapisan-lapisan berikutnya mengurangi ukuran data, dan semua informasi dikumpulkan di lapisan terakhir untuk membuat prediksi akhir. Proses ini dilatih dengan data sehingga model dapat mengenali pola pada gambar-gambar baru. Teachable Machine menyederhanakan pembuatan model ini tanpa perlu banyak memahami teknologi dibalikinya.

2.5 Receiver Operating Characteristic (ROC)

Kurva ROC sering digunakan untuk mengevaluasi kinerja klasifikasi karena secara efektif menilai algoritma. Kurva ini membandingkan sensitivitas True Positive Rate (TPR), yang direpresentasikan di sepanjang sumbu vertikal atau sumbu y, dengan spesifisitas False Positive Rate (FPR). Rumus sensitivitas dan spesifisitas disajikan dalam Persamaan 1 dan 2.

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \tag{1}$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \tag{2}$$

Untuk menghitung dan Dalam menentukan penggunaan parameter epoch mana yang lebih baik, perhitungan AUC (area di bawah kurva) digunakan. AUC mewakili luas di bawah kurva. Nilai AUC selalu berkisar antara 0 dan 1. AUC dihitung berdasarkan luas gabungan trapesium yang terbentuk oleh titik – titik sensitivitas dan spesifisitas[11].

2.6 Evaluasi Model

Dalam studi ini, evaluasi model juga dilakukan untuk menilai kesesuaian klasifikasi menggunakan matriks kebingungan (confusion matrix). Penyesuaian ini bertujuan untuk mengevaluasi hasil kinerja model, di mana matriks kebingungan juga dapat menunjukkan jumlah kesalahan prediksi untuk kelas minoritas maupun mayoritas[12].

Tabel 4. Confusion Matrix

		Prediction	
		Positive	Negative
Actual	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

2.6.1 Accuracy

Accuracy adalah ukuran yang menunjukkan seberapa baik sebuah model klasifikasi dapat memprediksi kelas yang benar dari data yang diberikan[13].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

2.6.2 Recall

Recall adalah matrik yang menunjukkan berapa banyak instansi positif sejati yang diklasifikasikan dalam suatu kategori tertentu dibandingkan dengan jumlah total instansi aktual yang termasuk dalam kategori tersebut[13].

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

2.6.3 Precision

Precision adalah rasio antara jumlah label yang diidentifikasi dengan benar dan jumlah total label yang diidentifikasi[14].

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (5)$$

2.6.4 F1-Score

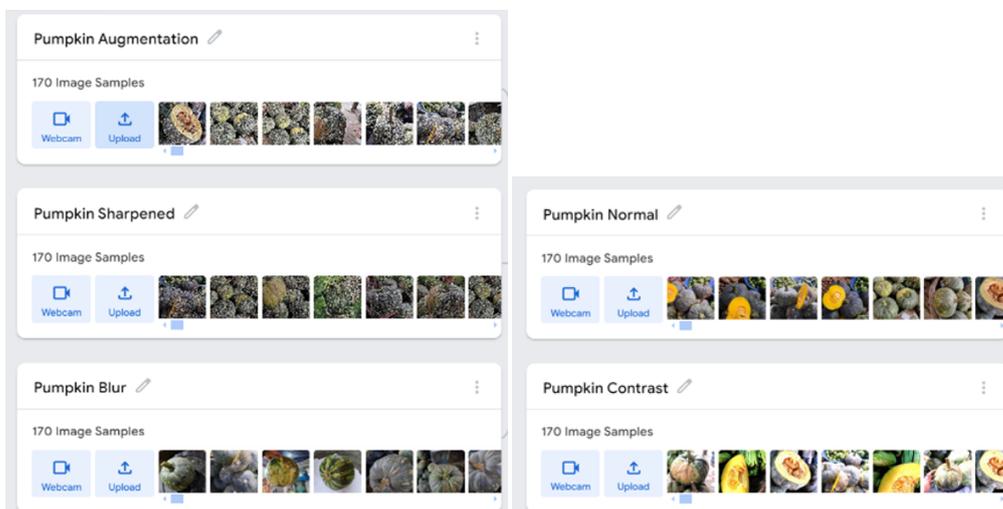
F1-Score adalah kombinasi dari nilai presisi dan recall, yang digunakan sebagai nilai pengukuran[15].

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

3. Hasil dan Pembahasan

3.1 Pengumpulan Dataset

Dataset ini terdiri dari gambar-gambar labu yang dikategorikan ke dalam 5 kelas yaitu labu yang diperbesar, labu yang dipersempit, labu yang kabur, labu normal, dan labu dengan kontras. Setiap kelas mengandung 170 gambar berukuran 224×224 piksel dalam format *.jpg.



Gambar 2. Dataset Labu

3.2 Pre-Processing

Pada tahap ini, data yang terkumpul akan mengalami augmentasi dan dibagi menjadi 3 kelas: labu kontras, labu yang dipersempit, dan labu yang kabur. Kelas augmentasi labu merupakan kombinasi dari gambar-gambar yang dipersempit dan yang kontras. Selain itu, dalam langkah pra-pemrosesan ini, pembagian data secara otomatis diatur oleh Teachable Machine dengan pembagian data pelatihan sebesar 85% dan data pengujian sebesar 15%.

```

import cv2
import os

def adjust_contrast_brightness(image, alpha=1.0, beta=0):
    """Fungsi untuk menyesuaikan kontras dan pencahayaan pada gambar."""
    adjusted = cv2.convertScaleAbs(image, alpha=alpha, beta=beta)
    return adjusted

def apply_contrast_brightness(input_dir, output_dir, alpha=1.0, beta=0):
    """Fungsi untuk menerapkan penyesuaian kontras dan pencahayaan pada semua gambar dalam direktori input."""
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    for filename in os.listdir(input_dir):
        file_path = os.path.join(input_dir, filename)
        image = cv2.imread(file_path)

        if image is None:
            continue

        adjusted_image = adjust_contrast_brightness(image, alpha, beta)

        cv2.imwrite(os.path.join(output_dir, filename), adjusted_image)
        print(f"Adjusted {filename} and saved to {output_dir}")

input_dir = './src_final/train_fix/Pumpkin_200'
output_dir = './src_final/train_fix2/Pumpkin_200_contrast'

apply_contrast_brightness(input_dir, output_dir, alpha=1.5, beta=5)
    
```

Gambar 3. Kode Augmentasi Contrast

```

import cv2
import os
import numpy as np

def sharpen_image(image, scale_factor=1.0):
    initial_kernel = np.array([[ -1, -1, -1],
                                [ -1,  9, -1],
                                [ -1, -1, -1]])
    adjusted_kernel = initial_kernel * scale_factor
    sharpened = cv2.filter2D(image, -1, adjusted_kernel)

    return sharpened

def apply_sharpening(input_dir, output_dir, scale_factor=0.7):
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    for filename in os.listdir(input_dir):
        file_path = os.path.join(input_dir, filename)
        image = cv2.imread(file_path)

        if image is None:
            continue

        sharpened_image = sharpen_image(image, scale_factor)
        cv2.imwrite(os.path.join(output_dir, filename), sharpened_image)
        print(f"Sharpened {filename} and saved to {output_dir}")

input_dir = './src2/train_not_blur2'
output_dir = './src2/sharpened4'

apply_sharpening(input_dir, output_dir, scale_factor=0.7)
    
```

Gambar 4. Kode Augmentasi Sharpened

```

import cv2
import os

def apply_blur(image, sigma=1.0):
    blurred = cv2.GaussianBlur(image, (0, 0), sigma)
    return blurred

def apply_blur_to_folder(input_dir, output_dir, sigma=1.0):
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    for filename in os.listdir(input_dir):
        file_path = os.path.join(input_dir, filename)
        image = cv2.imread(file_path)

        if image is None:
            continue

        blurred_image = apply_blur(image, sigma)
        output_path = os.path.join(output_dir, filename)
        cv2.imwrite(output_path, blurred_image)
        print(f"Blurred {filename} and saved to {output_path}")

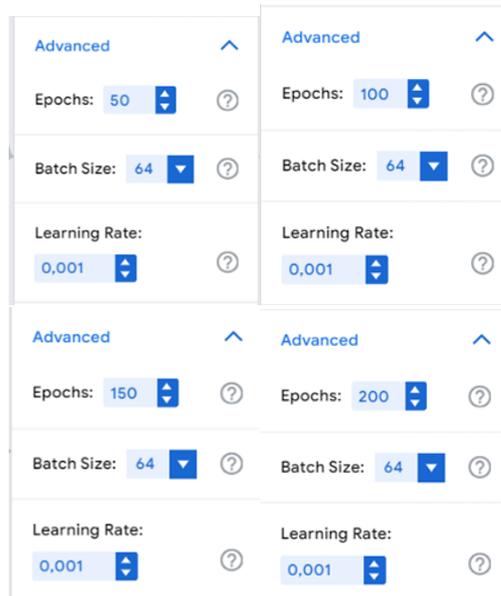
input_dir = './src2/train_not_blur2'
output_dir = './src2/blur_image_100%'

apply_blur_to_folder(input_dir, output_dir, sigma=1.0)
    
```

Gambar 5. Kode Augmentasi Blur

3.3 Implementasi Model

Dalam proses implementasi model, peneliti menggunakan berbagai parameter untuk membandingkan hasil optimal. Ini termasuk mengatur epoch dari 50 hingga 200, dan ukuran batch 64 dan 128 untuk setiap eksperimen di setiap epoch. Selain itu, laju pembelajaran tetap konstan pada 0.001 untuk semua uji coba.



Gambar 6. Konfigurasi Penyetelan Parameter

Setelah penyetelan parameter, implementasi model dapat dilanjutkan dengan melatihnya. Hasil pelatihan model ditunjukkan dalam Gambar 7, 8, 9, dan 10.

Accuracy per class

CLASS	ACCURACY	# SAMPLES
Pumpkin Augmentati...	0.77	26
Pumpkin Sharpened	0.73	26
Pumpkin Blur	0.92	26
Pumpkin Normal	0.85	26
Pumpkin Contrast	0.85	26

Gambar 7. Accuracy Per Class Dengan Epoch 50

CLASS	ACCURACY	# SAMPLES
Pumpkin Augmentati...	0.92	26
Pumpkin Sharpened	0.73	26
Pumpkin Blur	0.96	26
Pumpkin Normal	0.88	26
Pumpkin Contrast	0.73	26

Gambar 8. Accuracy Per Class Dengan Epoch 100

CLASS	ACCURACY	# SAMPLES
Pumpkin Augmentati...	0.65	26
Pumpkin Sharpened	0.77	26
Pumpkin Blur	0.92	26
Pumpkin Normal	0.81	26
Pumpkin Contrast	0.88	26

Gambar 9. Accuracy Per Class Dengan Epoch 150

CLASS	ACCURACY	# SAMPLES
Pumpkin Augmentati...	0.88	26
Pumpkin Sharpened	0.73	26
Pumpkin Blur	0.85	26
Pumpkin Normal	0.88	26
Pumpkin Contrast	0.92	26

Gambar 10. Accuracy Per Class Dengan Epoch 200

3.4 Evaluasi Model

Pada langkah berikutnya, model yang telah dibuat sebelumnya dievaluasi. Evaluasi ini menggunakan matriks kebingungan, yang berfungsi untuk mengukur kinerja suatu model atau algoritma. Dengan menggunakan matriks kebingungan, kita dapat mengidentifikasi tingkat kesalahan dari berbagai algoritma dengan memeriksa metrik seperti accuracy, precision, recall, dan F1-Score.

Class	Pumpkin Augm...	Pumpkin Shar...	Pumpkin Blur	Pumpkin Norm...	Pumpkin Cont...
Pumpkin Augm...	20	6	0	0	0
Pumpkin Shar...	7	19	0	0	0
Pumpkin Blur	0	0	24	2	0
Pumpkin Norm...	0	0	1	22	3
Pumpkin Cont...	0	0	0	4	22
Prediction	Pumpkin Augm...	Pumpkin Shar...	Pumpkin Blur	Pumpkin Norm...	Pumpkin Cont...

Gambar 11. Confusion Matrix Dengan Epoch 50

Class	Pumpkin Augm...	24	2	0	0	0
	Pumpkin Shar...	7	19	0	0	0
	Pumpkin Blur	0	0	25	1	0
	Pumpkin Norm...	0	0	1	23	2
	Pumpkin Cont...	2	0	0	5	19
	Prediction	Pumpkin Augm...	Pumpkin Shar...	Pumpkin Blur	Pumpkin Norm...	Pumpkin Cont...

Gambar 12. Confusion Matrix Dengan Epoch 100

Class	Pumpkin Augm...	17	9	0	0	0
	Pumpkin Shar...	6	20	0	0	0
	Pumpkin Blur	0	0	24	2	0
	Pumpkin Norm...	0	0	1	21	4
	Pumpkin Cont...	0	0	0	3	23
	Prediction	Pumpkin Augm...	Pumpkin Shar...	Pumpkin Blur	Pumpkin Norm...	Pumpkin Cont...

Gambar 13. Confusion Matrix Dengan Epoch 150

Class	Pumpkin Augm...	23	3	0	0	0
	Pumpkin Shar...	7	19	0	0	0
	Pumpkin Blur	0	0	22	4	0
	Pumpkin Norm...	1	0	1	23	1
	Pumpkin Cont...	0	0	0	2	24
	Prediction	Pumpkin Augm...	Pumpkin Shar...	Pumpkin Blur	Pumpkin Norm...	Pumpkin Cont...

Gambar 14. Confusion Matrix Dengan Epoch 200

Secara keseluruhan, peningkatan jumlah epoch cenderung meningkatkan kinerja pada beberapa kelas tertentu, namun peningkatan ini tidak merata di semua kelas. Secara khusus, model yang dilatih dengan 100 dan 200 epoch menunjukkan hasil yang secara umum lebih baik. Peningkatan yang signifikan teramati pada kelas-kelas seperti “augmentation” dan “blur”, yang menunjukkan kenaikan yang signifikan dalam

akurasi prediksi. Namun, kelas-kelas lain tidak secara konsisten mendapatkan manfaat dari peningkatan epoch. Variasi ini menyoroti bahwa meskipun lebih banyak epoch dapat menghasilkan kinerja model yang lebih baik, untuk mencapai hasil terbaik diperlukan penyetelan parameter yang cermat dan evaluasi berkelanjutan. Penting untuk memantau kinerja di semua kelas untuk memastikan bahwa model tidak overfitting atau underperforming di area tertentu, memerlukan pendekatan pelatihan yang seimbang dan teliti.

Tabel 5. Hasil Evaluasi Klasifikasi dengan Epoch 50

Class	Accuracy	Precision	Recall	F1-Score
Pumpkin Augmentation	77%	74%	77%	75%
Pumpkin Sharpened	73%	76%	73%	74%
Pumpkin Blur	92%	96%	92%	94%
Pumpkin Normal	85%	78%	95%	86%
Pumpkin Contrast	85%	100%	84%	91%

Tabel 6. Hasil Evaluasi Klasifikasi dengan Epoch 100

Class	Accuracy	Precision	Recall	F1-Score
Pumpkin Augmentation	92%	72%	92%	81%
Pumpkin Sharpened	73%	90%	73%	81%
Pumpkin Blur	96%	96%	96%	96%
Pumpkin Normal	88%	79%	95%	86%
Pumpkin Contrast	73%	90%	79%	84%

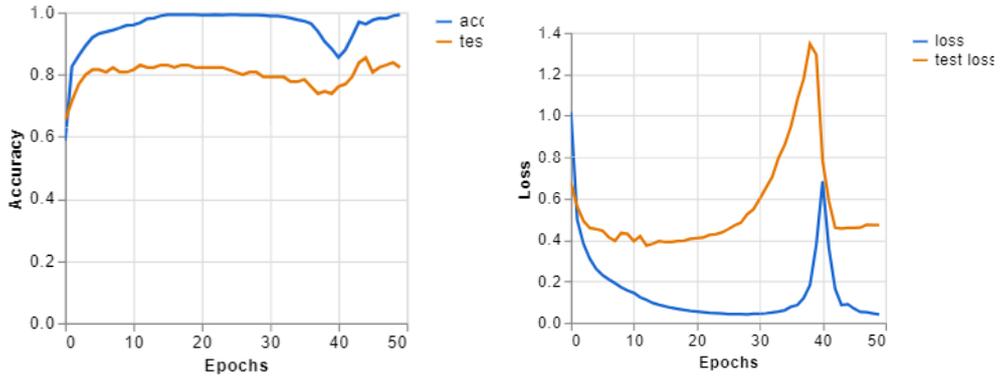
Tabel 7. Hasil Evaluasi Klasifikasi dengan Epoch 150

Class	Accuracy	Precision	Recall	F1-Score
Pumpkin Augmentation	65%	74%	65%	69%
Pumpkin Sharpened	77%	69%	77%	72%
Pumpkin Blur	92%	96%	92%	94%
Pumpkin Normal	81%	80%	80%	80%
Pumpkin Contrast	88%	85%	88%	87%

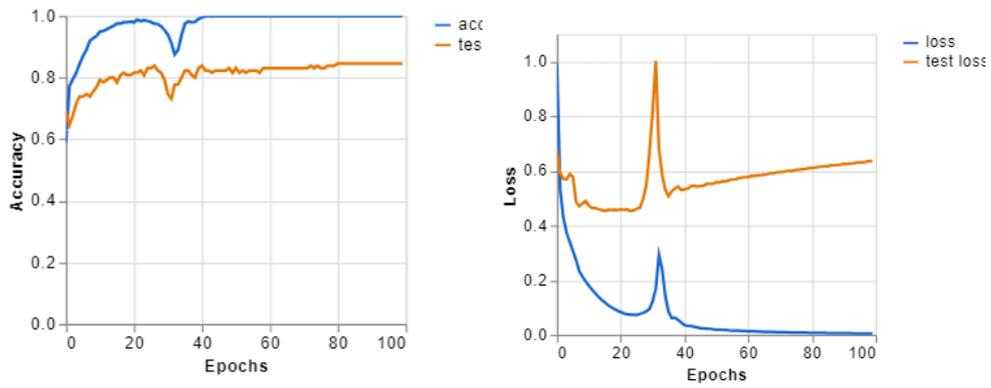
Tabel 8. Hasil Evaluasi Klasifikasi dengan Epoch 200

Class	Accuracy	Precision	Recall	F1-Score
Pumpkin Augmentation	88%	74%	88%	80%
Pumpkin Sharpened	73%	86%	73%	79%
Pumpkin Blur	85%	95%	84%	90%
Pumpkin Normal	88%	79%	92%	85%
Pumpkin Contrast	92%	96%	92%	94%

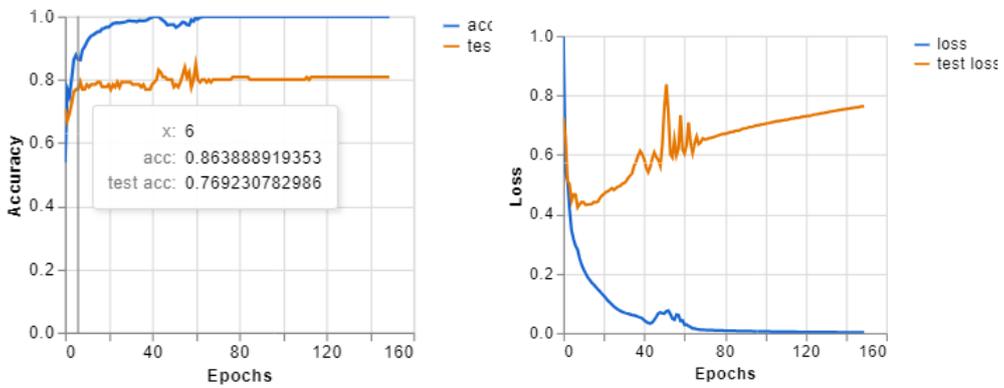
Dari hasil pelatihan model dengan konfigurasi yang sama (epoch 50, ukuran batch 64, LR 0.001), teramati variasi kinerja yang signifikan di antara berbagai jenis augmentasi gambar labu. Pada epoch 50, augmentasi blur menunjukkan kinerja terbaik dengan accuracy 92%, precision 96%, recall 92%, dan F1-Score 94%. Selain itu, augmentasi normal dan contrast juga menunjukkan kinerja kuat dengan akurasi masing-masing 85%, presisi 78% dan 100%, recall 95% dan 84%, serta F1-Score 86% dan 91%. Sebaliknya, augmentasi sharpened dan augmentation menunjukkan kinerja sedikit lebih rendah dengan nilai accuracy dan F1-Score berkisar antara 73% hingga 77%.



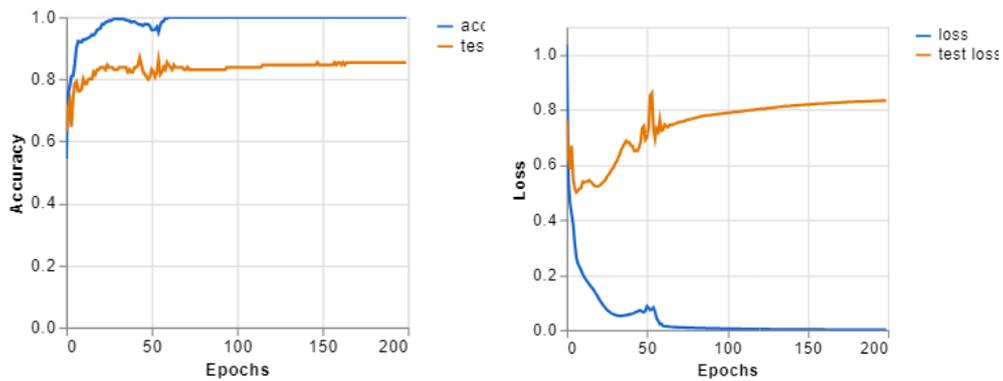
Gambar 15. ROC Accuracy dan Loss Dengan Epoch 50



Gambar 16. ROC Accuracy dan Loss Dengan Epoch 100



Gambar 17. ROC Accuracy dan Loss Dengan Epoch 150



Gambar 18. ROC Accuracy dan Loss Dengan Epoch 200

Pada konfigurasi batch size 64, learning rate 0.001, dan epoch 50, grafik ROC accuracy menunjukkan bahwa akurasi pelatihan (garis biru) meningkat signifikan dan stabil di sekitar 100% setelah sekitar 10 epoch. Namun, akurasi pengujian (garis oranye) tidak berubah di sekitar 60% sampai 70%, yang mengindikasikan kemungkinan overfitting. Grafik ROC loss menunjukkan bahwa kerugian pelatihan menurun, tetapi kerugian pengujian mengalami puncak tinggi sekitar epoch 35, menunjukkan masalah dalam proses pengujian atau terdapat anomali data. Pada konfigurasi dengan epoch 100, ROC accuracy memperlihatkan bahwa akurasi pelatihan terus meningkat mendekati 100%, sementara akurasi pengujian stabil di sekitar 80% setelah 20 epoch, yang juga menunjukkan overfitting. ROC loss menunjukkan penurunan signifikan pada kerugian pelatihan di awal, namun kerugian pengujian mengalami fluktuasi besar sekitar epoch 5, menandakan adanya masalah validasi atau anomali data.

Untuk epoch 150, grafik ROC accuracy menunjukkan akurasi pelatihan stabil di sekitar 86%, namun akurasi pengujian fluktuatif di sekitar 77%, yang menunjukkan masalah generalisasi. Grafik ROC loss memperlihatkan penurunan tajam di awal pelatihan, dengan fluktuasi kerugian pengujian setelah sekitar 40 epoch, menunjukkan overfitting dan ketidakstabilan selama validasi. Dengan epoch 200, ROC Accuracy menunjukkan akurasi pelatihan yang stabil tinggi, sementara akurasi pengujian fluktuatif dengan nilai lebih rendah, mengindikasikan overfitting. Grafik ROC loss memperlihatkan penurunan tajam pada kerugian pelatihan di awal, namun kerugian pengujian cenderung meningkat seiring waktu, menunjukkan overfitting yang signifikan. Dalam analisis hasil eksperimen dengan parameter batch size 64, learning rate 0.001, dan berbagai jumlah epoch, terlihat pola yang konsisten menunjukkan potensi overfitting pada model. Pada setiap pengujian, akurasi pelatihan relatif stabil di 100%, sementara akurasi pengujian stabil di tingkat yang lebih rendah, di antara 60% hingga 80%.

Hal ini menunjukkan bahwa model cenderung belajar dengan sangat baik dari data pelatihan tetapi tidak mampu menggeneralisasi dengan efektif pada data yang belum pernah dilihat sebelumnya. Selain itu, grafik kerugian menunjukkan fluktuasi yang signifikan pada grafik test loss setelah beberapa epoch, yang mengindikasikan bahwa model dapat mulai mengalami overfitting dengan mempelajari noise atau detail yang tidak relevan dari data pelatihan. Dari hasil analisis ini, konfigurasi batch size 64, learning rate 0.001, dan epoch 100 tampaknya menjadi yang paling efektif dalam klasifikasi pengenalan labu. Meskipun model masih mengalami overfitting, akurasi pengujian yang stabil di sekitar 80% menunjukkan performa yang relatif lebih baik dibandingkan konfigurasi lainnya. Namun, untuk mencapai performa yang lebih optimal, diperlukan penggunaan penyempurnaan dan penyesuaian parameter model untuk mengatasi masalah overfitting ini serta meningkatkan kemampuan model dalam menghadapi data yang tidak terlihat sebelumnya.

Tabel 9. Hasil Sensitivity dan Specificity Per Epoch

Class	Sensitivity				Specificity			
	50	100	150	200	50	100	150	200
Pumpkin Augmentation	76%	92%	65%	88%	87%	83%	88%	87%
Pumpkin Sharpened	73%	73%	76%	73%	88%	86%	83%	94%
Pumpkin Blur	92%	96%	92%	84%	98%	98%	96%	92%
Pumpkin Normal	95%	95%	95%	95%	97%	98%	98%	98%
Pumpkin Contrast	84%	79%	88%	92%	100%	96%	100%	100%

Pada hasil sensitiviy dan spesificity berbagai epoch menunjukkan variasi kinerja model di setiap epoch yang diuji. Pada epoch ke 50, kelas “Pumpkin Blur” dan “Pumpkin Normal” menunjukkan performa yang sangat baik dengan sensitivity dan spesificity yang tinggi, yaitu masing-masing 92% dan 98%, serta 95% dan 97%. Sebaliknya, sensitivity pada kelas “Pumpkin Augmentation” dan “Pumpkin Sharpened” lebih rendah pada epoch ini, dengan nilai masing – masing 76% dan 73%, meskipun spesificity relatif tinggi pada 87% dan 88%. Pada epoch ke 100, sensitivity “Pumpkin Augmentation” mengalami peningkatan signifikan menjadi 92%, meskipun spesificity menurun menjadi 83%. Di sisi lain, kelas “Pumpkin Contrast” menunjukkan penurunan sensitivity menjadi 79%, tetapi tetap mempertahankan spesifisitas yang tinggi di 96%. Pada epoch 150, terdapat penurunan sensitivity yang signifikan pada “Pumpkin Augmentation” menjadi

65%, meskipun specificity tetap stabil di 88%. Sementara itu, sensitivity “Pumpkin Blur” dan “Pumpkin Normal” tetap tinggi dengan specificity mendekati nilai maksimum. Pada epoch ke 200, sensitivity pada “Pumpkin Contrast” mengalami peningkatan kembali menjadi 92%, dengan spesifisitas tetap di 100%. Kelas lainnya menunjukkan hasil yang konsisten, dengan “Pumpkin Normal” tetap mempertahankan specificity tertinggi di 98%. Secara keseluruhan, sensitivity dan specificity menunjukkan kecenderungan perbaikan pada sebagian besar kelas seiring dengan bertambahnya epoch, meskipun terdapat beberapa fluktuasi yang perlu diperhatikan.

4. Analisa Hasil dan Model Klasifikasi

Setelah melakukan tahap implementasi dan evaluasi, maka bisa dilanjutkan ke tahap analisa hasil dan model klasifikasi yang menggunakan parameter berbeda. Hasil menunjukkan bahwa “Pumpkin Blur” secara konsisten tampil sebagai model terbaik di semua epoch, dengan sensitivity yang sangat tinggi sebesar 96% pada epoch 100, dan specificity di atas 90% pada semua threshold, meskipun sedikit menurun pada epoch 200 menjadi 84%. “Pumpkin Normal” menunjukkan sensitivity sangat tinggi dengan kestabilan di 95% di semua threshold, dan specificity yang konsisten, dengan penurunan performa yang teramati pada epoch 150 di mana akurasi menurun menjadi 81%. “Pumpkin Contrast” menunjukkan performa optimal pada threshold tinggi, dengan beberapa specificity konsisten 100% pada epoch 50 sampai 100, serta peningkatan precision pada epoch 100 dan 150. “Pumpkin Augmentation” dan “Pumpkin Sharpened” menunjukkan fluktuasi performa yang signifikan berdasarkan epoch dimana pada “Pumpkin Augmentation” mengalami penurunan akurasi menjadi 65% dan recall menjadi 65% pada epoch 150, sedangkan “Pumpkin Sharpened” menunjukkan precision tinggi pada epoch 100, mencapai 90%, tetapi dengan akurasi dan recall yang bervariasi. Secara keseluruhan, “Pumpkin Blur” adalah model yang paling stabil dan efektif dengan performa konsisten, diikuti oleh “Pumpkin Normal” yang menunjukkan kinerja baik namun dengan beberapa penurunan pada epoch lebih tinggi, dan “Pumpkin Contrast” yang juga menunjukkan performa baik dengan sensitivitas dan spesifisitas tinggi di threshold yang lebih tinggi.

5. Simpulan

Dari rangkaian tahapan pengujian yang telah dilakukan, didapatkannya suatu kesimpulan bahwa algoritma deep learning CNN pada Google Teachable Machine bisa diaplikasikan dalam proses untuk augmentasi gambar untuk identifikasi labu, sehingga dari penelitian ini dapat memberikan hasil yang bervariasi untuk melakukan identifikasi dan klasifikasi tanaman di bidang pertanian terutama dalam menganalisa hasil dengan penggunaan parameter yang berbeda seperti penyetulan parameter epoch dan melakukan augmentasi gambar seperti contrast, sharpened, dan blur sebelum dijadikan kelas. Dimana dengan faktor augmentasi gambar ini dapat membantu dalam mengenali fitur penting dan meningkatkan sensitivitas terhadap detail. Penelitian ini juga diharapkan dapat memberikan wawasan praktis tentang penerapan teknologi pembelajaran mesin dalam kehidupan sehari – hari. Pendekatan ini diharapkan dapat membantu petani dalam mengidentifikasi jenis labu dengan cepat dan akurat, mengurangi kesalahan manusia, dan meningkatkan produktivitas pertanian.

Berdasarkan analisis dan pengujian dengan berbagai parameter epoch, pada kelas “Pumpkin Blur” menghasilkan hasil terbaik dengan akurasi sebesar 96% pada epoch 100. Sedangkan pada F1-Score “Pumpkin Blur” pada epoch 100 dan “Pumpkin Contrast” pada epoch 200 memiliki skor F1-Score tertinggi yaitu 96% dan 94%. Namun, “Pumpkin Blur” pada epoch 100 menunjukkan F1-Score tertinggi di antara semua kelas dan epoch. Untuk meningkatkan performa model, perlu penyesuaian parameter lebih lanjut agar model dapat menghadapi data baru dengan lebih baik. Tujuannya adalah menerapkan pembelajaran mesin menggunakan Google Teachable Machine untuk mengidentifikasi labu dari berbagai data gambar labu, melatih model untuk mengenali berbagai jenis labu dengan akurasi tinggi sesuai dengan parameter yang telah ditetapkan. Harapan ke depannya penelitian ini dapat dikembangkan lebih lanjut dengan penyesuaian parameter yang lebih cermat untuk mencapai performa optimal dalam menghadapi data yang belum pernah dilihat sebelumnya.

Sumber dana

Tidak ada dana yang digunakan dalam penelitian ini. Semua hasilnya murni diperoleh dari kerja keras dan dedikasi peneliti dalam menyelesaikan program ini.

Pustaka

- [1] C. Chazar and M. H. Rafsanjani, "Penerapan teachable machine pada klasifikasi machine learning untuk identifikasi bibit tanaman," *Pros. Semin. Nas. Inov. dan Adopsi Teknol.*, vol. 2, no. 1, pp. 32–40, 2022.
- [2] Y. Pratama, U. Lestari, and A. Hamzah, "Pemanfaatan aplikasi teachable machine untuk pengenalan binatang menggunakan konsep convolutional neural network (cnn)," *J. Scr.*, vol. 10, no. 1, pp. 10–20, 2022, online available: <https://ejournal.akprind.ac.id/index.php/script/article/view/4067>.
- [3] D. I. Salintohe, Hasniati, and I. A. Musdar, "Implementasi machine learning untuk mengidentifikasi tanaman hias pada aplikasi tierra," *Jtriste*, vol. 9, no. 1, pp. 1–15, 2022.
- [4] Y. Jumaryadi, A. M. Ihsan, and B. Priambodo, "Klik: Kajian ilmiah informatika dan komputer klasifikasi jenis buah-buahan menggunakan citra digital dengan metode convolutional neural networks," *Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 3, pp. 1737–1746, 2023.
- [5] M. M. Ali, T. Hariyati, M. Y. Pratiwi, and S. Afifah, "Metodologi penelitian kuantitatif dan penerapannya dalam penelitian," *Educ. J.*, vol. 2, no. 2, pp. 1–6, 2022.
- [6] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognit.*, vol. 137, p. 109347, 2023.
- [7] A. Kebaili, J. Lapuyade-Lahorgue, and S. Ruan, "Deep learning approaches for data augmentation in medical imaging: A review," *J. Imaging*, vol. 9, no. 4, 2023.
- [8] E. Goceri, "Medical image data augmentation: techniques, comparisons and interpretations," *Springer Netherlands*, vol. 56, no. 11, 2023.
- [9] R. A. Tilasefana and R. E. Putra, "Penerapan metode deep learning menggunakan algoritma cnn dengan arsitektur vgg net untuk pengenalan cuaca," *J. Informatics Comput. Sci.*, vol. 5, no. 1, pp. 48–57, 2023.
- [10] T. P. Ananda, S. V. Widyasari, M. I. Muttaqin, and A. Stefanie, "Identifikasi tingkat kematangan buah pepaya menggunakan metode convolutional neural network (cnn)," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 3, pp. 2094–2097, 2023.
- [11] L. Qadrini, A. Sepperwali, and A. Aina, "Decision tree dan adaboost pada klasifikasi penerima program bantuan sosial," *J. Inov. Penelit.*, vol. 2, no. 7, pp. 1959–1966, 2021.
- [12] W. I. Sabilla and C. B. Vista, "Implementasi smote dan under sampling pada imbalanced dataset untuk prediksi kebangkrutan perusahaan," *J. Komput. Terap.*, vol. 7, no. 2, pp. 329–339, 2021.
- [13] A. Kusuma and H. N. Irmanda, "Analisis sentimen pada ulasan aplikasi indodax di google play store menggunakan metode support vector machine," 2022.
- [14] S. A. P. Perdana, T. B. Aji, and R. Ferdiana, "Aspect category classification dengan pendekatan machine learning menggunakan dataset bahasa indonesia (aspect category classification with machine learning approach using indonesian language dataset)," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 10, no. 3, pp. 229–235, 2021.
- [15] S. Clara, D. L. Prianto, R. A. Habsi, E. F. Lumbantobing, and N. Chamidah, "Implementasi seleksi fitur pada algoritma klasifikasi machine learning untuk prediksi penghasilan pada adult income dataset," *Semin. Nas. Mhs. Ilmu Komput. dan Apl. Jakarta-Indonesia*, vol. 2, no. 1, pp. 741–747, 2021.