

Perbandingan Algoritma SVM dan CNN menggunakan PCA untuk Klasifikasi Kematangan Jeruk Keprok

(Comparison of SVM and CNN Algorithms Using PCA for Mandarin Orange Classification)

Sunarso Sunarso^{1*}, Chairani Chairani², Joko Triloka³, Rio Kurniawan⁴

Institut Informatika dan Bisnis Darmajaya, Bandar Lampung, Indonesia^{1,2,3,4}

Sunarso494@gmail.com^{1,3,4}, chairani@mail.darmajaya.ac.id²



Riwayat Artikel:

Diterima pada 25 Maret 2025

Revisi 1 pada 10 April 2025

Revisi 2 pada 20 April 2025

Revisi 3 pada 01 Mei 2025

Disetujui pada 15 Mei 2025

Abstract

Purpose: This study aims to compare the SVM and CNN machine learning algorithms by combining PCA as data reduction to see which level of accuracy is higher with orange objects.

Methodology/approach: created using the waterfall model, the system used to create the model is matlab ver r2022a, using the help of the python programming language to separate the datasets used, the datasets used come from kaagle including the following (<https://www.kaggle.com/datasets/raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables>), and Orange disease dataset(<https://www.kaggle.com/datasets/jonathansilva2020/orange-diseases-dataset>).

Results/findings: The results obtained from the Matlab test using the CNN and PCA algorithms obtained an accuracy of 76.4% and the SVM and PCA classification models obtained an accuracy of 98.89%.

Conclusions: This research was successful with the results of combining the SVM and PCA algorithms which had high accuracy results compared to CNN and PCA.

Limitations: In this study, the focus is only on comparing the SVM and CNN algorithms with the help of PCA to see which one has the higher level of accuracy between the two. The dataset was only taken from Kaagle, and the software used to create the model was Matlab.

Contribution: This research is expected to be a reference for creating models in the future that can be applied to the classification process of automated products.

Keywords: *Classification, CNN, Mandarin Orange, PCA, SVM.*

How to Cite: Sunarso, S., Chairani, C., Triloka, J., Kurniawa, R. (2025). Perbandingan Algoritma SVM dan CNN menggunakan PCA untuk Klasifikasi Kematangan Jeruk Keprok. *Jurnal Ilmu Siber dan Teknologi Digital*, 3(2), 203-221.

1. Pendahuluan

Jeruk merupakan buah yang sering dikonsumsi oleh kebanyakan masyarakat Indonesia, dimana yang dilansir dari Kementerian Koordinator Bidang Perekonomian Republik Indonesia pada tahun 2021, dimana produksi buah jeruk siam mencapai 2,4 juta ton pada tahun tersebut (Limanseto, 2022). Banyak orang menyukai buah jeruk karena rasanya yang begitu manis dan segar, selain itu jeruk juga mengandung vitamin c yang begitu tinggi. Selain dimakan langsung jeruk juga bisa diolah menjadi berbagai olahan seperti minuman, makanan, obat dan lain-lain.

Karena tingginya peminat buah jeruk sendiri menurut data yang disebutkan oleh Kementrian Koordinator Bidang Perekonomian Republik Indonesia sebelumnya maka pemilihan buah yang segar menjadi suatu yang sangat penting karena dapat mempengaruhi rasanya. Pada pengklasifikasian buah

jeruk kebanyakan industri dan mayoritas masyarakat Indonesia menggunakan metode manual seperti pemilahan satu persatu buah jeruk berdasarkan kematangannya (Siskandar et al., 2020). Penggunaan teknik pengklasifikasian manual memiliki beberapa kekurangan yaitu akurasi yang dihasilkan tidak konsisten jika dilakukan dengan jumlah yang cukup banyak. Hal tersebut bisa terjadi karena pensortiran/pengklasifikasiannya oleh para pekerja bersifat subjektif jadi setiap pekerja memiliki pandangan yang berbeda dalam pengklasifikasiannya sendiri (Siskandar et al., 2020). Oleh karena itu, pengembangan sistem klasifikasi otomatis menjadi solusi yang penting untuk menjamin kualitas produk secara lebih efisien dan objektif.

Terdapat berbagai algoritma yang bisa diimplementasikan untuk klasifikasi gambar dan diantaranya adalah Convolutional Neural Network (CNN) dan Support Vector Machine (SVM). Klasifikasi Penyakit pada Citra Buah Jeruk Menggunakan Convolutional Neural Network (CNN) dengan Arsitektur Alexnet. Pada penelitian tersebut menggunakan dataset dari Kaagle.com dengan jumlah data sebanyak 1790 citra buah jeruk dengan 4 kelas, setiap kelasnya terdiri dari :fresh, blackspot, canker, dan greening, dengan gambar pelatihan yang diubah menjadi 227x227 pixel. Dilakukan pemisahan data training dan data testing yaitu 90%:10%, 80%:20%, 70%:30%, setiap pemrosesan mengeksekusi 300 epoch. Didapatkan dataset dengan 90% data training dan 10% data testing memperoleh akurasi yang tinggi yaitu 94,34% (Nisya, Wulansari, & Wartariyus, 2023; Suroto & Friadi, 2023).

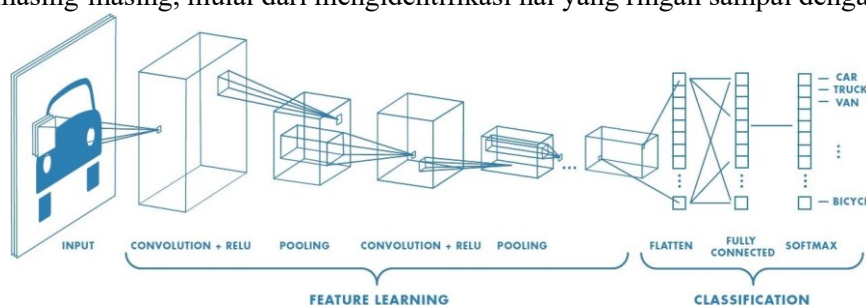
Klasifikasi buah juga dilakukan dengan melakukan Perbandingan Algoritma SVM dan CNN. Pada penelitian ini menggunakan dataset dari Kaagle.com dan didalam dataset tersebut memiliki 17 label buah dengan jumlah total data yaitu 11.219. Didapatkan hasil dari pengujian yaitu model klasifikasi SVM mendapatkan hasil akurasi sebesar 93,09% dan model klasifikasi dengan CNN mendapatkan hasil akurasi sebesar 96,87% (Suyudi, Sudadio, & Suherman, 2022; Utomo, Azizah, & Pangestu, 2022).

Dari hasil penelitian yang sudah pernah dilakukan diatas dimana algoritma CNN lebih unggul tingkat akurasinya dibanding SVM, pada penelitian ini nantinya akan dilakukan penggabungan antara algoritma PCA dan CNN serta PCA dan SVM dengan menggunakan bantuan tools Matlab. Karena pada penelitian yang sudah pernah dilakukan semuanya menggunakan dataset dari Kaagle.com, maka Dataset yang digunakan pada penelitian ini akan menggunakan dataset dari situs Kaagle.com pula. Tujuan dari penelitian ini nantinya adalah untuk membandingkan seberapa besar pengaruh dari penggabungan antara PCA dengan algoritma CNN dan SVM, serta membandingkan lebih akurat mana penggabungan antara PCA dengan CNN atau PCA dengan SVM dalam kasus klasifikasi tingkat kematangan buah jeruk (Suroto & Friadi, 2024).

2. Tinjauan Pustaka dan Pengembangan Hipotesis

2.1 Convolutional Neural Network

Algoritma CNN merupakan pengembangan dari metode sebelumnya yaitu Multilayer Perceptron (MLP) yang didesain untuk tujuan mengolah data dua dimensi (Kurniadi, Prasetyo, Ahmad, Aditya Wibisono, & Sandya Prasvita, 2021). Algoritma ini bekerja dengan mengidentifikasi hal-hal yang sifatnya visual, misalnya saja mengenali objek pada gambar (Saputra, Mulyana, & Yel, 2022). Sistem ini terdiri dari tiga bagian, yakni Convolutional Layer, Pooling Layer, dan Fully-Connected (FC) layer dengan tugas masing-masing, mulai dari mengidentifikasi hal yang ringan sampai dengan kompleks.



Gambar 2. 1 Sistematika algoritma CNN

(<https://www.trivusi.web.id/2022/04/algoritma-cnn.html>)

2.1.1 Convolutional Layer

Convolutional Layer sendiri adalah bagian dari tahapan arsitektur CNN itu sendiri. Pada tahapan ini dilakukan operasi konvolusi output dari layer sebelumnya. Operasi Konvolusi sendiri adalah operasi yang dilakukan pada dua fungsi argumen bernilai nyata dan operasi ini mengadaptasi fungsi output sebagai feature map dari inputan citra (Diabetik, 2020). Input dan Output dinilai sebagai dua argumen bernilai riil. Pada konteks citra, operasi konvolusi bekerja dengan cara melakukan pergerakan pada kernel ke berbagai posisi sesuai dengan ukuran citra (Affifah et al., 2022).

2.1.2 Stride

Stride merupakan suatu parameter yang menentukan besaran dari pergeseran filter pada proses convolutional filter. Jika besaran stride yaitu 3x3 maka convolutional filter akan bergeser sebanyak 3 pixel secara vertikal maupun horizontal (Mianah, Diah Arie Widhining K, & Farrady Alif Fiolana, 2023). Semakin Kecil stride yang digunakan maka detail informasi yang didapatkan dari inputan semakin banyak, tapi memerlukan waktu yang cukup lama juga (Rifai, Muttaqin, & Irmayanti, 2024).

2.1.3 Padding

Padding adalah suatu parameter yang menentukan jumlah pixel (berisi nilai 0) dimana nantinya akan ditambahkan pada setiap sisi dari inputan. Tujuannya sendiri yaitu untuk memanipulasi dimensi output dari convolutional layer agar tetap sama dengan dimensi input (Yenusi, Suryasatriya Trihandaru, & Setiawan, 2023). Nantinya akan meningkatkan performa dari model convolutional layer dimana nantinya akan fokus pada informasi yang sebenarnya yang berada diantara zero padding. Untuk menghitung dimensi dari convolutional layer bisa menggunakan rumus sebagai berikut:

$$\text{Output} = \frac{W+N+2P}{S} + 1 \dots\dots\dots (8)$$

Keterangan :

W : Panjang/Tinggi Input
N : Panjang/Tinggi Filter
P : Zero Padding
S : Stride

2.1.4 Pooling Layer

Pooling layer biasanya berada setelah convolutional layer. Pooling layer sendiri terdiri dari sebuah filter dengan ukuran dan stride tertentu yang bergeser pada seluruh area feature map (Handoko, Timotius, & Utomo, 2022). Tujuan dari penggunaan pooling layer yaitu untuk mengurangi dimensi dari feature map (downsampling), sehingga mempercepat komputasi karena parameter yang harus di update semakin sedikit dan mengatasi overfitting (Nur Ridho, Mellyka, Saputra, & Padmo Azam Masa, 2024).

2.1.5 Relu Layer

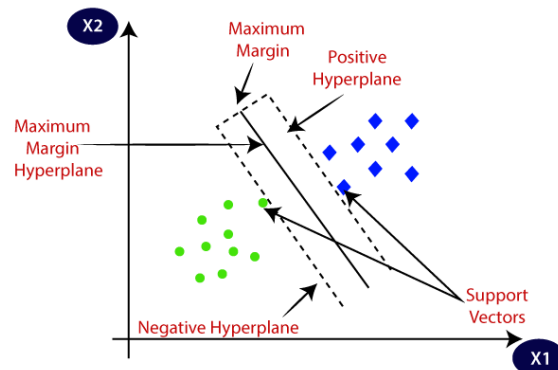
ReLU atau Rectified Activation Function merupakan non-saturating activation function yang dapat meningkatkan non-linearitas decision function dan network secara keseluruhan, tanpa harus mempengaruhi bidang-bidang reseptif pada convolution layer (Muhammad Yunus, 2020). ReLU sangat efektif untuk menghapus nilai negative pada feature/activation map dan dijadikan 0 (Riati, Yuhandri, & Nurcahyo, 2024).

Fully Connected

Fully-connected adalah lapisan dimana semua neuron aktivitas dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya (Pangestu, Basuki Rahmat, & Fetty Tri Anggraeny, 2020). Setiap aktivitas dari lapisan sebelumnya perlu diubah menjadi data satu dimensi sebelum dapat dihubungkan ke semua neuron di lapisan Fully-Connected (Alwanda, Ramadhan, & Alamsyah, 2020).

2.2 Support Vector Machine

Support Vector Machine (SVM) adalah algoritma *Machine Learning* yang diawasi dan dapat digunakan untuk klasifikasi dan regresi (Rizal, Girsang, & Prasetyo, 2019). Cara kerja SVM didasarkan pada SRM atau Structural Risk Minimization yang dirancang untuk mengolah data menjadi Hyperplane yang mengklasifikasikan ruang input menjadi dua kelas (Styawati, Hendrastuty, & Isnain, 2021). Cara kerja algoritma SVM diawali dengan pengelompokan kasus-kasus linier yang dapat dipisahkan dengan hyperplane dan dibagi menurut kelasnya.



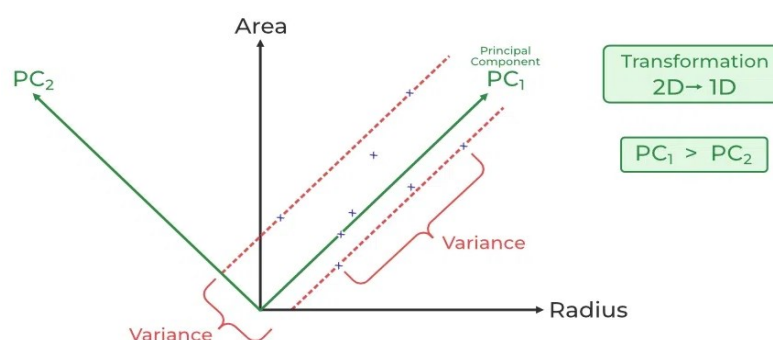
Gambar 2. 2 Model SVM

(<https://insancs.medium.com/support-vector-machine-svm-implementation-using-python-4442e9a5babc>)

Pada gambar 2.2 merupakan model sederhana dari algoritma SVM, dimana model ini terdapat dua pola yang berbeda dan dua pola tersebut dipisahkan oleh SVM. Dimana kedua pola tersebut dipisahkan sesuai dengan nilai datanya, jika datanya bernilai negative maka akan berada pada negative hyperlane dan jika datanya bernilai positif maka akan berada pada positive hyperlane hal tersebut bisa terjadi secara sempurna oleh hyperlane berdimensi d (Nugroho, Witarto, & Handoko, 2003).

2.3 Principal Component Analysis

Principal component analysis (PCA) adalah suatu teknik statistik multivariat yang secara linear mengubah bentuk sekumpulan variabel asli menjadi kumpulan variabel yang lebih kecil (Nasution, Nababan, Syaliman, Novelan, & Jannah, 2019). PCA digunakan untuk mengurangi dimensi kumpulan data dengan menemukan kumpulan variabel baru yang lebih kecil dari kumpulan variabel awal dan tetap mempertahankan sebagian besar informasi sampel yang nantinya berguna untuk regresi dan klasifikasi data (Borman & Priyopradono, 2018).



Gambar 2. 3 Principal Component Analysis

(<https://www.geeksforgeeks.org>)

PCA menganalisis table data yang mewakili pengamatan dijelaskan oleh beberapa variabel dependen, yang secara umum saling berkorelasi. Tujuannya adalah untuk mengekstrak informasi penting dari data

tabel dan untuk mengekspresikan informasi ini sebagai satu set baru variabel ortogonal yang disebut komponen utama. Komponen utamanya adalah kombinasi linier dari variabel asli dalam kumpulan data dan diurutkan berdasarkan tingkat kepentingannya. Total varians yang ditangkap oleh semua komponen utama sama dengan total varians pada dataset asli. Komponen utama (Principal Component) pertama menangkap variasi terbanyak dalam data, namun komponen utama kedua menangkap varian maksimum yang ortogonal terhadap komponen utama pertama, dan seterusnya seperti yang bisa dilihat pada gambar 2.2.

2.4 Confusion Matrix

Confusion matrix adalah alat yang digunakan untuk menggambarkan kinerja model klasifikasi pada data uji yang sudah diketahui hasil sebenarnya. Tujuan utama dari confusion matrix yaitu untuk memvisualisasikan dan menganalisis hasil prediksi yang dibuat oleh model, sehingga dapat lebih mudah memahami kelebihan dan kekurangan model dalam mengklasifikasikan data (Rina, 2023). Pada gambar 2.15 terdapat empat 4 istilah untuk merepresentasikan hasil proses klasifikasi pada confusion matrix. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN) (Suryadewiansyah, Endra, & Tju, 2020). Dimana TP merupakan data positif yang diprediksi benar, TN merupakan data negatif yang diprediksi benar, FP merupakan kondisi dimana data negatif atau data yang salah diprediksi positif atau benar, dan FN merupakan kondisi dimana data tersebut benar namun diprediksi salah (Nugroho, 2019).

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2. 1 Confusion Matrix Table

(<https://esairina.medium.com/memahami-confusion-matrix-accuracy-precision-recall-specificity-dan-f1-score-610d4f0db7cf>)

2.4.1 Akurasi

Akurasi adalah metrik evaluasi yang mengukur seberapa baik model membuat prediksi yang benar dari total prediksi yang dilakukan (Pratiwi, Handayani, & Sarjana, 2021). Dalam konteks klasifikasi, akurasi memberikan gambaran mengenai seberapa sering model memprediksi kelas yang benar, baik itu kelas positif maupun negatif (Dabbo, Yusuf Bisilisin, & Abstrak, 2024). Untuk menghitung nilai akurasi dapat menggunakan rumus matematika berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2.4.2 Presisi

Presisi adalah tingkat keakuratan antara informasi yang diminta oleh user dan respon yang diberikan oleh sistem (Hendriyana & Yazid Hilman Maulana, 2020). Dan rumus yang dipakai untuk menghitung presisi sebagai berikut :

$$Precision = \frac{TP}{TP + FP}$$

2.4.3 Recall

Recall adalah tingkat kesuksesan sebuah sistem dalam menemukan kembali sebuah informasi (Hendriyana & Yazid Hilman Maulana, 2020). Rumus yang dipakai untuk menghitung recall sebagai berikut:

$$Recall = \frac{TP}{TP + FP}$$

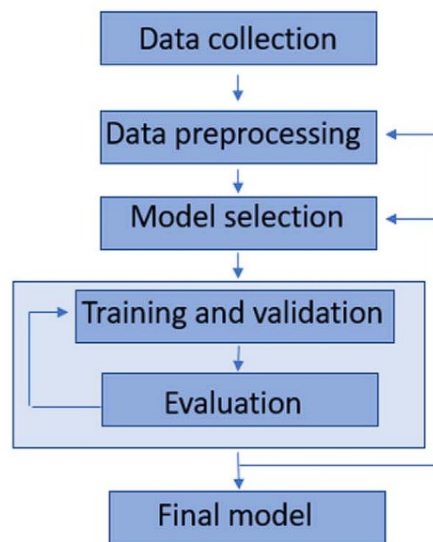
2.4.4 F1-Score

F1-Score adalah rata – rata dari nilai Presisi dan recall yang didapatkan (Taslim, Gunadi, & Tjondrowiguno, 2019). Untuk rumus yang dipakai sebagai berikut:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3. Metodologi Penelitian

Pada penelitian ini menggunakan metode Machine Learning dengan kategori *Supervised Learning*. Alur rancangan machine learning bisa dilihat pada gambar 3.1



Gambar 3. 1 Workflow Machine Learning

(<https://fastdatascience.com/data-science-project-management/workflows-pipelines-ml-ai/>)

3.1 Data Collection atau Pengumpulan data

Penelitian ini menggunakan data yang sudah ada sebelumnya yang didapat dari situs Kaagle.com. Data yang didapatkan memiliki label kematangan jeruk yang mendukung untuk penelitian ini. Kematangan jeruk yang tercakup pada penelitian ini yaitu Matang, Mentah, dan Busuk. Dataset yang diambil berasal dari Fresh and Stale Images of Fruits and Vegetables (<https://www.kaggle.com/datasets/raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables>) , dan Orange disease dataset (<https://www.kaggle.com/datasets/jonathansilva2020/orange-diseases-dataset>).

Dalam Dataset Fresh and Stale Images of Fruits and Vegetables terdapat 6 citra buah yaitu appel, banana, bitter melon, capsicum, orange, dan tomato. Dataset tersebut sudah terbagi menjadi dua kondisi buah yaitu segar dan busuk, jumlah citra dari dataset keseluruhan yaitu 14.700 citra. Karena dalam penelitian ini hanya menggunakan jeruk sebagai objeknya maka hanya data buah jeruk yang diambil, dan jumlah citra buah jeruk dari dataset Fresh and Stale Images of Fruits and Vegetables yaitu sebanyak 3.061 citra yang terbagi dari 1.466 buah segar dan 1.595 buah busuk.

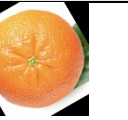
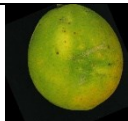
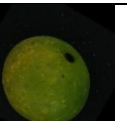
Tabel 3. 1 Contoh Dataset Fresh and Stale Images of Fruits and Vegetables

Kematangan	Contoh Citra			
Matang				
Busuk				

Sumber: (Penulis, 2025)

Orange disease dataset merupakan dataset yang berisikan beberapa jenis penyakit buah jeruk, jadi terbagi menjadi empat kondisi yaitu blackspot, canker, fresh dan greening. Pada dataset ini sudah terdapat label test dan train. Jumlah citra dari dataset ini yaitu 1.789 citra, dan pada label test terdapat sebanyak 624 citra dan pada label train terdapat 1.165 citra. Karena pada penelitian ini hanya memakai kondisi jeruk yang matang, mentah, dan busuk, maka dalam dataset ini hanya diambil kondisi fresh dan greening, dimana greening sendiri sebagai sample untuk citra mentah. Jadi diambil citra buah jeruk fresh sebanyak 552 citra dan greening sebanyak 545 citra.

Tabel 3. 2 Contoh Dataset Orange disease

Kematangan	Contoh Citra			
Matang				
Mentah				

Sumber: (Penulis, 2025)

3.2 Preprocessing Data

Pada preprocessing data ini dataset yang sudah di dapatkan tadi akan dilakukan resize dan pembagian data:

3.2.1 Resize Data

Ukuran pixel yang akan ditetapkan nantinya pada penelitian ini yaitu 100x100 piksel, serta akan dilakukan pengubahan format data dari jpg ke png pada dataset Orange disease pada data buah jeruk mentah, dilakukannya perubahan format data ke png karena keseluruhan data yang ada pada dataset Fresh and Stale Images of Fruits and Vegetables maupun dataset Orange disease menggunakan format png kecuali data buah jeruk mentah.

3.2.2 Pembagian data

Pada penelitian ini dataset Fresh and Stale Images of Fruits and Vegetables (<https://www.kaggle.com/datasets/raghavpotdar/fresh-and-stale-images-of-fruits-and-vegetables>) , dan Orange disease dataset (<https://www.kaggle.com/datasets/jonathansilva2020/orange-diseases-dataset>) dibagi menjadi dua model data dengan masing – masing pembagiannya yaitu 80% data training dan 20% data testing. Dimana data training untuk melatih model, dan data testing digunakan untuk menguji data yang sudah dilatih. Jumlah data awal bisa dilihat pada tabel 3.3

Tabel 3. 3 Total dataset awal yang diperoleh

Dataset	Kematangan			Total
	Mentah	Matang	Busuk	
Fresh and Stale Images of Fruits and Vegetables	-	1466	1595	3061
Orange disease	545	552	-	1097
Total	545	2018	1595	4158

Sumber: (Penulis, 2025)

Dan pada tabel 3.4 dataset setiap kematangan di buat sama yaitu 500

Tabel 3. 4 Dataset setelah dilakukan perubahan

Dataset	Kematangan			Total
	Mentah	Matang	Busuk	
Fresh and Stale Images of Fruits and Vegetables	-	400	500	900
Orange disease	500	100	-	600
Total	500	500	500	1500

Sumber : (Penulis, 2025)

Tabel 3. 5 Pembagian model dataset

Model Data	Dataset
Data Training	1200
Data Testing	300
Total	1500

Sumber : (Penulis, 2025)

3.2.3 Konversi warna RGB ke Grayscale

Grayscale adalah proses pengolahan citra untuk mengubah citra berwarna yang mempunyai nilai RGB diubah menjadi citra grayscale (keabuan). Format citra grayscale disebut skala keabuan karena pada umumnya warna yang dipakai adalah warna hitam sebagai warna minimal dan warna putih sebagai warna maksimalnya, sehingga warna antaranya adalah abu-abu (Muwardi & Fadlil, 2018).



Gambar 3. 2 Contoh konversi citra dataset rgb ke grayscale

Sumber : (Penulis, 2025)

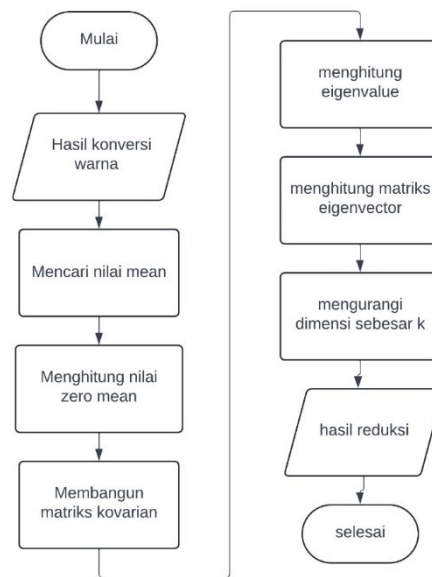
3.3 Model Selection

Pada Penelitian ini nantinya akan menggunakan dimension reduction dan klasifikasi model. Pemilihan Dimension reduction atau mereduksi dimensi sendiri bertujuan untuk mengurangi dimensi dataset yang ada dan nantinya akan digunakan pada saat klasifikasi. Algoritma yang dipakai untuk mereduksi data pada penelitian ini yaitu algoritma PCA. Tujuan klasifikasi pada penelitian ini sendiri yaitu untuk menentukan tingkat kematangan buah jeruk berdasarkan warna kulit buah. Algoritma yang dipakai dalam proses klasifikasi sendiri meliputi dua algoritma yaitu CNN dan SVM.

3.4 Training and Validation

3.4.1 Reduksi Data menggunakan PCA

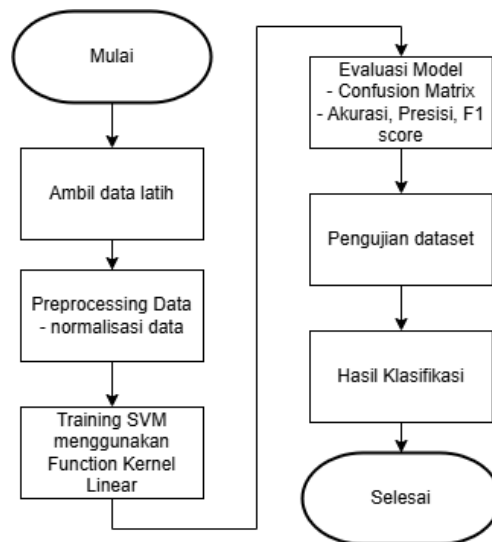
Metode PCA akan membentuk sekumpulan dimensi baru yang kemudian di ranking berdasarkan varian datanya. PCA akan menghasilkan Principal component yang didapat dari dekomposisi eigen value dan eigen vektor dari matriks kovariansi.



Gambar 3. 3 Flowchart alur proses PCA
Sumber : (Penulis, 2025)

3.4.2 Klasifikasi pada Algoritma SVM

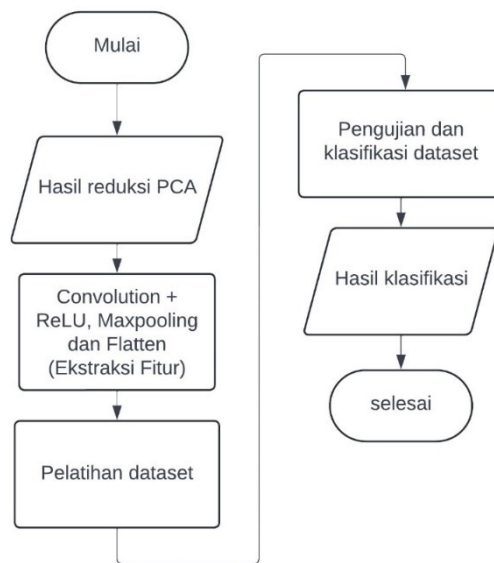
Setelah dilakukannya pengelompokan data serta reduksi data maka selanjutnya akan dilakukan klasifikasi pada algoritma SVM, untuk alur program bisa dilihat pada gambar 3.4.



Gambar 3. 4 Flowchart Alur Algoritma SVM
Sumber : (Penulis, 2025)

3.4.3 Klasifikasi pada algoritma CNN

Setelah didapatkan hasil reduksi data dari algoritma PCA selanjutnya dilakukan klasifikasi CNN, alur pengujian dataset menggunakan algoritma CNN bisa dilihat pada gambar 3.4 berikut:



Gambar 3. 5 Flowchart proses klasifikasi pada algoritma CNN
Sumber : (Penulis, 2025)

3.5 Evaluation

Setelah proses proses klasifikasi selesai nantinya akan dilakukan pengujian model dari data yang dihasilkan sebelumnya, pengujian ini nantinya menggunakan Confusion matrix akurasi sebagai acuan perbandingan. Dimana didalamnya terdapat nilai True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Dimana TP merupakan data positif yang diprediksi benar, TN merupakan data negatif yang diprediksi benar, FP merupakan kondisi dimana data negatif atau data yang salah diprediksi positif atau benar, dan FN merupakan kondisi dimana data tersebut benar nammun diprediksi salah (Nugroho, 2019). Selain Confusion matrix ada juga bahan evaluasi lain seperti Akurasi, Presisi, Recall, dan F1-Score untuk rumusnya sebagai berikut:

1. Akurasi

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Presisi

$$Precision = \frac{TP}{TP + FP}$$

3. Recall

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score

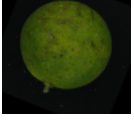
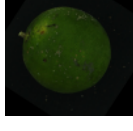
$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4. Hasil dan Pembahasan

4.1 Data Collection

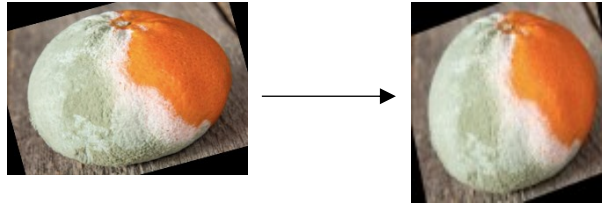
Karena terdapat dua dataset yang berbeda yaitu dataset Orange Disease dan dataset Fresh and Stale Images of Fruits and Vegetables oleh karena itu dilakukan penggabungan dataset menjadi satu kemudian jumlah dari citra dari masing – masing kondisi berjumlah 500 dengan total keseluruhan dataset yang dipakai yaitu 1500 citra dan didapatkan hasil dari penggabungan seperti pada tabel 4.1 berikut

Tabel 4. 1 Contoh dataset yang sudah digabung

Kondisi	Contoh dataset				
Busuk					
Matang					
Mentah					

4.2 Preprocessing Data

Karena pada dataset yang diambil memiliki resolusi dan jenis citra yang berbeda – beda maka akan dilakukan penyesuaian citra agar semua citra sama yang bertujuan agar pada saat pemrosesannya lebih cepat. Pada penelitian ini, dilakukan cropping dengan ukuran 100x100 pixels menggunakan bantuan program python untuk proses croppingnya.



Gambar 4. 1 Contoh perubahan citra
Sumber : (Penulis, 2025)

4.3 Model Selection, Training and Validation

4.3.1 Pembuatan Model CNN dan PCA

Pada tahap awal yang dilakukan sebelum memproses dataset yaitu mengubahnya menjadi grayscale terlebih dahulu, hal ini digunakan supaya pada saat proses klasifikasi nanti dapat lebih efisien.

```
imds.ReadFcn = @(loc) imresize(rgb2gray(imread(loc)), imageSize);
```

Gambar 4. 1 kode untuk mengubah gambar rgb ke grayscale
Sumber : (Penulis, 2025)



Gambar 4. 2Perubahan citra
Sumber : (Penulis, 2025)

Tahap selanjutnya yaitu pengaplikasian PCA untuk mereduksi dataset supaya pada saat proses klasifikasi dapat mengefisiensi waktu.

```
numComponents = 50;

[trainFeatures, trainLabels] = applyPCA(imdsTrain, imageSize, numComponents);
[testFeatures, testLabels] = applyPCA(imdsTest, imageSize, numComponents);
```

Gambar 4. 3Kode untuk menerapkan PCA
Sumber : (Penulis, 2025)

Pada gambar 4.6 adalah kode untuk menerapkan PCA, dimana numComponents = 50; adalah jumlah komponen yang ingin dipertahankan sebesar 50% dari jumlah komponen aslinya.

```

layers = [
    imageInputLayer([1 1 numComponents], 'Normalization', 'none')

    convolution2dLayer(1, 32, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(1, 'Stride', 1)

    convolution2dLayer(1, 64, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(1, 'Stride', 1)

    convolution2dLayer(1, 128, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(1, 'Stride', 1)

    fullyConnectedLayer(numel(categories))
    softmaxLayer
    classificationLayer];

```

Gambar 4. 4Kode untuk membangun arsitektur CNN
Sumber : (Penulis, 2025)

Pada pembuatan model CNN didalamnua terdapat jaringan yang melibatkan beberapa lapisan konvolusi dengan ukuran filter 1x1 yang digunakan untuk mengekstraksi fitur dari data, diikuti oleh lapisan batch normalization untuk menormalkan output dan lapisan ReLU untuk menambahkan non-linearitas. Jaringan ini mengulangi proses konvolusi, normalisasi, ReLU, dan pooling beberapa kali dengan jumlah filter yang meningkat (32, 64, dan 128) untuk menangkap fitur yang semakin kompleks.

Di akhir arsitektur, terdapat lapisan fully connected yang menghubungkan semua neuron sebelumnya ke neuron pada lapisan ini, jumlah neuron pada lapisan ini sesuai dengan jumlah kategori kelas yang ada. Lapisan softmax kemudian mengubah skor menjadi probabilitas untuk setiap kelas, dan lapisan classification layer terakhir menghitung loss untuk mengoptimalkan model selama pelatihan.

```

% Set training
options = trainingOptions('adam', ...
    'InitialLearnRate', 1e-4, ...
    'MaxEpochs', 20, ...
    'MiniBatchSize', 32, ...
    'ValidationData', {testFeatures, testLabels}, ...
    'ValidationFrequency', 30, ...
    'Verbose', true, ...
    'Plots', 'training-progress', ...
    'OutputFcn', @(info) writeProgressToLog(info, logFilePath));

```

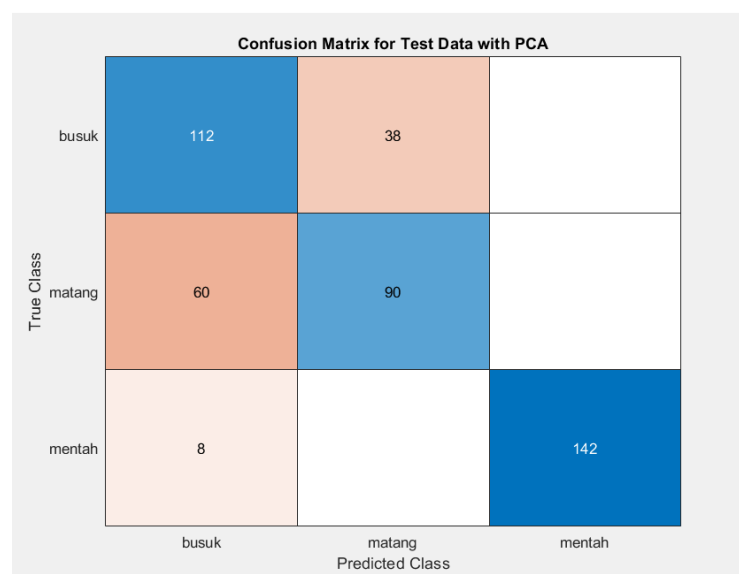
Gambar 4. 5kode untuk training dataset
Sumber : (Penulis, 2025)

Training data dilakukan selama 20 epoch, di mana setiap epoch melibatkan satu kali pengulangan melalui seluruh set data pelatihan. Selain itu, data pelatihan dipecah menjadi mini-batch berukuran 32 sampel per batch, yang memungkinkan pembaruan bobot lebih sering tanpa harus menunggu seluruh dataset selesai diproses.

Tabel 4. 2 Hasil training Adam

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss
1	1	00:00:02	18.75	24.67	1.6038	1.6049
1	30	00:00:02	56.25	50.89	0.9866	1.1506
2	50	00:00:03	65.62	62.0	0.8403	0.9027
2	60	00:00:03	65.62	62.0	0.767	0.9027
3	90	00:00:04	68.75	65.78	0.6875	0.7816
3	100	00:00:04	65.62	65.62	0.672	0.7313
4	120	00:00:05	78.12	68.22	0.5541	0.7136
4	150	00:00:05	81.25	69.33	0.5251	0.6713
5	180	00:00:06	75.0	70.89	0.527	0.6424
7	200	00:00:07	68.75	72.22	0.5143	0.6225
7	210	00:00:07	78.12	72.22	0.4491	0.6077
8	240	00:00:08	96.88	72.44	0.3796	0.5962
9	250	00:00:08	84.38	72.22	0.5437	0.5868
9	270	00:00:09	71.88	72.44	0.45	0.5962
10	300	00:00:09	78.12	72.22	0.3999	0.5873
11	330	00:00:10	84.38	72.44	0.4919	0.5804
11	350	00:00:10	87.5	72.0	0.3368	0.5734
12	360	00:00:11	81.25	72.0	0.389	0.5753
13	390	00:00:11	87.5	71.56	0.3368	0.5715
14	420	00:00:12	87.5	71.33	0.315	0.5681
15	450	00:00:13	78.12	71.33	0.3615	0.5681
15	480	00:00:14	90.62	71.11	0.4441	0.5634
16	510	00:00:14	87.5	71.11	0.3564	0.5634
17	540	00:00:15	87.5	71.11	0.338	0.5618
18	550	00:00:15	96.88	71.33	0.2305	0.5604
19	600	00:00:17	93.75	71.56	0.2869	0.5579
20	630	00:00:17	96.88	71.56	0.2407	0.5571
20	640	00:00:17	90.62	71.33	0.3172	0.5573

Sumber : (Penulis,2025)



Gambar 4. 6Confusion Matrix CNN dan PCA

Sumber : (Penulis, 2025)

Dari hasil confusion matrix tersebut selanjutnya dapat digunakan untuk mengukur kinerja model seperti akurasi, recall, f1-score, dan precision. Hasilnya seperti gambar berikut

```
Test Accuracy: 0.76444
Classification Report:
-----
Class      | Precision | Recall | F1 Score
-----
busuk      | 0.6222   | 0.7467 | 0.6788
matang     | 0.7031   | 0.6000 | 0.6475
mentah     | 1.0000   | 0.9467 | 0.9726
-----
Macro Average:
Macro Avg  | 0.7751   | 0.7644 | 0.7663
```

Gambar 4. 7Hasil Klasifikasi
Sumber : (Penulis, 2025)

Seperti yang terlihat pada Gambar 4.10 dapat dilihat bahwa nilai rata – rata *accuracy* sebesar 76,4%, *Precision* 77,51%, *Recall* 76,44%, *F1 Score* 76,63%. Dilihat dari hasil klasifikasi diatas bahwa kombinasi antara PCA dengan CNN dengan dataset yang diubah ke grayscale tingkat akurasi belum cukup tinggi dalam menentukan tingkat kematangan buah jeruk keprok.

4.3.2 Pembuatan model PCA dan SVM

Setelah ekstraksi feature dilakukan selanjutnya dilakukan pengaplikasian PCA dengan reduksi data 50%, hal ini bertujuan supaya pada saat klasifikasinya tidak memerlukan waktu lama dan tidak terlalu banyak data yang hilang dalam dataset tersebut. Berikut codenya :

```
numComponents = round(size(featuresTrain, 2) / 2);
[coeff, scoreTrain] = pca(featuresTrain, 'NumComponents', numComponents);
scoreTest = featuresTest * coeff(:, 1:numComponents); |
```

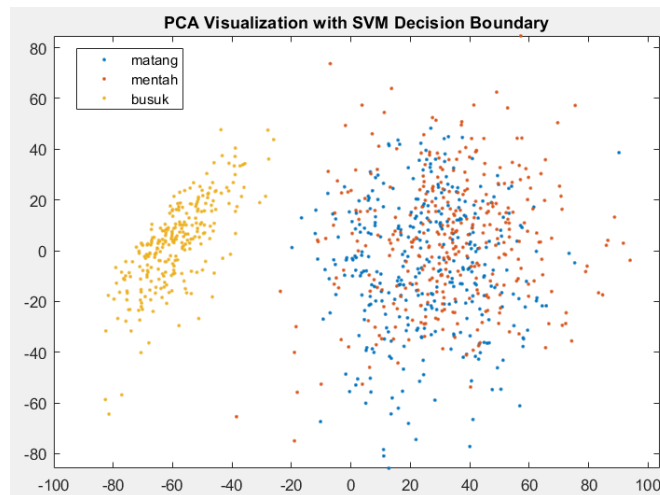
Gambar 4. 8Code PCA
Sumber : (Penulis, 2025)

Setelah proses semua selesai selanjutnya dilakukan klasifikasi menggunakan SVM. Berikut codenya

```
% Train the SVM classifier
classifier = fitcecoc(featuresTrain, labelsTrain);

% Predict the labels of the test set
predictedLabels = predict(classifier, featuresTest);
```

Gambar 4. 9Code klasifikasi SVM
Sumber : (Penulis, 2025)



Gambar 4. 10Decision Boundary
Sumber : (Penulis, 2025)

Dari Decision Boundry diatas bisa dilihat bahwa kategori dari dataset jeruk busuk paling terlihat perbedaannya dibanding matang dan mentah, maka bisa dipastikan akurasi untuk kategori busuk sangat bisa dibedakan oleh svm berbeda dengan matang dan mentah hampir menyatu di satu area yang sama. Untuk hasil klasifikasi bisa dilihat diconfusion matrix pada gambar 4.17 berikut.

Confusion Matrix				
True Class	busuk	matang	mentah	
	busuk	150		
	matang		149	1
	mentah	1	3	146
	busuk	matang	mentah	

Gambar 4. 11Hasil Confusion Matrix SVM dan PCA
Sumber : (Penulis, 2025)

Sesuai dari decision boundry diatas bahwasannya kategori busuk memiliki tingkat akurasi yang sangat tinggi dengan akurasi 100%, untuk kategori matang dan busuk masih tergolong cukup tinggi juga dengan prediksi matang yang benar sebanyak 149 dan hanya 1 salah prediksi, begitu juga dengan mentah jumlah data yang benar sejumlah 146, dan 4 prediksi yang salah.

```

Test Result:
Accuracy: 98.89%

Results:
Category    Precision    Recall    F1 Score    Support
matang      98.03%      99.33%    98.68%      150
mentah      96.05%      97.33%    96.69%      150
busuk       98.68%      100.00%   99.34%      150

Mean Results:
Precision: 97.59%
Precision: 100.91%
Precision: 98.23%
Recall: 98.89%
F1 Score: 98.23%
F1 Score: 99.89%
F1 Score: 98.56%

```

Gambar 4. 12 Nilai dari klasifikasi SVM dan PCA

Hasil dari klasifikasi SVM dan PCA didapat bahwa tingkat akurasi dari penggabungan dua algoritma tersebut dengan pembagian data training dan testing sebesar 70% dan 30% memiliki tingkat akurasi sebesar 98.89%, *Precision* 97.58%, *Recall* 98.88%, dan *F1-Score* 98.23%. Dari hasil tersebut bisa disimpulkan bahwa penggabungan kedua algoritma ini memiliki tingkat akurasi yang sangat tinggi untuk mengklasifikasi tingkat kematangan buah jeruk keprok.

4.4 Evaluation

Setelah dilakukan pengujian didapat sebuah data akurasi sebagai berikut

Tabel 4. 3 Tabel Perbandingan Hasil

	Split Data	Akurasi
CNN + PCA	70 : 30	76,4 %
SVM + PCA	70 : 30	98,89 %

Sumber ; (Penulis, 2025)

Dari data diatas didapat kesimpulan bahwa untuk kombinasi algoritma yang mendapatkan akurasi yang tinggi yaitu SVM + PCA dengan tingkat akurasi sebesar 98,89% dengan split data antara data training dan data testing sebesar 70:30. Dan hasil dari klasifikasi menggunakan CNN + PCA mendapatkan akurasi sebesar 76,4%. Bisa disimpulkan bahwasannya pada penelitian ini kombinasi antara SVM dan PCA mendapatkan tingkat akurasi yang lebih besar dibanding CNN dan PCA yang memiliki selisih akurasi sebesar 23,16%.

5. Kesimpulan Dan Saran

5.1 Kesimpulan

Penerapan gabungan algoritma antara SVM dan PCA kemudian CNN dan PCA sebagai tolak ukur perbandingan akurasi dalam melakukan proses klasifikasi sukses dilakukan menggunakan Software MATLAB, hal ini bisa disimpulkan bahwa software MATLAB masih bisa dipakai untuk alat pembuatan model klasifikasi berupa algoritma Artificial Intelegent (AI). Dengan melakukan perbandingan algoritma Support Vector Machine (SVM) dan Convolutional Neural Network dengan bantuan Principal Component Analysis sebagai pereduksi datanya didapatkan hasil akurasi sebesar 76,4% untuk kombinasi CNN dan PCA kemudian pada model klasifikasi SVM dan PCA mendapatkan hasil akurasi sebesar 98,89%. Dari hasil pengujian model tersebut dapat disimpulkan bahwa

penggabungan algoritma PCA yang memiliki akurasi yang tinggi yaitu SVM dengan hasil akurasi sebesar 98,9%.

5.2 Saran

Secara keseluruhan model algoritma yang dibangun sudah berjalan dengan baik untuk studi kasus yang sedang dibahas mungkin ada beberapa saran untuk kedepannya untuk bisa dikembangkan lagi:

1. Menggunakan variasi dataset lainnya sesuai studi kasus yang ingin dibahas

Referensi

- Affifah, D. D., Permanasari, Y., Matematika, R. P., Matematika, F., Ilmu, D., & Alam, P. (2022). Bandung Conference Series: Mathematics Teknik Konvolusi pada Deep Learning untuk Image Processing. *Bandung Conference Series: Mathematics*, 2(2), 103–112. Retrieved from <https://doi.org/10.29313/bcsm.v2i2.4527>
- Alwanda, M. R., Ramadhan, R. P. K., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *Jurnal Algoritme*, 1(1), 45–56. <https://doi.org/10.35957/algoritme.v1i1.434>
- Borman, R. I., & Priyopradono, B. (2018). Implementasi Penerjemah Bahasa Isyarat Pada Bahasa Isyarat Indonesia (BISINDO) Dengan Metode Principal Component Analysis (PCA). *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1), 103–108. <https://doi.org/10.30591/jpit.v3i1.631>
- Dabbo, P., Yusuf Bisilisin, F., & Abstrak, I. A. (2024). Klasifikasi Motif Kain Tenun Sabu Rajjua Menggunakan Convolutional Neural Network (CNN) Berbasis Citra. *KETIK : Jurnal Informatika*, 1(06), 11–18. Retrieved from <https://jurnal.faatuatua.com/index.php/KETIK/article/view/88>
- Diabetik, D. R. (2020). Implementasi Convolutional Neural Network (CNN) Untuk Dereksi Retinopati Diabetik, 1(1), 669–678.
- Handoko, A. B., Timotius, I. K., & Utomo, D. (2022). Klasifikasi Citra X-Ray Covid-19 Menggunakan Three-layered CNN Model. *Techné : Jurnal Ilmiah Elektroteknika*, 21(2), 155–168. <https://doi.org/10.31358/techne.v21i2.316>
- Hendriyana, H., & Yazid Hilman Maulana. (2020). Identification of Types of Wood using Convolutional Neural Network with Mobilenet Architecture. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(1), 70–76. <https://doi.org/10.29207/resti.v4i1.1445>
- Kurniadi, B. W., Prasetyo, H., Ahmad, G. L., Aditya Wibisono, B., & Sandya Prasvita, D. (2021). Analisis Perbandingan Algoritma SVM dan CNN untuk Klasifikasi Buah. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*, (September), 1–11.
- Limanseto, H. (2022). Terus Dorong Peningkatan Konsumsi Buah Nusantara, Pemerintah Gelar Kembali Gelar Buah Nusantara (GBN) ke-7 Tahun 2022 - Kementerian Koordinator Bidang Perekonomian Republik Indonesia. *Kementerian Koordinator Bidang Perekonomian Republik Indonesia*, 7–8. Retrieved from <https://www.ekon.go.id/publikasi/detail/4450/terus-dorong-peningkatan-konsumsi-buah-nusantara-pemerintah-gelar-kembali-gelar-buah-nusantara-gbn-ke-7-tahun-2022>
- Mianah, A. N., Diah Arie Widhining K, & Farrady Alif Fiolana. (2023). Klasifikasi Helm Keselamatan Menggunakan Metode Convolutional Neural Network (CNN). *Journal Zetroem*, 5(2), 94–102. <https://doi.org/10.36526/ztr.v5i2.2765>
- Muwardi, F., & Fadlil, A. (2018). Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, 3(2), 124. <https://doi.org/10.26555/jiteki.v3i2.7470>
- Nisya, I. S., Wulansari, O. D. E., & Wartariyus, W. (2023). Rancang Bangun Game Edukasi Bencana Alam Menggunakan Metode MDLC. doi:<https://doi.org/10.35912/jisted.v2i1.2374>
- Nasution, M. Z., Nababan, A. A., Syaliman, K. U., Novelan, M. S., & Jannah, M. (2019). Penerapan Principal Component Analysis (PCA) dalam Penentuan Faktor Doinan yang Mempengaruhi Pengidap Kanker Serviks. *Jurnal Mantik Penusa*, 3(1), 204–210.
- Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support Vector Machine –Teori dan Aplikasinya dalam Bioinformatika. *Proceeding of Indonesian Scientific Meeting in Central Japan, 2003.*, 842–847. <https://doi.org/10.1109/CCDC.2011.5968300>
- Nur Ridho, A., Mellyka, G., Saputra, F., & Padmo Azam Masa, A. (2024). Implementasi Metode

- Convolutional Neural Network (CNN) Untuk Klasifikasi Gambar Mobil dan Motor Menggunakan Keras. *Jurnal Jambo Digitech*, 1(1), 54–64. Retrieved from <https://jurnal.ildikti13.id/index.php/jjd/article/view/24>
- Pangestu, A. R., Basuki Rahmat, & Fetty Tri Anggraeny. (2020). Implementasi Algoritma Cnn Untuk Klasifikasi Citra Lahan Dan Perhitungan Luas. *Jurnal Informatika Dan Sistem Informasi (JIFoSI)*, 1(1), 166–174.
- Pratiwi, B. P., Handayani, A. S., & Sarjana, S. (2021). Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi Wsn Menggunakan Confusion Matrix. *Jurnal Informatika Upgris*, 6(2), 66–75. <https://doi.org/10.26877/jiu.v6i2.6552>
- Riati, I., Yuhandri, & Nurcahyo, G. W. (2024). Penerapan Convolutional Neural Network Untuk Mengidentifikasi Penyakit Tanaman Kelapa Sawit. *Jurnal KomtekInfo*, 11(4), 237–246. <https://doi.org/10.35134/komtekinfo.v11i4.554>
- Rifai, A. K., Muttaqin, M. R., & Irmayanti, D. (2024). Pemanfaatan Algoritma Convolutional Neural Network Dengan Untuk Mendeteksi Penyakit Pada Tumbuhan Jagung. *Sistematis : Jurnal Ilmiah Sistem Informasi*, 1(1), 18–26. Retrieved from <https://ejournal.rizaniamedia.com/index.php/sistematis>
- Rizal, R. A., Girsang, I. S., & Prasetyo, S. A. (2019). Klasifikasi Wajah Menggunakan Support Vector Machine (SVM). *REMIK (Riset Dan E-Jurnal Manajemen Informatika Komputer)*, 3(2), 1. <https://doi.org/10.33395/remik.v3i2.10080>
- Saputra, O., Mulyana, D. I., & Yel, M. B. (2022). Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Senjata Tradisional Di Jawa Tengah Dengan Metode Transfer Learning. *Jurnal SISKOM-KB (Sistem Komputer Dan Kecerdasan Buatan)*, 5(2), 45–52. <https://doi.org/10.47970/siskom-kb.v5i2.282>
- Siskandar, R., Indrawan, N. A., Kusumah, B. R., Santosa, S. H., Irmansyah, I., & Irzaman, I. (2020). Penerapan Rekayasa Mesin Sortir Sebagai Penentu Kematangan Buah Jeruk Dan Tomat Merah Berbasis Image Processing. *Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering)*, 9(3), 222. <https://doi.org/10.23960/jtep-l.v9i3.222-236>
- Styawati, S., Hendrastuty, N., & Isnain, A. R. (2021). Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine. *Jurnal Informatika: Jurnal Pengembangan IT*, 6(3), 150–155. <https://doi.org/10.30591/jpit.v6i3.2870>
- Suryadewiansyah, M. K., Endra, T., & Tju, E. (2020). Naïve Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 8(2), 81–88. <https://doi.org/10.25077/TEKNOSI.v8i2.2022.81-88>
- Suroto, S., & Friadi, J. (2023). Pengukuran Tingkat Capability IT Governance pada PT. Sarana Citranusa Kabil Menggunakan Framework Cobit 2019. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(2), 81–90. doi:[10.35912/jisted.v1i1.1915](https://doi.org/10.35912/jisted.v1i1.1915)
- Suroto, S., & Friadi, J. (2024). Manajemen Risiko Teknologi Informasi Pada Aplikasi CMS di PT. Sarana Citranusa Kabil - Batam Menggunakan ISO31000:2018. *Jurnal Ilmu Siber dan Teknologi Digital*, 2(1), 61–73. doi:[10.35912/jisted.v2i1.2351](https://doi.org/10.35912/jisted.v2i1.2351)
- Suyudi, I., Sudadio, S., & Suherman, S. (2022). Pengenalan Bahasa Isyarat Indonesia menggunakan Mediapipe dengan Model Random Forest dan Multinomial Logistic Regression. doi:<https://doi.org/10.35912/jisted.v1i1.1899>
- Taslim, M. M., Gunadi, K., & Tjondrowiguno, A. N. (2019). Deteksi Rumus Matematika pada Halaman Dokumen Digital dengan Metode Convolutional Neural Network. *Jurnal Infra*, 7(2), 123–129.
- Utomo, K. B., Azizah, A., & Pangestu, M. A. (2022). Peran Computer Assited Test dalam Implementasi Penilaian di SD Negeri 005 Palaran. doi:<https://doi.org/10.35912/jisted.v1i1.1529>
- Yenusi, Y. N., Suryasatriya Trihandaru, & Setiawan, A. (2023). Comparison of Convolutional Neural Network (CNN) Models in Face Classification of Papuan and Other Ethnicities. *JST (Jurnal Sains Dan Teknologi)*, 12(1), 261–268. <https://doi.org/10.23887/jstundiksha.v12i1.46861>