

Klasifikasi Kematangan Manggis berdasarkan Warna Manggis dengan *Convolutional Neural Networ* dan *Color Autocorrelogram* pada Fitur Warna Manggis

Varraz Hazzandra A, 13521020¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

Abstract—Teknologi pengolahan digital dan pembelajaran mendalam digunakan dalam banyak hal, termasuk penyortiran manggis berdasarkan tingkat kematangannya. Penyortiran tersebut seringkali dilakukan secara manual. *Image processing* mampu membedah warna cita manggis. *Deep learning* memiliki subset metode yang bernama *Convolutional Neural Network* (CNN) yang sering dipakai dalam kasus klasifikasi citra.

Keywords—CNN, Manggis, CAC, MobileNet

I. INTRODUCTION

Manngis termasuk ke dalam buah yang sangat ppular untuk dikonsumsi. Mutu manggis menjadi hal yang harus diperhatikan sebelum diedarkan sebagai produk Mutu suatu produk pangan dapat ditentukan dari warnanya (Abdullah dkk., 2001). Manggis memiliki beberapa warna yang dapat diidentifikasi sebagai Tingkat kematangan yang menentukan kelayakan konsumsi.

Image processing merupakan konsep yang banyak diterapkan dalam kebutuhan pengolahan visual. Dengan demikian, *image processing* dapat diterapkan dalam penentuan kelayakan manggis yang dinilai dari warna pada manggis.

Kemampuan *image processing* dapat ditemukan dalam *convolutional neural network* (CNN). CNN umum digunakan dalam klasifikasi citra.

II. LANDASAN TEORI

A. KEMATANGAN MANGGIS

Beberapa elemen dapat menjadi fokus untuk menentukan tingkat kematangan manggis. Warna menjadi salah satu elemen tersebut. Menurut Kementerian Pertanian (2021), terdapat tujuh level kematangan manggis. Tabel 2.1 menunjukkan Tingkat kematangan manggis berdasarkan sumber tersebut.

Tabel 2.1 Tingkat Kematangan Manggis

Tahap	Gambar	Warna	Status
0		Kuning kehijauan	Belum siap dipetik
1		Hijau kekuningan	Belum siap dipanen
2		Hijau kekuningan dengan bercak merah	Sudah siap dipanen-ekspor
3		Merah kecoklatan	Sudah siap dipanen-antar pulau
4		Merah keunguan	Sudah siap dipanen-antar pulau dan pasar lokal

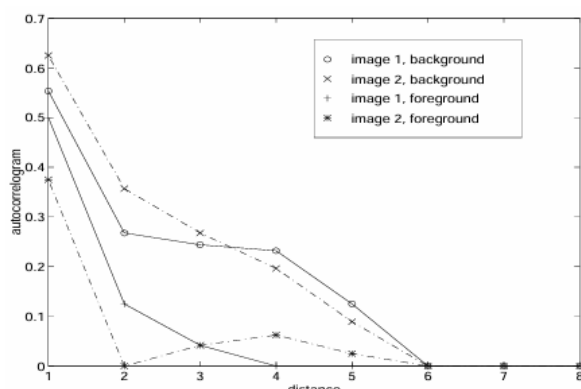
5		Ungu kemerahan	Sudah siap dipanen-pasar lokal
6		Ungu kehitaman	Masak-Pasar lokal

B. COLOR AUTOCORRELOGRAM

Color Auto Correlogram (CAC) adalah suatu algoritma ekstraksi fitur yang memetakan hubungan spasial antara warna warna identik pada citra (Huang dkk., 1997). CAC menghitung jarak antar piksel yang menggunakan warna yang sama. Hal tersebut menjadi pembeda CAC dengan histogram warna biasa yang umumnya hanya menghitung frekuensi warna. Fitur Deskripsi spasial yang diberikan CAC memberi keuntungan pada proses klasifikasi citra dengan warna sebagai fitur utama.



Gambar 2.1 Citra 1 dan 2

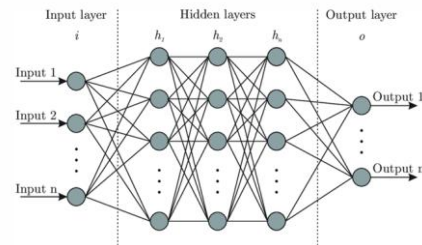


Gambar 2. 2 Diagram Correlogram Citra A dan B

C. DEEP LEARNING

Pembelajaran mendalam (*deep learning*) merupakan cabang dari pembelajaran mesin yang memanfaatkan cara kerja jaringan neural pada manusia untuk

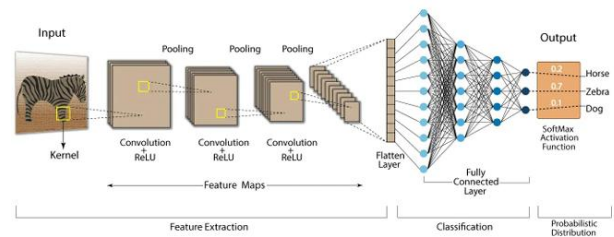
melakukan berbagai tugas, seperti klasifikasi, regresi, dan representasi. Terdapat tumpukan neuron yang membentuk suatu lapisan yang mengolah masukan data untuk suatu output. Lapisan tersebut berjumlah ratusan hingga ribuan untuk membentuk suatu jaringan neural. *Deep learning* menjadi konsep dasar untuk beberapa arsitektur pada tiap tugas, khususnya klasifikasi citra. CNN merupakan salah satu arsitektur berprinsip *deep learning*. CNN umumnya digunakan untuk input bertipe citra.



Gambar 2.3 Jaringan Saraf Tiruan yang Berprinsip *Deep Learning*

D. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan algoritma *deep learning* yang sering digunakan untuk mengolah data berstruktur seperti grid. Citra menjadi salah satu tipe data yang sering menjadi input CNN. CNN memiliki beberapa operasi untuk ekstraksi fitur langsung terhadap piksel pada citra.

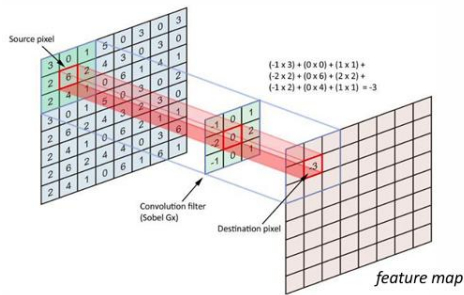


Gambar 2.4 Arsitektur CNN

CNN merupakan arsitektur yang terdiri dari tiga lapisan utama.

1. Convolutional Layer

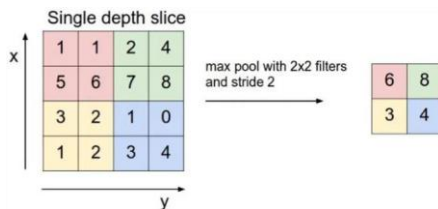
Lapisan konvolusional menerapkan operasi konvolusi terhadap citra masukan. Operasi tersebut memanfaatkan tapis yang melingkup beberapa pixel pada citra. Operasi yang dilakukan pada tiap pixel citra input menghasilkan pixel yang membentuk *feature map*. Proses pada lapisan ini dapat dilihat pada gambar 5.2.



Gambar 2.5 Proses pada Convolutional Layer

2. Pooling Layer

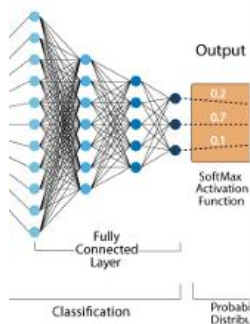
Lapisan *pooling* mengurangi ukuran spasial pada matriks hasil konvolusi. Pengurangan ukuran tersebut mengurangi daya komputasi untuk mengolah data. Terdapat dua cara pengolahan matriks pada lapisan ini, yaitu *max pooling* dan *average pooling*. *Max pooling* menghasilkan nilai maksimum dari pixel-pixel yang terlingkup dalam ukuran kernel. *Average pooling* menghasilkan nilai rata-rata dari pixel-pixel tersebut.



Gambar 2.6 Max Pooling

3. Fully Connected Layer

Fully Connected Layer merupakan lapisan terakhir pada CNN. Lapisan ini menghasilkan vektor dimensi jumlah kelas yang diprediksi oleh jaringan. Tiap elemen pada vektor merupakan nilai probabilitas tiap kelas. Klasifikasi dilakukan pada lapisan ini. Pada lapisan ini, terdapat fungsi *softmax* untuk menentukan hasil klasifikasi.



Gambar2.7 Fully Connected Layer

III. IMPLEMENTASI

A. Persiapan Data

Data yang dikumpulkan berupa gambar manggis untuk tiap tingkat kematangan berdasarkan warna. Dataset manggis tersebut didapat dari situs Kaggle bertajuk 'Mangosteen Computer Vision Dataset'. Terdapat 5 kelas (Tahap 0, tahap 1-2, tahap 3, tahap 4, tahap 5-6) dengan total citra 2478. Data sudah dibagikan untuk tiap jenis kebutuhan. Data latih memakan 70% jumlah citra. Data validasi memanfaatkan 20% jumlah citra total. 10% citra keseluruhan dimanfaatkan sebagai data uji. Data sudah melalui tahap augmentasi pada dataset sehingga augmentasi tidak dilakukan lagi. Pada tahap ini, dilakukan penerapan *color autocorrelogram* dalam ekstraksi fitur warna kulit manggis. Hal ini bertujuan untuk memodifikasi input yang akan diolah oleh model *pretrained CNN*.

```
DEF EXTRACT_CAC_FEATURES (IMAGE_PATH) :

    # REDUKSI DIMENSI

    IMG = CV2.RESIZE (IMG, (100, 100))

    IMG = CV2.CVT_COLOR (IMG,
    CV2.COLOR_BGR2RGB)

    # KUANTISASI WARNA

    DIV = 256 // 4

    QUANTIZED = (IMG // DIV) * DIV + (DIV
    // 2)

    PIXELS = QUANTIZED.RESHAPE (-1, 3)

    # HISTOGRAM CAC

    HIST, _ = NP.HISTOGRAMDD (PIXELS,
    BINS= (4, 4, 4),
    RANGE= ((0, 256), (0, 256), (0, 256)))

    FEATURE_VECTOR = HIST.FLATTEN ()

    # NORMALISASI

    NORM = NP.LINALG.NORM (FEATURE_VECTOR)

    IF NORM == 0: RETURN FEATURE_VECTOR

    RETURN FEATURE_VECTOR / NORM
```

Reduksi dimensi dilakukan untuk mempercepat proses komputasi pada matriks dengan menjaga informasi warna dominan pada kulit manggis. Citra dikonversi ke format RGB dan dimensi warna citra dikurangi menjadi 64 warna. Representasi hasil operasi tersebut berupa histogram yang kemudian diubah menjadi vektor elemen warna. Vektor dinormalisasi untuk menghasilkan representasi distribusi kepadatan warna kulit manggis.

B. Modelling

Inisiasi model dilakukan pada tahap ini. Bagian ekstraksi fitur keseluruhan memanfaatkan model *pre-trained* MobileNetV3Small. Proses pelatihan memanfaatkan model tersebut. Data yang dilatih sudah melalui proses *color autocorrelogram* untuk menjadi data input bagi model.

```
def build_hybrid_model(num_classes):
    # Input Gambar
    input_image = layers.Input(shape=(224, 224, 3), name="input_image")
    base_model = MobileNetV3Small(input_tensor=input_image, weights='imagenet', include_top=False, pooling='avg')
    base_model.trainable = False
    x1 = base_model.output
    x1 = layers.Dropout(0.2)(x1)

    # Input CAC
    input_cac = layers.Input(shape=(64,), name="input_cac")
    x2 = layers.Dense(64, activation='relu')(input_cac)
    x2 = layers.Dense(32, activation='relu')(x2)

    # Concat Layer
    combined = layers.Concatenate()([x1, x2])
    z = layers.Dense(64, activation='relu')(combined)
    output = layers.Dense(num_classes, activation='softmax')(z)

    model = Model(inputs=[input_image, input_cac], outputs=output)
    return model

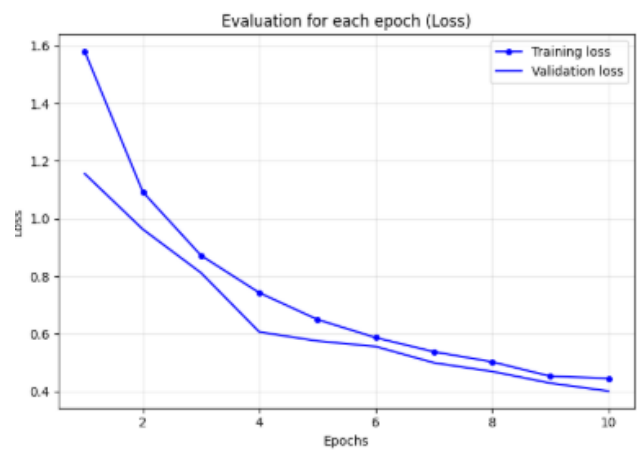
model = build_hybrid_model(num_classes = 5)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

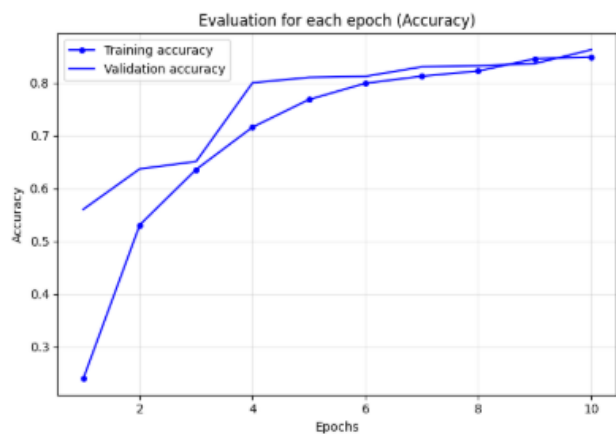
history = model.fit(training_data, validation_data=val_gen, epochs=10)
```

IV. HASIL

Gambar 4.1 dan gambar 4.2 menunjukkan histori pelatihan model MobileNetV3Small dengan CAC sebagai ekstraktor fitur warna untuk gambar input.



Gambar 4.1 Histori loss function dalam 10 epoch

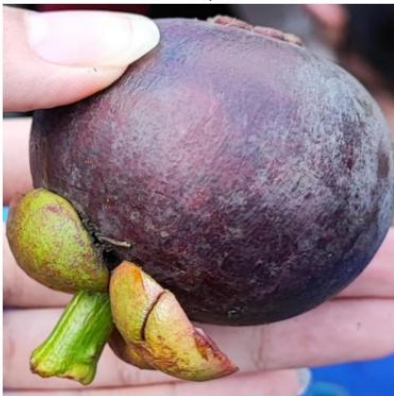


Gambar 4.2 Histori Akurasi dalam 10 epoch

Tabel 4.1 menunjukkan hasil pelatihan model dengan memanfaatkan CAC sebagai ekstraksi fitur warna. Tabel 4.2 menunjukkan hasil pengujian dengan gambar input.

Akurasi	Presisi	F1-Score	Recall
83.47%	84.78%	83.70%	83.47%

Tabel 4.2 Hasil Prediksi

Manggis Tahap 0	Prediksi: Tahap 0 (96.8%) 
Manggis Tahap 1-2	Prediksi: Tahap 1-2 (94.1%) 
Manggis Tahap 3	Prediksi: Tahap 4 (49.7%) 
Manggis Tahap 4	Prediksi: Tahap 4 (91.9%) 

Manggis Tahap 5-6	Prediksi: Tahap 5-6 (49.2%) 
----------------------	--

V. KESIMPULAN

Pendekatan CNN dengan CAC sebagai esktraktor warna pada data sebelum data dilatih memberikan hasil yang cukup baik dengan akurasi sebesar 83,47%. Namun, terdapat kesalahan prediksi pada gambar. Perlu dilakukan persiapan data yang lebih presisi terhadap 7 tahapan sesuai panduan dari Kementerian Pertanian (2021). Pelatihan tanpa CAC juga dibutuhkan untuk memberi hasil yang lebih jelas terkait perbandingan penggunaan CAC dan tidak.

UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa atas rahmatNya yang membuat penulis menyelesaikan makalah ini. Penulis juga mengucapkan terimakasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen pengajar mata kuliah pemrosesan citra digital tahun ajaran 2025/2026 yang telah membawakan ajaran materi dan bimbingan selama satu semester ini

REFERENSI

- [1] R. Munir, Convolutional Neural Network, Bandung: Program Studi Teknik Informatika, 2024
- [2] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 760–766
- [3] Direktorat Buah dan Florikultura, *Buku Lapang Budidaya Manggis*. Jakarta: Direktorat Jenderal Hortikultura, Kementerian Pertanian, 2021

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025



Varraz Hazzandra Abrar - 13521020