

Rancang Bangun Klasifikasi Buah Menggunakan CNN

1st Hardiansyah Firdaus
Electrical Engineering
UIN Sunan Gunung Djati
Bogor, Indonesia
fhardi00@gmail.com

2nd Haddy Firdaus
Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
haddyfirdaus@gmail.com

3rd M.Rizki Ramdani
Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
mriram260@gmail.com

4th Aldi Muhamad Fauzi
Electrical Engineering
UIN Sunan Gunung Djati
Purwakarta, Indonesia
aldimfauzi08@gmail.com

Abstrak—Perkembangan teknologi AI(Artificial Intelegent) atau kecerdasan buatan sudah merajalela dikehidupan. Salah satunya adalah *Deep Learning* yang sedang dikembangkan oleh peneliti dunia. *Deep Learning* adalah metode pembelajaran yang dirancang untuk terus menganalisis data dengan struktur logika yang mirip dengan bagaimana manusia mengambil keputusan.

Kata kunci—CNN, gray, pixel, buah

I. PENDAHULUAN

Indonesia dikenal sebagai salah satu negara yang banyak memproduksi tanaman hortikultura. Hal tersebut karena iklim tropis yang dimiliki Indonesia mendukung tanaman hortikultura (Anggiasari et al., 2016). Tanaman hortikultura terutama sayur sayuran dan buah-buahan merupakan merupakan bahan pangan asal tumbuhan yang mengandung kadar air tinggi dan dapat dikonsumsi dalam keadaan segar atau diolah (Supriati et al., 2008).

Buah salah satu komoditas hortikultura yang mempunyai peranan penting bagi pembangunan pertanian di Indonesia. Peran buah dalam tubuh sangat penting untuk metabolisme tubuh karena mengandung banyak vitamin dan mineral [Sinta, 2011].

Mengetahui banyaknya klasifikasi yang ditemukan pada sayuran, diperlukan adanya pendekatan digital agar dapat mengenali beragam jenis Buah dengan cepat dan mudah. Salah satu caranya adalah dengan menggunakan teknologi Deep Learning, yang merupakan salah satu bidang ilmu baru dalam Machine Learning dan memiliki perkembangan yang cukup pesat (Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. (2016).

Klasifikasi buah menggunakan Deep Learning Convolutional Neural Network (CNN) telah menjadi state-of-the-art dalam penelitian ini. Dalam pendekatan ini, terdapat beberapa perbedaan signifikan dibandingkan dengan metode tradisional dalam klasifikasi sayuran. CNN memungkinkan ekstraksi fitur otomatis dari gambar buah tanpa memerlukan fitur-fitur yang telah ditentukan terlebih dahulu oleh ahli domain (Wibowo & Hartanto, 2020). Arsitektur CNN yang terdiri dari lapisan konvolusi dan pooling memungkinkan model untuk secara hierarkis mempelajari fitur-fitur dari gambar sayuran, mulai dari fitur-fitur sederhana seperti tepi hingga fitur-fitur kompleks seperti tekstur dan bentuk (Krizhevsky et al., 2012). Keunggulan CNN juga terletak pada kemampuannya dalam menangani jumlah dan variasi buah yang berbeda, serta kemampuan skalabilitasnya yang memungkinkan model untuk mengklasifikasikan dengan akurasi yang tinggi bahkan pada sayuran yang belum pernah

dilihat sebelumnya (Yang, J., & Ye, Y. (2021). Selain itu, CNN juga memungkinkan pemanfaatan transfer learning, di mana model yang telah dilatih pada dataset besar seperti ImageNet dapat digunakan sebagai titik awal untuk melatih model klasifikasi sayuran, mengurangi waktu dan sumber daya yang dibutuhkan (Wibowo & Hartanto, 2020). Dengan tingkat akurasi yang tinggi, metode Deep Learning CNN telah terbukti melampaui kinerja metode tradisional dalam mengklasifikasikan berbagai jenis buah (Deng, Y., Zhang, Y., & Zhang, Y. 2020.).

Berdasarkan latar belakang masalah tersebut, maka dapat dirumuskan sebuah masalah yaitu bagaimana mengimplementasikan konsep Deep Learning dengan menggunakan CNN untuk melakukan klasifikasi jenis buah. Penelitian ini memiliki tujuan agar masyarakat dapat terbantu untuk mengidentifikasi sayuran yang dibutuhkan. Dengan adanya berbagai jenis sayuran yang ada saat ini, maka diperlukan adanya pendekatan teknologi untuk mengklasifikasi jenis buah tersebut. Terdapat berbagai teknologi yang dapat digunakan, terutama pada bidang Machine Learning yang memiliki berbagai Neural Networks yang akan dimanfaatkan untuk membuat model deteksi, guna mengidentifikasi suatu gambar dari jenis buah. Diantaranya adalah RNN (Recurrent Neural Network), GAN (Generative Adversarial Network), dan CNN (Convolutional Neural Network).

II. LANDASAN TEORI

A. Deep Learning

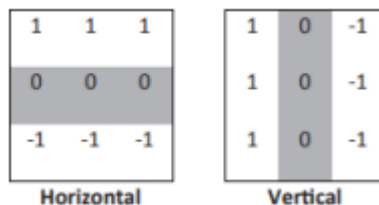
Deep learning merupakan salah satu bidang dari machine learning yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik deep learning memberikan arsitektur yang sangat kuat untuk supervised learning, dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik (O'Shea & Nash, 2015).

B. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak

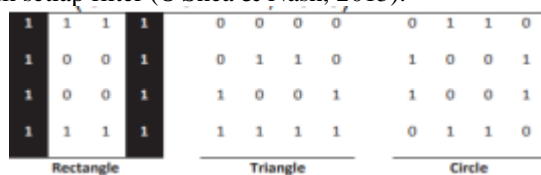
menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kuniyoshi Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeCun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeCun pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi ImageNet Large Scale Visual Recognition Challenge 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode Deep Learning, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode Machine Learning lainnya seperti SVM pada kasus klasifikasi objek pada citra. CNN menggunakan operasi konvolusi sebagai pengganti perkalian matriks umum, yang digunakan setidaknya pada satu lapisannya.

Sebuah CNN terdiri dari beberapa layer, yaitu pada Lapisan pertama akan memiliki filter yang mencari tepi horizontal dan filter lain untuk tepi vertikal. Filter ini ditunjukkan pada Gambar 2.1 sebagai matriks 3×3 . Jadi, kita tahu berapa banyak filter untuk digunakan di lapisan conv pertamadan juga apa filter ini. Ukuran 3×3 dipilih untuk filter karena ini adalah ukuran yang baik di mana struktur tepi horizontal dan vertikal jelas (O'Shea & Nash, 2015).

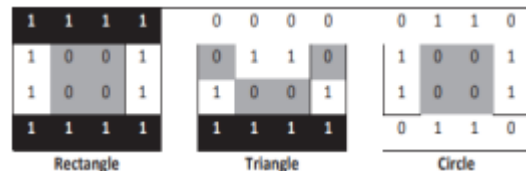


Gambar 2.1 Filter untuk mengenali tepi horizontal dan vertikal ukuran 3×3

Setelah menerapkan filter ini di atas matriks pada Gambar 2.1, lapisan konvolusi akan dapat mengenali tepi vertikal pada Gambar 2.2 dan tepi horizontal pada Gambar 2.3 (Wicaksono, 2020). Lapisan ini mampu mengenali tepi horizontal dan vertikal dalam persegi panjang. Itu juga mengenali tepi horizontal di dasar segitiga, tetapi tidak ada tepi dalam lingkaran. Pada saat ini, CNN memiliki dua kandidat untuk menjadi persegi panjang yang merupakan bentuk yang memiliki setidaknya satu sisi. Meskipun yakin bahwa bentuk ketiga tidak bisa persegi panjang CNN harus menyebarkannya ke lapisan lain sampai membuat keputusan di lapisan terakhir, karena menggunakan dua filter dalam lapisan konv pertama, ini menghasilkan dua output, satu untuk setiap filter (O'Shea & Nash, 2015).



Gambar 2.2 Tepi vertikal yang diakui berwarna hitam



Gambar 2.3 Tepi horizontal yang dikenali berwarna hitam

C. Tensorflow

Nama *Tensorflow* terdiri dari dua kata. Yang pertama adalah tensor yang merupakan unit data yang digunakan TF dalam komputasinya. Kata kedua adalah flow yang mencerminkan bahwa ia menggunakan paradigma aliran data. Akibatnya, TF membangun grafik komputasi yang terdiri dari data yang direpresentasikan sebagai tensor dan operasi yang diterapkan padanya. Untuk membuat hal-hal lebih mudah dipahami, ingatlah bahwa daripada menggunakan variabel dan metode, TF menggunakan tensor dan operasi (Setiawan & Herdianto, 2018).

III. METODE DAN IMPLEMENTASI

Deskripsi pekerjaan meliputi Dataset, Data training Flowchart program, penjelasan program, dan tampilan deteksi.

A. Dataset

Data ini diperlukan untuk pengujian data training, yaitu kumpulan gambar yang akan di training untuk pengolahan gambar. Data set itu sendiri diambil dari <https://www.kaggle.com/code/muhammeddalkran/fruit-360-dataset-with-cnn-in-keras#Model-Evaluation>. Total dataset pada sumber tersebut yaitu sebanyak 90.483 gambar. Namun pada percobaan kali ini jenis gambar yang digunakan hanya 5 jenis saja seperti pada gambar 3.1, Ukuran dari gambar adalah 100×100 piksel.

Name	Date modified	Type	Size
Avocado	6/23/2023 10:46 PM	File folder	
Banana	6/23/2023 10:47 PM	File folder	
Blueberry	6/23/2023 10:47 PM	File folder	
Cocos	6/23/2023 10:47 PM	File folder	
Dates	6/23/2023 10:47 PM	File folder	

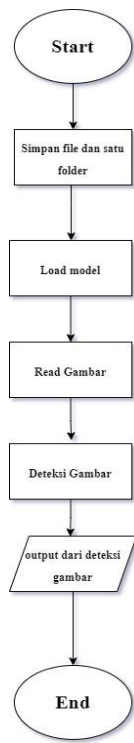
Gambar 3.1 Dataset Test pada Folder Komputer.

B. Data Training

Data training sebanyak 67.692 gambar akan dijadikan model H5 file, yang nantinya pada saat deteksi buah akan di load terlebih dahulu untuk modelnya. File yang berisi data training tersebut dinamai dengan 'Fruits_360.h5'.

C. Flowchart Program

Flowchart pada Gambar 3.2 adalah urutan dari jalannya program yang detail akan dibahas pada bab selanjutnya.



Gambar 3.2 Flowchat program

D. Penjelasan Program

Program di jalankan dengan menggunakan *software* Pycharm dengan pemograman bahasa python. Import fungsi load model dan image yang nanti akan digunakan untuk load deta model dari library crash dimana beguna untuk menyusun dan memproses dataset. Lalu import library numpy yang berguna untuk proses penghitungan matriks dan import matplotlib.pyplot yang berguna untuk grafik 2D. Load model dari file “Fruit_306.h5” yang berisikan data dari gambar-gambar yang telah ditraining.

- File juga pastikan sudah diupload ke dalam 1 folder dan satukan dengan file program.
- Import library yang diperlukan seperti gambar dibawah ini.

```

import os
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import tensorflow as tf
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications import ResNet50
from tensorflow import keras
# Menguji coba model
import numpy as np
from keras.preprocessing import image
import cv2
import tkinter as tk
from tkinter import filedialog
  
```

Gambar 3.3 Library yang digunakan

Library matplotlib diperlukan untuk grafik, plot, diagram, dan visualisasi lainnya dengan mudah. Library ini mendukung berbagai jenis plot, seperti scatter plot, line plot, bar plot, histogram, dan banyak lagi. Library NumPy menyediakan objek array multidimensi yang efisien, bersama dengan berbagai fungsi matematika untuk memanipulasi array tersebut. Library OS yang digunakan untuk berinteraksi dengan sistem operasi. Library

Tesortflow adalah library open source yang fokus pada komputasi numerik menggunakan grafik aliran tensor. Lalu library Opencv ini menyediakan fungsi-fungsi yang kuat untuk memanipulasi gambar dan video, mendeteksi objek, melacak objek, mengenali wajah, mengukur jarak, dan banyak lagi

c. Tahapan *preprocessing*

Pada tahapan ini menggunakan *ImageDataGenerator* untuk tahapan *preprocessing*. *ImageDataGenerator* adalah sebuah kelas dalam library Keras yang digunakan untuk memperluas dataset gambar dengan berbagai operasi augmentasi. Augmentasi data adalah proses mengubah variasi data pelatihan dengan melakukan transformasi seperti rotasi, pemotongan, pergeseran, zoom, dan lainnya. Tujuannya adalah meningkatkan keberagaman data pelatihan, sehingga model pembelajaran mesin dapat belajar dengan lebih baik dan memiliki kemampuan generalisasi yang lebih baik. Berikut program yang digunakan dalam tahap *preprocessing*.

```

# Menggunakan ImageDataGenerator untuk preprocessing
datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    validation_split=0.2
)

# Menyiapkan data train dan data validation
train_data = datagen.flow_from_directory(
    base_dir,
    class_mode='categorical',
    subset='training',
    target_size=IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    seed=SEED
)

valid_data = datagen.flow_from_directory(
    base_dir,
    class_mode='categorical',
    subset='validation',
    target_size=IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    seed=SEED
)

# Image Augmentation
data_augmentation = tf.keras.Sequential(
    [
        tf.keras.layers.RandomFlip("horizontal"),
  
```

Gambar 3.4 Tahap *Preprocessing* dengan *ImageDataGenerator*

d. Tahapan Pemilihan Aksitektur

Arsitektur Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf tiruan (neural network) yang dirancang khusus untuk tugas pengolahan citra dan pemrosesan visual.

Program yang digunakan dalam perancangan kali ini dapat dilihat pada gambar berikut. Tahap ini dimulai dari pembuatan aksitektur sampai membuat plot akurasi dan *loss* model.

```

# Compiling model
cnn_model.compile(
    loss='categorical_crossentropy',
    optimizer=tf.keras.optimizers.Adam(),
    metrics=['accuracy']
)

# Training model CNN
cnn_hist = cnn_model.fit(
    train_data,
    epochs=10,
    validation_data = valid_data
)

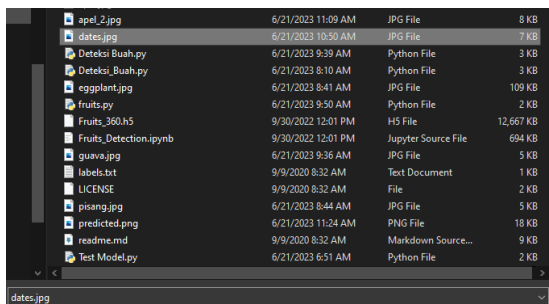
# Membuat plot akurasi model CNN
plt.figure(figsize=(10,4))
plt.plot(cnn_hist.history['accuracy'])
plt.plot(cnn_hist.history['val_accuracy'])
plt.title('CNN model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.grid(True)
plt.show()
  
```

Gambar 3.5 Tahap Pemilihan Arsitektur

e. Pendeteksian Gambar

Tahapan kali ini mendeteksi gambar dari file yang sudah disiapkan, dan file tersebut di ambil dari Google. Pilih

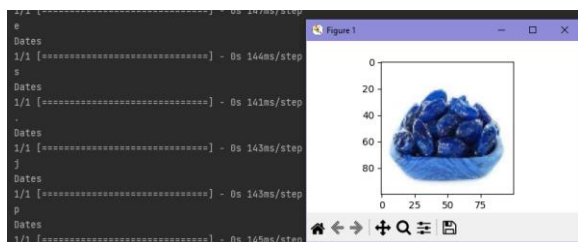
gambar yang akan dideteksi, untuk pendeteksian kali ini menggunakan pisang sebagai masukannya.



Gambar 3.6 Pemilihan Gambar yang akan di Deteksi

f. Hasil deteksi jenis buah-buahan

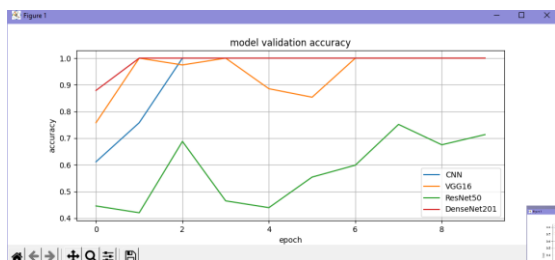
Output yang ditampilkan dari program yang telah dibuat dapat dilihat pada Gambar 3.7.



Gambar 3.7 Hasil Deteksi Gambar.

IV. PEMBAHASAN

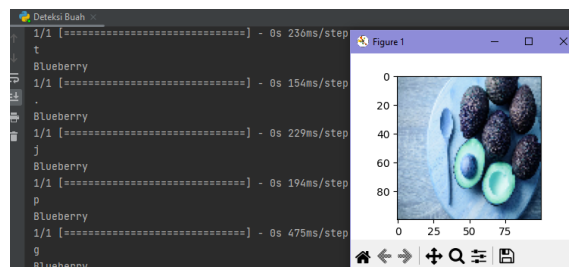
Hasil dari pengujian kali ini mendapatkan hasil yang maksimal atau akurasi yang cukup tepat, bisa dilihat pada grafik berikut.



Gambar 4.1 Model Akurasi Validasi

Dapat dilihat pada grafik diatas, garis yang konsisten atau tidak berfluktuasi itu adalah model DenseNet201, jika dibandingkan model-model lainnya seperti CNN, VGG16, dan ResNet50 hasilnya naik turun atau tidak konsisten, hal ini sangat berpengaruh terhadap hasil pendeteksian gambar. Model arsitektur yang digunakan untuk percobaan pendeteksian jenis buah yaitu model DenseNet201, karena memiliki kinerja yang lebih baik. Hal ini diperkuat karena DenseNet201 menggunakan struktur koneksi padat yang merupakan inovasi utama dari arsitektur DenseNet. Pada struktur ini, setiap lapisan memiliki koneksi langsung ke semua lapisan berikutnya dalam arsitektur. Hal ini memungkinkan aliran informasi yang lebih langsung dan efisien melalui jaringan.

Lalu pada program terdapat Variabel "epoch" dalam konteks pemodelan arsitektur Python mengacu pada jumlah kali yang model akan melalui keseluruhan dataset pelatihan selama proses pelatihan. Setiap kali model melewati seluruh dataset pelatihan, itu dihitung sebagai satu epoch. Hal ini terbukti ketika menggunakan nilai epochnya 2, maka hasil output yang ditampilkan adalah Blueberry meskipun foto yang ditampilkan itu sesuai dengan masukan, hal ini dikarenakan jumlah model yang dilewati oleh arsitektur model tidak seluruhnya dilakukan. Berikut hasil dari pengujian yang tidak sesuai dengan masukan gambar/jenis buah.



Gambar 4.2 Hasil Deteksi Gambar

Dapat dilihat pada Gambar 4.2 hasil output yang ditampilkan pada terminal berbeda dengan masukan, masukan kali ini menggunakan alpukat sebagai gambar yang akan dideteksi, sedangkan hasil output yang ditampilkan pada terminal itu menyatakan blueberry.

REFERENSI

- [1] O'Shea, K., & Nash, R. (2015). An Introduction to Extreme learning machines, Neural and Evolutionary Computing: Cornell University.
- [2] Pujoseno, J. (2018). Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Alat Tulis. Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia
- [3] Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C. DeepFruits: Sistem Deteksi Buah Menggunakan Deep Neural Networks. Sensor . 2016; 16(8):1222.
- [4] Wibowo, A., & Hartanto, R. (2020). Vegetable Classification using Convolutional Neural Network. International Journal of Computer Applications, 180(16), 18-22.
- [5] Sinta, "Pengertian Buah-buahan Lokal," vol. 11, no. 2, pp. 10–14, 2011.
- [6] Yang, J., & Ye, Y. (2021). Vegetable Recognition and Classification Based on Deep Learning. In 2021 5th International Conference on Automation, Electronics and Electrical Engineering (AEEE) (pp. 68-72). IEEE.
- [7] Deng, Y., Zhang, Y., & Zhang, Y. (2020). Vegetable Classification Based on Deep Learning Convolutional Neural Network. In 2020 4th International Conference on Advances in Artificial Intelligence (ICAAI) (pp. 70-74). IEEE.
- [8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems (pp. 1097-1105).
- [9] Wicaksono, A. F. (2020). Tutorial Dasar Tensorflow. [Online]. Available: https://ir.cs.ui.ac.id/alfan/tutorial/tf_intro.html.
- [10] Nuraeny, F., Nawansih, O., Sitanggang, M., & Merry Monika. Terbitan, : Teknologi Hasil Pertanian Fakultas Pertanian Universitas Lampung, 2018.
- [11] Setiawan D. 2006. "Peranan Sektor Unggulan Terhadap Pertumbuhan Ekonomi Daerah: Pendekatan Input-Output Multiregional Jawa Timur, Bali Dan Nusa Tenggara Barat. [Tesis] IPB."
- [12] Anggiasari, N. M., Y. Indriani, dan T. Endaryanto. 2016. "Sikap Dan Pengambilan Keputusan Pembelian Sayuran Organik Oleh Konsumen

Di Kota Bandar Lampung.” Jurnal Ilmu-ilmu Agribisnis (JIIA) Volume 4 N: :391-397.

- [13] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In International Conference on Machine Learning (ICML) (pp. 1597–1607).
- [14] Supriati, Y., Yulia, Y & Nurlela, I. (2008). Taman Sayur dan 19 Desain Menarik. Penebar Swadaya. Jakarta.