

Klasifikasi Tingkat Roasting Biji Kopi Dengan Metode CNN

Ilham Alfiantama¹, Michael Ilham Kresnawan², Andres Putra Handoko³

^{1,2,3}Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri

E-mail: ¹[*ilhamalfianacpah@gmail.com](mailto:ilhamalfianacpah@gmail.com), ²elhamkresnawan@gmail.com,

³andresputrahandoko@gmail.com

Abstrak – Penelitian ini mengusulkan penggunaan *Convolutional Neural Network (CNN)* untuk mengklasifikasikan tingkat roast pada biji kopi berdasarkan gambar digital. Tingkat roast kopi dapat mempengaruhi rasa dan aroma kopi, sehingga penentuan tingkat roast yang tepat menjadi penting dalam industri kopi. *CNN* merupakan metode *Deep Learning* yang dapat memproses citra dengan akurasi tinggi. Dataset berisi 1600 gambar biji kopi dengan empat tingkat roast, yaitu *dark roast*, *green roast*, *medium roast*, dan *light roast*. Proses training dilakukan dengan *Visual Studio Code* dan library *TensorFlow*, *Keras*, *Open CV*, dan *Numpy*. Hasil evaluasi menunjukkan akurasi mencapai 99% pada epoch ke-10. Sistem ini diharapkan membantu penikmat kopi mengenali tingkat roast kopi dengan lebih akurat.

Kata Kunci — Biji Kopi, *CNN*, *Deep Learning*, Klasifikasi, Tingkat Roast

1. PENDAHULUAN

Kopi adalah salah satu bahan minuman yang paling banyak dikonsumsi oleh masyarakat Indonesia. Kopi diyakini memiliki nilai ekonomi yang tinggi untuk meningkatkan ekonomi masyarakat dan sebagai sumber devisa Negara. Menurut data Badan Pusat Statistik (BPS) pada tahun 2021, Indonesia memproduksi biji kopi mencapai 774,6 ribu ton yang dimana pada tahun sebelumnya produksi kopi hanya sebesar 753,9 ribu ton, hal ini menunjukkan adanya peningkatan produksi kopi di Indonesia sebesar 2,75% [1].

Aroma dan rasa kopi dipengaruhi oleh tingkat penyangraian (*roasting*). Proses *roasting* merupakan salah satu tahapan yang penting dalam pengolahan kopi. Proses penyangraian (*roasting*) dapat menghasilkan berbagai level, ada 4 level *roasting* yang dilakukan yaitu *unroasted*, *light roasted*, *medium roasted*, dan *dark roasted*. Untuk mengetahui tingkat sangrai biji kopi dapat dilihat secara kasat mata dari warna hasil sangrai. Namun hal ini dianggap kurang akurat dan kurang optimal karena memungkinkannya terjadi *human error*, akibatnya sulit untuk menentukan tingkat sangrai biji kopi yang baik dan tepat. Adapun tindakan yang dapat dicapai untuk membantu penanganan masalah tersebut yaitu dengan memanfaatkan teknologi pengenalan citra metode *Deep Learning*.

Deep Learning (Pembelajaran Dalam) merupakan Teknik *Machine Learning* yang berbasis Jaringan Syaraf Tiruan (JST) atau evolusi dari JST. Salah satu algoritma *Deep Learning* yang merupakan pengembangan dari *Multi Layer Perceptron (MLP)* adalah *Convolutional Neural Network (CNN)* [2]. Penelitian pernah dilakukan oleh Joelyan Vicky, Frisca Ayu Serta Bagas Julianto (2023) dimana penelitian tersebut mengimplementasikan metode *CNN* untuk mendeteksi penyakit pada daun alpukat. Penelitian tersebut menggunakan data berjumlah 212 data training serta 52 data testing. Hasil testing dari penelitian tersebut mendapatkan hasil akurasi sebesar 96% pada epoch ke 10. Perbedaan dengan penelitian ini terletak pada objek penelitian. Penelitian ini menggunakan biji kopi sebagai objeknya[3]. Penelitian menggunakan metode yang serupa tetapi dengan objek yang berbeda juga pernah dilakukan oleh saudara Arvi Arkadia, Sekar Ayu Damayanti, Desta Sandya Prasvita (2021). Penelitian ini mengklasifikasikan objek buah Mangga Badami untuk menentukan tingkat kematangan dengan metode *CNN*. Dataset yang digunakan berjumlah 204 citra, terdiri dari 35 Mangga Badami Busuk, 75 Mangga Badami Mentah, dan 94 Mangga Badami Matang. Sedangkan dalam penelitian ini menggunakan objek biji kopi untuk menentukan tingkat kematangan [4].

Dari penjelasan tersebut, penulis akan merancang sistem klasifikasi tingkat *roasting* pada biji kopi berdasarkan gambar digital menggunakan pemrosesan gambar *Convolutional Neural Network (CNN)*. Metode ini bertujuan untuk mengetahui jenis *roasting* biji kopi dari gambar digital untuk menentukan jenis *roasting* itu sendiri. Tujuannya agar penikmat kopi atau pecinta kopi yang awam mengenai jenis *roasting* kopi dapat mengenali jenis level atau kategori *roasting* kopi.

2. METODE PENELITIAN

Kecerdasan Buatan (*Artificial Intelligence* atau AI) didefinisikan sebagai kecerdasan yang diperlihatkan oleh suatu entitas buatan. Sistem seperti ini biasanya disebut sebagai komputer. Kecerdasan diciptakan dan tertanam dalam mesin (komputer) sehingga berfungsi layaknya manusia. Kecerdasan buatan didasarkan pada prinsip bahwa kecerdasan manusia dapat didefinisikan sedemikian rupa sehingga mesin dapat dengan mudah menirunya dan menjalankan tugas, dari yang paling sederhana hingga yang kompleks. Tujuan kecerdasan buatan meliputi pembelajaran, penalaran, dan persepsi.

2.1 Dataset Roasting Biji Kopi

Dalam penelitian ini, peneliti menggunakan data tingkat roasting pada biji kopi yang berupa foto atau gambar. Data ini berjumlah 1600 foto. Beberapa data foto tersebut akan dibagi menjadi 2 yaitu data yang nantinya digunakan untuk *training* dan data yang digunakan untuk *testing*. Baik data training maupun data testing akan dibagi lagi dengan 4 kelas tingkat roasting yaitu dark roast, green roast, medium roast, dan light roast. Untuk data training, data yang terdapat pada masing masing kelas berjumlah 300 data. Sedangkan untuk data testing, masing masing kelas terdapat 100 data. Ukuran dari masing masing foto juga telah di samakan yaitu dengan ukuran 224 x 224. Dataset yang digunakan pada penelitian ini didapatkan dari *Kaggle dataset* dengan judul *Coffee Bean Dataset Resized (224 X 224)*. Berikut contoh dataset yang digunakan pada penelitian ini.



Gambar 1. Gambar Biji Kopi Dengan Tingkat Roasting Yang Berbeda

Pada gambar 1 dapat kita lihat bahwa perbedaan dari setiap tingkat roasting terdapat pada warnanya. Gambar sebelah kanan merupakan gambar biji kopi dengan tingkat *roasting unroasted*. Pada gambar biji kopi dengan tingkat *roasting unroasted* dapat kita lihat cenderung berwarna hijau. Sedangkan pada gambar kanan yang merupakan biji kopi dengan tingkat *roasting light roast* terlihat bahwa warna dari biji kopi tersebut telah memudar dan agak berwarna kecoklatan.

2.2 Convolutional Neural Network

Convolutional Neural Network merupakan algoritma *deep learning*, yang paling umum diterapkan pada saat menganalisis gambar visual. CNN merupakan *multilayer perceptron* yang setiap *neuronnya* saling terhubung untuk lapisan berikutnya. CNN juga mampu untuk menemukan pola hirarkis dalam data dan mengumpulkan piksel yang kompleks dari piksel sederhana. Dibandingkan dengan algoritma klasifikasi gambar lainnya, CNN menggunakan pra pemrosesan yang lebih sedikit dan CNN juga mempelajari filter yang dalam algoritma gambar biasa [5].

2.3 Confusion matrix

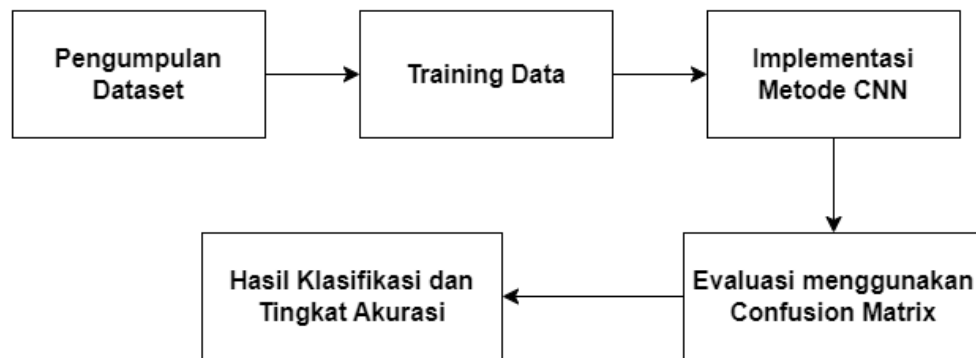
Sebuah perhitungan *matrix* yang biasa digunakan untuk mendapatkan sebuah nilai akurasi presisi dan *recall*. Perhitungan ini lazim di gunakan dalam metode *machine learning* atau pun lain yang berkaitan dengan metode klasifikasi. Untuk rumus yang digunakan sebagai berikut.[6]

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+TN+FN} \dots\dots\dots(1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \dots\dots\dots(2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \dots\dots\dots(3)$$

2.4 Alur/tahapan penelitian



Gambar 2. Alur Tahapan Penelitian

Berikut penjelasan alur tahapan penelitian:

1. Dalam penelitian ini yang dilakukan adalah mencari dan mengumpulkan dataset.
2. Kemudian akan dilakukan proses *training* dengan mengimplementasikan metode CNN pada dataset tersebut.
3. Setelah itu peneliti akan melakukan uji coba pada hasil *training* dan kemudian akan dilakukan proses perhitungan akurasi, *recall* serta presisi dari hasil *testing*.
4. Kemudian akan dilakukan evaluasi dengan menggunakan *confusion matrix*.
5. Selain itu peneliti juga akan menghitung waktu komputasi dari model tersebut.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi

Dalam penelitian ini, peneliti menggunakan *software Visual Studio Code* dengan tambahan ekstensi *Jupyter* untuk mempermudah proses penelitian dan dapat menganalisa satu demi satu proses yang sedang dilakukan. Selain itu terdapat beberapa *library* yang digunakan oleh peneliti. Diantaranya *TensorFlow*, *Keras*, *Open CV*, *Numpy*, *Scikit-learn* dan sebagainya. Beberapa tahapan proses yang dilakukan pada penelitian ini adalah sebagai berikut :

1. Melakukan *import Library*

Berbagai *library* yang dipakai pada penelitian ini akan di *import* terlebih dahulu dimana nantinya akan dipanggil pada proses proses selanjutnya. Seperti *library TensorFlow* yang nantinya dapat membantu untuk proses klasifikasi gambar dengan metode *Convolutional Neural Network*. Kemudian terdapat *library LabelEncoder* untuk mengubah label pada kelas yang awalnya merupakan kata akan diubah menjadi angka sehingga memudahkan dalam proses model nantinya. Beberapa *library* lain yang digunakan pada penelitian ini dapat dilihat pada gambar 3 berikut.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, Flatten, Dense, Conv2D, MaxPool2D, Dropout
from tensorflow.keras.optimizers import Adam
import numpy as np
import cv2
import glob
import matplotlib.pyplot as plt
import seaborn
```

Gambar 3. Proses *Import Library*

2. Proses pemanggilan dataset dan *praprocessing*

Dataset yang terdapat didalam folder lokal komputer akan dipanggil. Agar proses klasifikasi nantinya berjalan lebih cepat dan tidak menghambat jalannya proses, maka data yang berupa gambar akan dilakukan proses *praprocessing* yaitu dengan merubah ukuran dari masing masing gambar. Pada penelitian ini kami akan merubah semua ukuran dari gambar menjadi 224x224.

```
for label in label_list:
    for imagePath in glob.glob(imagePaths+label+'\\*.png'):
        print(imagePath)
        image = cv2.imread(imagePath)
        image = cv2.resize(image, (224, 224))
        data.append(image)
        labels.append(label)
```

Gambar 4. Proses *preprocessing*

3. Proses Perubahan Label Menjadi Angka
Pada proses ini label yang terdapat pada sistem akan dirubah menjadi angka yang awalnya merupakan kata. Proses ini nantinya akan memudahkan proses model.
4. *Split* dataset
Membagi dataset menjadi data yang akan digunakan untuk *training* dan data yang akan digunakan untuk *testing*. Proses ini menggunakan *library split* dengan tambahan parameter ukuran *test* yaitu 0.2 serta *random state* dengan nilai 42.
5. Membangun arsitektur CNN
Proses ini akan membuat atau membangun model CNN yang akan dilakukan pada saat proses *training* dan *testing* nantinya. Dimana dalam membangun model ini membutuhkan beberapa *layer* agar nantinya klasifikasi akan menghasilkan nilai sesuai dengan yang diharapkan.

```
model = Sequential()
# Extracted Feature Layer
model.add(InputLayer(input_shape=(224, 224, 3)))
model.add(Conv2D(filters=32, kernel_size=3, strides=1,
padding='same', activation='relu'))
model.add(MaxPool2D(pool_size=2, padding='same'))
model.add(Conv2D(filters=64, kernel_size=3, strides=1,
padding='same', activation='relu'))
model.add(MaxPool2D(pool_size=2, padding='same'))
model.add(Conv2D(filters=128, kernel_size=3, strides=1,
padding='same', activation='relu'))
model.add(MaxPool2D(pool_size=2, padding='same'))
model.add(Dropout(0.25))
model.add(Flatten())
# Fully Connected Layer
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(4, activation='softmax'))
```

Gambar 5. Pembangunan model CNN

6. Proses *training*
Setelah membuat beberapa kode untuk membangun model dari CNN, akan dilakukan proses *training* pada data *training* yang sudah tersedia. Peneliti menggunakan *10 epoch* dalam proses *training* ini. Selain itu data *testing* yang sudah dibagi sebelumnya akan dibuat sebagai *validation data*.
7. Evaluasi
Setelah proses *training* dijalankan, akan tampil proses *training* pada setiap *epoch* dengan beberapa rincian lain yaitu nilai *loss*, akurasi, *val_loss* serta *val accuracy*. Beberapa nilai yang keluar tersebut akan dianalisa oleh peneliti dan kemudian akan dievaluasi apakah model yang digunakan sudah menghasilkan nilai yang sesuai dengan harapan atau belum. Berikut hasil proses *training* yang telah dilakukan oleh peneliti.

```
Epoch 1/10 [-----] 486 25/step - loss: 0.6765 - accuracy: 0.4540 - val_loss: 0.0096 - val_accuracy: 0.0107
Epoch 2/10 [-----] 516 25/step - loss: 0.2821 - accuracy: 0.7990 - val_loss: 0.1381 - val_accuracy: 0.0267
Epoch 3/10 [-----] 546 25/step - loss: 0.1858 - accuracy: 0.9281 - val_loss: 0.0938 - val_accuracy: 0.0310
Epoch 4/10 [-----] 576 25/step - loss: 0.1043 - accuracy: 0.9592 - val_loss: 0.0411 - val_accuracy: 0.0708
Epoch 5/10 [-----] 606 25/step - loss: 0.0537 - accuracy: 0.9812 - val_loss: 0.0298 - val_accuracy: 0.0882
Epoch 6/10 [-----] 636 25/step - loss: 0.0534 - accuracy: 0.9846 - val_loss: 0.0298 - val_accuracy: 0.0708
Epoch 7/10 [-----] 666 25/step - loss: 0.0255 - accuracy: 0.9885 - val_loss: 0.0207 - val_accuracy: 0.0875
Epoch 8/10 [-----] 696 25/step - loss: 0.0208 - accuracy: 0.9885 - val_loss: 0.0263 - val_accuracy: 0.0710
Epoch 9/10 [-----] 726 25/step - loss: 0.0128 - accuracy: 0.9948 - val_loss: 0.0258 - val_accuracy: 0.0853
Epoch 10/10 [-----] 756 25/step - loss: 0.0102 - accuracy: 0.9957 - val_loss: 0.0260 - val_accuracy: 0.0875
```

Gambar 6. Hasil dari proses Training

Dapat dilihat bahwa nilai yang didapatkan oleh peneliti pada *epoch 10* mendapatkan tingkat akurasi mencapai 99%. Selain menganalisa dari hasil proses *training*, peneliti juga mengevaluasi model menggunakan *classification report*.

```
8/8 [=====] - 2s 270ms/step
      precision    recall  f1-score   support

   Dark         0.97         0.98         0.98         62
   Green         1.00         1.00         1.00         60
   Light         1.00         1.00         1.00         60
   Medium        0.98         0.97         0.97         58

 micro avg         0.99         0.99         0.99        240
 macro avg         0.99         0.99         0.99        240
weighted avg         0.99         0.99         0.99        240
samples avg         0.99         0.99         0.99        240
```

Gambar 7. Classification Report

Pada Gambar 7 peneliti menggunakan *classification report* untuk mengevaluasi model. Terdapat beberapa nilai yang ditampilkan oleh *classification report* diantaranya *precision*, *recall*, serta *f1 score* pada setiap kelas.

8. Proses *Testing* menggunakan data acak

Setelah menganalisa model, peneliti melakukan *testing* terhadap model dengan memanggil file gambar acak yang kemudian akan diproses oleh model. Output yang diberikan berupa tampilan gambar beserta dengan klasifikasinya.



Gambar 8. Hasil dari proses klasifikasi

4. SIMPULAN

Dari Hasil Implementasi yang dilakukan oleh peneliti menghasilkan beberapa kesimpulan diantaranya:

1. Model CNN dapat menunjukkan hasil yang sangat baik dalam mengklasifikasikan tingkat roasting pada biji kopi.
2. Sistem dapat mengimplementasikan CNN untuk mengklasifikasikan tingkat roasting pada biji kopi.

5. SARAN

Berdasarkan hasil implementasi model CNN kedalam proses klasifikasi tingkat roasting pada biji kopi masih terdapat beberapa kesalahan. Terdapat beberapa model lain yang mungkin nantinya akan menghasilkan nilai yang lebih besar daripada model CNN dalam mengklasifikasikan tingkat roasting biji kopi. Selain itu model CNN mungkin bisa dikombinasikan dengan model *praprocessing* lain seperti deteksi tepi, hsv dan sebagainya. Dengan menambahkan proses *praprocessing* selain yang dilakukan pada penelitian ini akan mendapatkan nilai yang lebih tinggi.

DAFTAR PUSTAKA

- [1] A. Ardiansyah and N. F. Hasan, "Deteksi dan Klasifikasi Penyakit Pada Daun Kopi Menggunakan Yolov7," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 12, no. 1, pp. 30–35, 2023, doi: 10.32736/sisfokom.v12i1.1545.
- [2] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 3, no. 2, pp. 49–56, 2018.
- [3] J. V. P. Putra, F. Ayu, and B. Julianto, "Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN," *Stain. (Seminar Nas. Teknol. Sains)*, vol. 2, no. 1, pp. 155–162, 2023.

- [4] A. Arkadia, S. Ayu Damayanti, and D. Sandya Prasvita, “Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN,” *Pros. Semin. Nas. Mhs. Bid. Ilmu Komput. dan Apl.*, vol. 2, no. 2, pp. 158–165, 2021.
- [5] E. Syahbana, F. I. Terapan, and U. Telkom, “Klasifikasi Tingkat Kematangan Biji Kopi Dengan Algoritma Artificial Intelligence Berbasis Website,” pp. 0–3.
- [6] U. P. Sanjaya *et al.*, “Optimasi Convolutional Neural Network dengan Standard Deviasi untuk Klasifikasi Pneumonia pada Citra X-rays Paru,” vol. 7, no. 3, pp. 40–47, 2023.