Laporan Presentasi: Analisis Model CNN untuk Klasifikasi Fashion MNIST

Muhammad Faiz Anindyo Widodo 1103213011



Latar Belakang

Fashion MNIST adalah dataset yang terdiri dari gambar-gambar pakaian dengan tujuan untuk menggantikan dataset MNIST yang sudah lama digunakan. MNIST terdiri dari digit tangan, sementara Fashion MNIST menyediakan tantangan yang lebih realistis dengan gambar fashion seperti kaos, sepatu, dan tas. Klasifikasi gambar dengan model deep learning seperti Convolutional Neural Networks (CNN) telah menunjukkan hasil yang sangat baik pada dataset ini.



Tujuan

Proyek ini bertujuan untuk membangun dan mengevaluasi model CNN untuk klasifikasi gambar menggunakan dataset Fashion MNIST. Dengan menggunakan teknik-teknik yang lebih canggih seperti augmentasi data, batch normalization, dan dropout, model ini diharapkan dapat mencapai akurasi di atas 90%.



Image Classification

Image classification dalam machine learning adalah proses di mana sebuah algoritma atau model dilatih untuk mengkategorikan atau mengklasifikasikan gambar ke dalam salah satu dari beberapa kategori atau kelas yang telah ditentukan. Proses ini melibatkan beberapa langkah utama, mulai dari pengumpulan dan pengolahan data gambar hingga pelatihan model dan evaluasi kinerja.



CNN

Convolutional Neural Networks (CNN) adalah jenis jaringan saraf tiruan yang terutama digunakan dalam pengenalan dan klasifikasi gambar. CNN memanfaatkan lapisan konvolusi untuk menangkap fitur-fitur spasial dari gambar, seperti tepi, sudut, dan pola.



Mnist Fashion Dataset

MNIST Fashion Dataset adalah dataset yang dirancang sebagai pengganti atau pelengkap dari dataset MNIST yang sudah sangat dikenal. MNIST Fashion Dataset berisi gambar-gambar artikel fashion (seperti sepatu, kaos, tas, dll.) dan sering digunakan sebagai benchmark dalam tugas-tugas computer vision dan machine learning.

Mengimpor Library dan Dataset

```
# Mengimpor library yang diperlukan import tensorflow as tf from tensorflow.keras.models import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization from tensorflow.keras.preprocessing.image import ImageDataGenerator import numpy as np import matplotlib.pyplot as plt

[23] # Mengimpor dataset Fashion MNIST dari Keras mnist = tf.keras.datasets.fashion_mnist

# Memuat dataset dan membaginya menjadi set pelatihan dan set pengujian (train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```



Melimit Data

```
# Menggunakan hanya 150 data pertama dari set pelatihan dan pengujian untuk pelatihan cepat

train_images, train_labels = train_images[:150], train_labels[:150]

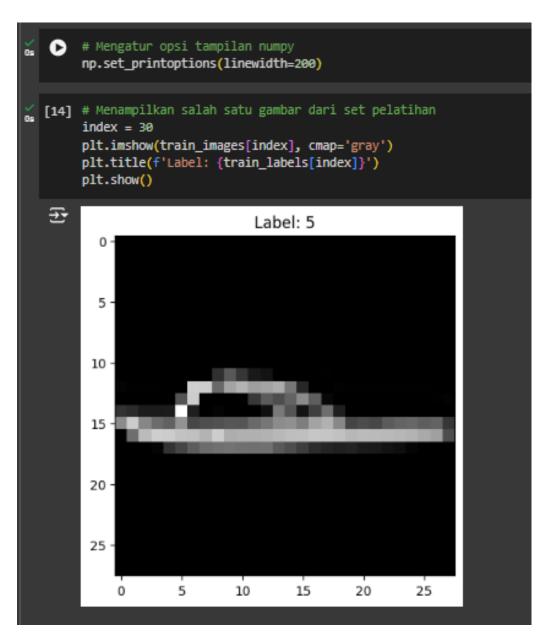
test_images, test_labels = test_images[:150], test_labels[:150]

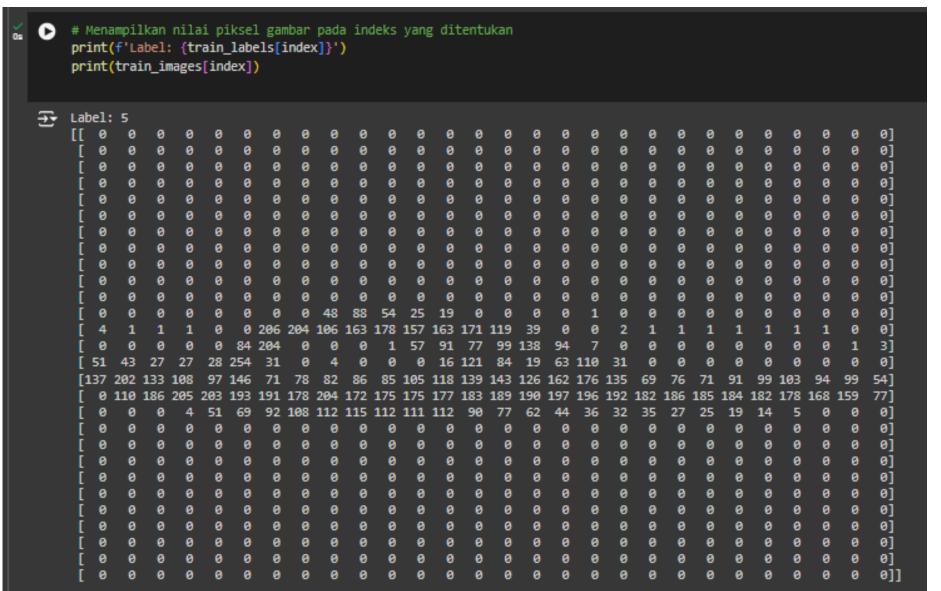
* Menampilkan bentuk data pelatihan dan pengujian
    print(f'Train images shape: {train_images.shape}')
    print(f'Train labels shape: {train_labels.shape}')
    print(f'Test images shape: {test_images.shape}')
    print(f'Test labels shape: {test_labels.shape}')

* Train images shape: (60000, 28, 28)
    Train labels shape: (10000, 28, 28)
    Test labels shape: (10000, 28, 28)
    Test labels shape: (10000, 28, 28)
```



Menampilkan Gambar dan Nilai Piksel







Preprocessing dan Augmentasi

```
[26] # Preprocessing data: merubah bentuk dan melakukan normalisasi
        train_images = train_images.reshape((train_images.shape[0], 28, 28, 1)) / 255.0
        test_images = test_images.reshape((test_images.shape[0], 28, 28, 1)) / 255.0
os [27] # Augmentasi Data
       datagen = ImageDataGenerator(
           rotation_range=10,
            width_shift_range=0.1,
           height_shift_range=0.1,
            shear_range=0.1,
            zoom_range=0.1,
           horizontal_flip=True
        datagen.fit(train_images)
os [33] model = Sequential([
           Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)), # Convolutional layer dengan 32 filter 3x3 dan ReLU activation
           MaxPooling2D((2, 2)), # Max pooling layer dengan filter 2x2
           Conv2D(64, (3, 3), activation='relu'), # Convolutional layer dengan 64 filter 3x3 dan ReLU activation
           MaxPooling2D((2, 2)), # Max pooling layer dengan filter 2x2
           Flatten(), # Meratakan output dari lapisan sebelumnya menjadi vektor
           Dense(128, activation='relu'), # Fully connected layer dengan 128 unit dan ReLU activation
           Dense(10, activation='softmax') # Output layer dengan 10 unit untuk klasifikasi 10 kategori dan softmax activation
       ])
```



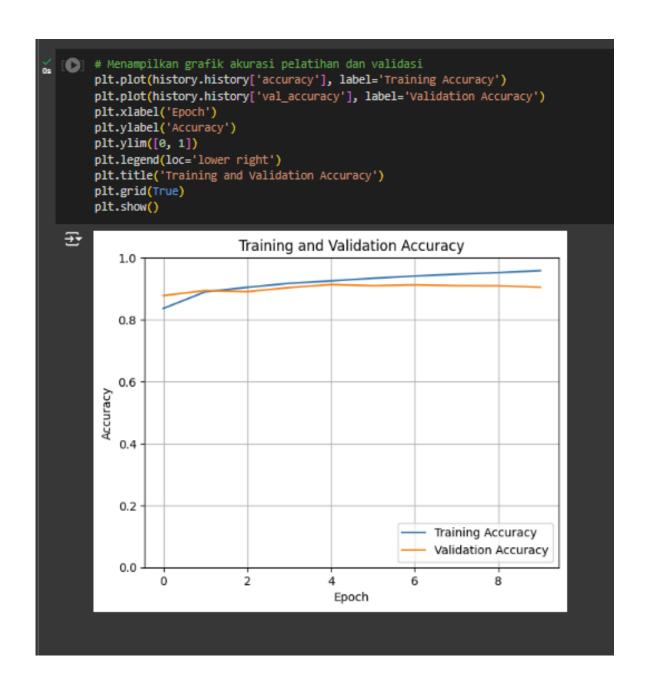
Compile, Train, dan Evaluasi Model

```
[34] # Compile model
 model.compile(optimizer='adam',
     loss='sparse_categorical_crossentropy',
     metrics=['accuracy'])
[35] # Train model
 history = model.fit(training_images, training_labels, epochs=10, validation_data=(test_images, test_labels))

→ Epoch 1/10
 Epoch 2/10
 Epoch 4/10
 Epoch 7/10
 Epoch 9/10
 [36] # Evaluasi Model
 test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)
 print(f'\nAkurasi pada data uji: {test_acc:.4f}')
₹ 313/313 - 2s - loss: 0.2885 - accuracy: 0.9045 - 2s/epoch - 8ms/step
 Akurasi pada data uji: 0.9045
```



• Grafik Akurasi





• Hasil

- Model dilatih dengan augmentasi data selama 10 epoch.
- Akurasi model pada data uji mencapai 90%, menunjukkan kemampuan yang baik dalam klasifikasi gambar fashion.
- Grafik akurasi pelatihan dan validasi menunjukkan model yang stabil dan tidak overfit.
- Prediksi pada beberapa gambar dari dataset pengujian menunjukkan bahwa model dapat mengklasifikasikan gambar dengan cukup akurat.



- CNN dengan augmentasi data, batch normalization, dan dropout dapat mencapai akurasi tinggi pada dataset Fashion MNIST.
- Teknik-teknik ini membantu dalam mencegah overfitting dan meningkatkan generalisasi model.
- Hasil ini menunjukkan bahwa model yang dibangun cukup kuat untuk tugas klasifikasi gambar fashion dan dapat digunakan dalam aplikasi dunia nyata.