UAS MACHINE LEARNING

IMAGE CLASSIFICATION MNIST ANGKA DATASET

Angelica Sharon Amelia Simanjuntak
(1103210032)

LATAR BELAKANG

Dataset MNIST adalah kumpulan gambar digit tulisan tangan yang populer dalam bidang machine learning dan computer vision, terdiri dari 70.000 gambar grayscale berukuran 28x28 piksel, dengan 60.000 gambar untuk pelatihan dan 10.000 untuk pengujian. Setiap gambar memiliki label angka (0-9). Dalam kode TensorFlow/Keras, gambar dinormalisasi dari nilai piksel 0-255 menjadi 0-1 dan label dikonversi menjadi one-hot encoding, sehingga memudahkan pelatihan model neural network dengan tujuan mencapai akurasi tinggi pada klasifikasi digit.

TUJUAN

Proyek ini bertujuan untuk membangun serta mengevaluasi klasifikasi yang dapat mengidentifikasi dari angka dengan nilai akurasi yang tinggi, Pada proyek ini diharapkan nilai akurasinya lebih dari 90%.

CNN MODEL

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan neural yang dirancang khusus untuk pemrosesan data grid seperti gambar. CNN terdiri dari lapisan-lapisan konvolusi yang menggunakan filter (kernel) untuk mengekstraksi fitur-fitur penting dari gambar, seperti tepi, tekstur, dan pola. Setelah lapisan konvolusi, lapisan pooling digunakan untuk mengurangi dimensi data dan mengendalikan overfitting.

MODEL SEQUENTIAL

model sequential merupakan model paling sederhana di Keras, yang memungkinkan kita untuk membangun model lapis demi lapis secara linear. Beberapa lapisan (layers) sequential yang disusun secara berurutan:

- Layers.Flatten

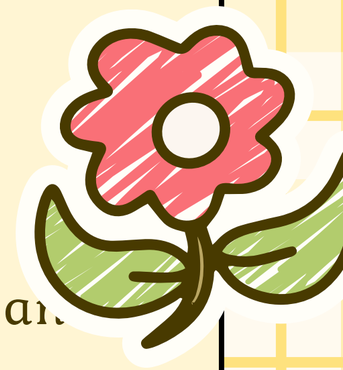
Layer Flatten meratakan matriks 2D ini menjadi vektor 1D sehingga bisa digunakan sebagai input untuk lapisan dense berikutnya.

- Layers.Dense

Lapisan ini memiliki 128 neuron. Setiap neuron di lapisan ini terhubung dengan setiap neuron di lapisan sebelumnya.

- Layers.Dropout

Dropout adalah teknik regulasi yang digunakan untuk mencegah overfitting.



CODE

```
[ ]  import tensorflow as tf
     from tensorflow.keras import layers, models
     import numpy as np
     import matplotlib.pyplot as plt
```

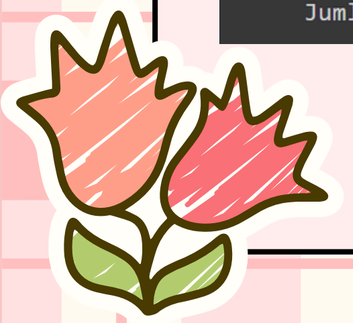
Mengimpor library TensorFlow, yang merupakan sebuah platform open-source untuk machine learning dan deep learning.

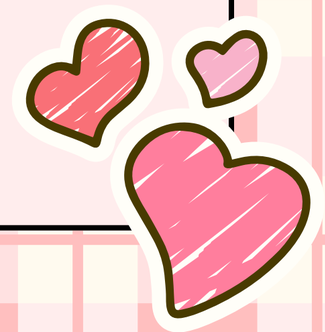
```
# Memuat dataset MNIST
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()

print("Jumlah data train: ", x_train.shape[0])
print("Jumlah data test: ", x_test.shape[0])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/t
11490434/11490434 [==============================] - 0s 0us/step
Jumlah data train:  60000
Jumlah data test:  10000
```

Memuat dan menampilkan jumlah data yang ada di data train dan data test.





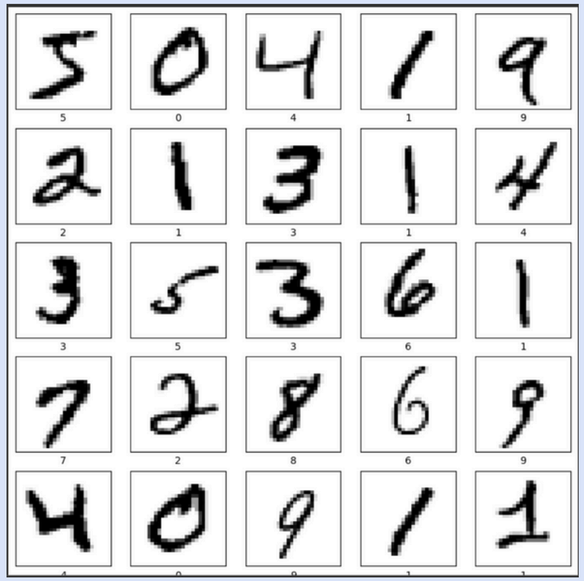
CODE

Menampilkan gambar dengan range 25, dengan masing-masing ukuran gambar yaitu 10x10

```python
# Menampilkan gambar dari dataset MNIST
def show_sample_images(x, y):
    plt.figure(figsize=(10, 10))
    for i in range(25):
        plt.subplot(5, 5, i + 1)
        plt.xticks([])
        plt.yticks([])
        plt.grid(False)
        plt.imshow(x[i], cmap=plt.cm.binary)
        plt.xlabel(y[i])
    plt.show()

show_sample_images(x_train, y_train)
```

OUTPUT



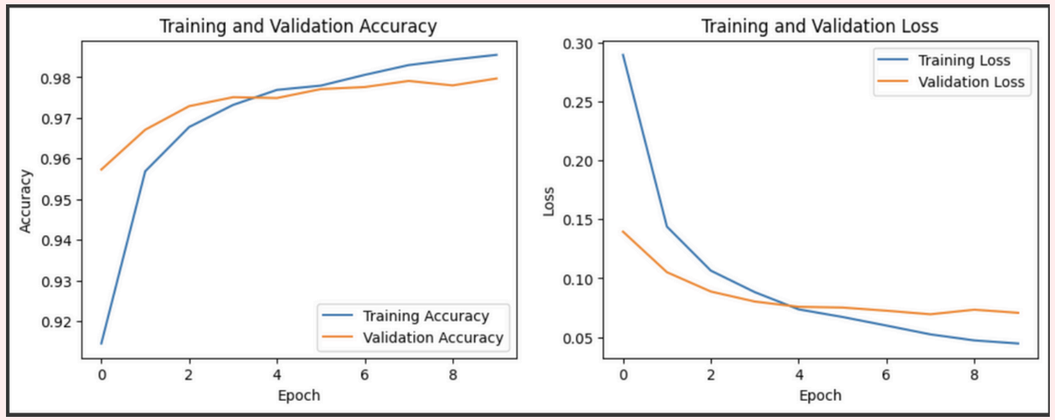
CODE

Pada proyek ini model dilatih dengan augmentasi data selama 10 epoch

```
[79] # 7. Melatih model dengan data training
     history = model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test))

⇄ Epoch 1/10
     1875/1875 [==============================] - 7s 3ms/step - loss: 0.2899 - accuracy: 0.9145 - val_loss: 0.1396 - val_accuracy: 0.9573
     Epoch 2/10
     1875/1875 [==============================] - 5s 3ms/step - loss: 0.1439 - accuracy: 0.9569 - val_loss: 0.1052 - val_accuracy: 0.9671
     Epoch 3/10
     1875/1875 [==============================] - 6s 3ms/step - loss: 0.1065 - accuracy: 0.9678 - val_loss: 0.0887 - val_accuracy: 0.9729
     Epoch 4/10
     1875/1875 [==============================] - 6s 3ms/step - loss: 0.0882 - accuracy: 0.9732 - val_loss: 0.0802 - val_accuracy: 0.9751
     Epoch 5/10
     1875/1875 [==============================] - 6s 3ms/step - loss: 0.0736 - accuracy: 0.9769 - val_loss: 0.0757 - val_accuracy: 0.9749
     Epoch 6/10
     1875/1875 [==============================] - 6s 3ms/step - loss: 0.0671 - accuracy: 0.9779 - val_loss: 0.0751 - val_accuracy: 0.9771
     Epoch 7/10
     1875/1875 [==============================] - 6s 3ms/step - loss: 0.0598 - accuracy: 0.9806 - val_loss: 0.0725 - val_accuracy: 0.9776
     Epoch 8/10
     1875/1875 [==============================] - 6s 3ms/step - loss: 0.0523 - accuracy: 0.9830 - val_loss: 0.0694 - val_accuracy: 0.9791
     Epoch 9/10
     1875/1875 [==============================] - 6s 3ms/step - loss: 0.0472 - accuracy: 0.9843 - val_loss: 0.0733 - val_accuracy: 0.9780
     Epoch 10/10
     1875/1875 [==============================] - 6s 3ms/step - loss: 0.0446 - accuracy: 0.9855 - val_loss: 0.0707 - val_accuracy: 0.9797
```

Test accuracy: 0.9805999994277954
Accuracy yang dihasilkan pada model ini adalah 98%



TERIMA KASIH