Nama:Andreas Hasiholan Sinaga

NIM : 1103213033

CNN Dengan Dataset Fashion Mnist

Fashion MNIST adalah dataset gambar yang sering digunakan untuk tugas klasifikasi dalam pembelajaran mesin, khususnya pada domain visi komputer. Dataset ini dikembangkan oleh Zalando sebagai alternatif untuk MNIST klasik, dengan fokus pada gambar artikel mode yang lebih kompleks. Fashion MNIST terdiri dari 70.000 gambar grayscale berukuran 28 × 28 piksel, yang mencakup 60.000 gambar untuk pelatihan dan 10.000 gambar untuk pengujian. Dataset ini terbagi ke dalam 10 kategori, seperti T-shirt/top, celana panjang, pullover, gaun, mantel, sandal, kemeja, sepatu olahraga, tas, dan sepatu bot, dengan setiap kategori dilabeli menggunakan angka 0 hingga 9.

```
# Pengaturan Hyperparameter dan Proses Training
kernel_sizes = [3, 5, 7]  # Pilihan ukuran kernel
epochs_options = [5, 50, 100 ]  # Jumlah epoch yang diuji
optimizers = [optim.SGD, optim.RMSprop, optim.Adam]  # Optimizer yang digunakan
early_stop_patience = 10  # Patokan early stopping
learning_rate = 0.001  # Tingkat pembelajaran (learning rate)

results = []
```

Pada proses pelatihan model CNN ini, terdapat beberapa hyperparameter utama yang diatur untuk mengoptimalkan performa model. **Ukuran kernel** diatur pada pilihan [3, 5, 7], yang menentukan dimensi filter dalam operasi konvolusi untuk mengekstraksi fitur dari gambar input. Ukuran kernel yang berbeda memungkinkan model menangkap pola visual dengan skala yang bervariasi. Selain itu, jumlah **epoch** yang diuji adalah [5, 50, 100], yang menentukan seberapa banyak iterasi penuh pelatihan dilakukan menggunakan dataset. Jumlah epoch yang lebih tinggi dapat meningkatkan kemampuan model untuk belajar, namun berisiko menyebabkan overfitting jika tidak dikontrol dengan baik.

Hyperparameter lainnya mencakup jenis **optimizer** yang digunakan, yaitu SGD, RMSprop, dan Adam, yang bertugas memperbarui bobot model secara efisien selama pelatihan. Untuk mencegah overfitting, diterapkan mekanisme **early stopping** dengan patience sebesar 10, yang akan menghentikan pelatihan lebih awal jika validation loss tidak membaik selama 10 epoch berturut-turut. Selain itu, **learning rate** diatur pada nilai 0.001 untuk mengontrol besarnya langkah dalam pembaruan bobot model. Hasil dari setiap kombinasi hyperparameter dicatat dalam variabel results, sehingga memungkinkan evaluasi performa dari berbagai konfigurasi.

```
Final Results:
Kernel: 3, Pooling: max, Optimizer: SGD, Final Train Loss: 5.0000, Final Val Loss: 0.3857
Kernel: 3, Pooling: max, Optimizer: SGD, Final Train Loss: 50.0000, Final Val Loss: 0.1239
Kernel: 3, Pooling: max, Optimizer: SGD, Final Train Loss: 100.0000, Final Val Loss: 0.1236
Kernel: 3, Pooling: max, Optimizer: SGD, Final Train Loss: 250.0000, Final Val Loss: 0.1525
Kernel: 3, Pooling: max, Optimizer: SGD, Final Train Loss: 350.0000, Final Val Loss: 0.1357
Kernel: 3, Pooling: max, Optimizer: RMSprop, Final Train Loss: 5.0000, Final Val Loss: 0.3434
Kernel: 3, Pooling: max, Optimizer: RMSprop, Final Train Loss: 50.0000, Final Val Loss: 0.1485
Kernel: 3, Pooling: max, Optimizer: RMSprop, Final Train Loss: 100.0000, Final Val Loss: 0.1525
Kernel: 3, Pooling: max, Optimizer: RMSprop, Final Train Loss: 250.0000, Final Val Loss: 0.1155
Kernel: 3, Pooling: max, Optimizer: RMSprop, Final Train Loss: 350.0000, Final Val Loss: 0.1216
Kernel: 3, Pooling: max, Optimizer: Adam, Final Train Loss: 5.0000, Final Val Loss: 0.2856
Kernel: 3, Pooling: max, Optimizer: Adam, Final Train Loss: 50.0000, Final Val Loss: 0.1837
Kernel: 3, Pooling: max, Optimizer: Adam, Final Train Loss: 100.0000, Final Val Loss: 0.1633
Kernel: 3, Pooling: max, Optimizer: Adam, Final Train Loss: 250.0000, Final Val Loss: 0.1704
Kernel: 3, Pooling: max, Optimizer: Adam, Final Train Loss: 350.0000, Final Val Loss: 0.2181
Kernel: 3, Pooling: avg, Optimizer: SGD, Final Train Loss: 5.0000, Final Val Loss: 0.5057
Kernel: 3, Pooling: avg, Optimizer: SGD, Final Train Loss: 50.0000, Final Val Loss: 0.2451
Kernel: 3, Pooling: avg, Optimizer: SGD, Final Train Loss: 100.0000, Final Val Loss: 0.1641
Kernel: 3, Pooling: avg, Optimizer: SGD, Final Train Loss: 250.0000, Final Val Loss: 0.1404
Kernel: 3, Pooling: avg, Optimizer: SGD, Final Train Loss: 350.0000, Final Val Loss: 0.1792
Kernel: 3, Pooling: avg, Optimizer: RMSprop, Final Train Loss: 5.0000, Final Val Loss: 0.3096
Kernel: 3, Pooling: avg, Optimizer: RMSprop, Final Train Loss: 50.0000, Final Val Loss: 0.1128
Kernel: 3, Pooling: avg, Optimizer: RMSprop, Final Train Loss: 100.0000, Final Val Loss: 0.0935
Kernel: 3, Pooling: avg, Optimizer: RMSprop, Final Train Loss: 250.0000, Final Val Loss: 0.1206
Kernel: 3, Pooling: avg, Optimizer: RMSprop, Final Train Loss: 350.0000, Final Val Loss: 0.0874
Kernel: 3, Pooling: avg, Optimizer: Adam, Final Train Loss: 5.0000, Final Val Loss: 0.3542
Kernel: 3, Pooling: avg, Optimizer: Adam, Final Train Loss: 50.0000, Final Val Loss: 2.3026
Kernel: 3, Pooling: avg, Optimizer: Adam, Final Train Loss: 100.0000, Final Val Loss: 0.0794
Kernel: 3, Pooling: avg, Optimizer: Adam, Final Train Loss: 250.0000, Final Val Loss: 0.0778
Kernel: 3, Pooling: avg, Optimizer: Adam, Final Train Loss: 350.0000, Final Val Loss: 0.0941
Kernel: 5, Pooling: max, Optimizer: SGD, Final Train Loss: 5.0000, Final Val Loss: 0.3907
Kernel: 5, Pooling: max, Optimizer: SGD, Final Train Loss: 50.0000, Final Val Loss: 0.1217
Kernel: 5, Pooling: max, Optimizer: SGD, Final Train Loss: 100.0000, Final Val Loss: 0.1175
Kernel: 5, Pooling: max, Optimizer: SGD, Final Train Loss: 250.0000, Final Val Loss: 0.1035
Kernel: 5, Pooling: max, Optimizer: SGD, Final Train Loss: 350.0000, Final Val Loss: 0.1264
Kernel: 5, Pooling: max, Optimizer: RMSprop, Final Train Loss: 5.0000, Final Val Loss: 0.4346
Kernel: 5, Pooling: max, Optimizer: RMSprop, Final Train Loss: 50.0000, Final Val Loss: 0.1760
Kernel: 5, Pooling: max, Optimizer: RMSprop, Final Train Loss: 100.0000, Final Val Loss: 0.1730
Kernel: 5, Pooling: max, Optimizer: RMSprop, Final Train Loss: 250.0000, Final Val Loss: 0.2209
Kernel: 5, Pooling: max, Optimizer: RMSprop, Final Train Loss: 350.0000, Final Val Loss: 0.1788
Kernel: 5, Pooling: max, Optimizer: Adam, Final Train Loss: 5.0000, Final Val Loss: 0.4116
Kernel: 5, Pooling: max, Optimizer: Adam, Final Train Loss: 50.0000, Final Val Loss: 0.2341
Kernel: 5, Pooling: max, Optimizer: Adam, Final Train Loss: 100.0000, Final Val Loss: 0.1546
Kernel: 5, Pooling: max, Optimizer: Adam, Final Train Loss: 250.0000, Final Val Loss: 0.1970
Kernel: 5, Pooling: max, Optimizer: Adam, Final Train Loss: 350.0000, Final Val Loss: 0.1271
Kernel: 5, Pooling: avg, Optimizer: SGD, Final Train Loss: 5.0000, Final Val Loss: 0.4796
Kernel: 5, Pooling: avg, Optimizer: SGD, Final Train Loss: 50.0000, Final Val Loss: 0.2144
Kernel: 5, Pooling: avg, Optimizer: SGD, Final Train Loss: 100.0000, Final Val Loss: 0.1696
```

Hasil eksperimen model CNN menggunakan dataset Fashion MNIST menunjukkan bahwa pemilihan parameter seperti ukuran kernel, jenis pooling, dan optimizer memiliki pengaruh yang signifikan terhadap performa model. Dalam eksperimen ini, ukuran kernel yang digunakan adalah 3, 5, dan 7. Kernel ukuran besar, seperti 7, cenderung memberikan hasil yang lebih baik karena mampu menangkap pola visual yang lebih kompleks dari data. Namun, kernel yang lebih kecil, seperti 3 dan 5, juga menunjukkan performa yang kompetitif pada beberapa konfigurasi, khususnya saat dikombinasikan dengan max pooling dan optimizer RMSprop atau Adam.

Dua jenis pooling yang diuji adalah max pooling dan average pooling. Max pooling memberikan hasil yang lebih stabil dengan nilai validation loss yang rendah, terutama untuk kernel berukuran kecil hingga sedang. Sementara itu, average pooling memberikan variasi

performa yang lebih besar. Namun, pada kombinasi tertentu, seperti kernel 7 dengan optimizer Adam, average pooling menghasilkan validation loss paling rendah. Hal ini menunjukkan bahwa pemilihan jenis pooling perlu disesuaikan dengan kombinasi parameter lainnya untuk mencapai hasil optimal.

Optimizer yang digunakan dalam eksperimen ini adalah SGD, RMSprop, dan Adam. Dari ketiga optimizer tersebut, Adam memberikan performa terbaik, terutama saat dikombinasikan dengan kernel 7 dan average pooling, menghasilkan nilai validation loss terendah sebesar **0.0778**. RMSprop juga memberikan hasil yang baik pada beberapa konfigurasi, khususnya dengan average pooling dan kernel kecil. Berdasarkan hasil eksperimen, konfigurasi terbaik untuk model CNN pada dataset Fashion MNIST adalah kernel **7**, pooling **average**, dan optimizer **Adam.**