

PENERAPAN DEEP LEARNING UNTUK KLASIFIKASI GAMBAR CIFAR-10

ARTIFICIAL INTELLIGENCE



Dosen Pengampu :

Yuyun Umaidah, M.kom

Disusun Oleh :

Moh. Dicky Purnama

2210631170031

Fabian Nur Rohman

2210631170118

KELAS D

TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SINGAPERBANGSA KARAWANG 2024

Pendahuluan

Pada tugas ini, kita akan menggunakan metode deep learning untuk mengklasifikasikan gambar dari dataset CIFAR-10. Dataset CIFAR-10 terdiri dari 60.000 gambar berwarna dalam 10 kelas yang berbeda, masing-masing berukuran 32x32 piksel. Dengan menggunakan jaringan konvolusi (Convolutional Neural Network atau CNN), kita akan melatih model untuk mengenali dan mengklasifikasikan gambar-gambar ini dengan akurasi tinggi. CNN dipilih karena kemampuannya dalam menangkap pola dan fitur pada gambar.

Tujuan

1. Membangun dan melatih model CNN untuk klasifikasi gambar pada dataset CIFAR-10.
2. Mengevaluasi performa model menggunakan metrik akurasi.
3. Visualisasi hasil pelatihan dan evaluasi model.
4. Menyediakan analisis dan interpretasi hasil.

Persiapan Data :

1. Import Library

Pertama, kita akan mengimpor pustaka yang diperlukan untuk pelatihan model, pemrosesan data, dan visualisasi.

```
# Import libraries
import tensorflow as tf
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
```

2. Memuat dan Normalisasi Dataset

Dataset CIFAR-10 diunduh dan dibagi menjadi data latih dan data uji. Kemudian, kita melakukan normalisasi pada data gambar agar memiliki nilai antara 0 dan 1, serta mengubah label menjadi bentuk one-hot encoding.

```
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# Normalize the image data
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Convert labels to categorical (one-hot encoding)
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)
```

3. Visualisasi Data

Menampilkan beberapa gambar dari dataset untuk memahami jenis gambar yang ada dalam CIFAR-10. Ini membantu kita untuk mengetahui distribusi dan variasi gambar dalam setiap kelas.

```
for i in range(9):
    plt.subplot(330 + 1 + i)
    plt.imshow(x_train[i])
plt.show()
```

Arsitektur Model :

4. Membangun Arsitektur Model

Model CNN dibangun dengan beberapa lapisan konvolusi, pooling, dan dense untuk menangkap fitur dari gambar dan melakukan klasifikasi. Berikut adalah penjelasan detail dari tiap lapisan dalam model:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPooling2D, Flatten, Dropout

# Define the model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    Flatten(),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])
```

5. Kompilasi Model

Model dikompilasi dengan optimizer Adam dan loss function categorical crossentropy. Metode evaluasi yang digunakan adalah akurasi.

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

6. Menampilkan Arsitektur Model

Ringkasan arsitektur model ditampilkan untuk memahami jumlah parameter dan struktur lapisan.

```
model.summary()
```

Pelatihan Model :

7. Melatih Model

Model dilatih menggunakan data latih selama 10 epoch dengan batch size sebesar 64. Validasi dilakukan menggunakan data uji.

```
history = model.fit(x_train, y_train, epochs=10, batch_size=64, validation_data=(x_test, y_test))
```

8. Visualisasi Performa Pelatihan

Grafik akurasi dan loss selama pelatihan dan validasi ditampilkan untuk memahami performa model. Visualisasi ini penting untuk memonitor apakah model mengalami overfitting atau underfitting.

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

Evaluasi dan Prediksi :

9. Evaluasi Model

Model dievaluasi menggunakan data uji untuk mendapatkan akurasi akhir.

```
test_loss, test_acc = model.evaluate(x_test, y_test, verbose=2)
print(f'Test accuracy: {test_acc}')
```

10. Prediksi dan Visualisasi Hasil

Model melakukan prediksi pada data uji dan hasilnya divisualisasikan untuk beberapa gambar. Ini membantu untuk memeriksa hasil prediksi secara visual dan memahami kinerja model pada gambar individu.

```
predictions = model.predict(x_test)

# Show a few prediction results
import numpy as np

# Function to display image with predicted and true label
def display_prediction(index):
    plt.imshow(x_test[index])
    plt.title(f"True: {np.argmax(y_test[index])}, Predicted: {np.argmax(predictions[index])}")
    plt.show()

# Display predictions for first 5 test images
for i in range(5):
    display_prediction(i)
```

Hasil dan Analisis :

Akurasi Model

- **Akurasi Pelatihan:** Model menunjukkan peningkatan akurasi yang konsisten selama pelatihan, mencapai akurasi pelatihan sekitar 0.8.
- **Akurasi Validasi:** Akurasi pada dataset uji (validasi) mencapai sekitar 0.7 setelah 10 epoch pelatihan. Ini menunjukkan bahwa model dapat mengenali gambar dengan baik, namun masih ada ruang untuk perbaikan lebih lanjut.

Grafik Performa

- **Akurasi:** Grafik akurasi menunjukkan bahwa model tidak mengalami overfitting yang signifikan, dengan kurva akurasi pelatihan dan validasi yang meningkat secara paralel.
- **Loss:** Grafik loss menunjukkan penurunan yang stabil pada nilai loss selama pelatihan, baik pada data pelatihan maupun data validasi.

Prediksi

Hasil prediksi untuk beberapa gambar uji menunjukkan bahwa model mampu mengklasifikasikan gambar dengan benar, meskipun terdapat beberapa kesalahan prediksi. Ini adalah beberapa contoh hasil prediksi yang divisualisasikan:

- **Gambar 1:** True label: 3, Predicted: 3
- **Gambar 2:** True label: 8, Predicted: 8
- **Gambar 3:** True label: 1, Predicted: 1
- **Gambar 4:** True label: 0, Predicted: 0
- **Gambar 5:** True label: 6, Predicted: 6

Kesimpulan

Model CNN yang dibangun menunjukkan performa yang baik dalam mengklasifikasikan gambar CIFAR-10 dengan akurasi uji sekitar 70%. Proses normalisasi data, arsitektur model yang efektif, serta visualisasi hasil merupakan komponen penting dalam analisis ini. Untuk meningkatkan performa lebih lanjut, kita dapat mempertimbangkan optimalisasi hyperparameter, penggunaan teknik augmentasi data, atau penerapan arsitektur model yang lebih kompleks seperti ResNet atau VGG.

Rekomendasi dan Peningkatan

1. **Optimalisasi Hyperparameter:** Melakukan pencarian hyperparameter menggunakan Grid Search atau Random Search untuk menemukan kombinasi terbaik dari parameter seperti learning rate, batch size, dan jumlah lapisan.
2. **Augmentasi Data:** Menggunakan teknik augmentasi data seperti rotasi, translasi, dan flipping untuk meningkatkan jumlah dan keragaman data latih.
3. **Arsitektur Model yang Lebih Kompleks:** Menerapkan arsitektur model yang lebih canggih seperti ResNet, VGG, atau EfficientNet untuk meningkatkan akurasi.
4. **Penggunaan Transfer Learning:** Menggunakan model pre-trained dan melakukan fine-tuning untuk meningkatkan performa klasifikasi.

Dengan pemahaman yang lebih dalam tentang data dan teknik deep learning, kita dapat terus meningkatkan kemampuan model dalam tugas klasifikasi gambar dan aplikasi lainnya di masa depan.