

Pendeteksian Gambar Model Sepatu Menggunakan Model Convolutional Neural Network

Muhammad Varendra, NPM 2206052673

FMIPA, Mathematics

December 20, 2024

Abstract

Convolutional Neural Networks (CNN) dapat dilihat sebagai kerangka kerja yang mereplikasi cara otak manusia mengenali pola visual dalam data. Dengan memanfaatkan hierarki lapisan, CNN mengabstraksi fitur dari representasi sederhana ke kompleks, seperti mendeteksi tepi pada gambar hingga memahami objek secara keseluruhan. Pendekatan ini memungkinkan CNN tidak hanya mengenali pola dalam gambar, tetapi juga beradaptasi untuk berbagai jenis data grid lainnya, seperti suara atau teks berstruktur. CNN, pada intinya, adalah upaya abstraksi sistematis untuk mengubah data mentah menjadi wawasan bermakna melalui proses belajar otomatis.

1 Introduction

Convolutional Neural Networks (CNN) telah merevolusi bidang kecerdasan buatan, khususnya dalam tugas-tugas yang melibatkan analisis data visual dan spasial. Terinspirasi dari cara kerja korteks visual manusia, CNN unggul dalam mengenali pola dan struktur pada data, seperti gambar, dengan memanfaatkan ekstraksi fitur secara hierarkis. Pendekatan ini memungkinkan CNN memproses data kompleks dengan sedikit kebutuhan rekayasa fitur secara manual. Selama bertahun-tahun, CNN telah menjadi elemen penting dalam berbagai aplikasi, seperti pengenalan citra, deteksi objek, analisis video, hingga pencitraan medis, yang mendorong inovasi di berbagai industri. Pendahuluan ini akan membahas konsep dasar, arsitektur, dan dampak transformasi CNN dalam menyelesaikan permasalahan dunia nyata.

2 Literature Review

2.1 Data Preprocessing

Preprocessing data merupakan tahap penting dalam penerapan Convolutional Neural Networks (CNN), karena kualitas data masukan sangat memengaruhi kinerja model. Proses ini bertujuan untuk membersihkan, menyiapkan, dan mengoptimalkan data sebelum dimasukkan ke dalam jaringan. Dalam konteks CNN, preprocessing melibatkan berbagai langkah, seperti normalisasi piksel untuk memastikan nilai input berada dalam rentang tertentu, augmentasi data untuk memperluas variasi dataset, serta penghapusan noise untuk meningkatkan kejelasan fitur. Teknik lainnya meliputi resizing gambar untuk menyamakan dimensi input dan konversi format warna, seperti dari RGB ke grayscale jika diperlukan. Dengan preprocessing yang baik, CNN dapat lebih efisien dalam memahami pola pada data, mengurangi risiko overfitting, dan meningkatkan generalisasi model terhadap data baru.

2.2 Introduction to Dataset and Visualization

Visualisasi data merupakan langkah penting dalam proses analisis dan pengembangan model machine learning, khususnya dalam proyek yang melibatkan data citra. Salah satu teknik visualisasi yang umum adalah dengan menampilkan gambar dari dataset menggunakan fungsi plotting, seperti `plt.imshow()` dari library Matplotlib. Teknik ini memungkinkan peneliti untuk memeriksa kualitas data, mengidentifikasi pola, serta memastikan format data sudah sesuai sebelum digunakan dalam pelatihan model. Dalam studi ini, setiap gambar dalam dataset divisualisasikan dalam

skala abu-abu (grayscale) untuk menyederhanakan analisis intensitas piksel. Proses ini tidak hanya membantu memahami karakteristik dataset, tetapi juga menjadi langkah awal yang krusial untuk menentukan langkah preprocessing dan augmentasi data yang diperlukan.

2.3 Normalisasi Data

Dalam proses preprocessing dataset untuk model deep learning, langkah pertama adalah mendefinisikan daftar nama kelas yang merepresentasikan kategori dalam data. Sebagai contoh, variabel class names berisi nama-nama kelas seperti 'T-shirt/top', 'Trouser', hingga 'Ankle boot', yang sesuai dengan dataset Fashion MNIST yang terdiri dari 10 kategori pakaian dan alas kaki. Langkah berikutnya adalah normalisasi nilai piksel gambar. Normalisasi dilakukan dengan membagi setiap nilai piksel dalam dataset train images dan test images dengan 255.0, sehingga nilai piksel yang awalnya berada dalam rentang 0 hingga 255 menjadi 0 hingga 1. Proses ini penting untuk meningkatkan efisiensi dan stabilitas komputasi, karena model deep learning lebih mudah beroperasi dengan data berskala kecil. Selain itu, normalisasi juga membantu mempercepat konvergensi model selama pelatihan serta meningkatkan generalisasi model terhadap data baru. Dengan demikian, kedua langkah ini merupakan tahap awal yang krusial dalam memastikan kualitas data untuk pelatihan model klasifikasi citra.

2.4 Reshape Images

Kode ini bertujuan untuk mengubah dimensi gambar dalam dataset pelatihan (trainimages) dan pengujian (testimages) agar sesuai dengan format yang dibutuhkan oleh model deep learning, khususnya Convolutional Neural Networks (CNN). Dalam hal ini, gambar yang awalnya memiliki dimensi dua dimensi (28x28 piksel) perlu diubah menjadi format 4D dengan menambahkan dimensi saluran (channel). Fungsi reshape() digunakan untuk mengubah bentuk data, di mana trainimages.reshape((trainimages.shape[0], 28, 28, 1)) dan testimages.reshape((testimages.shape[0], 28, 28, 1)) mengatur agar setiap gambar memiliki ukuran 28x28 piksel dan satu saluran (grayscale). Format ini penting karena model CNN mengharapkan input data dalam bentuk 4D, yang terdiri dari jumlah gambar, tinggi, lebar, dan jumlah saluran. Dengan demikian, langkah ini memastikan data gambar dapat diproses dengan benar oleh model dan memfasilitasi ekstraksi fitur spasial yang efisien selama pelatihan.

3 Methodology

3.1 Library Installation

Kode ini membangun arsitektur model Convolutional Neural Network (CNN) menggunakan Keras, yang terdiri dari beberapa lapisan konvolusional dan lapisan dense untuk klasifikasi gambar. Pertama, model dimulai dengan lapisan konvolusi pertama Conv2D yang memiliki 32 filter dengan ukuran kernel 3x3 dan fungsi aktivasi ReLU, yang menerima input gambar berukuran 28x28 piksel dengan satu saluran (grayscale). Setelah itu, lapisan max-pooling MaxPooling2D dengan ukuran pool 2x2 diterapkan untuk mengurangi dimensi gambar, mempertahankan fitur penting, dan mengurangi kompleksitas komputasi. Selanjutnya, ada dua lapisan konvolusi tambahan dengan 64 filter masing-masing, juga diikuti dengan lapisan max-pooling, untuk semakin mengekstraksi fitur kompleks dari gambar. Setelah bagian konvolusional, model melanjutkan dengan lapisan Flatten untuk mengubah data 2D menjadi vektor 1D, yang kemudian dilanjutkan dengan dua lapisan dense. Lapisan pertama memiliki 64 neuron dengan aktivasi ReLU, sementara lapisan terakhir terdiri dari 10 neuron tanpa fungsi aktivasi, yang sesuai dengan jumlah kelas untuk klasifikasi (misalnya, 10 kelas dalam dataset Fashion MNIST). Model ini siap untuk dilatih dan digunakan untuk tugas klasifikasi gambar.

3.2 Train Model

Kode ini digunakan untuk menyusun dan mempersiapkan model CNN yang telah dibangun untuk pelatihan. Pertama, model dikompilasi menggunakan optimizer adam, yang merupakan algoritma optimisasi populer untuk pelatihan deep learning karena kemampuannya dalam menyesuaikan pembelajaran berdasarkan gradien dengan cara yang efisien. Selanjutnya, loss function yang digunakan adalah SparseCategoricalCrossentropy, yang cocok untuk masalah klasifikasi multi-kelas di mana label target diberikan dalam bentuk integer (bukan one-hot encoding). Fungsi loss ini mengukur perbedaan antara label yang diprediksi oleh model dan label sebenarnya, dan model

akan berusaha meminimalkan nilai loss tersebut selama pelatihan. Parameter `from_logits=True` menunjukkan bahwa output model belum melalui fungsi aktivasi softmax, sehingga kita perlu menerapkan softmax untuk mengubah output menjadi probabilitas. Terakhir, metrik yang digunakan untuk mengevaluasi performa model adalah accuracy, yang mengukur persentase prediksi yang benar dibandingkan dengan total data. Setelah langkah ini, model siap untuk dilatih menggunakan dataset yang telah disiapkan.

Proses ini digunakan untuk melatih model Convolutional Neural Network (CNN) yang telah dibangun dan dikompilasi sebelumnya. Selama pelatihan, model akan melalui beberapa iterasi (disebut epoch) di mana setiap epoch melibatkan pemrosesan seluruh dataset pelatihan. Pada setiap iterasi, model berusaha menyesuaikan bobot-bobotnya untuk meminimalkan perbedaan antara prediksi dan label yang sebenarnya, dengan tujuan untuk meningkatkan akurasi. Selain itu, data pengujian digunakan untuk memvalidasi kinerja model selama pelatihan, yang memungkinkan untuk mengevaluasi seberapa baik model mampu menggeneralisasi data yang belum pernah dilihat sebelumnya. Proses pelatihan ini bertujuan untuk meminimalkan loss function dan meningkatkan metrik accuracy, yang mengukur seberapa sering model memberikan prediksi yang benar. Hasil dari pelatihan ini memberikan gambaran tentang seberapa baik model bekerja, baik pada data pelatihan maupun data pengujian.

3.3 *Evaluasi Model*

Hasil yang ditunjukkan merupakan ringkasan dari pelatihan model Convolutional Neural Network (CNN) selama 5 epoch. Pada setiap epoch, terdapat dua metrik utama yang ditampilkan: accuracy (akurasi) dan loss untuk data pelatihan dan data validasi (pengujian).

Epoch 1: Pada epoch pertama, model mulai dengan akurasi 74.83

Epoch 2: Di epoch kedua, akurasi pelatihan meningkat menjadi 87.76

Epoch 3: Pada epoch ketiga, model mencapai akurasi pelatihan 89.67

Epoch 4: Di epoch keempat, akurasi pelatihan mencapai 90.88

Epoch 5: Pada epoch terakhir, model memperoleh akurasi pelatihan tertinggi 91.72

Secara keseluruhan, hasil ini menunjukkan bahwa model semakin baik dalam mempelajari pola dari data pelatihan, dan akurasi validasi yang stabil menunjukkan bahwa model dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

3.4 *Plot Training and Validation Accuracy Values*

Visualisasi ini digunakan untuk memantau kinerja model selama pelatihan dengan membandingkan akurasi dan loss pada data pelatihan (train) dan data validasi (test) di setiap epoch.

Plot Akurasi: Grafik pertama menunjukkan bagaimana akurasi model meningkat seiring bertambahnya jumlah epoch, baik pada data pelatihan maupun data validasi. Akurasi menggambarkan seberapa sering model memberikan prediksi yang benar, dan grafik ini membantu kita melihat apakah model terus meningkat dalam kemampuannya untuk mengklasifikasikan data yang benar. Jika garis untuk data validasi menurun sementara akurasi pelatihan meningkat, itu mungkin menandakan overfitting.

Plot Loss: Grafik kedua menunjukkan bagaimana loss model berkurang seiring berjalannya waktu. Loss mengukur kesalahan model dalam prediksi, jadi penurunan nilai loss menunjukkan bahwa model semakin akurat dalam membuat prediksi. Jika loss data validasi lebih tinggi dari loss data pelatihan, itu bisa menunjukkan bahwa model tidak menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

Kedua grafik ini memberikan gambaran yang jelas mengenai proses pelatihan model, termasuk bagaimana model berkembang, apakah mengalami overfitting atau underfitting, dan seberapa baik model dalam menggeneralisasi data baru.

4 Results and Final Conclusion

Model CNN yang telah dibangun untuk mendeteksi gambar menunjukkan kinerja yang sangat baik sepanjang proses pelatihan. Selama 5 epoch, model menunjukkan peningkatan yang konsisten baik dari sisi akurasi maupun loss. Pada awalnya, akurasi pelatihan berada di angka 74.83

Selain itu, meskipun akurasi terus meningkat, loss juga menunjukkan penurunan yang signifikan. Loss pelatihan dimulai pada 0.6959 dan berakhir di 0.2211, sementara loss pada data validasi juga menurun dari 0.3658 menjadi 0.2816. Penurunan ini mengindikasikan bahwa model semakin efisien dalam meminimalkan kesalahan selama proses pelatihan dan validasi.

Secara keseluruhan, model ini menunjukkan kemampuan yang baik dalam mengenali gambar dengan berbagai kelas yang ditentukan, seperti T-shirt, trousers, dress, dan lainnya. Proses pelatihan berjalan dengan stabil, tanpa tanda-tanda overfitting, karena akurasi pada data pelatihan dan validasi cukup seimbang. Dengan hasil ini, model CNN dapat dikatakan siap untuk diterapkan dalam deteksi gambar dengan tingkat akurasi yang cukup tinggi.

5 References

- LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
- Keras Documentation (2024). Keras: The Python deep learning library.
- He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 770-778.
- Google Colab Documentation. (2023).
- Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*.