

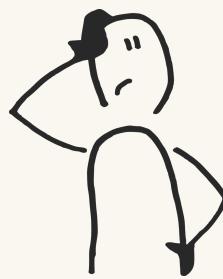
KEYLA'S PROJECT

Deep Learning Image Denoising Project

NAME

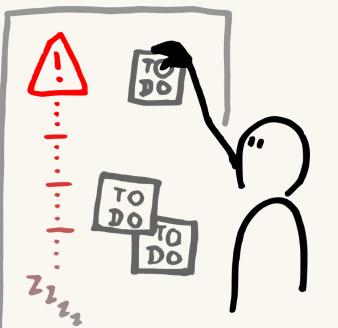
Keyla Faristha Rindani

Project Overview



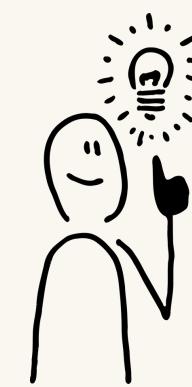
Problem

Gambar sering punya noise (bintik-bintik) yang bikin gambarnya buram dan susah diproses. Dataset ber-noise juga jarang tersedia, jadi model sulit belajar cara membersihkannya.



Solution

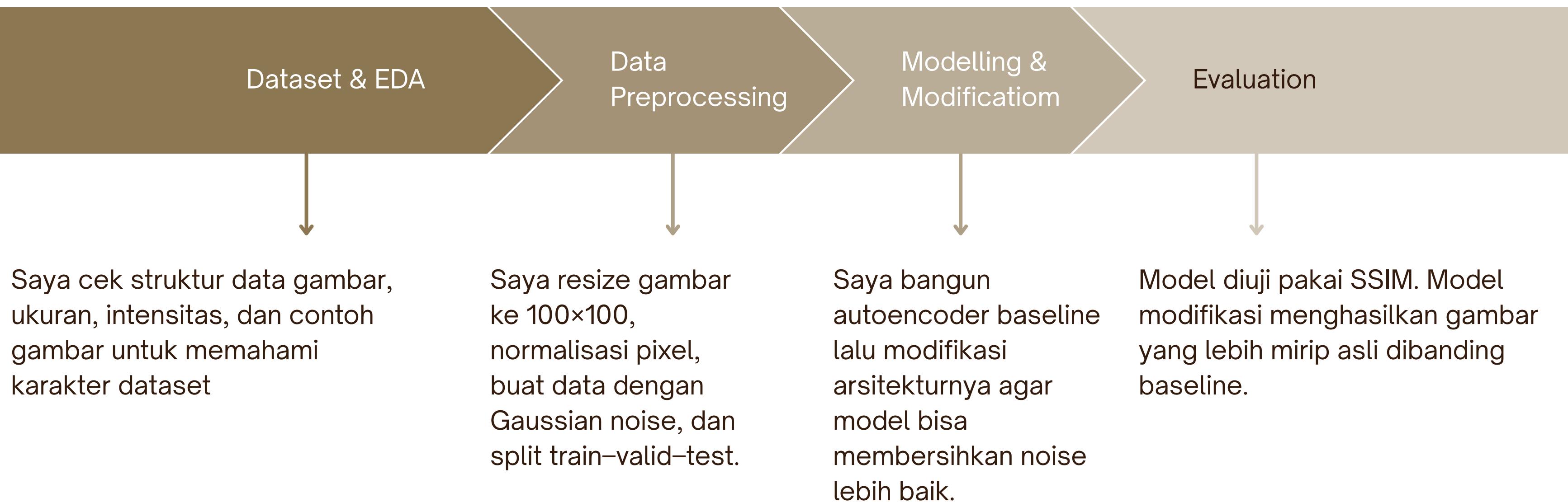
Saya membuat autoencoder yang bisa membersihkan gambar secara otomatis dan saya juga tambahkan Gaussian noise ke gambar, lalu latih model untuk belajar membedakan mana noise dan mana gambar asli.



Why It Matters

Gambar yang lebih bersih bikin analisis visual jadi lebih akurat, bisa berguna untuk kamera, CCTV, medis, atau model AI lain yang butuh input berkualitas.

Workflow



Dataset & EDA

- Sumber dataset: Google Drive dari dosen (folder A/B sesuai NIM).
- Jumlah data: ±1.074 gambar.
- Format: JPEG, RGB, ukuran 600×600 pixel.

Karakteristik gambar:

- Intensitas cenderung gelap
- Variasi kontras antar gambar cukup tinggi
- Setelah normalisasi berada pada rentang [0–1]

Masalah data:

- Tidak ada gambar ber-noise (harus dibuat manual)
- Perbedaan intensitas cukup signifikan sehingga perlu normalisasi



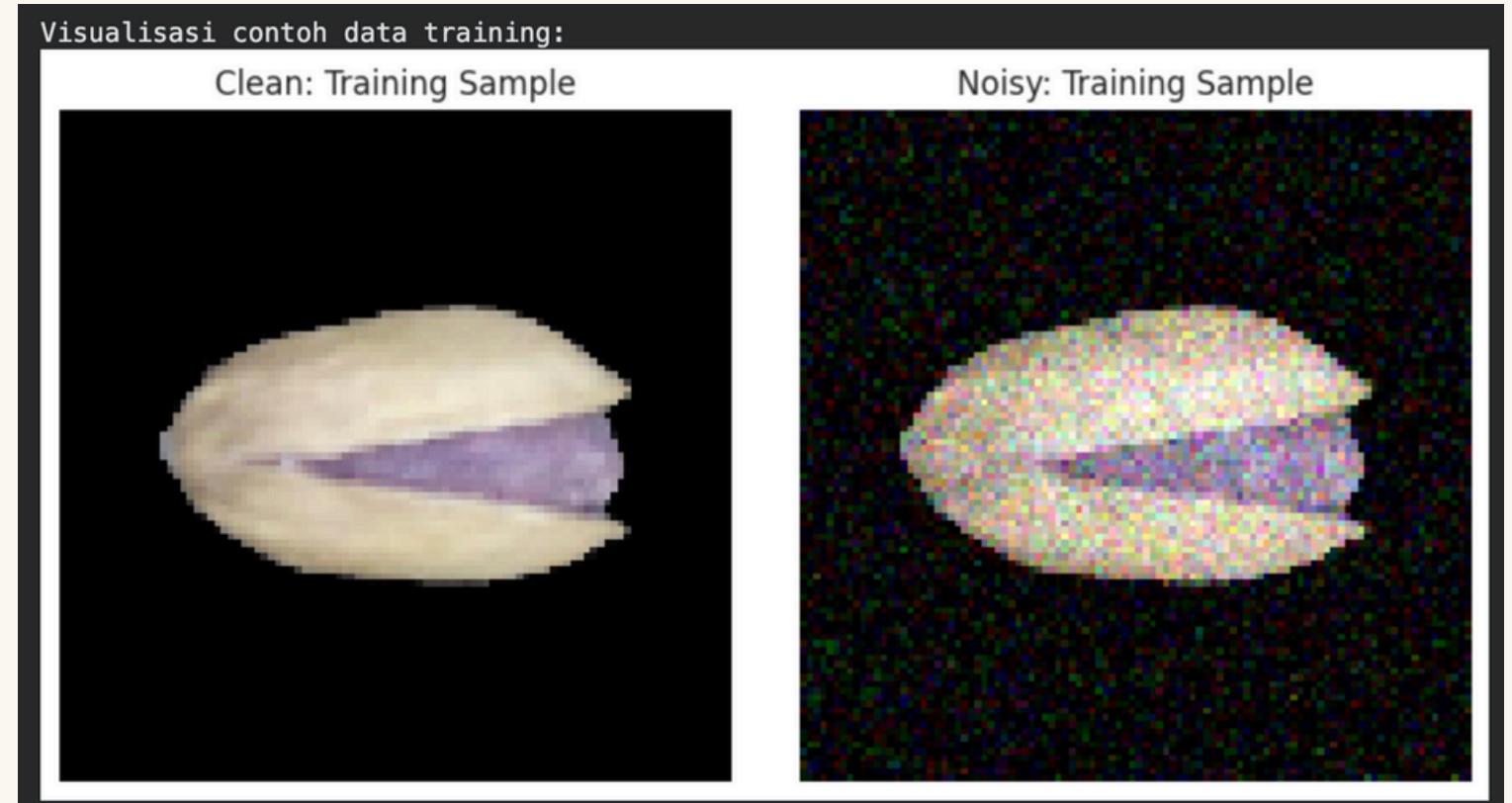
Data Preprocessing

Langkah-langkah preprocessing:

- Resize gambar: $600 \times 600 \rightarrow 100 \times 100$ pixel (agar komputasi lebih ringan)
- Normalisasi pixel ke rentang 0–1
- Membuat data noisy image menggunakan Gaussian noise (mean 0.0, std 0.1)
- Membagi dataset:
 - 80% training
 - 10% validation
 - 10% testing

Pembagian dataset:

- Training: 858 gambar (79.9%)
- Validation: 108 gambar (10.1%)
- Testing: 108 gambar (10.1%)



Modeling & Modification

Model: **Convolutional Autoencoder**

Kenapa memilih model ini?

- Karena autoencoder cocok untuk tugas rekonstruksi gambar dan secara natural bisa membuang noise saat encoding.

Arsitektur singkat:

- Encoder: beberapa layer Conv2D (kernel 3×3, ReLU) untuk menangkap pola penting
- Bottleneck: representasi kecil berisi fitur inti
- Decoder: Upsampling + Conv2D untuk mengembalikan ke ukuran asli
- Output: Sigmoid (karena pixel 0–1)
- Optimizer: Adam
- Loss function: MSE

Model baseline kemudian dimodifikasi dengan menambah depth/filter agar hasil denoising lebih halus.



Baseline Model

After Modification



Evaluasi / Hasil

Metrik utama: SSIM (Structural Similarity Index)

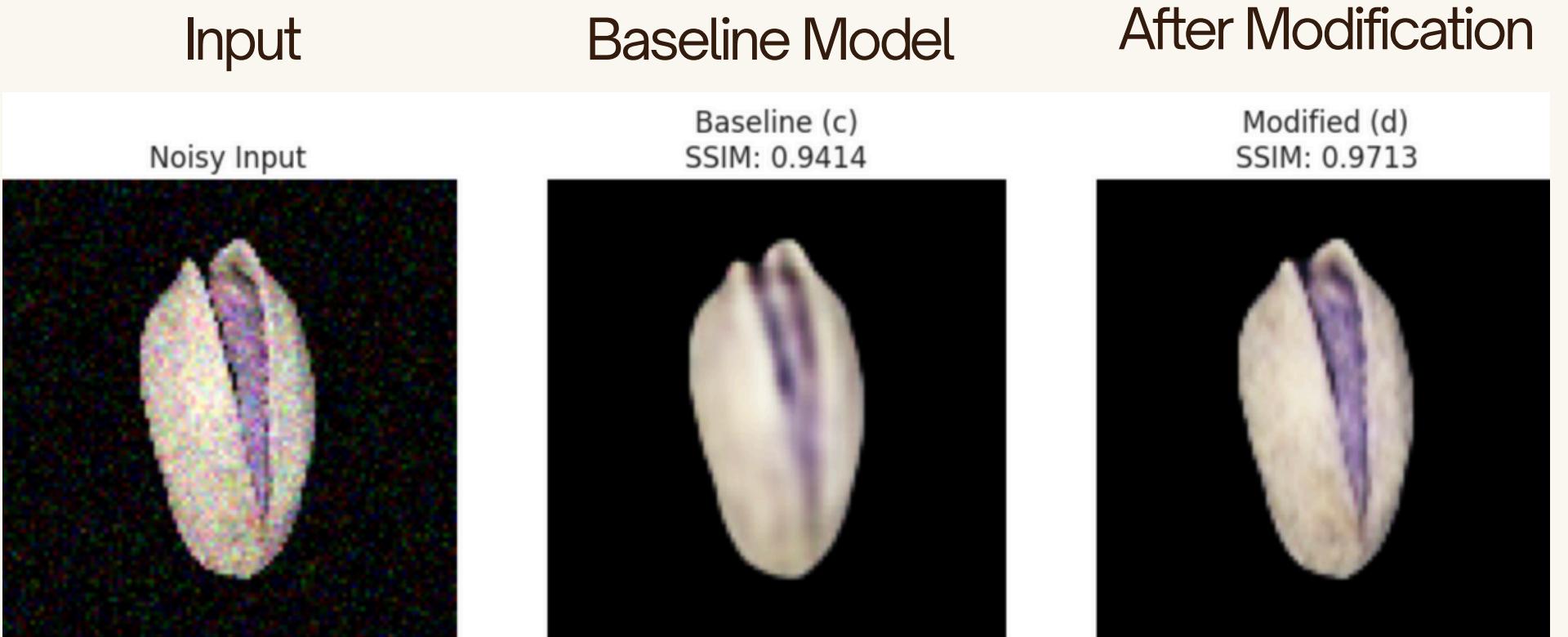
- SSIM mengukur kemiripan struktur gambar

Hasil evaluasi:

- Model baseline → SSIM lebih rendah
- Model modifikasi → SSIM lebih tinggi

Interpretasi:

Model modifikasi mampu mempertahankan struktur, tekstur, dan kontras gambar lebih baik. Hasil denoising lebih bersih dan lebih mirip gambar asli.



SOLUTION OVERVIEW



Project Summary

What We Created

- Model autoencoder berhasil mengurangi noise tanpa merusak struktur utama gambar.
- Modifikasi arsitektur memberikan peningkatan performa yang signifikan berdasarkan nilai SSIM.
- Model ini layak digunakan sebagai langkah awal untuk pipeline denoising pada tugas computer vision.

CHALLENGES FACED

Challenge 1

Tidak Ada Dataset Noisy Asli

Tantangan pertama adalah tidak adanya dataset gambar noisy yang sesungguhnya. Karena itu saya harus membuat noise sendiri menggunakan Gaussian noise (mean 0, std 0.1). Tantangannya adalah memastikan noise yang ditambahkan cukup realistik dan cukup ‘kotor’ untuk dilatih model, tapi tidak sampai merusak struktur gambar.

Challenge 2

Menentukan Arsitektur Autoencoder yang Pas

Autoencoder harus cukup dalam untuk menangkap pola visual dan membedakan mana detail gambar asli dan mana noise. Tapi kalau terlalu kompleks, model bisa overfit dan malah tidak generalisasi. Tantangannya adalah menemukan keseimbangan jumlah layer, jumlah filter, dan struktur encoder-decoder yang memberikan hasil denoising paling bersih tanpa overfitting.

What I Learned

Personal Skills Gained

Dari project ini, aku belajar banyak hal, baik teknis maupun cara berpikir sebagai data scientist. Secara teknis, aku jadi lebih paham workflow deep learning untuk image, mulai dari EDA gambar, preprocessing, nambahin noise, sampai bangun autoencoder dan memodifikasinya supaya hasil denoising lebih bagus.

Aku juga belajar gimana cara milih arsitektur yang pas, kenapa layer tertentu dipakai, dan gimana cara debugging model kalau hasil awalnya kurang oke. Selain itu, aku jadi ngerti cara evaluasi model gambar pakai SSIM, yang ngukur seberapa mirip kualitas visual hasilnya dengan gambar asli.

Di luar teknis, aku belajar untuk iterasi terus dengan mencoba, gagal, dan perbaiki lagi. Jadi aku belajar pentingnya memahami data dulu sebelum buru-buru modeling. Project ini juga ngelatih aku buat lihat masalah dari sudut pandang user, tujuan akhirnya bukan cuma bikin model jalan, tapi bikin gambar hasil denoising lebih bersih dan lebih usable.

KEYLA'S PROJECT

The End

NAME

Keyla Faristha Rindani