Laporan Akhir Eksplorasi Autoencoder pada Dataset Sign Language MNIST

Disusun oleh:
Ghulam Mushthofa 442023611060

Teknik Informatika

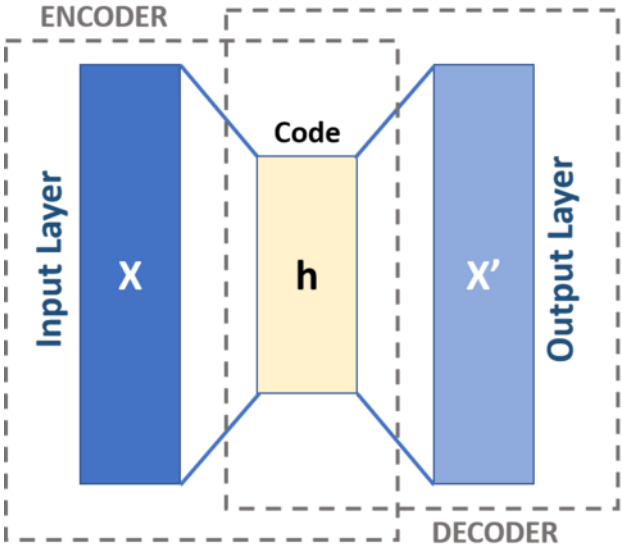
Universitas Darussalam Gontor

1 July 2025

1. Pendahuluan

Perkembangan teknologi pengenalan citra semakin pesat, terutama dalam bidang pembelajaran mesin dan deep learning. Salah satu metode yang menarik untuk dieksplorasi adalah Autoencoder, yaitu jaringan saraf buatan yang dilatih untuk menyalin input ke output-nya sendiri, melalui representasi tengah yang disebut latent space. Autoencoder sangat berguna untuk ekstraksi fitur, reduksi dimensi, dan rekonstruksi data.

Dalam laporan ini, kami mengeksplorasi penerapan Convolutional Autoencoder (CNN-AE) pada dataset Sign Language MNIST. Dataset ini berisi gambar tangan yang merepresentasikan alfabet bahasa isyarat Amerika (ASL) dalam format grayscale 28x28 piksel. Tujuan dari eksperimen ini adalah untuk melatih autoencoder yang mampu memahami dan merepresentasikan bentuk-bentuk gestur dalam bentuk kompresi, kemudian mampu membentuk kembali gambar tersebut seakurat mungkin.



2. Dataset dan Preprocessing

Dataset yang digunakan bernama Sign Language MNIST, tersedia di Kaggle. Dataset ini terdiri dari dua file CSV yaitu:

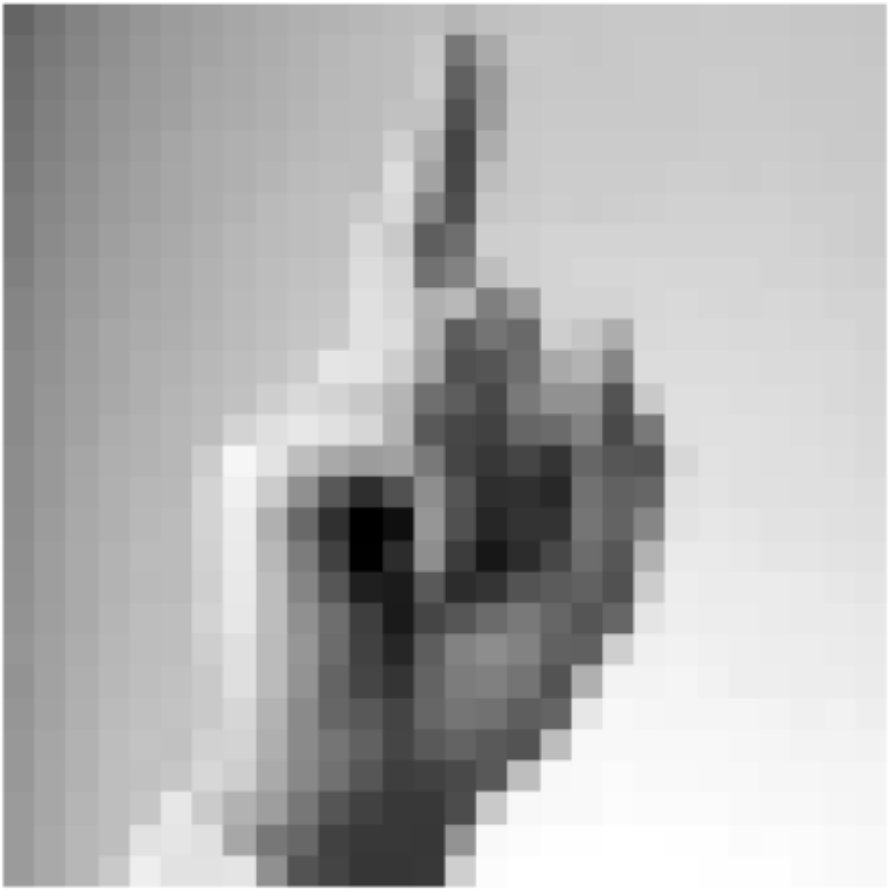
- sign_mnist_train.csv: 27.455 gambar untuk pelatihan
- sign_mnist_test.csv: 7.172 gambar untuk pengujian

Setiap baris dalam file CSV merepresentasikan satu gambar dengan format:

- Kolom pertama: label huruf (0-25, mewakili A-Z kecuali J dan Z)
- 784 kolom berikutnya: nilai piksel dalam grayscale (0–255)

Preprocessing dilakukan dengan mengubah piksel menjadi tensor berukuran (1, 28, 28), menormalisasi ke rentang [0,1], dan menghapus label karena tidak diperlukan dalam autoencoder (unsupervised learning).

Contoh gambar mentah dari dataset Sign Language MNIST



3. Arsitektur Model Autoencoder

Model yang digunakan berbasis CNN dan terdiri dari dua bagian utama: encoder dan decoder.

Encoder:

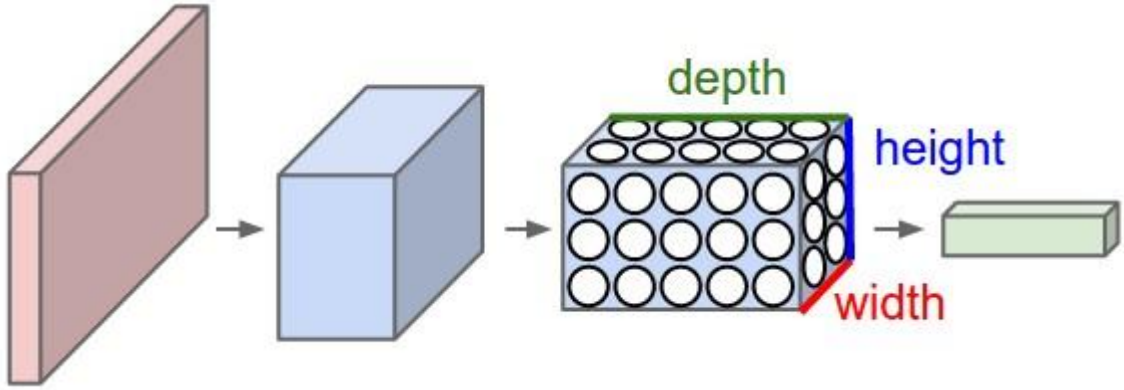
- Conv2D(1→16), kernel=3, padding=1 → ReLU → MaxPool(2×2)
- Conv2D(16→32), kernel=3, padding=1 → ReLU → MaxPool(2×2)

- Conv2D(32→64), kernel=3, padding=1 → ReLU → MaxPool(2×2)

Ukuran input (1, 28, 28) menjadi latent space (64, 3, 3) = 576 dimensi.

Decoder:

- ConvTranspose2D(64→32), kernel=3, stride=2
- ConvTranspose2D(32→16), kernel=3, stride=2, padding=1, output_padding=1
- ConvTranspose2D(16→1), kernel=3, stride=2, padding=1, output_padding=1 → Sigmoid



4. Training Model

Model dilatih menggunakan fungsi loss MSELoss() dan optimizer Adam dengan learning rate 0.001. Training dilakukan selama 20 epoch dengan batch size 64.

Selama pelatihan, loss menurun secara konsisten:

- Epoch 1: 0.0166
- ...
- Epoch 20: 0.0031

Hal ini menunjukkan bahwa model belajar dengan baik dan mampu melakukan rekonstruksi yang mendekati input asli.

```
Epoch [1/20], Loss: 0.0142
Epoch [2/20], Loss: 0.0070
Epoch [3/20], Loss: 0.0059
Epoch [4/20], Loss: 0.0053
Epoch [5/20], Loss: 0.0049
Epoch [6/20], Loss: 0.0046
Epoch [7/20], Loss: 0.0044
Epoch [8/20], Loss: 0.0042
Epoch [9/20], Loss: 0.0041
Epoch [10/20], Loss: 0.0039
Epoch [11/20], Loss: 0.0038
Epoch [12/20], Loss: 0.0037
Epoch [13/20], Loss: 0.0036
Epoch [14/20], Loss: 0.0036
Epoch [15/20], Loss: 0.0035
Epoch [16/20], Loss: 0.0034
Epoch [17/20], Loss: 0.0033
Epoch [18/20], Loss: 0.0033
Epoch [19/20], Loss: 0.0032
Epoch [20/20], Loss: 0.0032
```

5. Hasil Rekonstruksi

Setelah model dilatih, kami melakukan rekonstruksi gambar dari dataset uji. Gambar-gambar ini dibandingkan langsung dengan input aslinya. Secara visual, hasil rekonstruksi sangat mirip dengan input.

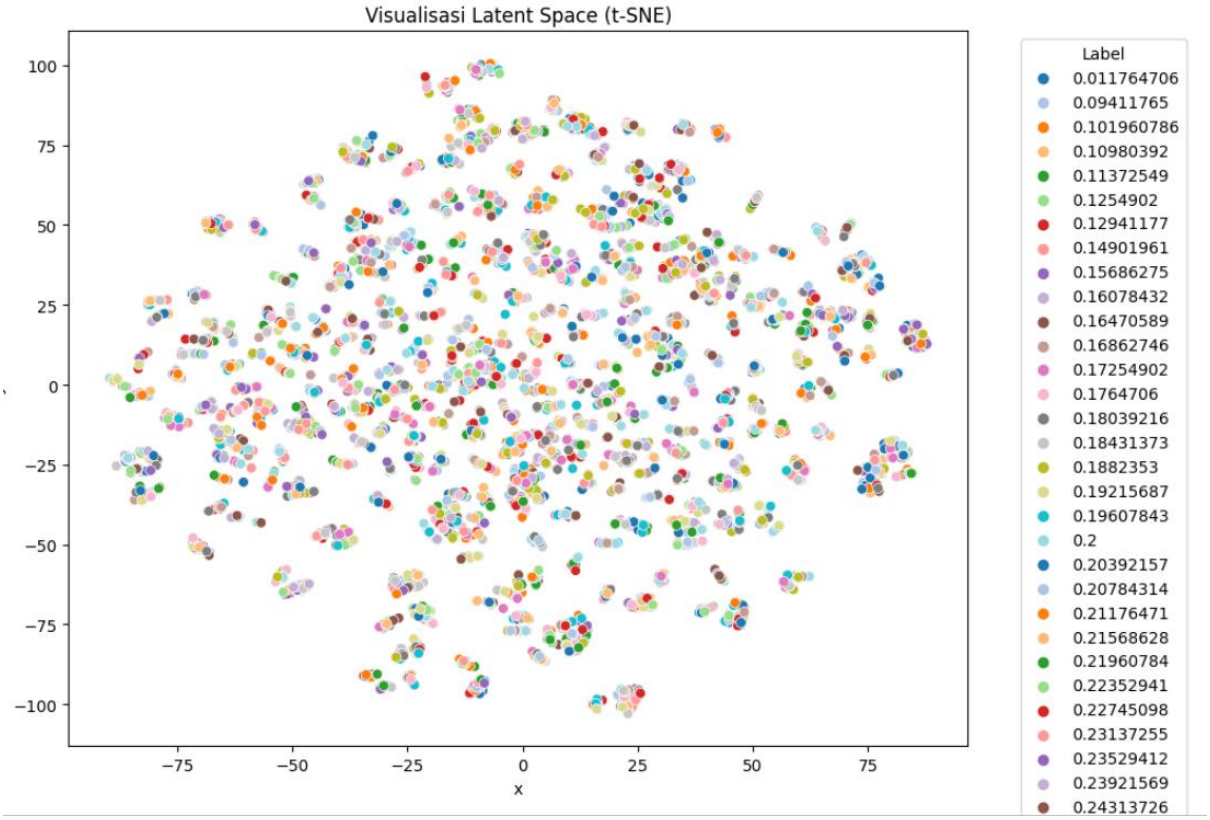
Beberapa gambar memang terlihat sedikit blur, namun secara keseluruhan bentuk dasar gestur tetap terlihat jelas.

6. Visualisasi Latent Space

Untuk memahami bagaimana model merepresentasikan gestur dalam latent space, kami menggunakan algoritma t-SNE untuk mereduksi dimensi dari 576 ke 2 dimensi.

Hasilnya divisualisasikan dalam scatter plot, di mana setiap titik mewakili satu gambar dan warna menunjukkan label aslinya.

Hasil visualisasi menunjukkan bahwa titik-titik dengan label sama cenderung berkelompok, yang berarti model berhasil mempelajari struktur gestur dalam latent space.



7. Evaluasi dan Analisis

Model autoencoder menunjukkan performa yang stabil:

- Rekonstruksi visual sangat baik
- Loss turun signifikan dalam waktu singkat
- Latent space menghasilkan pemisahan label yang cukup jelas

Namun, beberapa kekurangan yang diamati:

- Hasil blur pada beberapa kelas huruf
- t-SNE memerlukan waktu lama dan hasilnya sensitif terhadap parameter

Solusi untuk perbaikan ke depan:

- Menambahkan dropout atau batch normalization
- Eksperimen dengan arsitektur deeper encoder
- Menggunakan variational autoencoder (VAE)

8. Refleksi Pribadi

Selama mengerjakan eksplorasi Autoencoder ini, saya mendapatkan banyak pengalaman berharga yang memperdalam pemahaman saya tentang deep learning, khususnya dalam konteks unsupervised learning dan representasi fitur citra. Berikut adalah tiga poin utama dari refleksi pribadi saya:

1. Pemahaman Mendalam tentang Arsitektur Autoencoder
Saya jadi lebih paham bagaimana arsitektur encoder dan decoder bekerja secara sepasang. Encoder mengubah gambar menjadi bentuk vektor laten (compressed), sedangkan decoder berusaha membentuk kembali gambar tersebut. Tantangan utamanya adalah menjaga ukuran input dan output tetap konsisten, serta memilih parameter stride, padding, dan output_padding yang tepat pada ConvTranspose2D.

2. Pengalaman Menangani Error dan Debugging Model
Saya belajar menangani berbagai error selama proses pelatihan, seperti mismatch size pada loss function (input vs target), error broadcasting tensor, dan kesalahan indentasi dalam definisi model. Selain itu, saya juga jadi lebih terbiasa menggunakan tools seperti print(shape) dan view() untuk debugging saat terjadi ketidaksesuaian dimensi data.

3. Pentingnya Visualisasi dalam Unsupervised Learning
Tanpa label, sulit mengevaluasi performa model autoencoder hanya dari angka loss. Oleh karena itu, saya belajar pentingnya visualisasi, seperti:

- Gambar asli vs hasil rekonstruksi
- Visualisasi latent space menggunakan t-SNE
  Melalui ini, saya bisa menilai apakah model benar-benar belajar representasi yang bermakna.

9. Kesimpulan

Autoencoder berhasil diaplikasikan pada dataset Sign Language MNIST dengan hasil memuaskan. Model mampu memahami dan merekonstruksi gambar tangan dengan baik. Latent space yang dihasilkan juga memiliki struktur yang bermakna, terbukti dari visualisasi t-SNE.

Autoencoder membuka banyak peluang untuk digunakan dalam tugas-tugas lain seperti:

- Denoising image
- Data compression
- Anomaly detection

Dengan dasar ini, autoencoder dapat dikembangkan lebih lanjut menjadi sistem klasifikasi bahasa isyarat, yang dapat membantu masyarakat tuna rungu dan membangun aplikasi terapan berbasis computer vision.

10. Dokumentasi & Lampiran

Link dataset:
https://www.kaggle.com/datasets/datamunge/sign-language-mnist