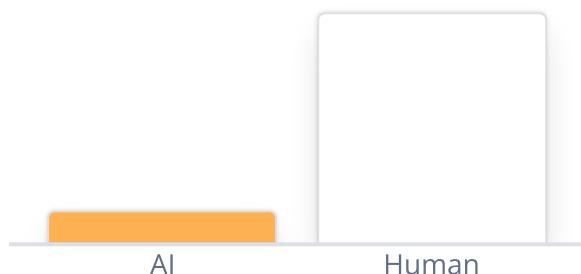


Results

13%

of text is likely AI ⓘ



AI-generated ⓘ  13%

Human-written ⓘ  87%

 Caution: Our AI Detector is advanced, but no detectors are 100% reliable, no matter what their accuracy scores claim. Never use AI detection alone to make decisions that could impact a person's career or academic standing.



Laporan Final Project

Kecerdasan Buatan (Lanjut)

Implementasi Generative Adversarial Network (GAN):

Menghasilkan Gambar Angka Tulisan Tangan

Kelompok 2

Anggota Kelompok

1. 2303010115 Adi Rosyadi Universitas Perjuangan Tasikmalaya 2. 2303010126 Juan Pernando A. Universitas Perjuangan Tasikmalaya 3. 23115668 Raditya M. W. H Universitas Amikom Yogyakarta 4. 23115646 Rifqi Aryo Mulyadi Universitas Amikom Yogyakarta 1. Latar Belakang Perkembangan Kecerdasan Buatan saat ini sangat pesan termasuk pada pengolahan citra. Deep Learning adalah salah satu yang menjadi fondasi digital dalam memecahkan masalah terkait kecerdasan buatan. Penerapan ini sudah sangat luas mulai dari data sintesis wajah hingga ke tulisan yang menyerupai tangan manusia[1]. Konsep Generative Adversarial Networks (GAN) merupakan Solusi dari deep learning untuk memecahkan masalah tersebut. GAN bekerja dengan dua cara yaitu Generator dan Diskriminator[2].

Pada pembelajaran GAN ini, Generator bekerja sebagai pembuat replika data baru dan Diskriminator yang menilai hasilnya. Melalui pengulangan lebih lanjut Generator terus diuji sampai Diskriminator sulit membedakan yang asli dengan yang palsu. Generator dan Diskriminator saling berlawanan satu sama lain sampai menghasilkan epoch atau waktu yang terus berlurang sampai hasilnya bagus.

Secara sederhananya GAN ini merupakan Generative Model untuk menciptakan data baru, hal ini bermanfaat untuk mengembangkan Kecerdasan Buatan yang semakin pesat. 2. Dataset

Projek ini menerapkan dataset didalam lingkungan deep learning data mnist diakui sebagai hal yang mendasar, terutama untuk peneliti yang ingin menguji sesuatu algoritma. Data mnist dibagi jadi dua bagian utama, antara lain:

- Data Pelatihan: Berjumlah 60.000 sampel gambar.
- Data Pengujian: Berjumlah 10.000 sampel gambar.

Kejelasan Teknis Citra

Setiap entitas gambar dalam dataset ini diatur dalam jenis tertentu. Dimensi Resolusi: Memiliki ukuran 28 x 28 piksel.

1. Skala Warna: Menggunakan format grayscale (skala abu-abu).

2. Kedalaman Piksel: Nilai intensitas setiap piksel berada pada rentang 0 (hitam) hingga 255 (putih). Metode Perolehan Data Data mnist tidak diunduh dari luar, melainkan sudah terintegrasi API dengan suatu data.

Berikut adalah potongan kode yang digunakan untuk memuat data tersebut:

Penyesuaian skala piksel ke dalam interval [-1, 1] merupakan langkah penting, yang disamakan dengan arsitektur model GAN. Hal ini dikarenakan lapisan output pada komponen Generator mengimplementasikan fungsi aktivasi Tanh, yang secara matematis memetakan nilai ke dalam rentang [-1, 1].

3. Metode 3.1 Alur Projek Pelaksanaan projek ini dilakukan dengan tahapan yang terintegrasi. Dimulai dari tahap perolehan data hingga ke validasi model, setiap langkahnya ditingkatkan untuk memperkirakan sebuah model. Untuk lebih dalam lagi data ini bekerja meliputi:

Perolehan Data: Tahap awal berupa pengumpulan dan pemuatan dataset MNIST ke dalam lingkungan pemrograman.

1. Pra-pemrosesan Citra: Dilakukan normalisasi dan penyesuaian dimensi data agar untuk nantinya diproses oleh algoritma. 2. Konstruksi Arsitektur Generator: Merancang susunan jaringan saraf yang bertanggung

Jawab menciptakan data sintetis. 3. Konstruksi Arsitektur Discriminator: Membangun unit jaringan saraf yang berfungsi sebagai penguji keaslian gambar.

4. Integrasi Model GAN: Disini di lakukan penggabungan komponen Generator dan Discriminator ke dalam sistem training saling berlawanan. 5. Fase Pelatihan: Menjalankan proses loop training untuk melatih

kemampuan kedua jaringan. 6. Interpretasi dan Analisis Output: dengan melakukan visualisasi terhadap hasil generasi angka serta mengevaluasi kualitas gambar yang diproduksi.

3.2 Perolehan dan Pra-pemrosesan Data Setelah sudah menjelaskan karakter dari dataset MNIST pada bagian sebelumnya, langkah

selanjutnya yaitu menerapkan serangkaian prosedur pra-pemrosesan. Tahap ini sangat penting untuk mempersiapkan data agar sesuai dengan traingin di arsitektur GAN. Proses dimulai dengan memuat data ke dalam memori, yang kemudian dilanjutkan dengan pengiriman skala piksel menjadi rentang [-1, 1]. Penerapan kerja dari nornalisasi ini diterapkan dengan kode sebagai berikut:

Penerapan kode tersebut bermanfaat sebagai prosedur validasi untuk memastikan bahwa intensitas piksel gambar telah terpetakan secara tepat, dalam interval [-1, 1]. Penyamaan ini bersifat wajib karena arsitektur Generator mengadopsi fungsi aktivasi Tanh, yang memerlukan input dan menghasilkan output dalam rentang numerik yang identik guna menjaga stabilitas. Untuk memperbaiki ketetapan perhinginan sistem selama dalam pelatihan, data citra yang telah dinormalisasi kemudian dikirim ke dalam objek tf.data.Dataset melalui pustaka TensorFlow. Prosedur penyusunan dataset ini mencakup langkah-langkah sebagai berikut:

3.3 Pengaturan Arsitektur Model 3.3.1 Bagian Generator

Generator bertugas untuk mengembangkan data sintesis citra tiruan. Proses ini meniru data mnist dengan acak, yang berperan sebagai keacakan atau ketidakpastian untuk menjamin adanya keragaman pada output gambar angka yang dihasilkan. Susunan tingkatan yang menyusun bagian Generator ini diterapkan melalui kode berikut:

3.3.2 Bagian Discriminator

Discriminator bekerja sebagai penguji dari hasil Generator. Tugas utamanya adalah melakukan proses untuk menentukan keaslian data. Pengaturan cara kerja dan susunan terkait Diskriminat, dicantumkan dalam kode sebagai berikut:

3.3.3 Penyatuan Sistem dan Algoritma GAN Setelah tahap perancangan masing-masing bagian selesai, GAN dijelaskan sebagai model generatif yang mengambil prinsip pembelajaran saling bersaing.

Pada pengerjaan ini, integrasi antara unit Generator dan Discriminator tidak menggunakan pendekatan model sekuensial konvensional. Sebagai gantinya, sistem melakukan siklus pelatihan kustom dengan memanfaatkan fungsi `tf.data.GradientTape`.
3.4
Prosedur Pelatihan Model Tahapan pelatihan model dilakukan melalui epoch sekitar 1000 epoch, projek ini menerapkan cara kerja Linear Decay pada penyimpanan standar noise agar menjaga stabilitas interaksi saling berlawanan. Untuk proses lebih mendalamnya meliputi hal seperti: Pengurangan Noise Dinamis; Parameter noise std disesuaikan secara bergerak maju dimulai dari nilai 0,10 pada awal pengulangan dan menyusut secara sejajar hingga mencapai 0,00 pada epoch

terakhir.
• Tahap Saling Berlawanan di setiap kelompok data, fungsi `train_step` diproses untuk melakukan pembaruan bobot pada komponen Generator dan Discriminator secara serentak.
• Memantau dan catatan kinerja data kerugian dari Generator (`g_1`) dan Discriminator (`d_1`) diperoleh pada setiap kelompok, dan dirata-ratakan di akhir setiap epoch.
• Hasil Akhir nya menyelesaikan 1000 epoch, fungsi `generate_images` dipanggil untuk mensahkan output akhir. Di tahap ini, model diharapkan telah mencapai titik menuju pusat.

4. Hasil dan Pembahasan Pemantauan dilakukan secara berskala untuk melihat suatu model dari setiap tahapan,

Perkembangan tersebut dapat dijelaskan ke dalam beberapa tahapan penting, dimulai dari tahap awal sebagai berikut:

• Tahap Awal(Epoch 1): Kondisi pada Epoch ini masih dalam keadaan acak dan buram



• Tahap Keseimbangan dan Pengurangan Noise (Epoch 200 – 500): Seiring dengan penurunan nilai noise std secara sedikit demi sedikit, kualitas nya sudah menunjukan kejelasan angka tulisan tangan.

• Tahap Perpindahan dan Susunan Pola (Epoch 600 – 900): Selama proses ini pengurangan reduksi mulai sangat terlihat. Generator bisa menghasilkan angka yang lebih bagus.

• Tahap Optimal (Epoch 1000): Di tahap final ini, model telah mencapai titik keseimbangan yang diharapkan. Citranya menunjukan kualitas yang jelas, meskipun secara pembelajaran selanjutnya dapat terus ditingkatkan hingga 6000 epoch, capaian ini sudah sangat tepat untuk membuktikan keberhasilan model dalam menghasilkan data yang nyata.

5. Pemecahan Kinerja dan Batasan Model

Hasil menunjukkan bahwa pembelajaran Generator dilakukan secara bertahap dan memakan waktu yang cukup lama. Meskipun model sudah menghasilkan data sintesis yang tepat, masih ada batasan terkait ketajaman dan kejelasan lebih lanjut, seperti adanya efek blur atau pada garis angka.



Keseimbangan sistem ini sangat dipengaruhi oleh penerapan strategi Linear Noise Decay. Cara kerja tersebut secara tepat untuk mengurangi dominasi sebelum waktunya oleh Discriminator, sehingga memberikan ruang bagi Generator untuk memisahkan informasi dari data ke arah perbaikan hingga mencapai titik pusat akhir yang seimbang.

6. Kesimpulan

Berdasarkan penelitian projek ini dapat disimpulkan dengan titik dasar sebagai berikut:

- Kemampuan Arsitektur GAN: Projek ini menyatakan bahwa kerangka kerja GAN memiliki ketetapan yang tinggi untuk memproses data visual yang rumit.
- Signifikan Dataset MNIST: MNIST terbukti menjadi acuan yang sangat cocok untuk tahap awal pembelajaran sistem generatif.
- Menetapkan Pengaturan Pelatihan: Kualitas citra dipengaruhi oleh epoch semakin tinggi epoch makan hasil semakin jelas.
- Kreativitas Sintesis Data: Jaringan Generator terbukti tidak melakukan proses langsung terhadap dataset referensi. Model ini berhasil mengambil inti dari penulisan angka, yang memungkinkannya untuk memproduksi entitas visual baru yang bersifat unik namun tetap mempertahankan angka yang tepat dan tetap dengan data asli.

7. Referensi

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

- [2] I. Goodfellow et al., "Generative Adversarial Nets," in Proc. Advances in Neural Information Processing Systems (NIPS), 2014, pp. 2672–2680.
- [3] Y. LeCun et al., "The MNIST database of handwritten digits," 1998.
- [4] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in Proc. International Conference on Learning Representations (ICLR), 2016.
- [5] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," in Proc. 30th International Conference on Neural Information Processing Systems (NIPS), 2016, pp. 2234–2242.

[6] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Amortised MAP Inference for Image Super-resolution," in Proc. 5th International Conference on Learning Representations (ICLR), 2017. (Referensi utama untuk teknik penambahan noise pada input discriminator).

- Riqqi Aryo Mulyadi: Implementasi kode dan training model • Juan Pernando: Analisis hasil dan pengujian
- Adi Rosyadi: Penyusunan laporan dan dokumentasi • Raditya M. W. H: Studi literatur dan referensi
- Link Colab: https://colab.research.google.com/drive/1oB20FhOKXRZ4-jzSzsPA7FRuvLyW_Gb9?usp=sharing