

AI Report

We are moderately confident this text is

AI Generated

AI Probability

69%

This number is the probability that the document is AI generated, not a percentage of AI text in the document.

Plagiarism



The plagiarism scan was not run for this document. Go to gptzero.me to check for plagiarism.

Laporan Final Project Kecerdasan Buatan (Lanjutan) - 1/4/2026

Cek Yuk

Laporan Final Project Kecerdasan Buatan (Lanjutan)

Implementasi Neural Style Transfer untuk Transformasi Gaya Artistik pada Citra Digital Menggunakan VGG-19.

Kelompok 10

Anggota Kelompok:

23.11.5571 | Aulia Resty Nur Aini

23.11.5603 | Nabillah Alimuddin

23.11.5604 | M. Dian Fauzi

23.11.5620 | Ledyvia Audiz Coranov

1. Latar Belakang

Perkembangan teknologi Artificial Intelligence (AI), khususnya dalam domain Generative Modelling, telah menghadirkan paradigma baru dalam penciptaan karya seni digital.

Salah satu tantangan utama dalam industri kreatif adalah menggabungkan konten visual realistis (seperti fotografi) dengan estetika artistik yang kompleks (seperti gaya lukisan impressionisme) tanpa memerlukan proses pelukisan manual yang memakan waktu lama.

Metode konvensional dalam pengolahan citra digital seringkali hanya memanipulasi warna atau filter sederhana, namun gagal menangkap esensi "gaya" atau tekstur dari menjadi karya seni.

Pada tahun 2016, Gatys et al.

memperkenalkan metode Neural Style Transfer (NST) yang memanfaatkan Convolutional Neural Networks (CNN) untuk memisahkan dan merekomposisi konten dan gaya dari dua citra yang berbeda [1].

Pendekatan ini memungkinkan komputer untuk "melukis ulang" menjadi Foto dengan gaya seniman ternama.

Penelitian serupa juga telah dikembangkan di Indonesia, seperti yang dilakukan oleh Yulian dkk. (2023) yang menerapkan NST dan Convolutional Autoencoder untuk pelestarian motif Batik [5].

Mengacu pada penelitian tersebut, penggunaan arsitektur Deep Learning terbukti efektif dalam merepresentasikan fitur visual artistik yang kompleks.

Dalam proyek akhir ini, kami mengembangkan system bernama "The Art Museum Generator".

Sistem ini mengimplementasikan arsitektur VGG-19 yang telah dilatih sebelumnya (pre-trained) pada dataset ImageNet.

Tujuannya adalah untuk mendemokratisasi akses terhadap penciptaan seni digital, memungkinkan pengguna awam untuk mentransformasi foto pribadi mereka menjadi karya seni dengan gaya pelukis maestro dunia (seperti Van Gogh atau Monet) melalui antarmuka berbasis web yang interaktif.

2. Metode:

1) Alur Pengerjaan (System Flowchart)

Tahapan pengerjaan proyek ini dimulai dari pengumpulan data hingga deployment aplikasi, dengan rincian sebagai berikut:

a. Pengumpulan Dataset: Mengurasi citra gaya (style images) dari repositori seni publik (Kaggle: Best Artworks of All Time).

b. Preprocessing Citra:

i. Resizing: Mengubah dimensi citra konten dan gaya menjadi maksimum 512 piksel (sisi terpanjang) untuk menjaga efisiensi komputasi GPU.

ii.

Normalization: Mengonversi nilai piksel ke rentang $[0, 1]$ dan menambahkan dimensi batch agar sesuai dengan input tensor VGG-19.

c. Pembangunan Model (VGG19): Memuat model VGG-19 tanpa top layer (klasifikasi), dan mendefinisikan intermediate layers untuk ekstraksi fitur:

i. Content Layer: block5_conv2

ii.

Style Layers: block1conv1, block2conv1, block3conv1, block4conv1, block5_conv1.

d. Proses Optimasi (Training Loop):

i. Menghitung Content Loss (perbedaan fitur konten) dan Style Loss (perbedaan korelasi Gram Matrix).

Dimana α (bobot konten) diset sebesar 10^4 dan β (bobot gaya) diset sebesar 10^{-2} .

ii.

5?5531100515a5R35a

Gram Matrix: Untuk merepresentasikan gaya, kami menghitung korelasi antar filter fitur menggunakan matriks Gram, yang memungkinkan model menangkap tekstur tanpa terikat pada posisi spasial objek [1], [5].

3. Dataset

Dataset yang digunakan dalam penelitian ini bernama Best Artworks of All Time.

Dataset ini diperoleh dari platform Kaggle, yang dipublikasikan oleh kontributor bernama Icaro (ikarus777).

Tautan: <https://www.kaggle.com/ikarus777/best-artworks-of-all-time>

Dataset "Best Artworks of All Time" merupakan koleksi citra digital yang menghimpun karya-karya lukisan dari 50 seniman paling berpengaruh dalam sejarah seni dunia.

Dataset ini difokuskan pada pengenalan pola artistik berdasarkan pelukisnya (artist-centric).

Secara rinci, dataset ini terdiri dari dual komponen utama:

Data Citra (Images):

- Berisi lebih dari 8.400 gambar lukisan.

Struktur direktori disusun berdasarkan nama seniman (setiap seniman memiliki situ folder tersendiri, misalnya folder VincentvanGogh, LeonardodaVinci, dll).

O Gambar tersedia dalam variasi ukuran asli dan ukuran yang telah diubah (resized) untuk mempermudah komputasi.

Metadata (artists.csv): File ini berisi informasi tekstual mengenai profil ke-50 seniman yang ada dalam dataset. Atribut kepada kolom yang terdapat dalam file ini meliputi:

id: Identifikasi unik untuk setiap seniman.

name: Nama lengkap seniman.

\bigcirc years: Tahun kelahiran dan kematian seniman.

\bigcirc genre: Aliran seni utama yang dianut (contoh: Impressionism, Renaissance, Surrealism).

\bigcirc nationality: Kebangsaan seniman.

bio: Biografi singkat seniman.

\bigcirc paintings: Jumlah total lukisan seniman tersebut yang tersedia dalam dataset ini.

Hasil Pengujian

4.1.

Skenario Pengujian

Pengujian dilakukan menggunakan lingkungan komputasi Google Colab dengan akselerator GPU T4.

Skenario pengujian meliputi:

Input: Foto pemandangan hutan (Resolusi asli 960×1280, di-resize ke 512px).

Style: Lukisan "Self-Portrait" karya Vincent van Gogh.

Parameter: Epoch = 5, Steps per Epoch = 50, Learning Rate = 0.02.

4.2.

Hasil Visual

A. Spesifikasi Model

Berdasarkan log eksekusi sistem, arsitektur VGG-19 yang digunakan memiliki total parameter sebanyak 20,024,384.

Sesuai dengan metode Neural Style Transfer (Gatys et al.), jumlah Trainable Params adalah 0, yang mengonfirmasi bahwa bobot model dibekukan (frozen) dan tidak terjadi proses backpropagation pada layer model, melainkan pada citra input.

B. Grafik Evaluasi Loss

Total waktu yang dibutuhkan untuk menyelesaikan proses pelatihan sebanyak 5 Epoch adalah sekitar 18,9 detik. Waktu eksekusi yang singkat ini membuktikan efisiensi kode saat dijalankan pada lingkungan GPU T4.

Grafik evaluasi merekam penurunan nilai Total Loss berdasarkan jumlah langkah (steps).

Tercatat penurunan yang signifikan dari awal hingga akhir pelatihan:

Loss Awal (Step 0): $\approx 7.4 \times 10^7$

Loss Akhir (Step 250): $\approx 4.5 \times 10^6$

Gambar 4.1.

Grafik Penurunan Total Loss selama 250 Steps (5 Epoch).

Grafik pada Gambar 4.1 memperlihatkan tren garis yang menurun tajam (steep descent) pada 50 langkah pertama, kemudian melandai secara bertahap menuju titik konvergensi di kisaran 4.5×10^{-6} .

C. Hasil Visualisasi (Kualitatif)

Gambar 4.2.

Hasil implementasi gaya Van Gogh pada objek Patung Garuda.

Secara visual, model berhasil mengaplikasikan karakteristik impasto (teknik cat tebal) khas Van Gogh.

Langit: Tekstur latar belakang lukisan yang abstrak berhasil menggantikan tekstur awan realistis, memberikan nuansa dramatis.

Objek Utama: Patung Garuda tetap mempertahankan geometri aslinya berkat ekstraksi fitur konten dari layer `block5_conv2`, namun warnanya telah beradaptasi dengan palet warna hangat (oranye/cokelat) dari lukisan referensi.

5. Analisa Hasil

5.1.

Analisa Dinamika Loss Function dan Konvergensi

Berdasarkan grafik penurunan Total Loss dengan total 250 langkah iterasi, dapat dianalisis karakteristik optimasi sebagai berikut:

Step 0-50: Pada rentang Step 0 hingga 50 (setara dengan Epoch 1), terjadi penurunan Loss yang paling ekstrem, yaitu dari angka 7.4×10^{-7} menjadi kisaran 1.0×10^{-7} .

Fenomena ini menunjukkan bahwa Adam Optimizer bekerja sangat efektif pada tahap awal untuk mengubah distribusi statistik citra konten (white noise/content) secara global agar mendekati representasi gaya referensi.

Step 50-250: Memasuki Step 50 hingga 250 (setara dengan Epoch 2 s.d. 5), kurva terlihat melandai (asymptotic). Pada fase ini, model tidak lagi mengubah struktur gambar secara drastis, melainkan berfokus pada penghalusan detail tekstur (fine-tuning texture).

Meskipun durasi pelatihan dibatasi hanya 5 epoch demi efisiensi, stabilitas kurva menunjukkan bahwa model telah mencapai titik konvergensi yang cukup baik untuk menghasilkan visual yang estetis.

5.2.

Analisa Arsitektur VGG-19 sebagai Feature Extractor

Keberhasilan sistem dalam menghasilkan citra artistik tanpa melatih ulang model (0 trainable parameters) memvalidasi efektivitas arsitektur VGG-19 varian 20 juta parameter sebagai pengekstraksi fitur semantik yang kuat.

Pemilihan Layer Konten (block5_conv2): Keputusan menggunakan layer yang cukup dalam (block5) untuk representasi konten terbukti krusial.

Layer ini menangkap representasi spasial tingkat tinggi (bentuk patung Garuda), bukan detail piksel (warna asli batu).

Inilah alasan mengapa meskipun seluruh tekstur batu digantikan oleh goresan kuas warna-warni, siluet dan bentuk 3D patung Garuda tetap kokoh dan tidak terdistorsi.

Pemilihan Layer Gaya (Multi-scale Style): Penggunaan kombinasi 5 layer gaya (dari block1 hingga block5) memungkinkan transfer gaya yang komprehensif.

Layer awal (block1, block2) memtransfer warna dan goresan halus, sementara layer akhir (block4, block5) memtransfer pola komposisi lukisan yang lebih abstrak.

5.3.

Analisa Trade-off Bobot Konten dan Gaya

Pada percobaan ini, digunakan rasio bobot konten (α) sebesar 10^4 dan bobot gaya (β) sebesar 10^{-2} .

Secara matematis, terdapat ketimpangan nilai yang besar (10,000 banding 0.01).

Namun, analisa telah visual menunjukkan bahwa rasio ini justru menghasilkan keseimbangan perseptual yang tepat.

Hal ini terjadi karena nilai numerik mentah dari Gram Matrix (Gaya) cenderung memiliki magnitudo yang jauh lebih besar daripada nilai MSE (Konten).

Jika kedua bobot disetarakan (misal 1:1), maka Style Loss akan mendominasi total Loss, yang mengakibatkan citra telah menjadi abstrak total dan bentuk patung Garuda akan hilang.

Konfigurasi bobot yang dipilih berhasil mempertahankan identitas objek (Garuda) sambil tetap mengaplikasikan estetika Van Gogh secara kuat.

5.4.

Efisiensi Komputasi

Waktu eksekusi rata-rata sebesar 3.8 detik per epoch (Total 18.9 detik) pada resolusi 512px menunjukkan efisiensi metode optimasi berbasis single-image.

Metode ini jauh lebih ringan dibandingkan metode pelatihan Image Transformation Network (seperti CycleGAN) yang bisa memakan waktu berjam-jam.

Hal ini membuktikan bahwa meskipun menggunakan arsitektur VGG-19 yang kompleks (20 juta parameter), metode optimasi langsung pada piksel tetap menjadi pendekatan yang paling efisien dari segi sumber daya komputasi untuk aplikasi Art Generator berbasis pengguna tunggal.

6. Kesimpulan

Berdasarkan telah implementasi system "The Art Museum Generator", dapat disimpulkan bahwa:

Efisiensi Arsitektur VGG-19:

Metode Transfer Learning dengan membekukan bobot VGG-19 (20,024,384 parameter) terbukti sangat efisien. Model mampu mengkstrak fitur konten dan gaya yang kompleks tanpa membebani komputasi, dengan waktu eksekusi total hanya 18.9 detik.

Kualitas Konvergensi Model:

Algoritma optimasi berhasil meminimalkan Total Loss secara signifikan dari 7.4×10^7 menjadi 4.5×10^6 dalam 250 langkah iterasi.

Penurunan drastis ini menunjukkan bahwa sistem mampu mengadaptasi distribusi gaya lukisan ke dalam foto dengan akurat dalam 5 epoch.

Keseimbangan Visual (Content vs Style):

Konfigurasi bobot konten (10^4) dan gaya (10^{-2}) menghasilkan citra keluaran yang optimal.

Sistem sukses mengaplikasikan tekstur artistik (seperti goresan kuas impasto) tanpa merusak struktur geometri objek asli, sehingga identitas foto tetap terjaga.

Aksesibilitas Teknologi:

Integrasi model ke dalam antarmuka interaktif (User Interface) berhasil menyederhanakan kompleksitas Deep Learning, memungkinkan pengguna awam untuk menciptakan karya seni generatif berkualitas tinggi secara instan dan mudah.

7. Referensi

- [1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2414–2423.
- [2] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," in Proceedings of the European Conference on Computer Vision (ECCV), Cham: Springer, 2016, pp. 694–711.
- [3] K. Liu, D. Yang, and Y. Ma, "Image Style Transfer Based on Dynamic Convolutional Manifold Alignment of Halo Attention," Electronics, vol. 12, no. 8, p. 1881, 2023.
- [4] T. Q. Nguyen et al., "Material Translation Based on Neural Style Transfer with Ideal Style Image Retrieval," Sensors, vol.

22, no.
19, p. 7317, 2022.

[5] C. Yulian, H. Santoso, I. Wasito, and Haryono, "Batik-Image Style Transfer Using Neural Style Transfer and Convolutional Autoencoder," JATISI (Jurnal Teknik Informatika dan Sistem Informasi), vol. 10, no. 4, pp. 593–607, Dec. 2023.

[6] Y. A. Irawan and A. Widjaja, "Pembangkitan Pola Batik dengan Menggunakan Neural Transfer Style dengan Penggunaan Cost Warna," JuTISI (Jurnal Teknik Informatika dan Sistem Informasi), vol. 6, no. 2, Aug. 2020.

8. Kontribusi & distribusi anggota kelompok

NIM	Nama	Kontribusi
-----	------	------------

---	---	---
-----	-----	-----

2311.5604	M. Dian Fauzi	• Implementasi VGG-19: Menulis kode inti untuk memuat model pre-trained dan membekukan bobot (freezing weights).
-----------	---------------	--

		• Logika Algoritma: Menyusun fungsi matematika Gram Matrix dan Training Loop untuk proses style transfer.
--	--	---

		• Environment Setup: Menyiapkan environment Google Colab (GPU) dan Cloudflare Tunnel agar program bisa berjalan online.
--	--	---

2311.5620	Ledyvia Audiz Coranov	• Akuisisi Dataset: Mengunduh dataset Best Artworks of All Time dari Kaggle dan memastikan integritas data (folder seniman terbaca dengan baik).
-----------	-----------------------	--

		• Data Loading & Testing: Memastikan fungsi load_img berjalan lancar dan menyiapkan skenario pengujian (gambar konten vs gaya).
--	--	---

		• Quality Assurance: Melakukan pengecekan plagiasi (Turnitin) dan pengecekan AI (ZeroGPT) pada laporan akhir untuk memastikan orisinalitas.
--	--	---

2311.5571	Aulia Resty Nur Aini	• Pengembangan UI: Membuat antarmuka web menggunakan Streamlit, termasuk tombol upload, dropdown pilihan seniman, dan layout halaman.
-----------	----------------------	---

- Visualisasi & Styling: Mengimplementasikan kode Matplotlib untuk grafik loss real-time serta menambahkan CSS untuk mempercantik tampilan.

- Presentasi: Menyusun slide Power Point (PPT).

| 23.11.5603 | Nabillah Alimuddin | • Penyusunan Laporan & Studi Pustaka: Mengompilasi seluruh hasil percobaan dalam bentuk laporan serta mencari referensi jurnal pendukung untuk memperkuat dasar teori.

- Analisa: Menganalisis makna grafik loss (konvergensi) dan mencari referensi jurnal pendukung untuk memperkuat dasar teori.

- Video: Membuat video penjelasan proyek, bedah kode serta demo aplikasi yang memperlihatkan alur kerja sistem dari input hingga hasil generate.

| --- | --- | --- |

High Human Impact ● ● ● ● ● High AI Impact

FAQs

What is GPTZero?

GPTZero is the leading AI detector for checking whether a document was written by a large language model such as ChatGPT. GPTZero detects AI on sentence, paragraph, and document level. Our model was trained on a large, diverse corpus of human-written and AI-generated text with support for English, Spanish, French, German, and other languages. To date, GPTZero has served over 10 million users around the world, and works with over 100 organizations in education, hiring, publishing, legal, and more.

When should I use GPTZero?

Our users have seen the use of AI-generated text proliferate into education, certification, hiring and recruitment, social writing platforms, disinformation, and beyond. We've created GPTZero as a tool to highlight the possible use of AI in writing text. In particular, we focus on classifying AI use in prose. Overall, our classifier is intended to be used to flag situations in which a conversation can be started (for example, between educators and students) to drive further inquiry and spread awareness of the risks of using AI in written work.

Does GPTZero only detect ChatGPT outputs?

No, GPTZero works robustly across a range of AI language models, including but not limited to ChatGPT, GPT-5, GPT-4, GPT-3, Gemini, Claude, and AI services based on those models.

What are the limitations of the classifier?

The nature of AI-generated content is changing constantly. As such, these results should not be used to punish students. We recommend educators to use our behind-the-scenes [Writing Reports](#) as part of a holistic assessment of student work. There always exist edge cases with both instances where AI is classified as human, and human is classified as AI. Instead, we recommend educators take approaches that give students the opportunity to demonstrate their understanding in a controlled environment and craft assignments that cannot be solved with AI. Our classifier is not trained to identify AI-generated text after it has been heavily modified after generation (although we estimate this is a minority of the uses for AI-generation at the moment). Currently, our classifier can sometimes flag other machine-generated or highly procedural text as AI-generated, and as such, should be used on more descriptive portions of text.

I'm an educator who has found AI-generated text by my students. What do I do?

Firstly, at GPTZero, we don't believe that any AI detector is perfect. There always exist edge cases with both instances where AI is classified as human, and human is classified as AI. Nonetheless, we recommend that educators can do the following when they get a positive detection: Ask students to demonstrate their understanding in a controlled environment, whether that is through an in-person assessment, or through an editor that can track their edit history (for instance, using our [Writing Reports](#) through Google Docs). Check out our list of [several recommendations](#) on types of assignments that are difficult to solve with AI.

Ask the student if they can produce artifacts of their writing process, whether it is drafts, revision histories, or brainstorming notes. For example, if the editor they used to write the text has an edit history (such as Google Docs), and it was typed out with several edits over a reasonable period of time, it is likely the student work is authentic. You can use GPTZero's Writing Reports to replay the student's writing process, and view signals that indicate the authenticity of the work.

See if there is a history of AI-generated text in the student's work. We recommend looking for a long-term pattern of AI use, as opposed to a single instance, in order to determine whether the student is using AI.