

Penggunaan Pola Dua Gambar dengan Metode *Fast Style Transfer* menggunakan TensorFlow

Liani Cikita Saragih¹⁾, Ayudya Octaviani²⁾, Meity Damaria³⁾, Novita Natalia⁴⁾, Puput Indri Sushanty⁵⁾

¹⁾²⁾³⁾⁴⁾⁵⁾ Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya
Kampus UPR Tunjung Nyaho, Jalan Yos Sudarso, Palangka Raya

¹⁾ lianicikita11@gmail.com

²⁾ ydyocta@gmail.com

³⁾ meitydamaria@gmail.com

⁴⁾ nvtnatalia25@gmail.com

⁵⁾ puputindri08@gmail.com

Abstrak

Era *Society 5.0* telah memunculkan kemajuan teknologi baru dalam seni visual, memungkinkan para seniman untuk bereksperimen dengan berbagai teknik dengan efisiensi yang lebih tinggi. *Fast Style Transfer* adalah salah satu pendekatan dalam transfer gaya gambar, memungkinkan penggabungan gaya atau estetika visual dari satu gambar ke gambar lain dengan cepat dan efisien melalui penggunaan jaringan saraf konvolusional. Metode ini memanfaatkan TensorFlow sebagai *framework* utama untuk implementasi, yang nantinya menghasilkan *output* berkualitas dengan kecepatan yang lebih efisien daripada metode tradisional. Penelitian ini menggunakan pendekatan eksperimental dengan langkah-langkah pengumpulan data, pengolahan data, implementasi *Fast Style Transfer*, evaluasi model, dan analisis hasil. Penelitian ini bertujuan untuk mengoptimalkan proses penggabungan pola dari dua gambar menggunakan metode *Fast Style Transfer* dengan memanfaatkan TensorFlow. Tahap awal melibatkan pengimporan berbagai *library* seperti TensorFlow, OpenCV, dan Matplotlib untuk pemrosesan gambar. Proses ini terdiri dari langkah-langkah penting seperti menentukan fungsi untuk memuat dan memproses gambar, memvisualisasikan gambar *input* dan *output*, serta menyimpan hasil akhir dalam format yang sesuai. Pada tahap akhir, *output* dari proses penggabungan pola dua gambar yang telah dilakukan dapat divisualisasikan dan disimpan untuk penggunaan lebih lanjut. Penelitian ini berhasil menggabungkan pola dari dua gambar dengan hasil yang memuaskan.

Kata kunci: *Fast Style Transfer*, Penggabungan Pola Gambar, Pengolahan Citra, TensorFlow.

Abstract

The Era *Society 5.0* has ushered in new technological advancements in visual arts, allowing artists to experiment with various techniques with higher efficiency. *Fast Style Transfer* is one approach in image style transfer, enabling the merging of visual styles or aesthetics from one image to another quickly and efficiently through the use of convolutional neural networks. This method leverages TensorFlow as the primary framework for implementation, ultimately yielding high-quality outputs at a faster pace compared to traditional methods. The research employs an experimental approach involving steps such as data collection, data processing, *Fast Style Transfer* implementation, model evaluation, and result analysis. The aim of this research is to optimize the process of merging patterns from two images using the *Fast Style Transfer* method with TensorFlow. The initial phase involves importing various libraries such as TensorFlow, OpenCV, and Matplotlib for image processing. This process consists of crucial steps such as defining functions to load and process images, visualizing input and output images, and saving the final results in appropriate formats. In the final stage, the output of merging patterns from two images can be visualized and saved for further use. The research successfully merges patterns from two images with satisfactory results.

Keywords: *Fast Style Transfer, Pattern Image Fusion, Image Processing, TensorFlow.*

1. PENDAHULUAN

Di era *Society 5.0*, seni visual terus berkembang dengan munculnya teknologi baru. Sebagai contoh, *software* untuk pengeditan gambar, fotografi digital, dan lain-lain. Hal tersebut memungkinkan para seniman untuk melakukan eksperimen dengan berbagai teknik yang nantinya akan menghasilkan karya seni yang unik dan menarik.

Pada seni visual, terdapat salah satu teknik yang populer yaitu transfer pola gambar. Teknik ini digunakan untuk menggabungkan pola atau gaya dari satu gambar ke gambar lainnya yang nantinya akan menghasilkan efek yang unik dan artistik. Berbagai aplikasi termasuk seni digital, desain grafis, dan fotografi telah menggunakan teknik ini. Namun, terdapat kendala pada metode transfer pola gambar tradisional yaitu seringkali lambat dan komputasi intensif bagi para seniman dan desainer yang ingin bereksperimen secara *real-time*.

Terdapat salah satu pendekatan yang digunakan untuk mengatasi keterbatasan metode transfer pola gambar tradisional, yaitu *Fast Style Transfer*. *Fast Style Transfer* adalah sebuah teknik dalam pemrosesan gambar yang memungkinkan untuk mentransfer gaya atau estetika visual dari satu gambar ke gambar lain. Metode ini menggunakan jaringan saraf konvolusional atau *Convolutional Neural Network (CNN)*. Untuk mempelajari pola dan gaya gambar dan kemudian menerapkannya ke gambar lain dengan cara yang cepat dan efisien.

Penelitian ini menggunakan jenis penelitian eksperimental, dengan langkah-langkah yaitu pengumpulan data, pengolahan data, implementasi *Fast Style Transfer*, evaluasi model, dan analisis hasil. Metode yang digunakan dalam penelitian ini adalah *Fast Style Transfer* yang pengimplementasiannya menggunakan TensorFlow. Pemilihan metode ini terbukti mampu menghasilkan *output* yang berkualitas dengan kecepatan yang lebih efisien dari metode tradisional.

Penerapan *Fast Style Transfer* dalam seni visual membuka peluang baru bagi para seniman dan desainer untuk mengeksplorasi berbagai gaya dan estetika dalam karya-karya yang dibuat dengan cara yang lebih efisien. Dengan kemampuannya untuk mentransfer gaya dari satu gambar ke gambar lainnya secara cepat, maka dapat dengan mudah bereksperimen dengan berbagai variasi gaya dan memperluas kreativitas tanpa terbatas oleh kendala waktu dan komputasi. Selain itu, keberhasilan metode ini dalam menghasilkan *output* berkualitas juga membuka peluang bagi pengembangan lebih lanjut dalam aplikasi seni visual, baik dalam konteks digital maupun tradisional, dengan memanfaatkan potensi teknologi untuk menciptakan karya-karya yang semakin inovatif dan menginspirasi.

Dengan memahami latar belakang ini, maka dibuat sebuah penelitian yang bertujuan untuk menggabungkan pola dua gambar dengan metode *Fast Style Transfer* menggunakan TensorFlow. Selain itu, penelitian ini juga bertujuan untuk mengoptimalkan proses transfer gaya antara dua gambar dengan memanfaatkan kemampuan TensorFlow dan teknik-teknik terkini dalam *deep learning*.

2. TINJAUAN PUSTAKA

Penelitian ini dilakukan oleh Satria Nusa Paradilaga (2020) dengan judul “Penerapan Kombinasi Pola Dua Gambar dengan Metode *Neural Style Transfer* untuk Aplikasi *Mobile*”. Penelitian ini bertujuan untuk menggabungkan pola dari dua gambar menggunakan metode *Neural Style Transfer* serta membangun sebuah aplikasi *mobile* yang mampu menerapkan teknik tersebut. Metode tersebut menggunakan *Convolutional Neural Network* untuk mengidentifikasi fitur atau konten dalam gambar dengan memanfaatkan lapisan-lapisan intermediat dalam arsitektur VGG19. Meskipun intermediat *layer* tersebut tidak mampu secara tepat mengekstraksi gaya dari gambar, korelasi fitur ditemukan melalui matriks gram. Model ini bertindak sebagai hasil ekstraksi konten dan gaya. Proses ekstraksi ini kemudian diikuti oleh algoritma transfer gaya yang melibatkan perhitungan kerugian dari konten dan gaya pada gambar menggunakan model, dan kemudian dilakukan pelatihan. Pelatihan ini dinilai untuk mendapatkan tahap pelatihan yang lebih baik, yang kemudian diimplementasikan pada aplikasi *mobile* melalui pembuatan layanan

web yang dapat diakses oleh aplikasi *mobile*. Pemilihan bobot untuk total variasi, konten, dan gaya sangat mempengaruhi hasil akhir gambar [1].

2.1 Machine Learning

Machine Learning merupakan ilmu mesin yang dikembangkan berdasarkan algoritma dan model secara statistik untuk menjalankan tugas dengan sendirinya tanpa arahan dari pengguna. Penggunaan *Machine Learning* pada sistem komputer digunakan untuk memproses data historis yang berukuran besar dan dapat melakukan identifikasi pola suatu data [2].

2.2 Deep Learning

Deep Learning adalah bagian dari *Machine Learning* yang memanfaatkan *artificial neural networks* dengan banyak lapisan (*layers*) untuk menganalisis dan mempelajari data. Dalam konteks ini, “*deep*” mengacu pada kompleksitas struktur yang terdiri dari beberapa lapisan, sementara “*learning*” merujuk pada proses pembelajaran terhadap data [3]. Tujuan utama *Deep Learning* adalah menciptakan representasi data yang lebih abstrak dan kompleks melalui serangkaian lapisan pengolahan data [4].

2.3 Penggabungan Pola Gambar

Penggabungan pola gambar adalah proses mengkombinasikan atau mentransfer atribut visual dari satu gambar ke gambar lainnya. Tujuan utamanya adalah untuk menciptakan gambar baru yang memadukan fitur-fitur atau gaya dari dua atau lebih gambar sumber, menghasilkan hasil yang unik dan menarik. Proses ini melibatkan analisis struktural dan konten visual dari gambar-gambar tersebut serta penggunaan algoritma dan teknik khusus dalam pemrosesan gambar dan grafika komputer. Dengan menggunakan pendekatan yang tepat, penggabungan pola gambar dapat menciptakan hasil yang estetis dan kreatif, serta memberikan kontribusi yang signifikan dalam pengembangan teknologi pemrosesan citra [5].

2.4 Style Transfer

Style Transfer atau transfer gaya adalah teknik dalam penglihatan komputer dan grafis yang melibatkan menghasilkan gambar baru dengan menggabungkan konten dari satu gambar dengan gaya dari gambar lain. Tujuan transfer gaya adalah untuk membuat gambar yang mempertahankan konten dari gambar asli sambil menerapkan gaya visual dari gambar lain [6].

2.5 Fast Style Transfer

Fast Style Transfer adalah sebuah teknik dalam pemrosesan gambar yang memungkinkan mentransfer gaya atau estetika visual dari satu gambar ke gambar lain dengan cepat dan efisien melalui pendekatan berbasis *deep learning*. Dalam teknik ini, model *neural network* menggunakan arsitektur *encoder-decoder* untuk mengekstraksi fitur-fitur konten dari gambar *input* melalui *encoder*, kemudian menerapkan gaya tersebut pada gambar *output* melalui *decoder*. Hal ini memungkinkan penciptaan gambar dengan struktur konten yang sama namun dengan gaya yang berbeda secara otomatis, menciptakan hasil yang unik dan menarik [7].

2.6 TensorFlow

TensorFlow adalah sebuah *framework deep learning open-source* yang dikembangkan oleh tim Google, juga berfungsi sebagai *library* untuk ilmu pengetahuan komputasi [8]. Dengan fleksibilitasnya, TensorFlow dapat diterapkan dalam berbagai bidang dan menyediakan antarmuka yang fleksibel untuk menerapkan berbagai algoritma *Machine Learning*, cocok untuk beragam sistem komputasi [9]. Meskipun memiliki kelebihan seperti dapat digunakan di berbagai perangkat dan tersedia secara gratis, TensorFlow juga memiliki kekurangan seperti pembaruan sistem yang sering memerlukan instalasi ulang dan fitur yang lebih terbatas bagi pengguna Windows dibandingkan dengan pengguna Linux.

2.7 TensorFlow Hub

TensorFlow Hub adalah sebuah platform yang menyediakan beragam model *Machine Learning* yang telah dilatih sebelumnya, serta modul-modul yang siap digunakan dalam pengembangan aplikasi *Machine Learning*. Platform ini memungkinkan para pengembang untuk dengan mudah mencari, menemukan, dan menggunakan model-model yang telah dilatih tersebut dalam berbagai proyek *Machine Learning* [10]. TensorFlow Hub menyediakan akses ke berbagai jenis model, termasuk model untuk tugas-tugas seperti klasifikasi gambar, pemrosesan bahasa alami, dan lainnya. Selain itu, TensorFlow Hub juga memfasilitasi kolaborasi antara para peneliti dan pengembang dengan memungkinkan untuk membagikan model-model yang telah dilatih dengan komunitas TensorFlow yang lebih luas.

2.8 Google Colab

Google Colab, atau dikenal sebagai Google Collaboratory, adalah sebuah alat yang disediakan secara gratis oleh Google untuk keperluan penelitian, yang memanfaatkan komputasi awan atau sistem penyimpanan awan [11]. Platform ini memungkinkan pengguna untuk menulis dan mengeksekusi kode Python di lingkungan *cloud* tanpa perlu menginstal atau mengonfigurasi apa pun di mesin lokal. Google Colab menyediakan akses ke sumber daya komputasi yang kuat, termasuk CPU, GPU, dan TPU, serta menyediakan penyimpanan *cloud* yang terintegrasi. Pengguna dapat membuat dan berbagi *notebook* interaktif yang berisi kode, teks naratif, grafik, dan visualisasi data.

2.9 Python

Python adalah bahasa pemrograman yang dinamis dan tinggi tingkatnya. Ia dikenal sebagai bahasa interpreter karena kemampuannya untuk langsung mengubah *source code* menjadi kode mesin saat program dijalankan. Python mendukung pendekatan Pemrograman Berorientasi Objek dalam pengembangan aplikasi, sambil tetap mudah dipelajari dan menyediakan beragam struktur data tingkat tinggi. Dengan sintaksis dinamis dan sifat interpretatifnya, Python cocok digunakan untuk skrip dan pengembangan aplikasi yang cepat. Bahasa ini juga mendukung berbagai pola pemrograman, seperti gaya berorientasi objek, imperatif, fungsional, dan prosedural. [12].

3. METODE PENELITIAN

Penelitian ini menggunakan jenis penelitian eksperimental, dimana melibatkan implementasi sebuah metode tertentu untuk mencapai tujuan tertentu. Kemudian, eksperimen dilakukan untuk mengevaluasi kinerja metode yang diusulkan. Selanjutnya, analisis terhadap hasil eksperimen dilakukan untuk memahami kondisi dari penelitian [13].

1. Pengumpulan Data: Data yang digunakan dalam penelitian ini terdiri dari dua jenis gambar, yaitu gambar gaya (*style images*) dan gambar konten (*content images*). Gambar gaya dipilih dari berbagai gaya lukisan terkenal dan karya seni visual lainnya, sedangkan gambar konten diambil dari berbagai sumber gambar fotografi.
2. Pengolahan Data: Setiap gambar dimuat dan diproses untuk memastikan konsistensi ukuran dan formatnya. Selanjutnya, gambar-gambar tersebut dinormalisasi agar memiliki rentang nilai piksel yang seragam dan disesuaikan dengan format yang diperlukan untuk pelatihan model.
3. Implementasi *Fast Style Transfer*: Implementasi *Fast Style Transfer* dilakukan menggunakan TensorFlow dengan memanfaatkan arsitektur model *deep learning* yang telah terbukti efektif dalam transfer gaya. Dipilih arsitektur yang telah teruji kinerjanya dalam literatur terkait dan melakukan konfigurasi *hyperparameter*, seperti *learning rate* dan jumlah iterasi.
4. Evaluasi Model: Untuk mengevaluasi kinerja model *Fast Style Transfer* yang dibuat, menggunakan metrik kualitatif, serta penilaian visual dari hasil penggabungan pola gambar. Pengujian dilakukan pada kumpulan data validasi yang terpisah, dan hasilnya dibandingkan dengan gambar-gambar referensi yang memiliki gaya yang diinginkan.
5. Analisis Hasil: Hasil dari eksperimen dievaluasi dan dianalisis untuk mengidentifikasi kelebihan dan kekurangan dari metode yang diusulkan. Selain itu, juga melakukan

perbandingan dengan metode *Fast Style Transfer* lainnya atau penelitian terkait untuk menilai kontribusi unik dari pendekatan yang dilakukan.

4. PEMBAHASAN

Proyek penelitian ini adalah proses penggabungan pola dari dua buah gambar dengan metode *Fast Style Transfer* yang menggunakan bahasa pemrograman Python untuk membuat *source code*, TensorFlow sebagai *framework*, TensorFlow Hub sebagai platform model, dan Google Colab sebagai tempat menulis dan mengeksekusi kode Python di penyimpanan *cloud*.

4.1 Mengimport *Libraries* dan *Modules*

```
import functools
import os
import cv2

from matplotlib import gridspec
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
import tensorflow_hub as hub

print("TF Version: ", tf.__version__)
print("TF Hub version: ", hub.__version__)
print("Eager mode enabled: ", tf.executing_eagerly())
print("GPU available: ", tf.config.list_physical_devices('GPU'))
```

Gambar 4.1 Kode untuk Mengimport *Libraries* dan *Modules*

Kode di atas menggunakan beberapa *library* yaitu TensorFlow, OpenCV, dan Matplotlib untuk melakukan pengolahan gambar menggunakan model *Fast Style Transfer*. Pertama, mengimpor beberapa modul yang dibutuhkan seperti `functools`, `os`, `cv2` (OpenCV), `matplotlib.pyplot`, `numpy`, `tensorflow`, dan `tensorflow_hub`. Modul-modul ini menyediakan fungsi-fungsi yang diperlukan untuk pembacaan gambar, visualisasi, manipulasi *array*, dan implementasi model *Fast Style Transfer*. Selanjutnya, mencetak informasi versi dari TensorFlow dan TensorFlow Hub yang sedang digunakan, serta memeriksa apakah mode *eager execution* aktif dan apakah GPU tersedia untuk komputasi. Hal ini penting untuk memastikan bahwa TensorFlow dapat berjalan dengan baik dan memanfaatkan sumber daya komputasi yang tersedia secara efisien.

4.2 Menentukan Fungsi *Loading* dan Visualisasi Gambar

```
def crop_center(image):
    """Mengembalikan gambar persegi yang dipotong."""
    shape = image.shape
    new_shape = min(shape[1], shape[2])
    offset_y = max(shape[1] - shape[2], 0) // 2
    offset_x = max(shape[2] - shape[1], 0) // 2
    image = tf.image.crop_to_bounding_box(
        image, offset_y, offset_x, new_shape, new_shape)
    return image

@functools.lru_cache(maxsize=None)
def load_image(image_url, image_size=(256, 256), preserve_aspect_ratio=True):
    """Memuat dan memproses gambar."""
    image_path = tf.keras.utils.get_file(os.path.basename(image_url)[-128:], image_url)
    img = tf.io.decode_image(
        tf.io.read_file(image_path),
        channels=3, dtype=tf.float32)[tf.newaxis, ...]
    img = crop_center(img)
    img = tf.image.resize(img, image_size, preserve_aspect_ratio=True)
    return img

def show_n(images, titles=('',)):
    n = len(images)
    image_sizes = [image.shape[1] for image in images]
    w = (image_sizes[0] * 6) // 320
    plt.figure(figsize=(w * n, w))
    gs = gridspec.GridSpec(1, n, width_ratios=image_sizes)
    for i in range(n):
        plt.subplot(gs[i])
        plt.imshow(images[i][0], aspect='equal')
        plt.axis('off')
        plt.title(titles[i] if len(titles) > i else '')
    plt.show()
```

Gambar 4.2 Kode untuk Menentukan Fungsi *Loading* dan Visualisasi Gambar

Kode di atas mendefinisikan beberapa fungsi yang digunakan untuk memuat, memproses, dan menampilkan gambar dalam lingkungan Python menggunakan TensorFlow dan Matplotlib. Fungsi 'crop_center' mengambil gambar sebagai *input* dan mengembalikan gambar persegi yang dipotong dari tengahnya. Ini dilakukan dengan menghitung dimensi baru untuk gambar persegi berdasarkan dimensi minimum dari lebar dan tinggi gambar asli, lalu menghitung *offset* yang diperlukan untuk memotong gambar agar tetap berada di tengah. Fungsi 'load_image' digunakan untuk memuat gambar dari URL yang diberikan, memprosesnya dengan memastikan ukuran gambar tetap konsisten dan sesuai dengan ukuran yang ditentukan, serta mengonversinya menjadi tensor float dengan tipe data `tf.float32`. Proses pemrosesan melibatkan pembacaan file gambar, dekompresi, pemangkasan gambar agar menjadi persegi, dan penyesuaian ukuran gambar sesuai dengan parameter yang diberikan. Selain itu, fungsi ini menggunakan 'functools.lru_cache' untuk menyimpan hasil pemrosesan gambar dalam *cache* agar tidak perlu memuat ulang gambar yang sama jika dipanggil kembali dengan URL yang sama. Fungsi 'show_n' digunakan untuk menampilkan satu atau lebih gambar dalam satu baris dengan judul opsional. Ini menggunakan Matplotlib untuk membuat subplot sesuai dengan jumlah gambar yang diberikan, kemudian menampilkan setiap gambar dengan judul yang sesuai.

4.3 Memuat Contoh Gambar

```
content_image_url = 'https://i.pinimg.com/564x/1b/b2/79/1bb27932b3e9de4315c29d6daaac8c0f.jpg'
style_image_url = 'https://i.pinimg.com/564x/8e/30/4a/8e304aac188acd1f61943e58752a2115.jpg'
output_image_size = 384

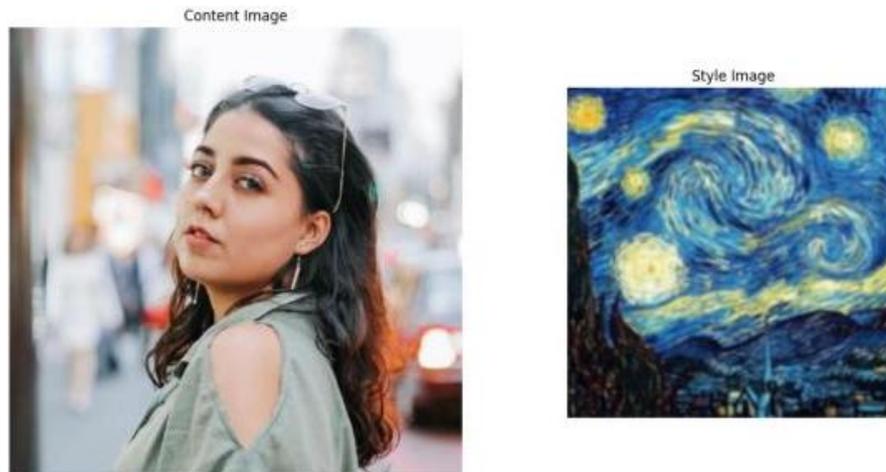
content_img_size = (output_image_size, output_image_size)
style_img_size = (256, 256)

content_image = load_image(content_image_url, content_img_size)
style_image = load_image(style_image_url, style_img_size)
style_image = tf.nn.avg_pool(style_image, ksize=[3,3], strides=[1,1], padding='SAME')
show_n([content_image, style_image], ['Content Image', 'Style Image'])
```

Gambar 4.3 Kode untuk Memuat Contoh Gambar

Kode di atas bertujuan untuk memuat dan menampilkan gambar-gambar yang akan digunakan dalam proses *Fast Style Transfer*. Pertama, URL gambar konten (`content_image_url`) dan gambar gaya (`style_image_url`) ditentukan. Selanjutnya, ukuran *output* gambar (`output_image_size`) dan ukuran gambar gaya (`style_img_size`) didefinisikan. Setelahnya, gambar konten dan gaya dimuat menggunakan fungsi `load_image()`, yang merupakan fungsi yang tidak terdefinisi dalam kode tersebut, dan disesuaikan ukurannya dengan `content_img_size` dan `style_img_size`. Namun, terdapat sebuah modifikasi pada gambar gaya, dimana gambar tersebut diubah menggunakan fungsi `tf.nn.avg_pool()` dengan parameter `ksize=[3,3]`, `strides=[1,1]`, dan `padding='SAME'`. Terakhir, gambar konten dan gaya ditampilkan menggunakan fungsi `show_n()`, yang juga merupakan fungsi yang tidak terdefinisi dalam kode tersebut, dengan label 'Content Image' untuk gambar konten dan 'Style Image' untuk gambar gaya.

Berikut adalah gambar konten (*content image*) dan gambar gaya (*style image*) yang digunakan.



Gambar 4.4 Content Image dan Style Image yang Digunakan

```
hub_handle = 'https://tfhub.dev/google/magenta/arbitrary-image-stylization-v1-256/2'
hub_module = hub.load(hub_handle)
```

Gambar 4.5 Kode untuk Mengimport dan Memuat Model

Kode di atas digunakan untuk mengimpor dan memuat model *pre-trained* untuk melakukan *stylization* gambar menggunakan TensorFlow Hub. Pada baris pertama, sebuah URL yang menunjukkan ke model TensorFlow Hub spesifik untuk *stylization* gambar, disimpan dalam variabel 'hub_handle'. URL ini mengacu pada model yang telah dilatih sebelumnya dengan TensorFlow, yang dapat mentransfer gaya dari satu gambar ke gambar lainnya. Pada baris kedua, fungsi 'hub.load()' digunakan untuk memuat model tersebut dari URL yang telah ditentukan. Model ini kemudian dapat digunakan untuk mentransfer gaya dari gambar-gambar gaya tertentu ke gambar-gambar konten yang diinginkan.

```
outputs = hub_module(content_image, style_image)
stylized_image = outputs[0]
```

Gambar 4.6 Kode untuk Menghasilkan *Output* dari Modul

Dalam kode di atas, terdapat dua variabel yang digunakan, yaitu ‘content_image’ dan ‘style_image’, yang masing-masing berisi gambar konten dan gambar gaya. Kedua gambar ini akan digunakan sebagai *input* untuk sebuah modul TensorFlow Hub, yang disimpan dalam variabel ‘hub_module’. Kemudian, hasil dari proses *Fast Style Transfer* disimpan dalam variabel ‘outputs’. Karena hasil dari modul TensorFlow Hub adalah sebuah list, maka ‘stylized_image’ diinisialisasi dengan elemen pertama dari list tersebut, yang merupakan gambar yang telah diaplikasikan gaya dari gambar gaya (*style image*) ke gambar konten (*content image*).

4.4 Visualisasi Gambar *Input* dan Gambar yang Dihasilkan

```
show_n([content_image], titles=['Original Content Image'])
show_n([style_image], titles=['Style Image'])
show_n([stylized_image], titles=['Stylized Image'])
```

Gambar 4.7 Kode untuk Memvisualisasikan Gambar *Input* dan Gambar *Output*

Dari kode di atas, terdapat tiga pemanggilan fungsi ‘show_n()’, masing-masing untuk menampilkan gambar pada tiga kondisi yang berbeda. Pada pemanggilan pertama, fungsi digunakan untuk menampilkan gambar konten asli untuk menjadi dasar atau konten yang akan diubah gayanya. Pada pemanggilan kedua, fungsi digunakan untuk menampilkan gambar gaya sebagai sumber gaya atau estetika visual yang akan ditransfer ke gambar konten. Pada pemanggilan ketiga, fungsi digunakan untuk menampilkan gambar hasil dari proses penggabungan pola, di mana gaya dari gambar gaya ditransfer ke gambar konten. Setiap pemanggilan fungsi ‘show_n()’ disertai dengan argumen ‘titles’ yang menentukan judul untuk setiap gambar yang ditampilkan, memberikan konteks yang jelas tentang kondisi dari setiap tampilan gambar.

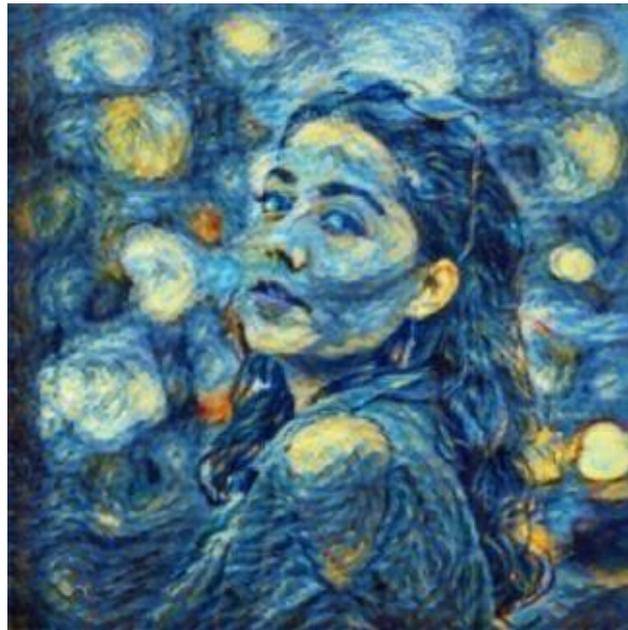
4.5 Menyimpan Hasil Akhir atau *Output*

```
output_image_path = 'stylized_image.jpg'
stylized_image_np = np.array(stylized_image[0] * 255, dtype=np.uint8)
cv2.imwrite(output_image_path, cv2.cvtColor(stylized_image_np, cv2.COLOR_RGB2BGR))
print(f"Stylized image telah disimpan di: {output_image_path}")
```

Gambar 4.8 Kode untuk Menyimpan Hasil Akhir atau *Output*

Kode di atas bertujuan untuk menyimpan gambar yang telah dihasilkan melalui proses stilisasi. Pertama, ditentukan jalur atau *path* tempat gambar hasil akan disimpan dengan menetapkan nilai variabel ‘output_image_path’ ke ‘stylized_image.jpg’. Selanjutnya, gambar hasil stilisasi yang disimpan dalam bentuk tensor (*stylized_image*) dikonversi menjadi array NumPy menggunakan ‘np.array()’. Array ini kemudian dikonversi menjadi tipe data uint8 dengan rentang nilai antara 0 dan 255, sesuai dengan format yang dibutuhkan oleh fungsi ‘cv2.imwrite()’ untuk menyimpan gambar. Gambar hasil ini kemudian disimpan menggunakan fungsi ‘cv2.imwrite()’, dengan proses konversi warna dari format RGB ke BGR menggunakan ‘cv2.cvtColor()’, karena OpenCV (cv2) membutuhkan gambar dalam format BGR untuk menyimpannya. Akhirnya, pesan yang memberi tahu lokasi penyimpanan gambar disampaikan melalui fungsi ‘print()’, dengan mencetak nilai variabel ‘output_image_path’. Dengan demikian, setelah eksekusi kode ini, gambar hasil stilisasi akan tersimpan dengan nama file ‘stylized_image.jpg’ di lokasi yang ditentukan.

Berikut adalah gambar *output* dari proses penggabungan pola dua gambar tersebut.



Gambar 4.9 Gambar *Output* dari Penggabungan Dua Pola Gambar

5. KESIMPULAN

Berdasarkan penelitian ini, dilakukan penggabungan pola dari dua gambar menggunakan metode *Fast Style Transfer* dengan menggunakan TensorFlow. Melalui langkah-langkah yang dijabarkan, penelitian ini berhasil mengoptimalkan proses transfer gaya antara dua gambar dengan memanfaatkan kemampuan TensorFlow dan teknik-teknik terkini dalam *deep learning*. Hasilnya, teknik *Fast Style Transfer* mampu menciptakan gambar baru dengan estetika visual yang unik dan menarik, menggabungkan gaya dari gambar gaya dengan konten dari gambar konten. Dengan demikian, penelitian ini tidak hanya menghadirkan solusi untuk kendala tradisional dalam transfer pola gambar, tetapi juga membuka potensi baru dalam eksplorasi kreativitas seni visual dengan memanfaatkan teknologi terkini.

Selain itu, penelitian ini juga menunjukkan bahwa TensorFlow, sebagai *framework Deep Learning*, bersama dengan sumber daya yaitu TensorFlow Hub dan Google Colab, menjadi alat yang sangat berguna dalam implementasi dan eksperimen dalam bidang pengolahan gambar. Dengan menggunakan teknologi ini, para peneliti dan praktisi seni visual dapat mengembangkan karya-karya yang inovatif dan menginspirasi dengan lebih efisien, mengatasi kendala tradisional dalam proses kreatif yang dibuat.

DAFTAR PUSTAKA

- [1] S. N. Paradilaga, "PENERAPAN KOMBINASI POLA DUA GAMBAR DENGAN METODE NEURAL STYLE TRANSFER UNTUK APLIKASI MOBILE," Universitas Atma Jaya Yogyakarta, 2020.
- [2] S. Junaidi *et al.*, *Buku Ajar Machine Learning*. Jambi: PT. Sonpedia Publishing Indonesia, 2024.
- [3] Noviyanti, Yuliana, F. S. D. Arianto, and L. F. M. Horhoruw, *Teknologi Deep Learning: Eksplorasi Kemampuan Metode Generative Adversarial Networks (GANS)*. Ponorogo: Uwais Inspirasi Indonesia, 2023.
- [4] Y. Heryadi and E. Irwansyah, *Deep Learning dan Aplikasinya di Bidang Informasi Geospasial*. Depok: PT. Artifisia Wahana Informa Teknologi, 2020.

-
- [5] R. Dijaya, *Buku Ajar Pengolahan Citra Digital*. Sidoarjo: UMSIDA Press, 2023.
- [6] Anonym, "Style Transfer," Papers With Code. Accessed: Apr. 19, 2024. [Online]. Available: <https://paperswithcode.com/task/style-transfer>
- [7] Hidayaturrehman, "Style Transfer: Memindahkan Style Lukisan ke dalam Gambar Lain dengan Menggunakan Machine Learning," BINUS UNIVERSITY. [Online]. Available: <https://socs.binus.ac.id/2020/11/22/style-transfer-memindahkan-style-lukisan-ke-dalam-gambar-lain-dengan-menggunakan-machine-learning/>
- [8] D. J. P. Manajang, S. R. U. A. Sompie, and A. Jacobus, "Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor," *J. Tek. Inform.*, vol. 15, no. 3, pp. 171–178, 2020, doi: <https://doi.org/10.35793/jti.15.3.2020.29775>.
- [9] I. B. A. Peling, I. M. P. A. Ariawan, and G. B. Subiksa, "Deteksi Bahasa Isyarat Menggunakan Tensorflow Lite dan American Sign Language (ASL)," *J. Krisnadana*, vol. 3, no. 2, pp. 90–100, 2024, doi: <https://doi.org/10.58982/krisnadana.v3i2.534>.
- [10] S. Beeharry, "Using Tensorflow hub to deploy machine learning models," Medium. Accessed: Apr. 19, 2024. [Online]. Available: <https://medium.com/@siamsoftlab/using-tensorflow-hub-to-deploy-machine-learning-models-4b5167dcbb4a>
- [11] M. Maesaroh, T. N. Padilah, and J. H. Jaman, "Penerapan Algoritma K-Means Clustering pada Pengelompokan Daerah Penyebaran Diare di Provinsi Jawa Barat," *JATI J. Mhs. Tek. Inform.*, vol. 7, no. 4, pp. 2783–2787, 2023, doi: <https://doi.org/10.36040/jati.v7i4.7208>.
- [12] A. Suharto, *Fundamental Bahasa Pemrograman Phyton*. Purbalingga: Eureka Media Aksara, 2023. doi: 9786234877724.
- [13] Rukminingsih, G. Adnan, and M. A. Latief, *Metode Penelitian Pendidikan*. Yogyakarta: Erhaka Utama, 2020.