

Design of a Generative AI Image Similarity Test Application and Handmade Images Using Deep Learning Methods

Rancang Bangun Aplikasi Uji Kemiripan Gambar AI Generative dan Gambar Buatan Tangan Menggunakan Metode Deep Learning

Rifqi Alfaesta Prawiratama¹

¹ Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

^{1*} yuuki0@umsida.ac.id

*: *Penulis korespondensi (corresponding author)*

Informasi Artikel

Received: July 2023

Revised: August 2023

Accepted: September 2023

Published: October 2023

Abstract

Purpose: The aim of this research is to develop a classification model using the Transformer approach, specifically the BEiT architecture, to differentiate between handmade images and AI Generative Art. The objective is to ensure the authenticity of art and address ethical and legal concerns related to AI Generative Art.

Design/methodology/approach: The study utilizes the BEiT architecture within the Transformer approach to create a classification model. The training process uses Bidirectional Encoder representation from Image Transformers (BEiT) to improve image classification. The primary datasets are collected through a Python image scraper program. The BEiT workflow includes Pre-training, Masking, Inpainting, and Interface Design with Gradio.

Findings/result: The Transformer model, using the BEiT architecture, achieves 96.34% accuracy and 0.0921 loss in differentiating handmade images and AI Generative Art. The model demonstrates a balanced precision and recall in each category, outperforming previous methods such as Convolutional Neural Network (CNN) and VGG16. The language used is clear, objective, and value-neutral, with a formal register and precise word choice. No changes in content were made. The Gradio interface was used to successfully test the model.

Originality/value/state of the art: The research presents a state-of-the-art classification model that uses the Transformer approach, specifically the BEiT architecture, to differentiate between handmade and AI Generative Art images. The research presents a state-of-the-art classification model that uses the Transformer approach, specifically the BEiT architecture, to differentiate between

handmade and AI Generative Art images. The text adheres to conventional structure and formatting features, including consistent citation and footnote style. The sentences and paragraphs create a logical flow of information with causal connections between statements. The text is free from grammatical errors, spelling mistakes, and punctuation errors. Additionally, the research is enhanced by the innovative approach to data collection using a Python image scraper program.

Abstrak

Keywords: Deep Learning; AI Generative; Transformers; BEiT; Image Classification

Kata kunci: Deep Learning; AI Generatif; Transformers; BEiT; Klasifikasi Gambar

Tujuan: Tujuan dari penelitian ini adalah untuk mengembangkan model klasifikasi menggunakan pendekatan Transformer, khususnya arsitektur BEiT, untuk membedakan antara gambar buatan tangan dan Seni Generatif AI. Tujuannya adalah untuk memastikan keaslian karya seni dan mengatasi masalah etika dan hukum yang terkait dengan Seni Generatif AI.

Perancangan/metode/pendekatan: Penelitian ini menggunakan arsitektur BEiT dalam pendekatan Transformer untuk membuat model klasifikasi. Proses pelatihan menggunakan representasi Bidirectional Encoder dari Image Transformers (BEiT) untuk meningkatkan klasifikasi gambar. Dataset utama dikumpulkan melalui program pengikis gambar Python. Alur kerja BEiT meliputi Pra-pelatihan, Masking, Inpainting, dan Desain Antarmuka dengan Gradio.

Hasil: Model Transformer, menggunakan arsitektur BEiT, mencapai akurasi 96,34% dan kehilangan 0,0921 dalam membedakan gambar buatan tangan dan AI Generative Art. Model ini menunjukkan presisi dan recall yang seimbang di setiap kategori, mengungguli metode sebelumnya seperti Convolutional Neural Network (CNN) dan VGG16. Bahasa yang digunakan jelas, objektif, dan netral dari nilai, dengan register formal dan pilihan kata yang tepat. Tidak ada perubahan dalam konten yang dibuat. Antarmuka Gradio digunakan untuk menguji model dengan sukses.

Keaslian/ state of the art: The research presents a state-of-the-art classification model that uses the Transformer approach, specifically the BEiT architecture, to differentiate between handmade and AI Generative Art images. The research presents a state-of-the-art classification model that uses the Transformer approach, specifically the BEiT architecture, to differentiate between handmade and AI Generative Art images. The text adheres to conventional

structure and formatting features, including consistent citation and footnote style. The sentences and paragraphs create a logical flow of information with causal connections between statements. The text is free from grammatical errors, spelling mistakes, and punctuation errors. Additionally, the research is enhanced by the innovative approach to data collection using a Python image scraper program.

1. Pendahuluan

Dewasanya di zaman modern ini teknologi semakin cepat berkembang dan teknologi juga memberikan efisiensi waktu dan keringanan dalam melakukan suatu aktivitas maupun kegiatan sehari-hari. Teknologi yang telah banyak berkembang saat ini adalah kecerdasan buatan (AI) [1]. Kecerdasan buatan (AI) sendiri adalah bidang ilmu komputer yang berfokus untuk melakukan pengembangan terhadap mesin dan sistem agar dapat melaksanakan tugas secara otomatis mirip seperti kemampuan berfikir manusia [2]. Kecerdasan buatan (AI) berupaya menciptakan mesin yang mampu belajar, berpikir dan menjalankan berbagai tugas seperti pengenalan suara, pemrosesan bahasa alami, penalaran, serta pengambilan keputusan [3]. Hal ini tentu memberi pengaruh besar pada bidang seni dan seni rupa, salah satunya adalah *AI Generative Art*. *AI Generative Art* sendiri adalah karya seni generatif menggunakan algoritma dan teknologi kecerdasan buatan (AI) untuk menciptakan karya seni yang unik [4].

Deep learning sendiri adalah bagian dari kecerdasan buatan (AI) yang menggunakan jaringan syaraf tiruan (*neural networks*) dengan banyak *layers* (*deep neural networks*) untuk memecahkan masalah klasifikasi yang kompleks [5]. Model *deep learning* sendiri terinspirasi oleh struktur dan fungsi dari otak manusia, dan mampu belajar secara otomatis dari data yang tidak terstruktur dan tidak berlabel [6]. *Deep learning* telah digunakan dalam berbagai aplikasi seperti pengenalan objek, pengenalan suara, dan banyak hal lainnya [7].

Pada karya seni buatan dari *AI Generative Art* juga melibatkan penggunaan algoritma *deep learning* yang dapat menghasilkan karya seni berdasarkan pola, aturan dan dataset yang diberikan. Algoritma tersebut dapat mempelajari data dari dataset seni yang ada, memahami elemen-elemen visual, dan menghasilkan karya yang serupa atau bahkan baru secara kreatif [8]. Hal ini memberikan kemampuan bagi mesin untuk menghasilkan karya seni dengan karakteristik yang unik dan seringkali mengeksplorasi kombinasi dan variasi tertentu yang dapat menghasilkan suatu karya yang belum pernah dipelajari oleh seniman manusia sebelumnya [9].

Penelitian lain yang telah membahas terkait perancangan metode *Deep Learning* sebagai klasifikasi pada citra adalah “Klasifikasi Citra Bunga dengan Menggunakan *Deep Learning*: CNN (*Convolution Neural Network*)” yang dilakukan oleh Intyanto dengan tujuan untuk membandingkan kinerja model *AlexNet* dan *ResNet* dalam klasifikasi citra bunga menggunakan *transfer learning*. Metode yang digunakan adalah CNN (*Convolution Neural Network*) dan hasil dari penelitian ini menunjukkan VGG16 lebih unggul dalam klasifikasi gambar bunga dengan akurasi 0.8 dan loss 0.52 dibandingkan dengan arsitektur penulis [10].

Penelitian lainnya adalah membandingkan performa model *deep learning* yang dilakukan oleh Baharuddin, Azis, dan Hasanuddin membahas tentang “Analisis Performa Metode *K-Nearest*

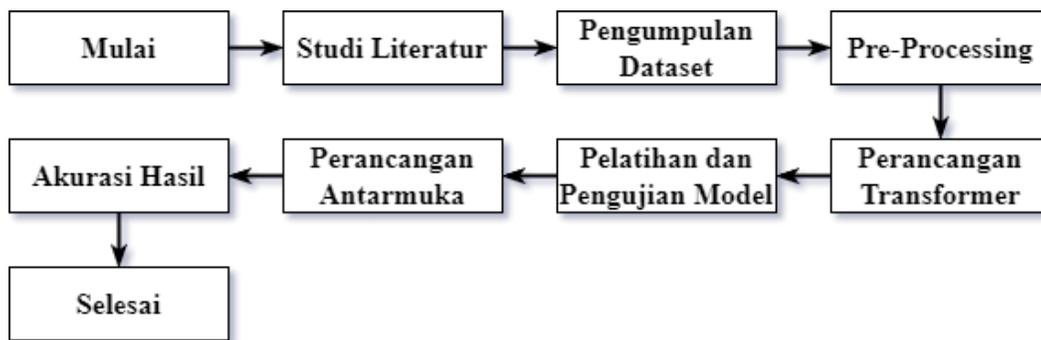
Neighbor untuk Identifikasi Jenis Kaca”. Penelitian ini bertujuan untuk menentukan nilai K yang optimal dalam metode KNN untuk mencapai performa klasifikasi yang terbaik, yang diukur melalui akurasi, presisi, *recall*, dan *F-Measure*. Metode yang digunakan adalah *K-Nearest Neighbor* (KNN) dan hasil dari penelitian ini menunjukkan *K-Nearest Neighbor* (KNN) efektif dalam mengidentifikasi jenis kaca dengan nilai ($K=3$) memiliki akurasi 64%, presisi 63%, *recall* 71%, dan *F-Measure* 67% [11].

Meskipun *AI Generative Art* membawa inovasi luar biasa dalam dunia seni, terdapat tantangan signifikan yang perlu diatasi. Dalam beberapa aspek, Karya-karya *AI Generative Art* meskipun dapat menciptakan visual yang menarik, karya *AI Generative Art* dapat kehilangan orisinalitasnya karena kurangnya elemen emosi, niat, dan pengalaman manusia yang terkandung dalam karya seni rupa [12]. Pertanyaan mendasar pun muncul mengenai apakah karya-karya ini memiliki hak yang setara untuk diakui sebagai seni. Lebih dari itu, masalah etika dan hukum dapat timbul ketika algoritma *AI Generative Art* melanggar hak kekayaan intelektual atau menggunakan data tanpa izin [13]. Untuk mengatasi masalah ini, dibutuhkan sebuah alat yang mampu secara jelas mendeteksi apakah suatu gambar mirip dengan karya hasil dari *AI Generative Art* atau karya buatan tangan. Dengan adanya alat tersebut diharapkan dapat meminimalkan risiko konflik hukum terkait hak cipta dan secara efektif melindungi hak seluruh seniman.

Dengan demikian, kehadiran model untuk mengklasifikasikan karya seni *AI Generative Art* dan karya seni buatan tangan menjadi krusial. Dan diharapkan akan membantu menjaga integritas seni dan mengatasi potensi dari berbagai masalah etika serta hukum yang mungkin dapat timbul kedepannya. Hal ini supaya dapat menciptakan landasan yang lebih kokoh demi perkembangan teknologi *AI Generative Art* secara etis dan adil, serta dapat berdampingan dengan karya seni lainnya.

Berdasarkan ringkasan dari penelitian terdahulu dan fenomena yang telah disajikan sebelumnya menunjukkan hasil dari nilai akurasi yang kurang memuaskan dalam melakukan klasifikasi gambar. Selain itu demi membantu menjaga integritas seni dan mengatasi potensi dari berbagai masalah etika penggunaan gambar tanpa izin. Oleh karena itu perlunya dilakukan penelitian lebih lanjut dengan tujuan untuk lebih memperhatikan pengembangan model yang efisien dengan melakukan perbandingan metode klasifikasi yang lebih komprehensif, pengembangan dataset yang lebih besar dan analisis fitur model yang lebih mendalam.

2. Metode/Perancangan



Gambar 1. Alur Penelitian

2.1. Studi Literatur

Studi pustaka memiliki peran penting dalam penelitian model *deep learning* karena dapat membantu mendalami pemahaman tentang konsep, metode, dan perkembangan terbaru dalam bidang ini. Dengan fokus pada referensi dari berbagai sumber seperti buku, jurnal ilmiah, prosiding konferensi dan artikel yang telah *dipeer-review*, akan menambah wawasan yang lebih dalam tentang konsep-konsep teoritis yang mendasari *deep learning*, metode yang relevan untuk pengembangan model *deep learning* serta temuan-temuan penelitian terbaru dengan masalah penelitian yang akan dihadapi kedepannya.

2.2. Pengumpulan Dataset

Jenis data yang digunakan dalam penelitian ini adalah data primer, yang diperoleh dengan melakukan pengunduhan langsung menggunakan program pengambil gambar (*image scraper*) yang dijalankan menggunakan bahasa pemrograman Python. Pada masing masing gambar karakter anime adalah gambar yang unik dan berbeda antara satu sama lain dan terdapat berbagai macam komposisi seperti karakter, latar belakang dan objek. Secara umum jumlah presentasi dari komposisi pada masing masing gambar karakter anime terdapat sekitar 70% menampilkan karakter dan 30% menampilkan latar belakang atau objek. Data dalam penelitian ini diklasifikasikan menjadi dua kategori yaitu gambar karakter anime buatan tangan dan *AI Generative* dengan jumlah total pada setiap kategorinya sebanyak 11000 gambar yang berformat JPG/JPEG (*Joint Photographic Experts Group*), PNG (*Portable Network Graphics*) dan GIF (*Graphics Interchange Format*). Gambaran visual dari data yang telah dikumpulkan sebelumnya disajikan dalam Gambar 2 dan Gambar 3.



Gambar 2. Dataset Buatan Tangan



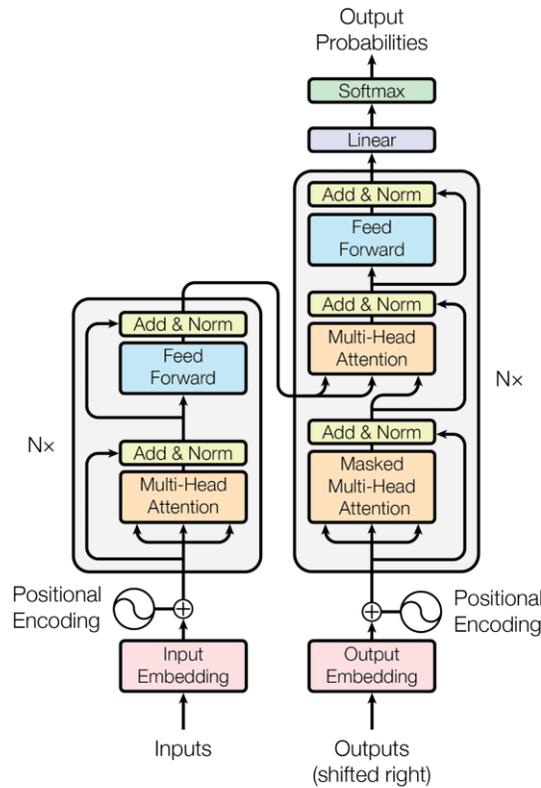
Gambar 3. Dataset Buatan AI Generative

2.3. Pre-Processing

Pre-processing adalah tahap awal dimana seluruh gambar yang ada di dataset diolah dan dipersiapkan agar dapat dilanjutkan ke tahap berikutnya [14]. Salah satu langkah *Pre-processing* yang diterapkan adalah dengan mengubah ukuran gambar (*resize*) sehingga semua gambar memiliki dimensi yang konsisten sesuai dengan konfigurasi yang sudah ditentukan sebelumnya.

2.4. Perancangan Transformer

Pendekatan yang diterapkan dalam penelitian ini adalah dengan memanfaatkan arsitektur Transformer untuk melaksanakan tugas klasifikasi sesuai dengan tujuan penelitian. Arsitektur Transformer ini terdiri dari dua layers diantaranya *Encoder* dan *Decoder*, dan skema arsitektur tersebut dapat dilihat pada Gambar 4.



Gambar 4. Arsitektur Transformer

Dalam kerangka Transformer terdapat dua tahapan utama yaitu Encoder dan Decoder.

1. Tahap *Encoder*

Encoder terdiri dari tumpukan $N = 6$ *layers* identik, di mana setiap *layers* terdiri dari dua *sublayer*:

- a. *Sublayer* pertama mengimplementasikan mekanisme *self-attention multi-head*. Mekanisme *multi-head* mengimplementasikan h *head* yang menerima versi kueri, kunci, dan nilai yang diproyeksikan secara linier (berbeda), masing-masing untuk menghasilkan h *output* secara paralel yang kemudian digunakan untuk menghasilkan hasil akhir.
- b. *Sublayer* kedua adalah jaringan *feed-forward* yang terhubung penuh dan terdiri dari dua transformasi linier dengan aktivasi *Rectified Linear Unit* (ReLU) di antaranya:

$$FFN(x) = \text{ReLU}(W_1x + b_1)W_2 + b_2 \tag{1}$$

Enam *layers encoder* Transformer menerapkan transformasi linier yang sama untuk semua kata dalam urutan *input*, tetapi setiap *layers* menggunakan parameter bobot (W_1 , W_2) dan bias (b_1 , b_2) yang berbeda untuk melakukan proses *encodernya* [15].

Setiap *sublayer* juga digantikan oleh *layers* normalisasi, *layernorm(.)*, yang menormalkan jumlah yang dihitung antara *input sublayer*, x , dan *output* yang dihasilkan oleh *sublayer* itu sendiri, *sublayer*(x).

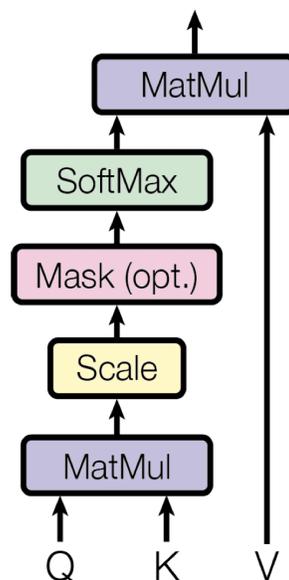
$$\tag{2}$$

$$\text{layernorm}(x + \text{sublayer}(x)) \quad (3)$$

2. Tahap *Decoder*

Decoder juga terdiri dari tumpukan $N = 6$ *layers* identik yang masing-masing terdiri dari tiga *sublayer*:

- a. *Sublayer* pertama menerima *output* sebelumnya dari tumpukan *decoder* menambahkannya dengan informasi posisi dan mengimplementasikan *self-attention multi-head* di atasnya. Sementara *encoder* dirancang untuk memperhatikan semua kata dalam urutan *input* terlepas dari posisinya dalam urutan tersebut, *decoder* dimodifikasi untuk hanya memperhatikan kata-kata sebelumnya [16]. Oleh karena itu prediksi untuk sebuah kata pada posisi i hanya dapat bergantung pada *output* yang diketahui untuk kata-kata yang ada sebelum kata tersebut dalam urutan. Pada mekanisme *multi-head attention* (yang mengimplementasikan beberapa fungsi *single-attention* secara paralel), hal ini dicapai dengan memperkenalkan sebuah *mask* atas nilai-nilai yang dihasilkan oleh perkalian berskala dari matriks Q dan K . *Masking* ini diimplementasikan dengan menekan nilai matriks yang seharusnya berhubungan dengan koneksi ilegal [17].



Gambar 5. Diagram *self-attention* Transformer

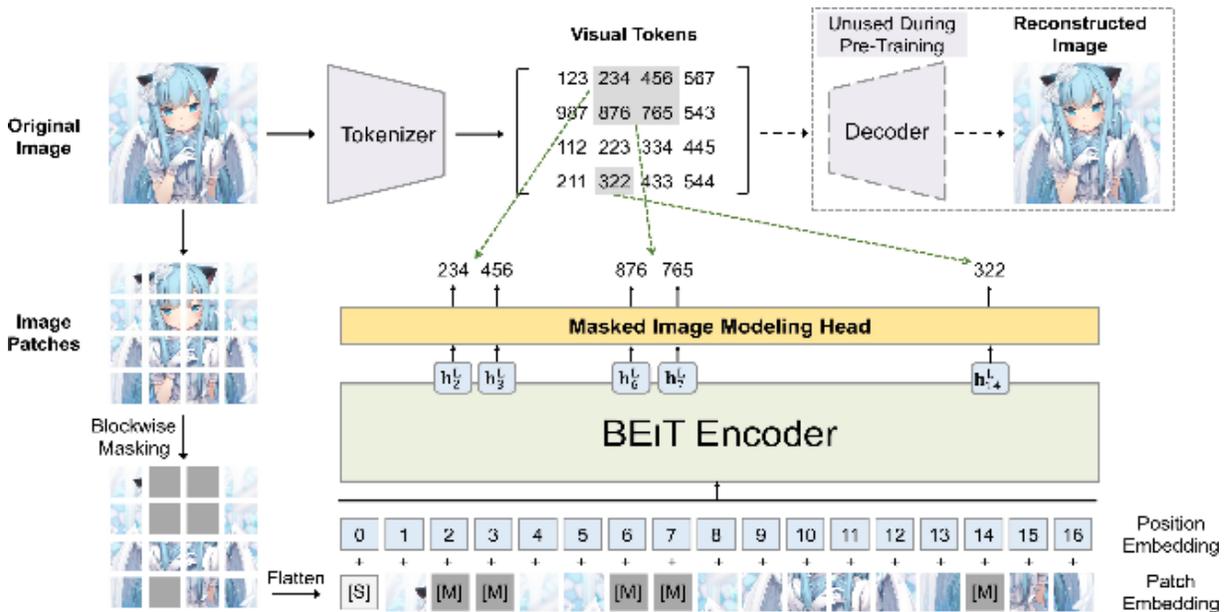
Pada diagram tersebut menunjukkan langkah-langkah dalam mekanisme *self-attention* yang digunakan dalam model transformer. Mekanisme *self-attention* memungkinkan model untuk dapat mempelajari hubungan antara posisi yang berbeda dalam urutan yang sama [18]. Dalam diagram ini menunjukkan enam langkah utama dalam setiap komponen, yaitu:

- **Q, K dan V:** Ini adalah singkatan dari *query*, *key* dan *value*. Mereka adalah bagian dari mekanisme *attention* dalam model Transformer.
- **MatMul (Q dan K):** *query* dan *key* adalah vektor yang mewakili posisi yang berbeda dalam urutan. MatMul menghitung *dot-product* antara *query* dan *key* untuk setiap posisi, *dot-product* sendiri adalah ukuran kesamaan antara dua vektor.

- **SoftMax:** *SoftMax* mendistribusikan bobot pada posisi yang berbeda berdasarkan seberapa mirip *Query* dan *Key*.
 - **Mask (optional):** *Mask* digunakan untuk mencegah model memperhatikan posisi yang tidak relevan.
 - **Scale:** *Scale* mengalikan bobot dengan $1/\sqrt{dk}$, di mana dk adalah dimensi dari *key*.
 - **MatMul (A dan V):** *attention* dan *value* adalah *vektor* yang mewakili posisi yang berbeda dalam urutan. *MatMul* menghitung *dot-product* antara *attention* dan *value* untuk setiap posisi.
- b. *Sublayer* kedua mengimplementasikan mekanisme *self-attention multi-head* yang serupa dengan yang diimplementasikan pada *sublayer* pertama *encoder*. Di sisi *decoder* mekanisme *multi-head* ini menerima pertanyaan dari *sublayer decoder* sebelumnya dan kunci serta nilai dari *output encoder*. Hal ini memungkinkan *decoder* untuk memperhatikan semua kata dalam urutan input.
- c. *Sublayer* ketiga mengimplementasikan jaringan *feed-forward* yang terhubung sepenuhnya. Serupa dengan yang diimplementasikan pada *sublayer* kedua *encoder*. Tiga *sublayer* pada sisi *decoder* juga memiliki koneksi residual di sekitarnya dan digantikan oleh *layers* normalisasi. Pengkodean posisi juga ditambahkan ke penyematan *input decoder* dengan cara yang sama seperti yang dijelaskan sebelumnya untuk *encoder*.

2.5. Pelatihan dan Pengujian Model

Proses *training* pada model Transformer ini menggunakan arsitektur BEiT (*Bidirectional Encoder representation from Image Transformers*) agar dapat meningkatkan kinerja klasifikasi gambar pada berbagai dataset yang telah tersedia sebelumnya. Arsitektur BEiT memiliki beberapa tahapan yang disebut juga sebagai alur kerja BEiT yang dapat dilihat pada Gambar 6.



Gambar 6. Tahapan Arsitektur BEiT

Tahapan pada alur kerja BEiT secara umum terdiri dari empat tahapan utama, yaitu:

1. Tahap *Pre-training*

Pada tahap ini gambar akan dibagi menjadi sejumlah *patch* berukuran kecil yang kemudian digunakan untuk mewakili gambar secara keseluruhan. Proses ini dilakukan untuk meningkatkan efisiensi pelatihan model BEiT. Selain itu proses ini juga dapat membantu model BEiT untuk mempelajari berbagai fitur visual dari gambar [19]. Dengan melihat gambar dari berbagai sudut dan perspektif, model BEiT dapat belajar untuk mengenali berbagai fitur visual, seperti bentuk, warna, dan tekstur.

2. Tahap *Masking*

Pada tahap ini sejumlah bagian dari gambar dihilangkan secara acak. Model BEiT kemudian dilatih untuk mengisi bagian-bagian yang hilang tersebut. Proses ini dilakukan untuk meningkatkan kemampuan model BEiT dalam mengenali berbagai fitur visual [20]. Dengan menghilangkan bagian-bagian dari gambar, model BEiT harus belajar untuk menggunakan informasi yang tersisa untuk mengisi bagian-bagian yang hilang.

3. Tahap *Encoding*

Pada tahap ini model BEiT mengubah gambar menjadi representasi vektor. Representasi vektor ini kemudian digunakan untuk berbagai tugas seperti klasifikasi, pengenalan objek, dan transformasi gambar. Representasi vektor yang dihasilkan oleh model BEiT adalah representasi yang lebih abstrak dari gambar yang asli. Representasi vektor ini dapat digunakan untuk berbagai tugas karena dapat diproses oleh model pembelajaran mesin [21].

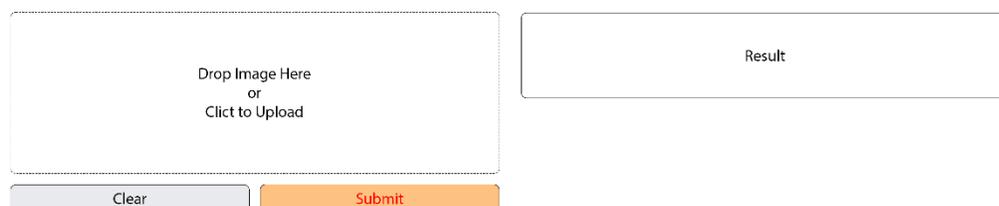
4. Tahap *Decoding*

Pada tahap ini representasi vektor diubah kembali menjadi gambar yang utuh kembali. Gambar yang dihasilkan kemudian dibandingkan dengan gambar asli. Proses ini dilakukan untuk menilai kinerja model BEiT dalam menghasilkan gambar yang mirip dengan gambar asli [20]

2.6. Prancangan Antarmuka dengan Gradio

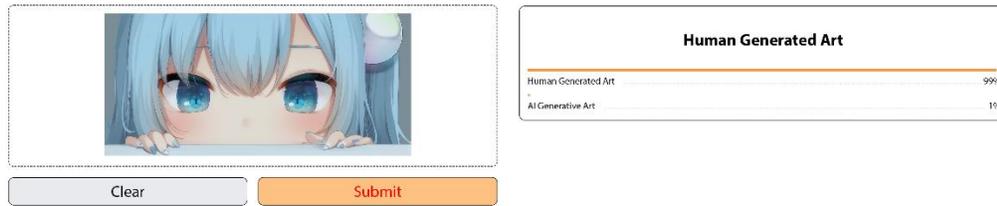
Agar pengguna dapat memanfaatkan fungsionalitas model deep learning dengan cepat dan mudah, antarmuka dirancang menggunakan Gradio. Hal ini karena Gradio memungkinkan pembuatan komponen antarmuka pengguna yang fleksibel dan dapat disesuaikan dengan cepat untuk memastikan kemudahan penggunaan tanpa mengorbankan efisiensi implementasi [22]. Rancangan antarmuka memiliki beberapa tahapan yang dapat dilihat pada Gambar 7 dan Gambar 8.

a. Rancangan antarmuka untuk mengupload gambar



Gambar 7. Halaman Upload Gambar

b. Rancangan antarmuka untuk menampilkan hasil deteksi



Gambar 8. Halaman Hasil Deteksi

2.7. Akurasi Hasil

Evaluasi hasil dilakukan untuk memverifikasi akurasi pada model yang telah dirancang sebelumnya. Tujuan utama dari evaluasi ini adalah untuk mengukur sejauh mana model Transformer mampu menjalankan proses atau tugas klasifikasi dengan baik dengan menguji tingkat keakuratan dan validitas hasil klasifikasi karya seni buatan tangan dan *AI Generative*. Tahap ini mencakup analisis hasil data dan temuan dari penelitian yang kemudian digunakan untuk menyusun kesimpulan akhir menggunakan rumus untuk menghitung akurasi pada model klasifikasi

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}} \times 100\% \quad (3)$$

3. Hasil dan Pembahasan

3.1. Konfigurasi Model

Parameter dan pengaturan yang digunakan selama pelatihan model *deep learning* ini disajikan pada tabel 1.

Tabel 1. Konfigurasi *pre-training model*

No.	Konfigurasi	Niai
1.	Path	beit-base-patch16-224
2.	Layers	12
3.	Head Attention	12
4.	Hidden Size	768
5.	Intermediate	3,072
6.	Dropout Rate	0.1
7.	Channels	3
8.	Input Size	224
9.	Epoch	1
10.	Training Batch	32
11.	Gradient Accumulation Steps	4
12.	Learning Rate	0.00005
13.	Optimizer Beta1	0.9
14.	Optimizer Beta2	0.999
15.	Optimizer Epsilon	0.00000001

3.2. Pelatihan Model

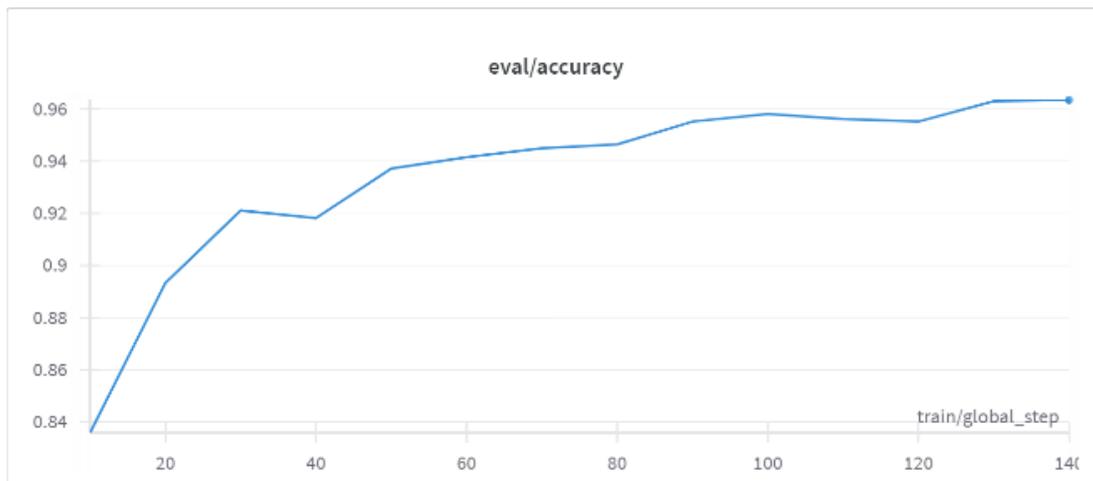
Pelatihan model *deep learning* ini menggunakan microsoft/beit-base-patch16-224 selama 1 epoch yang mengindikasikan bahwa model ini menggunakan konfigurasi dasar (base) dari arsitektur BEiT dengan ukuran patch input 16x16 pixel dan ukuran gambar input 224 x 224 pixel. Hasil dari training membutuhkan waktu 27 menit 33 detik menggunakan Python 3.10.6 pada sistem operasi Linux dengan spesifikasi GPU NVIDIA GeForce RTX 3080. Yang kemudian menghasilkan data yang disajikan pada Tabel 2.

Tabel 2. Hasil *training* model

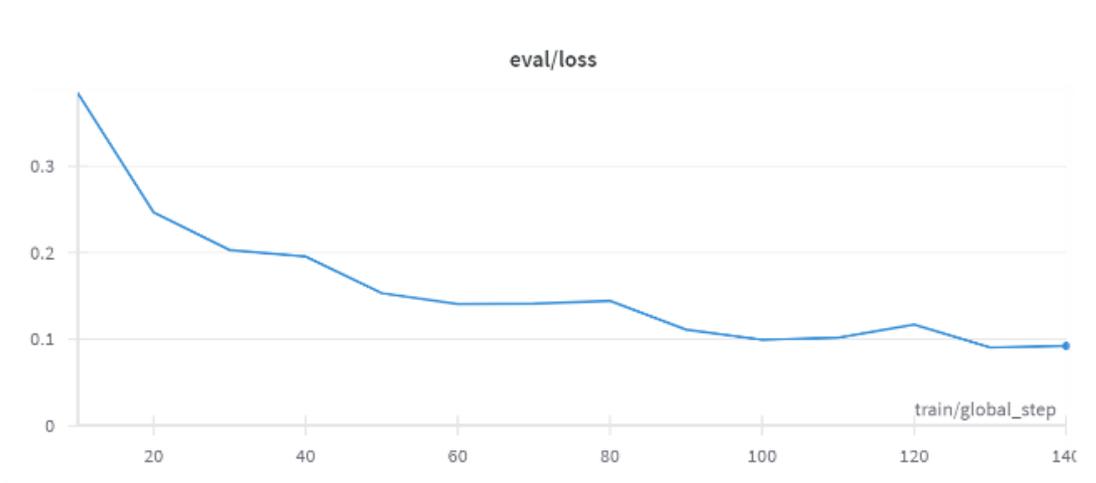
No.	Evaluasi	Niai
1.	Accuracy	0.9634
2.	Loss	0.0921
3.	Train Runtime	1,654.1677
4.	Train Samples per Second	144
5.	Epoch	1
6.	Global Step	144
7.	Learning Rate	0.00000011852415266911
8.	Train Loss	0.1213
9.	Total FLOPs	1,427,721,866,730,012,700

Dari data yang telah disajikan pada tabel sebelumnya, menunjukkan konfigurasi dan hasil *training* pada model dengan jelas. Pada setiap epoch model dapat mencapai akurasi hingga 96.34% dan *loss* sebesar 0.0921. Hasil ini memberikan gambaran mengenai performa model dalam memahami dan mengklasifikasikan gambar buatan tangan dan *AI Generative*. Konfigurasi yang digunakan seperti *learning rate*, *batch size* dan *optimizer* juga turut berperan dalam membentuk hasil akhir ini.

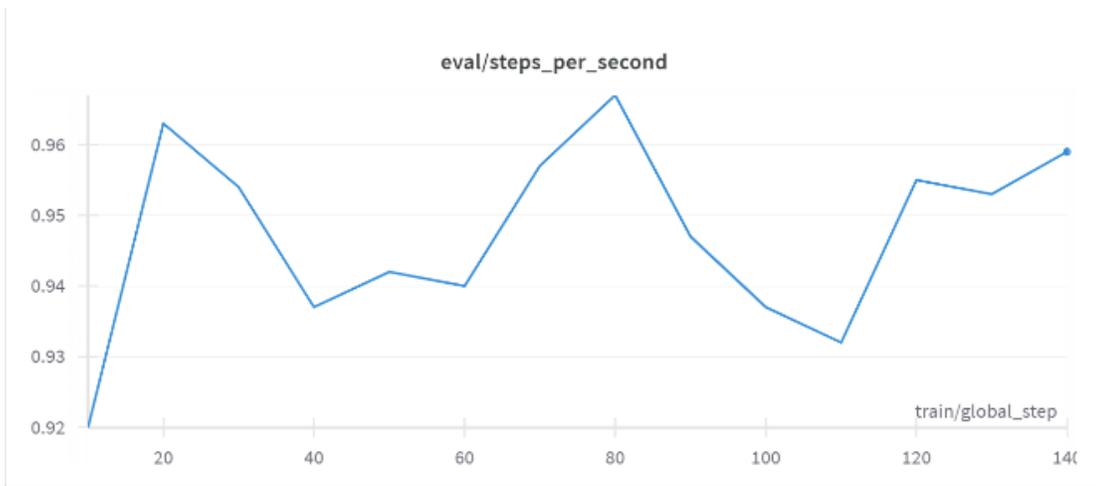
Selain tabel, grafik pembelajaran juga disertakan untuk memberikan visualisasi yang lebih lengkap terhadap perubahan akurasi dan *loss* selama proses pelatihan. Grafik ini menyajikan informasi dinamis yang dapat membantu pemahaman tentang tren dan perkembangan model dari awal hingga akhir pelatihan.



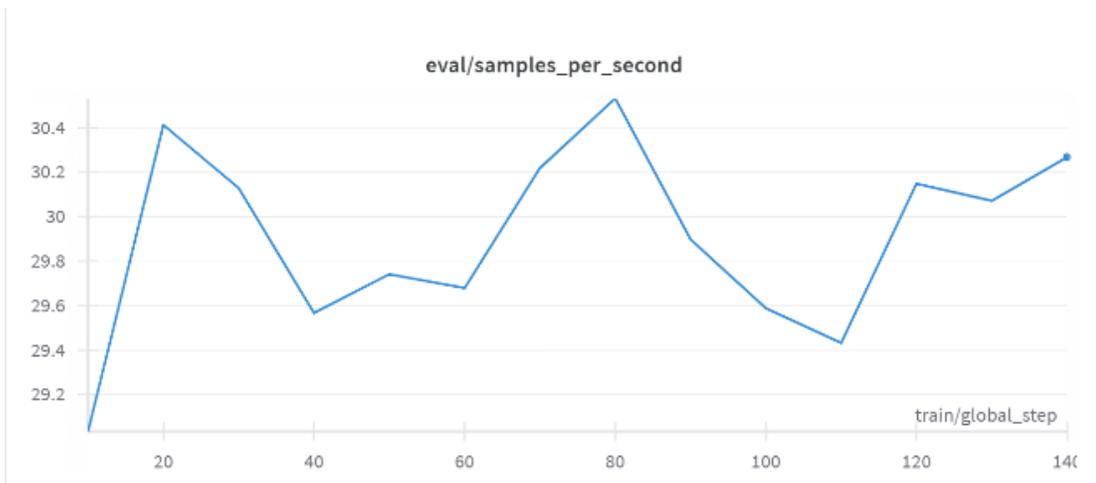
Gambar 9. Grafik akurasi



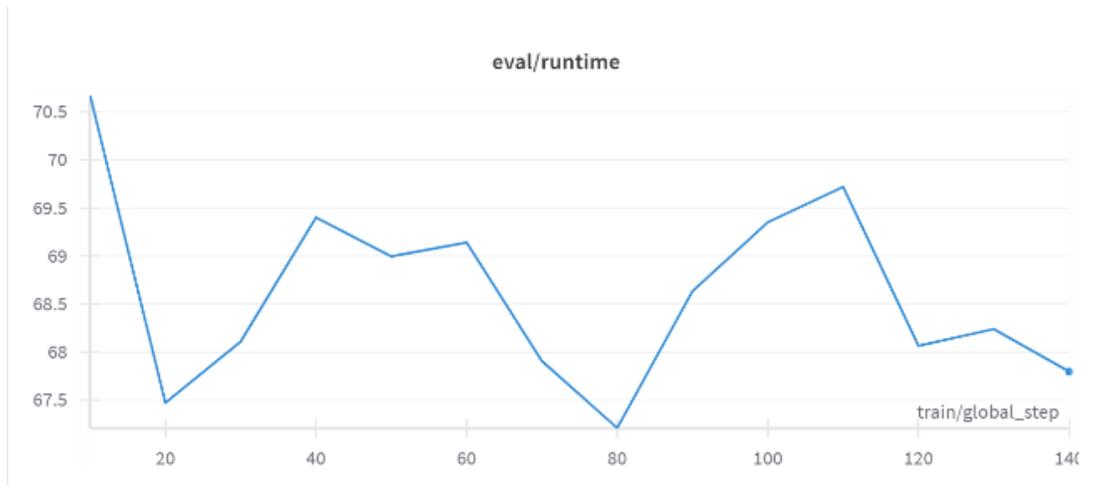
Gambar 10. Grafik loss



Gambar 11. Grafik steps per second



Gambar 12. Grafik samples per second



Gambar 13. Grafik runtime

Dari data tabel konfigurasi dan hasil training model beserta grafik training menjadi penjelasan yang kuat dalam menggambarkan langkah-langkah dan keunggulan model Transformer dan BEiT yang dikembangkan untuk tugas klasifikasi ini. Pemahaman mendalam terhadap konfigurasi, hasil training dan visualisasi grafik membentuk dasar yang komprehensif untuk interpretasi dan kesimpulan yang ditarik dari penelitian ini.

3.3. Pengujian Model

Pengujian model yang telah dilatih sebelumnya dilakukan menggunakan antarmuka Gradio, yang telah terbukti berfungsi dengan baik dalam memfasilitasi penggunaan model. Antarmuka Gradio memastikan kemudahan penggunaan dan memungkinkan evaluasi yang efisien terhadap model klasifikasi yang dikembangkan.



Gambar 14. Pengujian deteksi gambar buatan tangan



Gambar 15. Pengujian deteksi gambar buatan ai generative

Pengujian pada model ini dilakukan menggunakan sepuluh gambar acak untuk mengevaluasi kinerja model klasifikasi. Dalam setiap uji coba *F1 Score* dihitung berdasarkan hasil *precision* dan *recall* yang masing-masing dihitung dengan rumus.

$$\text{Precision} = \frac{(TP)}{(TP)+(FP)} \quad (4)$$

$$\text{Recall} = \frac{(TP)}{(TP)+(FN)} \quad (5)$$

$$F1 \text{ Score} = \frac{2x(\text{Presisi})x(\text{Recall})}{(\text{Presisi})+(\text{Recall})} \quad (6)$$

Tujuan utama pengujian ini adalah untuk mengevaluasi sejauh mana model mampu memberikan keseimbangan antara ketepatan (presisi) dan kelengkapan (*recall*) dalam mengklasifikasikan gambar buatan tangan dan gambar buatan manusia. Hasil akhir dari pengujian ini menghasilkan data yang disajikan pada Tabel 3.

Tabel 3. Hasil Pengujian Deteksi Gambar

Pengujian	Precision	Recall	F1-Score
1.	0.978	1	0.9881
2.	0.995	1	0.9975
3.	0.999	1	0.9995
4.	0.996	1	0.997
5.	0.99	1	0.9949
6.	1	1	1
7.	0.994	1	0.994
8.	0.999	1	0.9995
9.	0.984	1	0.9921
10.	0.991	1	0.9955

Dari data yang disajikan pada Tabel 3 menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall pada masing-masing kategori.

4. Kesimpulan dan Saran

Pada model klasifikasi yang telah dikembangkan menggunakan pendekatan Transformer khususnya dengan memanfaatkan arsitektur BEiT, menunjukkan hasil yang sangat memuaskan dalam pengujian klasifikasi gambar. *F1 Score* yang tinggi pada setiap pengujian mencerminkan keseimbangan yang baik antara presisi dan *recall*. Hal ini menunjukkan bahwa hasilnya dapat diandalkan untuk melakukan suatu deteksi. Dibandingkan dengan metode sebelumnya yang memanfaatkan *Convolutional Neural Network* (CNN) dan model VGG16 yang hanya mencapai akurasi 80%. Sementara itu metode *K-Nearest Neighbor* (KNN) pada penelitian lainnya menunjukkan akurasi sekitar 64%. Meskipun pencapaian tersebut dapat dianggap layak model Transformer menunjukkan performa yang unggul.

Daftar Pustaka

- [1] E. Supriyadi and D. Asih, "IMPLEMENTASI ARTIFICIAL INTELLIGENCE (AI) DI BIDANG ADMINISTRASI PUBLIK PADA ERA REVOLUSI INDUSTRI 4.0," *Jurnal RASI*, vol. 2, Jan. 2020, doi: 10.52496/rasi.v2i2.62.

-
- [2] A. Zein, “Kecerdasan Buatan Dalam Hal Otomatisasi Layanan,” *Jurnal Ilmu Komputer*, vol. 4, no. 2, pp. 16–25, Jan. 2021, [Online]. Available: <https://jurnal.pranataindonesia.ac.id/index.php/jik/article/view/96>
- [3] Emily A. Weiss, “Artificial Intelligence: Foundations, Concepts, and Ethical Considerations,” *Journal of Intelligent Systems*, vol. 28, no. 1, pp. 1–20, 2019.
- [4] M. Jovanović and M. Campbell, “Generative Artificial Intelligence: Trends and Prospects,” *Computer (Long Beach Calif)*, vol. 55, no. 10, pp. 107–112, 2022, doi: 10.1109/MC.2022.3192720.
- [5] V. Borisov, J. Haug, and G. Kasneci, “CancelOut: A Layer for Feature Selection in Deep Neural Networks,” 2019, pp. 72–83. doi: 10.1007/978-3-030-30484-3_6.
- [6] Koosha Sharifani and Mahyar Amini, “Machine Learning and Deep Learning: A Review of Methods and Applications,” *World Information Technology and Engineering Journal*, vol. 10, no. 07, pp. 3897–3904, 2023.
- [7] N. Yudistira, “Peran Big Data dan Deep Learning untuk Menyelesaikan Permasalahan secara Komprehensif,” *Expert*, vol. 11, no. 2, pp. 78–89, Jan. 2021, doi: 10.36448/expert.v11i2.2063.
- [8] Craig A. DeLarge, “The Role of Artificial Intelligence in Generative Art,” *Journal of Computational Creativity*, vol. 4, no. 2, pp. 145–162, 2019.
- [9] E. Zhou and D. Lee, “Generative AI, Human Creativity, and Art,” *SSRN Electronic Journal*, 2023, doi: 10.2139/ssrn.4594824.
- [10] G. W. Intyanto, “Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network),” *Jurnal Arus Elektro Indonesia*, vol. 7, no. 3, p. 80, Dec. 2021, doi: 10.19184/jaei.v7i3.28141.
- [11] M. M. Baharuddin, H. Azis, and T. Hasanuddin, “ANALISIS PERFORMA METODE K-NEAREST NEIGHBOR UNTUK IDENTIFIKASI JENIS KACA,” *ILKOM Jurnal Ilmiah*, vol. 11, no. 3, pp. 269–274, Dec. 2019, doi: 10.33096/ilkom.v11i3.489.269-274.
- [12] R. Srinivasan and K. Uchino, “Biases in Generative Art,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, New York, NY, USA: ACM, Mar. 2021, pp. 41–51. doi: 10.1145/3442188.3445869.
- [13] A. Ghosh and G. Fossas, “Can There be Art Without an Artist?,” Sep. 2022.
- [14] L. Hermawan and M. Bellaniar Ismiati, “Pembelajaran Text Preprocessing berbasis Simulator Untuk Mata Kuliah Information Retrieval,” *Jurnal Transformatika*, vol. 17, no. 2, p. 188, Jan. 2020, doi: 10.26623/transformatika.v17i2.1705.
- [15] R. Xiong *et al.*, “On Layer Normalization in the Transformer Architecture,” Feb. 2020.
- [16] K. Li *et al.*, “An Empirical Study of Transformer-Based Neural Language Model Adaptation,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, May 2020, pp. 7934–7938. doi: 10.1109/ICASSP40776.2020.9053399.
-

- [17] A. Gillioz, J. Casas, E. Mugellini, and O. A. Khaled, “Overview of the Transformer-based Models for NLP Tasks,” Sep. 2020, pp. 179–183. doi: 10.15439/2020F20.
- [18] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, “Self-supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation,” Apr. 2020.
- [19] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” Jul. 2019.
- [20] Z. Peng, L. Dong, H. Bao, Q. Ye, and F. Wei, “BEiT v2: Masked Image Modeling with Vector-Quantized Visual Tokenizers,” Aug. 2022.
- [21] S. Chaudhury and K. Sau, “RETRACTED: A BERT encoding with Recurrent Neural Network and Long-Short Term Memory for breast cancer image classification,” *Decision Analytics Journal*, vol. 6, p. 100177, Mar. 2023, doi: 10.1016/j.dajour.2023.100177.
- [22] T. Singh, D. C. Jhariya, M. Sahu, P. Dewangan, and P. Y. Dhekne, “Classifying Minerals using Deep Learning Algorithms,” *IOP Conf Ser Earth Environ Sci*, vol. 1032, no. 1, p. 012046, Jun. 2022, doi: 10.1088/1755-1315/1032/1/012046.