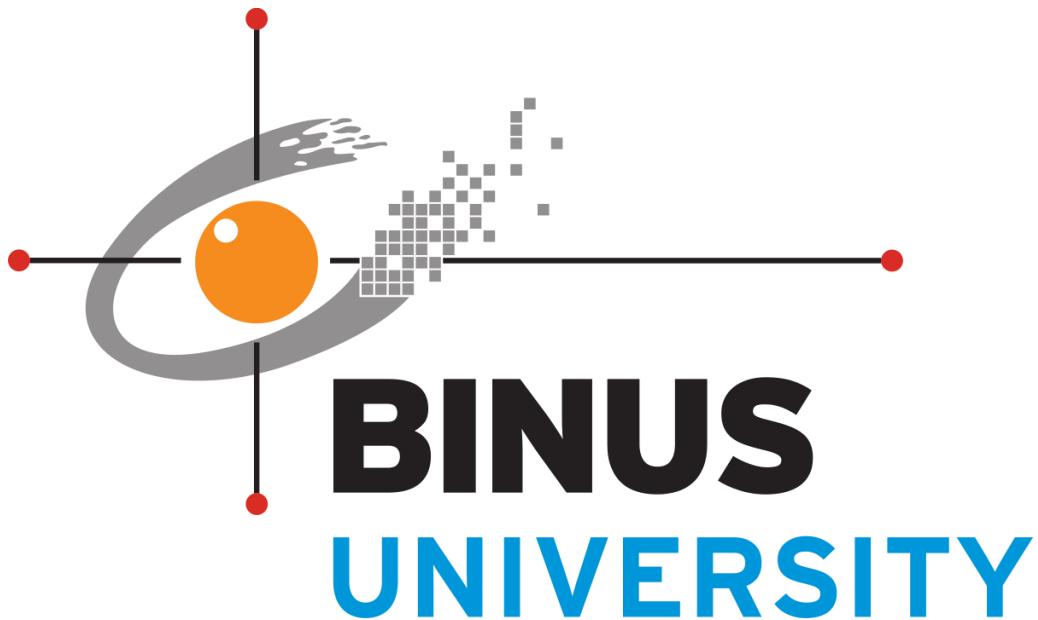


AI GENERATED ART DETECTION



Nama Anggota Kelompok:

- 2602058932 - Jovan Amarta Liem
- 2602070351 - Jonathan Surya Sanjaya
- 2602178911 - Cecillia Tjung

Daftar Isi

Problem Background	3
Literature Review	5
Hypothesis	18
Related Studies	19
Dataset	23
Methods	32
Experiments	35
Experiment Results	44
Conclusion	54
References.....	55

Problem Background

Penggunaan Artificial Intelligence (AI) di dalam dunia seni hingga saat ini masih isu sensitif dan menjadi perdebatan yang pelik terkhusus untuk generated AI. Banyak seniman yang menolak penggunaan AI karena mengurangi nilai seni yang dihasilkannya. Manusia sendiri menjadikan seni sebagai alat untuk menyajikan cerita, emosi, dan nilai personal dari pembuat karya seni dengan menuangkan kreativitasnya melalui lukisan[1]. Dengan nilai-nilai tersebut, seni yang dibuat oleh manusia mencerminkan pengalaman, perjuangan, dan pemikiran unik dari penciptanya sehingga jika pada akhirnya pembuatan seni dilakukan oleh mesin, nilai-nilai tersebut menjadi tergerus.

Berdasarkan hal tersebut, perlu adanya batasan khusus yang dijadikan aturan dan tolak ukur etika untuk menjaga nilai-nilai yang sudah ada[2]. Batasan tersebut dapat diterapkan agar pemanfaatan AI dapat digunakan secara lebih efektif. Jika batasan tersebut tidak diterapkan, maka terdapat dampak tertentu yang akan merugikan beberapa pihak. Contoh dampak dari penggunaan AI tanpa batasan tertentu adalah adanya plagiarisme dan penyalahgunaan gaya seniman, efek pada produksi dan konsumsi budaya yang menghambat kreativitas kolektif manusia, dan kerugian ekonomi untuk seniman. Hal tersebut dapat terjadi karena tidak tidak adanya *consent* dalam penggunaan AI untuk menggunakan gaya seniman tertentu seperti penggunaan Stable Diffusion yang mampu menghasilkan karya sangat mirip sementara seniman tak mendapat kompensasi apapun[3].

Dengan algoritma yang semakin canggih, bahkan sulit untuk membedakan lukisan generated AI dengan lukisan hasil tangan manusia jika hanya berdasarkan mata manusia. Dalam mengatasi hal tersebut, dibutuhkan alat untuk dapat melindungi karya seni. Solusi yang dapat ditawarkan untuk permasalahan tersebut adalah penggunaan teknologi yang tepat yang bertujuan untuk mendeteksi lukisan hasil AI dengan karya asli manusia. Untuk mendukung hal tersebut, baru-baru ini terdapat *art piracy* yang terjadi melalui situs eBay. Dilansir dari theguardian.com, terdapat sekitar 40 lukisan yang dijual di eBay dan teridentifikasi palsu[4]. Di antara lukisan tersebut terdapat lukisan yang diklaim milik pelukis terkenal yaitu Monet dan Renoir. Salah satu contoh lukisan

karya Monet berjudul “Forest With a Stream” , dijual dengan harga \$599.000, dan sebuah lukisan lain diklaim sebagai studi Claude Renoir seharga \$165.000.

Setelah dilakukan pemeriksaan menggunakan algoritma AI, lukisan tersebut dapat dideteksi palsu karena terdapat ketidaksesuaian dalam teknik dan gaya yang tidak konsisten. Penggunaan AI dalam mendeteksi karya seni palsu menunjukkan potensi besar dalam menjaga integritas pasar seni. Teknologi ini dapat membantu mengidentifikasi penipuan yang mungkin sulit dideteksi melalui metode tradisional, sehingga hal ini bisa merupakan perlindungan bagi kolektor dan pembeli karya seni. Oleh karena itu, kami membuat teknologi untuk mendeteksi lukisan generated AI dengan lukisan tangan manusia atas dorongan untuk menjaga pasar seni.

Literature Review

Aboutalebi, H., Mao, D., Fan, R., Xu, C., He, C., & Wong, A. (2024). DeepfakeArt Challenge: A benchmark dataset for generative AI art forgery and data poisoning detection [arXiv:2306.01272]. Diakses dari <https://arxiv.org/abs/2306.01272>

Penelitian ini berfokus pada masalah *copyright infringement* dalam konteks hasil seni oleh AI. Dalam paper ini, terdapat dataset sintetik yang dirancang untuk mensimulasikan skenario di dunia nyata jika pelanggaran hak cipta terjadi dengan mengeksplorasi hal tersebut seperti dalam proses deteksi.

Sebagai salah satu sumber dataset penelitian kami, paper ini menjelaskan bagaimana image yang dihasilkan dibuat (*AI Generated*), dimana pembuat mengambil source (gambar original) dari WikiArt dataset yang kemudian di generate dengan menggunakan bantuan model **Stable Diffusion II & ControlNet**. Dari dataset yang disediakan sendiri, kami mengambil 2 kategori, *inpainting* dan juga *style transfer*.

Inpainting

Image yang dibuat pada genre *inpainting* dibuat dengan menggunakan bantuan model **Stable Diffusion II** dengan prompt “Generate a painting compatible with the rest of the image”, dimana dalam pembuatanya sendiri, dibutuhkan bantuan masking yang dipilih secara random dengan tiga kategori, *side masking*, *diagonal masking*, dan *random masking*.

Style Transfer

Image yang dibuat pada genre *style transfer* menggunakan bantuan model **ControlNet Zhang and Agrawala** untuk mengalih / mengubah style source image dengan bantuan Canny Edges. Perubahan ini pun melibatkan sejumlah prompting:

1. *High - quality, detailed, realistic image.*
2. *A high - quality, detailed, cartoon-style drawing.*
3. *A high - quality, detailed, oil painting.*
4. *A high - quality, detailed, pencil drawing.*

Dari percobaan di atas sendiri, didapatkan hasil bahwa model **DINO - v2 ViT - L/14** adalah model dengan accuracy tertinggi sebesar sekitar 82%.

Model	Accuracy	Precision	Recall	F1	MAE
MultiGrain Berman et al. [2019]	81.20 %	97.68%	63.90 %	0.77	0.42
DINO-v1 XCiT-S/16 Caron et al. [2021]	80.84 %	97.65 %	63.18 %	0.76	0.32
DINO-v1 XCiT-S/8 Caron et al. [2021]	78.75 %	97.05 %	59.28 %	0.73	0.32
DINO-v1 ViT-S/8 Caron et al. [2021]	81.71 %	96.74 %	65.61 %	0.78	0.34
DINO-v1 ResNet-50 Caron et al. [2021]	81.11 %	96.22 %	64.78 %	0.77	0.38
DINO-v2 ViT-S/14 Oquab et al. [2023]	81.12 %	96.88 %	64.29 %	0.77	0.31
DINO-v2 ViT-B/14 Oquab et al. [2023]	81.88 %	95.65 %	66.78 %	0.79	0.30
DINO-v2 ViT-L/14 Oquab et al. [2023]	82.75 %	95.09 %	69.04 %	0.80	0.29
DINO-v2 ViT-g/14 Oquab et al. [2023]	81.20 %	95.05 %	65.81 %	0.78	0.29
SwinTransformer Liu et al. [2021]	78.02 %	96.17 %	58.34 %	0.72	0.35

Kusuma, S. W., Natalia, F., Ko, C. S., & Sudirman, S. (2024). DETECTION OF AI-GENERATED ANIME IMAGES USING DEEP LEARNING. *ICIC Express Letters, Part B: Applications*, 15(3), 295–301.
<https://doi.org/10.24507/icicelb.15.03.295>

Penelitian ini bertujuan untuk mendeteksi gambar pada anime yang dihasilkan oleh AI melalui pendekatan deep learning. Pada penelitian ini, dilakukan eksperimen untuk mengetahui kemampuan AI dalam membuat gambar anime secara cepat dan otomatis di mana hal ini memiliki potensi mengancam mata pencaharian seniman. Proses yang dilakukan dalam penelitian ini mencangkup data collection, preprocessing, dan model training.

Data Collection

Data yang digunakan dalam penelitian ini dibagi menjadi dua kelas: gambar asli anime buatan tangan manusia serta gambar anime yang dihasilkan oleh AI. Dataset tersebut diambil dari dataset publik dan generator AI berbasis anime. Data yang dihasilkan berkaitan dengan beragam variasi visual dan gaya untuk memastikan model dapat mendeteksi perbedaan gambar buatan manusia dengan generated images.

Preprocessing

Beberapa tahap preprocessing yang dilakukan sebagai berikut.

1. Resize gambar ke dalam ratio 1:1 dengan ukuran yang sama, yaitu 256x256.

2. Augmentation dilakukan dengan mengaplikasikan image transformation seperti flip, rotate, dan zoom.

Model Training

Sebelum dilakukan training, dataset dibagi menjadi 75% untuk training, dan 25% untuk data testing. Total dataset untuk training adalah 750 gambar dan dilakukan training menggunakan 2 model: MobileNetV2 dan MobileNetV3.

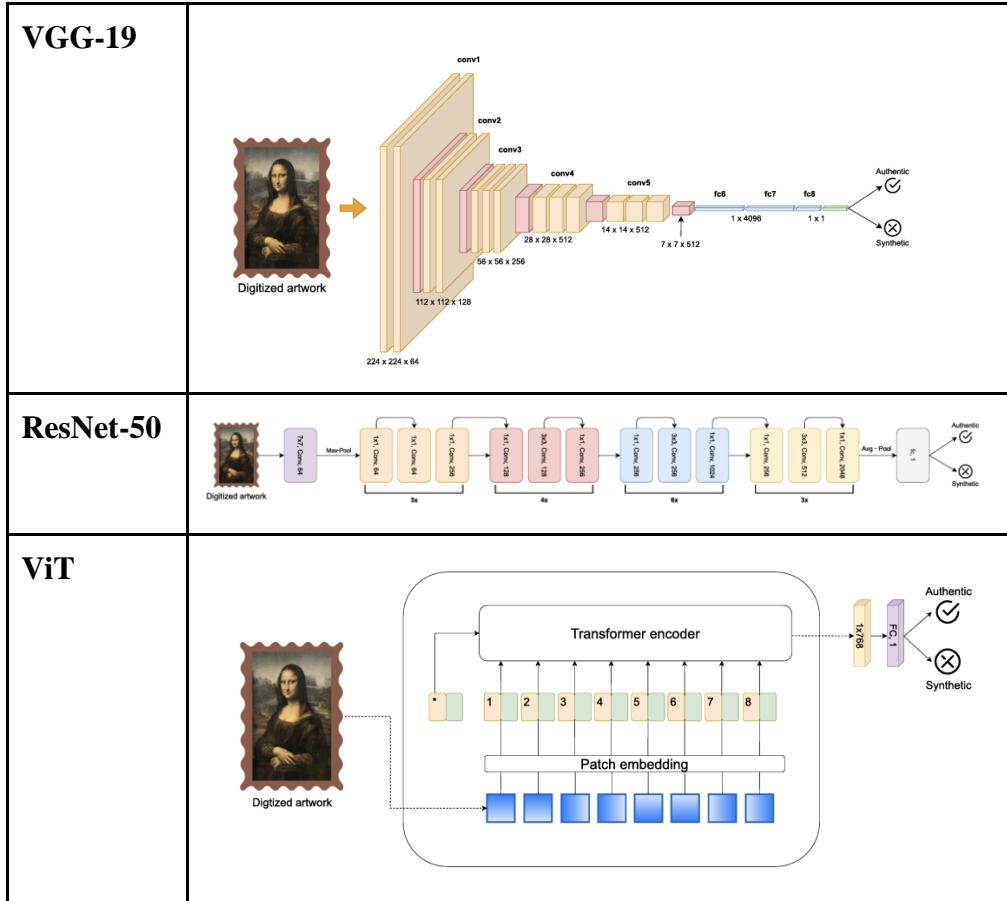
Ketika dievaluasi menggunakan data testing, terdapat 117 human images dan 133 AI images yang digunakan. Pada saat deteksi AI images, kedua model dapat mendeteksi dengan baik keseluruhan 133 gambar. Sementara pada saat melakukan deteksi pada human images, model MobileNetV2 dapat mendeteksi 109 dan MobileNetV2 mendeteksi 110. Dengan demikian, model MobileNetV3 mendeteksi lebih baik dengan akurasi 97.2% sesuai tabel di bawah ini.

	Accuracy	Precision	Recall	F1-score
MobileNetV2	96.8%	100%	94.3%	97.1%
MobileNetV3	97.2%	100%	95.0%	97.4%

Bianco, T., Castellano, G., Scaringi, R., & Vessio, G. (2023). Identifying AI-Generated Art with Deep Learning. <http://ceur-ws.org>

Paper ini menjelaskan deep learning dalam mengidentifikasi AI-generated art yang menunjukkan bahwa Vision Transformers memiliki kemampuan yang kuat untuk mengidentifikasi task tersebut. Dalam paper ini, diusulkan bahwa analisis feature importance dari perspective semantic dapat meningkatkan interpretabilitas model melampaui interpretasi visual dari activation maps.

Terdapat tiga model yang digunakan pada paper ini: VGG-19, ResNet-50, dan ViT.



Dengan arsitektur di atas, hasilnya menunjukkan bahwa ViT memiliki performa yang paling baik dibanding dengan dua model lainnya. Hal ini dapat terjadi karena ViT menggunakan transformer architecture dengan input berbasis patch. Setiap patch ini diratakan menjadi vektor satu dimensi lalu diberikan positional embedding untuk mempertahankan informasi spasial.

ViT dengan self-attention mechanism memungkinkan model ini menangkap relasi global antar bagian gambar sehingga dapat memahami struktur gambar secara keseluruhan. Ketiga model memberikan akurasi yang tinggi tapi ViT menunjukkan yang terbaik. Metric ditunjukkan pada gambar di bawah ini.

	Accuracy	Precision	Recall	F1
VGG-19	0.9581	0.9590	0.9562	0.9575
ResNet-50	0.9654	0.9645	0.9655	0.9650
ViT	0.9758	0.9752	0.9759	0.9755

Huang, Y. F., & Wang, C. T. (2014). Classification of painting genres based on feature selection. Lecture Notes in Electrical Engineering, 308, 159–164. https://doi.org/10.1007/978-3-642-54900-7_23

Paper ini menggunakan sistem klasifikasi genre lukisan, dengan menggunakan 4 deskriptor terkait fitur warna dan tekstur sesuai spesifikasi MPEG-7. Sistem ini memanfaatkan algoritma *Self-Adaptive Harmony Search(SAHS)* untuk memilih subset fitur relevan yang kemudian digunakan untuk melatih setiap classifier berbasis SVM (*Support Vector Machine*).

Background

Membahas sistem klasifikasi gambar pada fitur seperti warna, tekstur, dan bentuk. Untuk meningkatkan akurasi klasifikasi genre lukisan :

- Data fitur berlebihan diatasi dengan reduksi dimensi menggunakan PCA atau feature selection
- Classifier SVM, karena dikenal memberikan performa baik jika parameter dan fungsi kernel dipilih tepat

Arsitektur Sistem

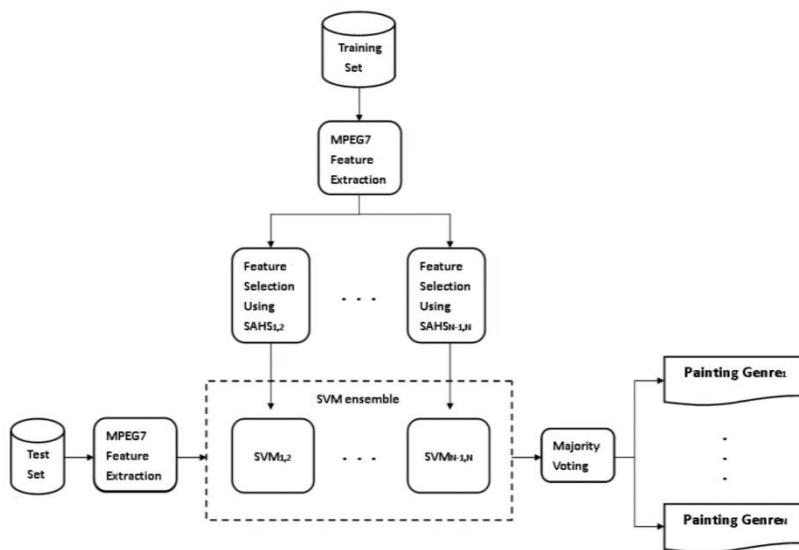
1. Training Phase
2. Ekstraksi fitur warna dan tekstur
 - Color Layout Descriptor
 - Color Structure Descriptor
 - Edge Histogram Descriptor
 - Homogeneous Texture descriptor

Total terdapat 186 dimensi fitur awal.

SAHS digunakan untuk memilih subset fitur optimal, yang kemudian digunakan untuk melatih model SVM dalam pendekatan satu-lawan-satu (*one-against-one*).

3. Testing Phase

Fitur yang diekstraksi dan dimasukkan dari classifier yang dilatih hasil prediksi digabung menggunakan strategi voting untuk menentukan genre lukisan



4. Model dengan 2 komponen utama:

a. Algoritma SAHS

Mencari fitur terbaik dengan cara mengukur korelasi antar fitur subset(intra-correlation) dan relevansi fitur terhadap kelasnya(inter-correlation)

b. Fungsi Objektif

Subset fitur yang baik memiliki intra correlation rendah dan inter correlation tinggi.

5. Hasil Eksperimen

Dataset :

- Cubism (72 lukisan)

- Fauvism (74 lukisan)
- Impressionism (74 lukisan)
- Naïve Art (49 lukisan)
- Pointillism (71 lukisan)
- Realism (71 lukisan)

1. Fitur Awal (168 Dimensi)

Table 1 Confusion matrix on the original feature set

	Cub	Fau	Imp	Naï	Poi	Rea	Recall
Cubism	16	3	1	0	0	2	72.7%
Fauvism	9	9	1	1	1	1	40.9%
Impressionism	6	2	13	0	1	0	59.1%
Naïve art	1	1	0	13	1	0	81.3%
Pointillism	0	2	4	0	15	1	68.2%
Realism	3	0	4	0	0	15	68.2%
<i>Precision</i>	45.7%	52.9%	56.5%	92.9%	83.3%	78.9%	64.3%

Akurasi : 64.3%, beberapa genre seperti FAUVISM sulit diklasifikasikan karena terlihat pada tabel recall <= 50%.

2. Seleksi Fitur Global

Table 2 Confusion matrix by the global selection strategy

	Cub	Fau	Imp	Naï	Poi	Rea	Recall
Cubism	16	3	2	0	0	1	72.7%
Fauvism	6	12	2	0	1	1	54.5%
Impressionism	4	1	15	1	1	0	68.2%
Naïve art	2	1	0	12	1	0	75.0%
Pointillism	1	1	3	0	16	1	72.7%
Realism	3	0	5	0	0	14	63.6%
<i>Precision</i>	50.0%	66.7%	55.6%	92.3%	84.2%	82.4%	67.5%

Fitur berkurang menjadi 54, akurasi meningkat menjadi 67.5%

3. Seleksi Fitur Lokal

Table 3 Number of features in each local feature set

Cubism	Cub	Fau	Imp	Naï	Poi	Rea
Fauvism	*	39	38	31	30	36
Impressionism		*	32	34	28	27
Naïve art			*	38	32	32
Pointillism				*	29	28
Realism					*	33
Cubism						*

Subset fitur spesifik dihasilkan untuk setiap pasangan genre (15 subset) akurasi menjadi 69.8%, lebih baik dibanding fitur global.

Metode paper ini mengungguli pendekatan lain dengan akurasi lebih tinggi dan algoritma seleksi fitur lebih efisien.

Table 5 Comparisons among all methods

	J. Zujovic et al. [12]	M. Culjak et al. [2]	J. Zujovic et al. [12]	M. Culjak et al. [2]	Ours
Dataset	Google search & CARLI collections	Google search & Artlex database	Google search & CARLI collections	Google search & Artlex database	Google search & Artlex database
Original feature	-	68	-	68	186
Genres	5	6	5	6	6
Dim. reduction	-	-	-	-	SAHS
Classifier	SVM	ANN	AdaBoost	SMO	SVM
Accuracy	57.8%	56.6%	68.3%	60.2%	69.8%

Kesimpulan

- Sistem klasifikasi genre lukisan menggunakan fitur warna dan tekstur dari spesifikasi MPEG-7 serta algoritma SAHS untuk memilih subset fitur optimal.
- Model SVM dengan strategi seleksi lokal memiliki akurasi terbaik dengan angka 69.8%
- Pendekatan ini lebih efektif dibanding metode yang ada sebelumnya.

Ivanova, Krassimira & Stanchev, Peter & Velikova, Evgenia & Vanhoof, Koen & Depaire, Benoît & Mitov, Iliya & Markov, Krassimir. (2010). Features for Art Painting Classification Based on Vector Quantization of MPEG-7 Descriptors.

Pada paper ini, kami berfokus untuk pencarian feature extraction yang bisa digunakan untuk menganalisis perbedaan antara art asil, *human drawn* dan art yang dibuat oleh AI, *AI Art Generated*.

Paper ini sendiri berfokus untuk membahas klasifikasi jenis lukisan seni menggunakan descriptor MPEG - 7 dan mengevaluasi seberapa efektif descriptor MPEG - 7 dalam merepresentasikan karakteristik visual sebuah lukisan.

Dalam penelitian ini sendiri, dijelaskan bahwa nama formal MPEG - 7 adalah “Multimedia Content Description Interface”, dimana MPEG - 7 sendiri mendeskripsikan image melalui banyak hal seperti dominant colors, edginess, texture, dan lain sebagainya. MPEG - 7 sendiri juga sering digunakan dalam kasus seperti image - to - image matching, searching for similarities, dan sketch queries.

Dari MPEG - 7 sendiri, berbagai macam descriptor yang digunakan adalah:

1. *Scalable Color (SC)* yang merepresentasikan color histogram dalam color space HSV yang diencode menggunakan Haar transform. Untuk merepresentasikan sebuah gambar, *Scalable Color* memerlukan vector dengan 64 attribute.
2. *Color Layout (CL)* adalah metode untuk menentukan distribusi spasial warna dengan menggunakan ruang warna YCbCr. Metode ini menggunakan koefisien DCT yang terkuantisasi untuk komponen Y, Cb, dan Cr, yaitu:
 - a. Komponen Y:
 - Koefisien DCT pertama (DY1).
 - Lima koefisien DCT berikutnya yang terkuantisasi (DY2- DY6).
 - b. Komponen Cb:
 - Koefisien DCT pertama (DCb1).
 - Dua koefisien DCT berikutnya yang terkuantisasi (DCb2 - DCb3).
 - c. Koefisien Cr:
 - Koefisien DCT pertama (DCr1).
 - Dua koefisien DCT berikutnya yang terkuantisasi (DCr2 - DCr3).

Se

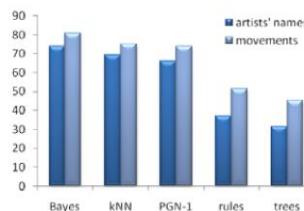
hingga jika ditotalkan, *Color Layout* memiliki 12 attributes.

3. *Color Structure (CS)*, yang menentukan baik isi warna maupun struktur dari konten. Deskriptor ini mengekspresikan struktur warna lokal dalam sebuah gambar melalui elemen struktur yang terdiri dari beberapa sampel gambar.

Hasilnya, vektor dengan 65 atribut digunakan untuk merepresentasikan color structure.

4. *Dominant Color (DC)*. Deskriptor ini direpresentasi ulang menjadi tiga vektor yang mendistribusikan hue, saturation, dan luminance yang terkuantisasi. Setelah terkuantisasi, hasilnya adalah vektor dengan 23 atribut (13 untuk hue, 5 untuk saturation, dan 5 untuk luminance).
5. *Edge Histogram (EH)*. Menentukan distribusi spasial dari lima jenis edges di area gambar lokal (4 edge terarah - vertikal, horizontal, 45 degree, 135 degree, dan satu tidak terarah). Edge histogram menghasilkan sebuah vektor dengan 80 atribut.
6. *Homogenous Texture (HT)*, yang merincikan tekstur area menggunakan energi dan deviasi energi dalam sekumpulan saluran frekuensi. Vektor dengan 60 atribut akan digunakan untuk merepresentasikan energi dan deviasi energi.

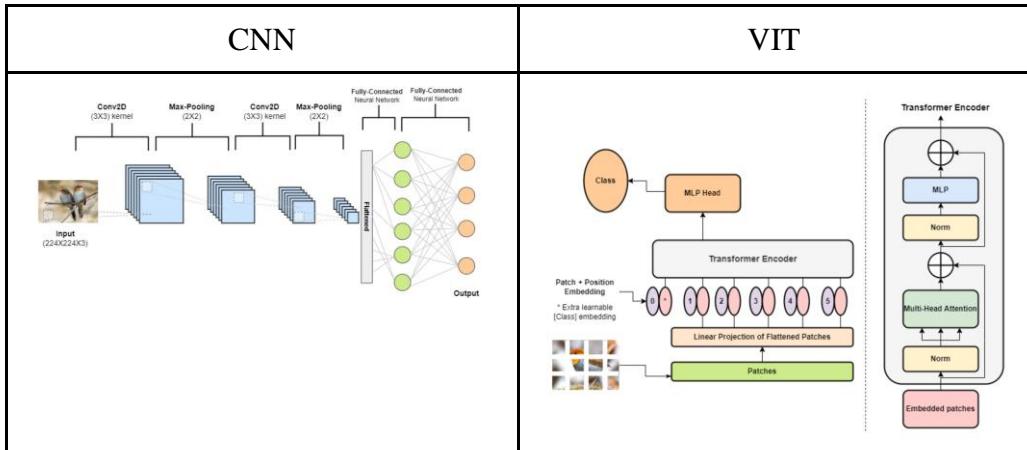
Overall accuracy dari eksperimen ini sendiri adalah sekitar 70 - 80%



Maurício, José & Domingues, Ines & Bernardino, Jorge. (2023). Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review. Applied Sciences. 13. 5521. 10.3390/app13095521.

Pada penelitian kali ini, kami berfokus mencari tahu alasan mengapa VIT bisa menghasilkan accuracy dan performa yang lebih baik dari model berbasis CNN (ResNet50V2 dan Xception).

Paper ini pertama - tama menjelaskan perbedaan arsitektur yang dimiliki oleh VIT dan CNN.



Jika kita lihat dari gambar sendiri, kita bisa melihat bahwa model berbasis CNN memproses gambar secara lokal menggunakan layar convolution, sedangkan ViT memproses gambar dengan patch - patch kecil yang lebih efisien untuk gambar dengan resolusi yang lebih besar dan tinggi.

Dari paper di atas, bisa disimpulkan beberapa hal sebagai berikut:

1. Performance ViT vs CNN

ViT Menunjukkan performa unggul dalam berbagai aplikasi, terutama pada dataset kecil karena mekanisme *self - attention* yang dimilikinya yang memungkinkan menangkap hubungan global antar elemen gambar. Namun, jika dilatih dengan data kecil tanpa *pre - training*. ViT memiliki kemampuan generalisasi yang lebih rendah dibandingkan dengan CNN.

2. Keterbatasan CNN

CNN memiliki kelemahan dalam *shift - invariance* yang membuatnya kurang efektif jika terdapat noise pada gambar, namun penambahan filter seperti *anti - aliasing* atau peningkatan kernel dapat meningkatkan generalisasi CNN.

3. Robustness

ViT lebih unggul dalam menangani gambar dengan noise dan gangguan karena kemampuannya dalam memahami konteks global. CNN lebih sensitif terhadap fitur *high - frequency*. Tetapi memiliki ketahanan bawaan terhadap translasi karena sifat *translation - invariance* - nya.

4. Mekanisme Multi - Head Attention (ViT)

Mekanisme ini memungkinkan ViT untuk menangkap lebih banyak informasi dari setiap piksel dan menurunkan komputasi karena perhitungan dilakukan secara paralel yang memberikan keuntungan khusus pada gambar dengan elemen sekunder yang memperjelas elemen utama.

Silva, R. S. R., Lotfi, A., Ihianle, I. K., Shahtahmassebi, G., & Bird, J. J. (2024). *ArtBrain: An explainable end-to-end toolkit for classification and attribution of AI-generated art and style* [arXiv:2412.01512]. Diakses dari <https://arxiv.org/abs/2412.01512>

Paper ini juga merupakan salah satu dari source dataset kami, dimana dalam pembuatan gambar *AI Generated*-nya, author mengambil gambar originalnya dari Artbench - 10 yang kemudian menggunakan bantuan dua model, yakni *Latent Diffusion* dan *Stable Diffusion*. Namun dalam penelitian ini kami hanya menggunakan gambar dari *Stable Diffusion*, dikarenakan *Stable Diffusion* memungkinkan negative prompting yang memungkinkan model untuk menghapus sebuah “frame” atau foto bingkai dari gambar yang dihasilkan seperti pada *Latent Diffusion*, dimana hal ini tentu bisa menyebabkan model kita terlalu bias / overfit pada gambar AI karena hal ini.

Prompting yang digunakan sendiri berdasarkan “A painting in art style . . .” yang memiliki sepuluh kategori, *art nouveau, baroque, expressionism, impressionism, post impressionism, realism, romanticism, surrealism, dan ukiyo e.*

Dari penelitian yang dilakukan, model mereka berhasil memperoleh accuracy sebesar kurang lebih 86%.

Table 2: Classification scores.

Generative Model	Art Style	F1-Score		
		ArtBench	MobileNet V2	Our Study
Latent Diffusion	Art Nouveau	-	0.9685	0.9875
	Baroque	-	0.9243	0.9565
	Expressionism	-	0.9667	0.9766
	Impressionism	-	0.7238	0.7627
	Post impressionism	-	0.7247	0.7551
	Realism	-	0.8808	0.9099
	Renaissance	-	0.9332	0.9542
	Romanticism	-	0.8906	0.9190
	Surrealism	-	0.9800	0.9885
	Ukiyo-e	-	0.9995	1.0000
Standard Diffusion	Art Nouveau	-	0.9990	1.0000
	Baroque	-	0.9861	0.9935
	Expressionism	-	0.9975	0.9985
	Impressionism	-	0.9885	0.9935
	Post impressionism	-	0.9870	0.9935
	Realism	-	0.9975	1.0000
	Renaissance	-	0.9874	0.994
	Romanticism	-	0.9975	0.9985
	Surrealism	-	0.9985	1.0000
	Ukiyo-e	-	1.0000	1.0000
Human	Art Nouveau	0.6660	0.6209	0.7043
	Baroque	0.7917	0.6948	0.7854
	Expressionism	0.5127	0.5122	0.5881
	Impressionism	0.4636	0.4728	0.5228
	Post impressionism	0.5175	0.4660	0.5759
	Realism	0.4525	0.4079	0.4991
	Renaissance	0.8159	0.7314	0.8111
	Romanticism	0.5980	0.4780	0.6050
	Surrealism	0.7737	0.7618	0.8251
	Ukiyo-e	0.9695	0.9721	0.9850
Overall Model Accuracy		-	0.8354	0.8693

Giannakas, Filippos & Troussas, Christos & Krouska, Akrivi & Sgouropoulou, C. & Voyatzis, Ioannis. (2021). XGBoost and Deep Neural Network Comparison: The Case of Teams' Performance. 10.1007/978-3-030-80421-3_37.

Pada penelitian ini, kami lebih berfokus pada performa XGBoost yang dapat menyaingi performa model deep learning, setelah membaca literature review tersebut, kami menemukan bahwa XGBoost memiliki performa lebih baik dalam accuracy learning dan prediksi dalam Binary classification task, dimana alasan lain yang mendukung performa XGBoost adalah data yang cocok dengan Tree based method.

Pada penelitian di atas sendiri, ditemukan bahwa accuracy yang didapatkan dengan XGBoost adalah 95.60% dan 93.08% sedangkan untuk DNN ditemukan hanya 80.50% dan 77.36%.

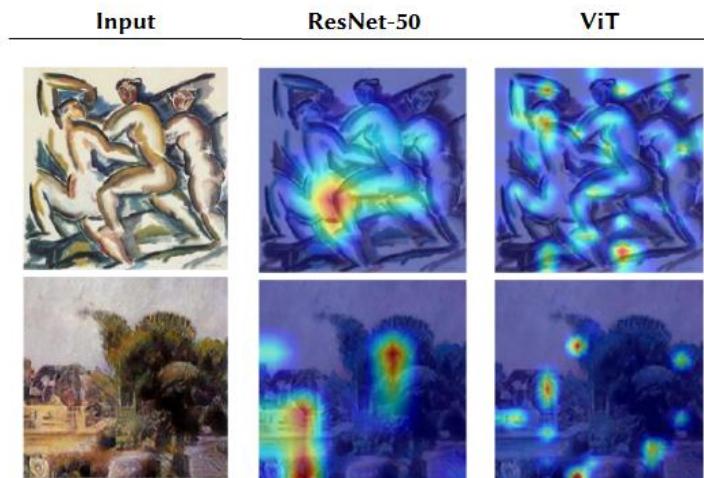
Kendati demikian, author menyampaikan bahwa rendahnya accuracy yang didapatkan oleh DNN menandakan bahwa pentingnya melakukan riset lanjut untuk menemukan parameter dan kualitas data yang cocok untuk meningkatkan performa.

Hypothesis

Berdasarkan literature review yang kami baca sebelumnya, kami menduga bahwa hasil terbaik akan terdapat pada model machine learning dengan feature extraction MPEG7 dan ViT untuk model deep learning.

Kami menduga model XGBoost dengan feature extraction MPEG7 akan menjadi yang paling baik diantara model machine learning dan feature extraction lainnya karena MPEG7 menggunakan banyak deskriptor diantaranya adalah scalable color, color layout, color structure, dominant color, edge histogram, homogeneous texture. Sedangkan, untuk model machine learning lainnya kami akan menggunakan feature extraction HSV (Hue, Saturation, Value) saja dan kombinasi antara HSV dengan Edge detection. Kami juga menduga model XGBoost akan menjadi yang terbaik dari model machine learning lainnya karena model XGBoost menggunakan gradient boosting framework dimana, setiap tree pada XGBoost akan memperbaiki yang sebelumnya.

Kemudian untuk model deep learning kami menduga bahwa model ViT akan menjadi yang paling bagus diantara model deep learning lainnya, karena model ViT (Vision Transformer) bekerja pada patch-patch kecil, sehingga dapat membuat model ViT lebih mudah mengerti bagian-bagian pada gambar dibandingkan model machine learning lainnya yang berbasis CNN.



Related Studies

ARTBRAIN: AN EXPLAINABLE END-TO-END TOOLKIT FOR CLASSIFICATION AND ATTRIBUTION OF AI-GENERATED ART AND STYLE

Table 2: Classification scores.

Generative Model	Art Style	F1-Score		
		ArtBench	MobileNet V2	Our Study
Latent Diffusion	Art Nouveau	-	0.9685	0.9875
	Baroque	-	0.9243	0.9565
	Expressionism	-	0.9667	0.9766
	Impressionism	-	0.7238	0.7627
	Post impressionism	-	0.7247	0.7551
	Realism	-	0.8808	0.9099
	Renaissance	-	0.9332	0.9542
	Romanticism	-	0.8906	0.9190
	Surrealism	-	0.9800	0.9885
	Ukiyo-e	-	0.9995	1.0000
Standard Diffusion	Art Nouveau	-	0.9990	1.0000
	Baroque	-	0.9861	0.9935
	Expressionism	-	0.9975	0.9985
	Impressionism	-	0.9885	0.9935
	Post impressionism	-	0.9870	0.9935
	Realism	-	0.9975	1.0000
	Renaissance	-	0.9874	0.994
	Romanticism	-	0.9975	0.9985
	Surrealism	-	0.9985	1.0000
	Ukiyo-e	-	1.0000	1.0000
Human	Art Nouveau	0.6660	0.6209	0.7043
	Baroque	0.7917	0.6948	0.7854
	Expressionism	0.5127	0.5122	0.5881
	Impressionism	0.4636	0.4728	0.5228
	Post impressionism	0.5175	0.4660	0.5759
	Realism	0.4525	0.4079	0.4991
	Renaissance	0.8159	0.7314	0.8111
	Romanticism	0.5980	0.4780	0.6050
	Surrealism	0.7737	0.7618	0.8251
	Ukiyo-e	0.9695	0.9721	0.9850
Overall Model Accuracy		-	0.8354	0.8693

Pada penelitian ini, author berhasil mendapatkan accuracy sebesar 86% dengan menggunakan Deep Learning berbasis AttentionConvNext dengan arsitektur model di bawah ini:

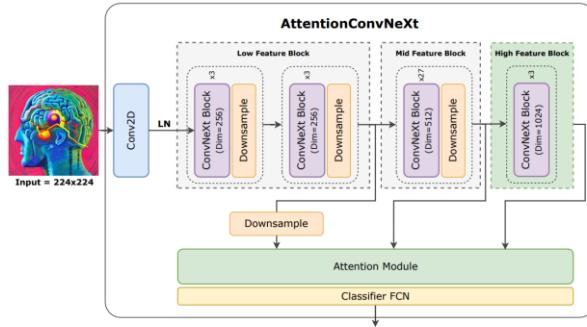


Figure 1: 'AttentionConvNeXt' model architecture design.

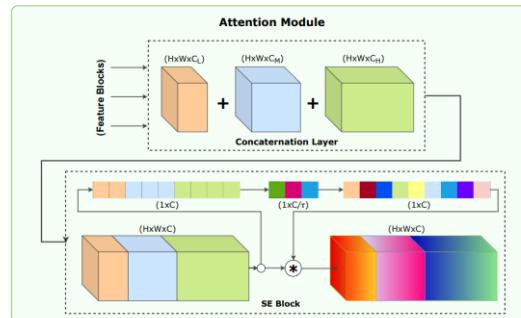


Figure 2: Attention Module Architecture.

Dengan dataset yang digunakan sebagai berikut:

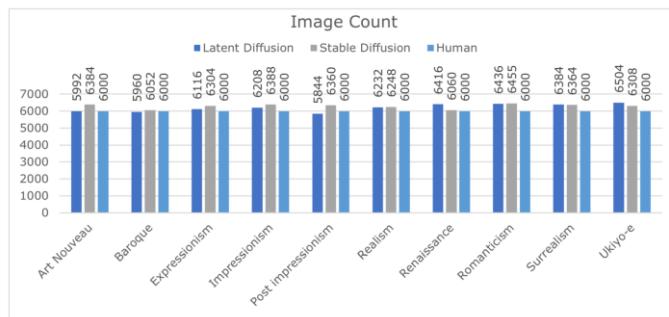


Figure 3: Image counts in each class.

Selain hanya mengandalkan akurasi model, author juga menggunakan artistic turing test untuk menilai keandalan model dalam mendeteksi apakah sebuah art yang ditunjukkan merupakan gambar original (human drawn) atau gambar AI (AI Generated).

10

Who or what do you think painted this image? *

(1 Point)



- A human artist
- A machine

Test dilakukan dengan menggunakan 25 human drawn dan Standard Diffusion dari AI - ArtBench.

Human Art Knowledge	AI Art Knowledge	AI-Art Detection Accuracy of Human			
		Novice	Beginner	Advanced	Expert
Novice		50.0%	58.8%	70.0%	-
(Response Count)		(12)	(8)	(1)	(0)
Beginner		45.5%	51.4%	62.0%	-
(Response Count)		(4)	(16)	(8)	(0)
Advanced		52.0%	55.0%	-	-
(Response Count)		(2)	(2)	(0)	(0)
Expert		-	-	80.0%	-
(Response Count)		(0)	(0)	(1)	(0)
Overall Human Accuracy		53.8%			
(Response Count)		(50)			
ArtBrain (AI) Accuracy		98%			

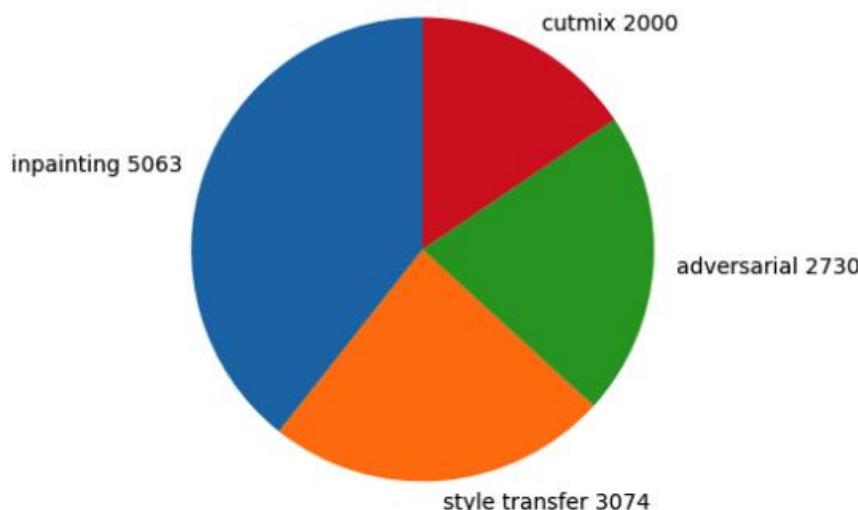
Hasil overall Human accuracy sendiri masih terbilang cukup rendah dengan modus pada kategori Beginner yakni hanya **53.8%** sedangkan model yang diusulkan berhasil memprediksi sebaik **98% accuracy**.

DeepfakeArt Challenge: A Benchmark Dataset for Generative AI Art Forgery and Data Poisoning Detection

Model	Accuracy	Precision	Recall	F1	MAE
MultiGrain Berman et al. [2019]	81.20 %	97.68%	63.90 %	0.77	0.42
DINO-v1 XCiT-S/16 Caron et al. [2021]	80.84 %	97.65 %	63.18 %	0.76	0.32
DINO-v1 XCiT-S/8 Caron et al. [2021]	78.75 %	97.05 %	59.28 %	0.73	0.32
DINO-v1 ViT-S/8 Caron et al. [2021]	81.71 %	96.74 %	65.61 %	0.78	0.34
DINO-v1 ResNet-50 Caron et al. [2021]	81.11 %	96.22 %	64.78 %	0.77	0.38
DINO-v2 ViT-S/14 Oquab et al. [2023]	81.12 %	96.88 %	64.29 %	0.77	0.31
DINO-v2 ViT-B/14 Oquab et al. [2023]	81.88 %	95.65 %	66.78 %	0.79	0.30
DINO-v2 ViT-L/14 Oquab et al. [2023]	82.75 %	95.09 %	69.04 %	0.80	0.29
DINO-v2 ViT-g/14 Oquab et al. [2023]	81.20 %	95.05 %	65.81 %	0.78	0.29
SwinTransformer Liu et al. [2021]	78.02 %	96.17 %	58.34 %	0.72	0.35

Pada penelitian ini, author menggunakan beberapa model sebagai percobaan, untuk hasil accuracy tertinggi sendiri didapat oleh model **DINO - v2 ViT-L/14 dengan accuracy 82.75%**.

Perbedaan dari eksperimen yang kami lakukan sendiri, author paper ini menggunakan keempat kategori seperti **cutmix, inpainting, style transfer, dan adversarial data poisoning**. Dengan distribusi datasetnya sendiri kurang seimbang (lebih didominasi oleh kategori inpainting)



(b) Distribution of categories of similar pairs

Dataset

AI - ArtBench10

Source: <https://www.kaggle.com/datasets/ravidussilva/real-ai-art/data>

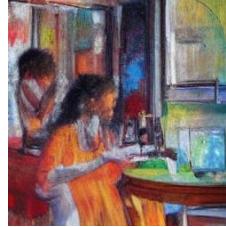
AI - ArtBench merupakan dataset yang terdiri dari 180.000 art images, dimana 60.000 diantaranya merupakan gambar original (human - drawn) yang diambil dari ArtBench - 10 dataset dan sisanya digenerate dengan menggunakan model **Stable Diffusion** dan **Latent Diffusion**.

Resolusi dari gambar di atas sendiri:

- Gambar Original: 256 x 256
- Latent Diffusion: 256 x 256
- Stable Diffusion: 768 x 768

Dataset yang disediakan sendiri terdiri dari tiga puluh kategori, dimana masing - masing 10 kategori (Latent Diffusion, Stable Diffusion, dan human - drawn) memiliki kategori:

Category	Human Drawn	Stable Diffusion	Latent Diffusion
Art Nouveau			
Baroque			
Expressionism			

Impressionism			
Post Impressionism			
Realism			
Renaissance			
Romanticism			
Surrealism			



Seperti yang bisa dilihat di perbandingan gambar di atas, kategori **Latent Diffusion** pada beberapa kesempatan memiliki frame / bingkai foto sehingga tidak digunakan pada eksperimen yang kami lakukan karena dapat menyebabkan bias terhadap kategori AI.

Alasan mengapa **Stable Diffusion** jarang memiliki frame pada gambar yang dihasilkannya sendiri disebabkan karena model tersebut memungkinkan negative prompting seperti “photo frame” untuk menghapus photo frame pada gambar yang dihasilkan.

Prompting yang digunakan untuk menghasilkan setiap gambar sendiri adalah “A painting in art style . . .” dengan menggunakan model **Text - to - Image** berbasis **Stable Diffusion** dan **Latent Diffusion**.

DeepfakeArt Challenge

Source: <https://www.kaggle.com/datasets/danielmao2019/deepfakeart>

Dataset DeepfakeArt Challenge Benchmark dirancang untuk mendukung pengembangan algoritma yang bertujuan untuk mendeteksi **forgery** (**Pemalsuan**) dan **poisoning (kontaminasi data)** yang dilakukan dengan menggunakan teknik **Generative AI**, seperti penggunaan konten berhak cipta (copyright) untuk membuat karya baru.



(a) Distribution of similar versus dissimilar pairs (b) Distribution of different categories of similar data
 Figure 2: Overall distribution of data in DeepfakeArt dataset.

Dataset yang disediakan sendiri terdiri dari kurang lebih 32.000 gambar, dimana kategori yang kami pakai sendiri berfokus pada kategori **similar data**, khususnya bagian **Inpainting**, dan **Style Transfer**.

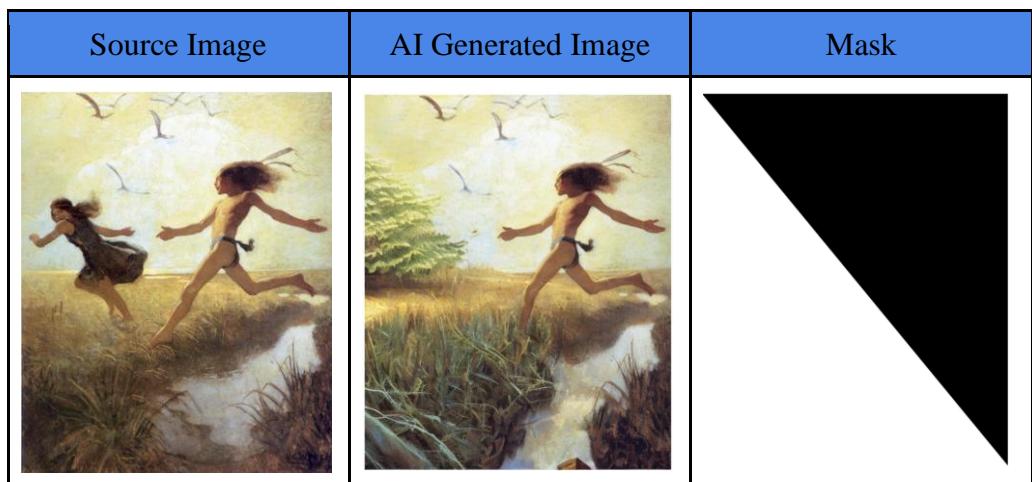
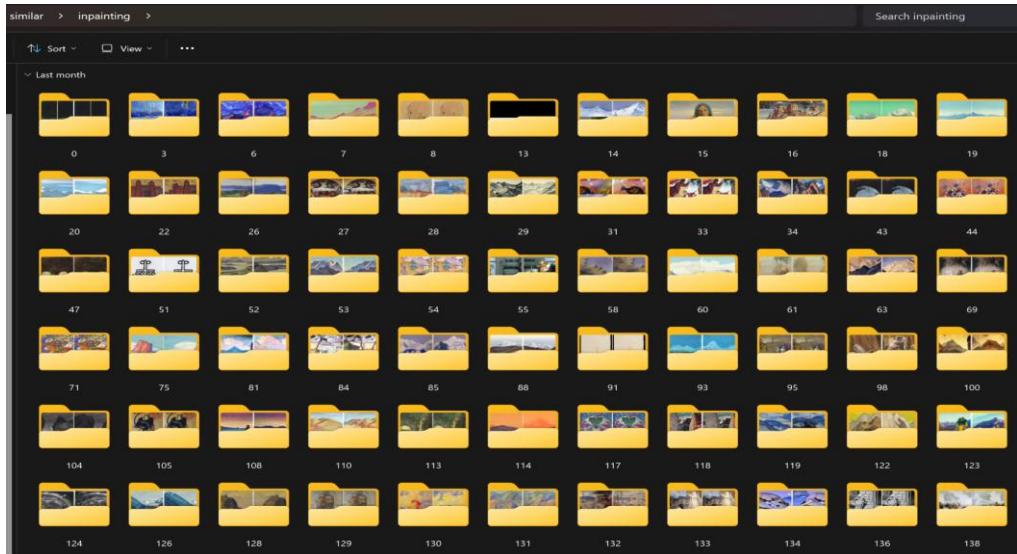
Inpainting

Source image yang digunakan untuk *inpainting* category berasal dari **WikiArt** yang diambil secara acak untuk mengenerate gambar forgery.

Dalam setiap folder (image) yang diambil sendiri terdapat tiga files yang terdiri dari:

1. **Source Image:** Image yang digunakan untuk membuat forgery image.
2. **Inpainting Image:** Image (forgery image) yang digenerate dengan menggunakan model **Stable Diffusion II**.
3. **Masking Image:** Image hitam - putih yang digunakan model sebagai pedoman untuk mengenerate bagian yang ingin diubah dan menghasilkan inpainting image. Masking yang diterapkan akan berada dalam range 40% - 60% image original. Dengan mengikuti salah satu skema berikut secara random:
 - a. **Side Masking:** Top side, bottom side, right side atau left side dari source image.
 - b. **Diagonal Masking:** Upper right, upper left, lower left, atau lower left diagonal side dari source image.
 - c. **Random Masking:** Part dari source image akan secara random dimask.

Prompt yang digunakan sendiri adalah “Generate a painting compatible with the rest of the image”.



Seperti yang bisa dilihat pada gambar di atas, area gambar original (gambar perempuan) yang sama dengan masking akan digenerate ulang dengan menggunakan Stable Diffusion II yang menghasilkan gambar pepohonan.

Style Transfer

Sama seperti inpainting, style transfer juga menggunakan **WikiArt** sebagai source image. Perbedaanya terletak pada penyusunan image didalam folder yang disediakan, dimana setiap image dikategorikan dalam beberapa genre seperti

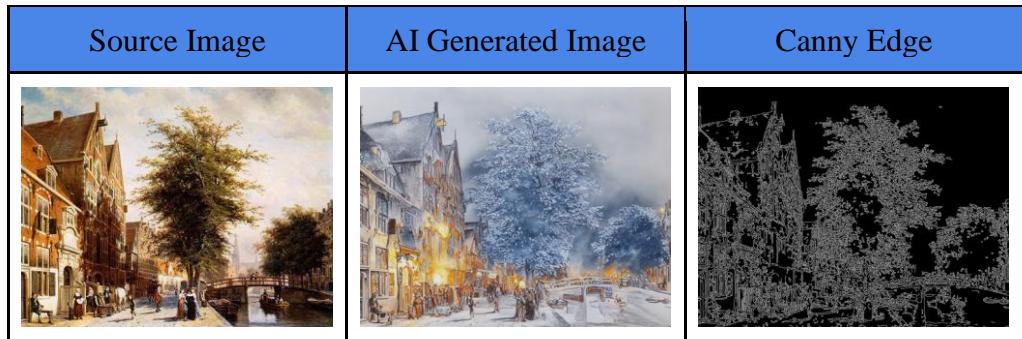
dataset sebelumnya, namun memiliki beberapa genre tambahan seperti Romanticism, Realism, Rococo, Minimalism, dan lain sebagainya.

Format dari setiap image sendiri adalah sebagai berikut:

1. **Source Image:** Image yang digunakan untuk membuat forgery image.
2. **Style Transferred Image:** Image yang digenerate ulang dari edge image dengan menggunakan **ControlNet**.
3. **Edge Image:** Image yang dibuat dengan bantuan Canny Edge Detection.

Prompt yang digunakan sendiri adalah sebagai berikut:

1. "a high-quality, detailed, realistic image".
2. "a high-quality, detailed, cartoon style drawing".
3. "a high-quality, detailed, oil painting".
4. "a high-quality, detailed, pencil drawing"



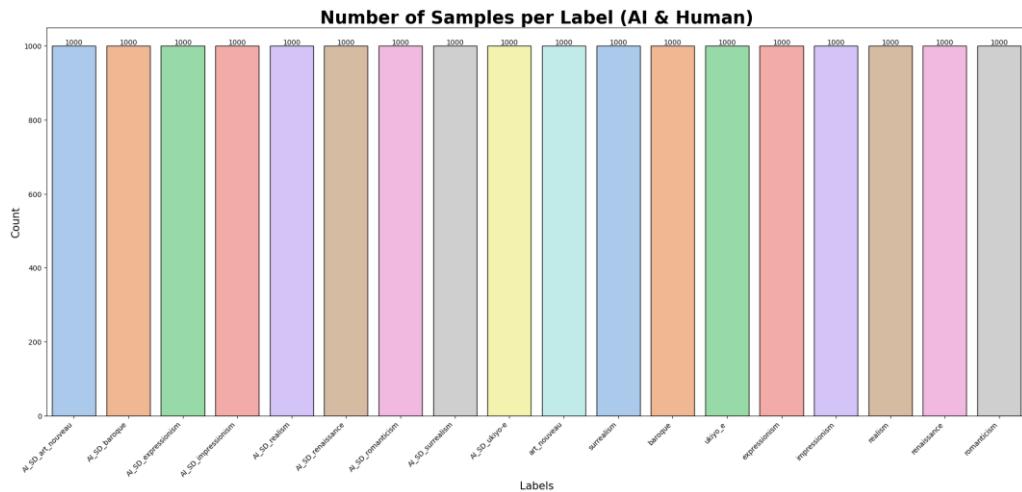
Dari gambar di atas, setelah mendapatkan edge dari Source image, maka ControlNet akan digunakan untuk menghasilkan gambar AI Generated.

Percobaan I

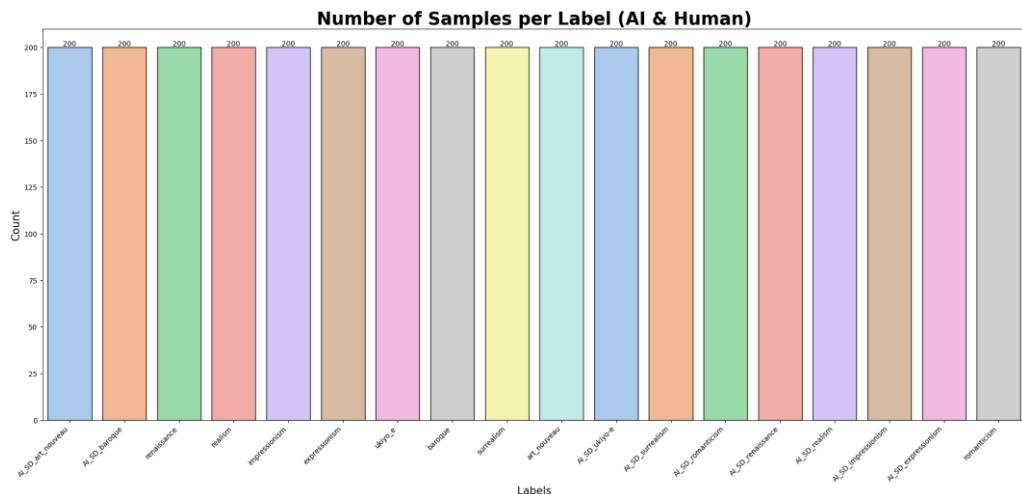
Pada percobaan pertama, kami baru menggunakan dataset dari source pertama kami (**AI - ArtBench**) yang kemudian diklasifikasikan menjadi 18 class (AI dan Human untuk masing - masing kategori), dimana untuk Machine Learning

(Feature Extraction) dan Deep Learning menggunakan 1000 images dari setiap folder, pengecualian metode MPEG - 7 dikarenakan memakan cukup RAM sehingga diperkecil hingga 200 images per foldernya.

Percobaan Machine Learning (Feature Extraction) & Deep Learning



Percobaan MPEG - 7

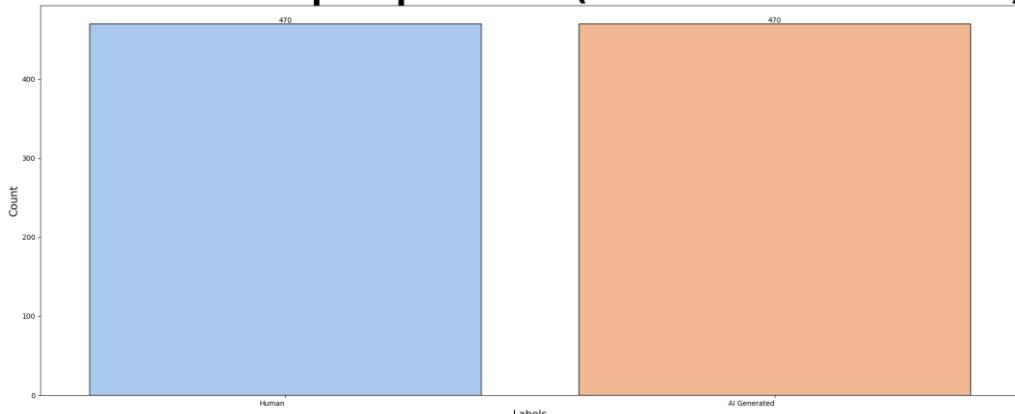


Percobaan II

Pada percobaan terakhir, kami menggabungkan kedua dataset di atas (AI - ArtBench & DeepfakeArt) yang kemudian lebih kami fokuskan menjadi 2 class untuk bertujuan apakah gambar tersebut merupakan AI Generated atau bukan. Dimana untuk masing - masing kategori:

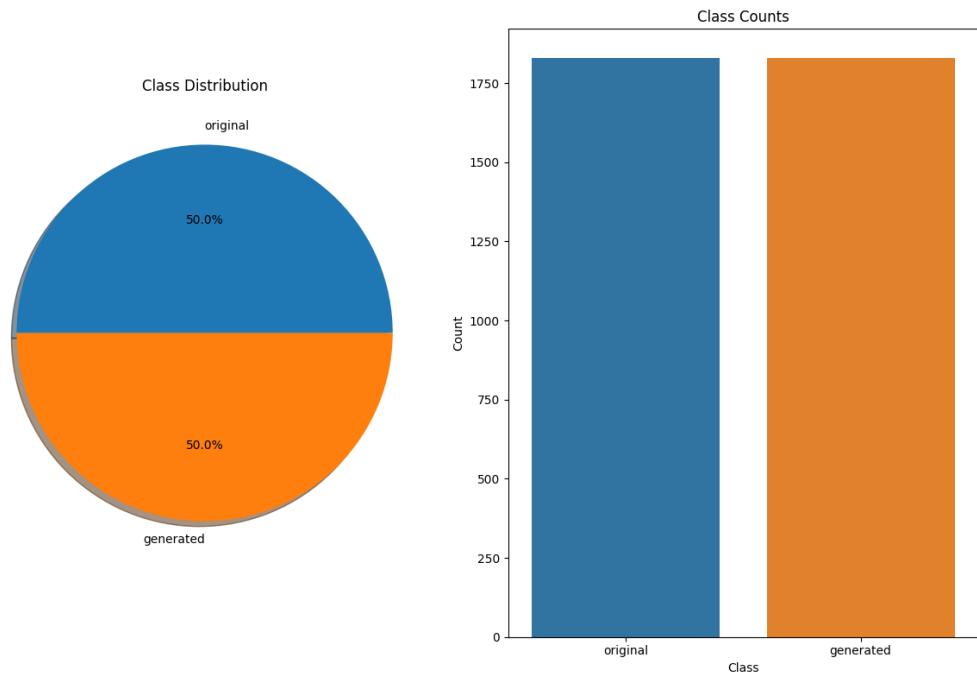
Machine Learning (Feature Extraction)

Number of Samples per Label (Human vs AI Generated)



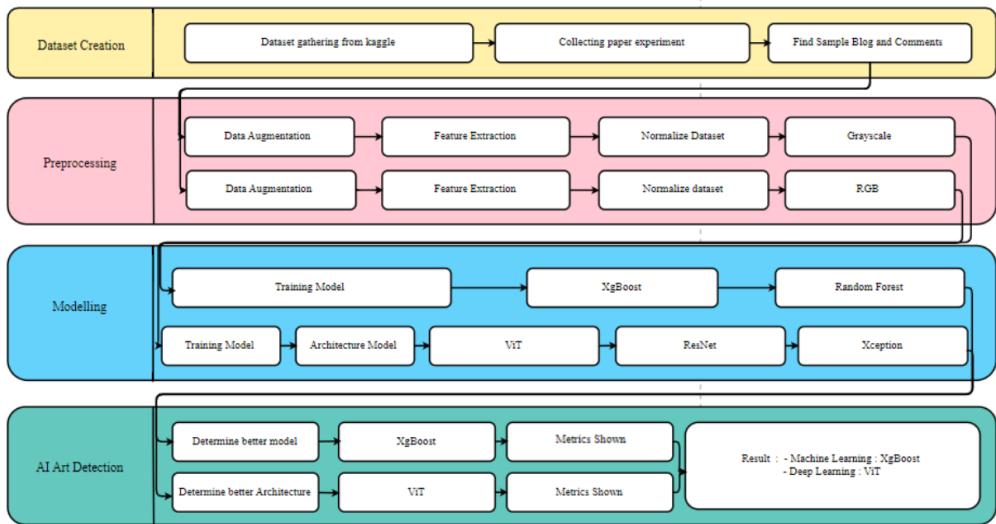
Untuk machine learning, kami mengambil **20** pada setiap kategori pada dataset pertama. Untuk dataset kedua, kami mengambil **20 images** untuk setiap class pada kategori **inpainting**, dan **10 images** untuk **AI** dan **human** drawn pada **style transfer** yang menghasilkan **total 470 images** untuk setiap class.

Deep Learning



Pada **dataset pertama**, kami mengambil 60 images dari setiap foldernya. Kemudian pada **dataset kedua** kategori inpainting, kami mengambil 750 images untuk setiap class (AI & Human drawn) dan 20 images untuk setiap folder yang ada pada style transfer (AI & Human drawn) yang menghasilkan total **1830 untuk setiap class**.

Methods



Dimulai dari **Penentuan topik**, kami memutuskan untuk berfokus pada **AI - Art Detection** karena menurut kami, seiring berkembangnya zaman, penggunaan AI juga semakin canggih, yang memungkinkan pihak yang tidak bertanggung jawab untuk melakukan kejahatan digital seperti **Deepfake** dan **Art Forgery** (merusak barang berhak cipta yang berfokus pada lukisan yang ada) dengan menggunakan bantuan AI. Kemudian kami melanjutkan metode kami seperti yang tertera pada gambar di atas:

Dataset Creation

Pada tahapan ini, kelompok kami berfokus pada pengumpulan data dan eksperimen terkait seperti:

- Dataset gathering from Kaggle: Disini kami berfokus pada pencarian dataset yang pada Kaggle dan dapat digunakan dalam penelitian kami.
- Collecting paper experiment: Di tahap ini, kami juga mencari eksperimen - eksperimen terkait yang memiliki topik serupa dalam mendeteksi AI - Art, dan juga feature extraction yang mungkin bisa digunakan dalam melakukan klasifikasi berbasis feature sebuah gambar seperti warna, edges, dan lain sebagainya serta model - model yang digunakan dalam mendeteksi gambar asli dan ai.

- Find Sample Blog and Comments: Sama seperti tahap sebelumnya, kami disini masih berfokus pada pencarian model yang bisa digunakan dalam mendeteksi gambar asli dan ai.

Preprocessing

Pada step ini, kami memproses data (images) yang akan digunakan dalam percobaan kami, dimana kami melakukan beberapa hal seperti:

- Data Augmentation: Agar model yang kami usulkan lebih robust terhadap perubahan - perubahan yang ada pada images yang diberikan kepada model, kami melakukan augmentasi seperti:
 1. Random horizontal flip.
 2. Random vertical flip.
 3. Random brightness.
 4. Random contrast.
- Feature Extraction: Agar model machine learning kami bisa melakukan prediksi yang baik, kami memerlukan feature yang bisa menggambarkan image dengan baik, beberapa diantaranya seperti:
 1. HSV
 2. HSV + Edge
 3. MPEG - 7
- Normalize Dataset: Agar dapat diterima oleh semua model, kami meresize image kami dengan ukuran 224 x 224 dan dibagi dengan /255.

Untuk prosesnya sendiri kami secara keseluruhan menggunakan image dalam mode RGB.

Modeling

Untuk model, kami memilih menggunakan:

Machine Learning

- Random Forest
- XGBoost

Deep Learning

- CNN:
 - a. Xception architecture.
 - b. ResNet50V2 architecture.
- ViT (Google / ViT - base - patch16 - 224)

Metrics

Untuk metrics sendiri, kami menggunakan **accuracy score** untuk melihat bagaimana performa model dan feature extraction dalam mendeteksi gambar yang diberikan, kami berfokus:

- Machine Learning: Kami berfokus menentukan feature extraction mana yang paling baik dalam mendeskripsikan gambar (HSV, HSV + Edge, dan MPEG - 7).
- Deep Learning: Disini fokus kami adalah menentukan arsitektur terbaik model dalam memprediksi gambar yang diberikan (Xception, ResNet50V2.

Experiments

Pada eksperimen yang kami lakukan, kami membuat 2 model machine learning dan 3 model deep learning. Kami membuat model XGBoost dan Random Forest untuk model machine learning, untuk model deep learning kami membuat model ViT, Xception, dan ResNet.

Kami menggunakan 3 cara feature extraction yang berbeda untuk model machine learning, diantaranya adalah HSV, kombinasi antara HSV dan Edge, dan MPEG7. Dibawah ini adalah contoh fitur yang diambil menggunakan feature extraction yang kami gunakan. Cara ini kami gunakan baik pada eksperimen pertama kami, maupun eksperimen yang terakhir kami lakukan.

Gambar yang digunakan:

Original Image

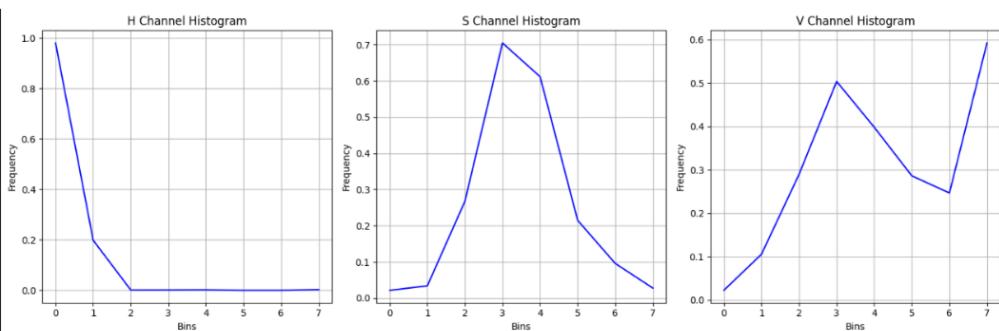


AI Generated Image

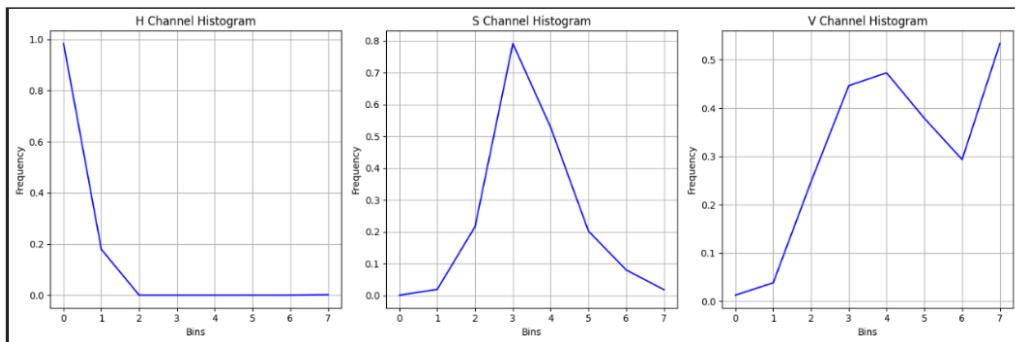


HSV adalah singkatan dari Hue, Saturation, Value, Hue merepresentasikan warna misalnya merah, hijau, biru dan lain sebagainya. Saturation adalah intensitas warna dari sebuah gambar, dan Value merepresentasikan tingkat kecerahan dari gambar. Berikut adalah contoh fitur yang diambil dari salah satu gambar menggunakan feature extraction HSV:

Original Image

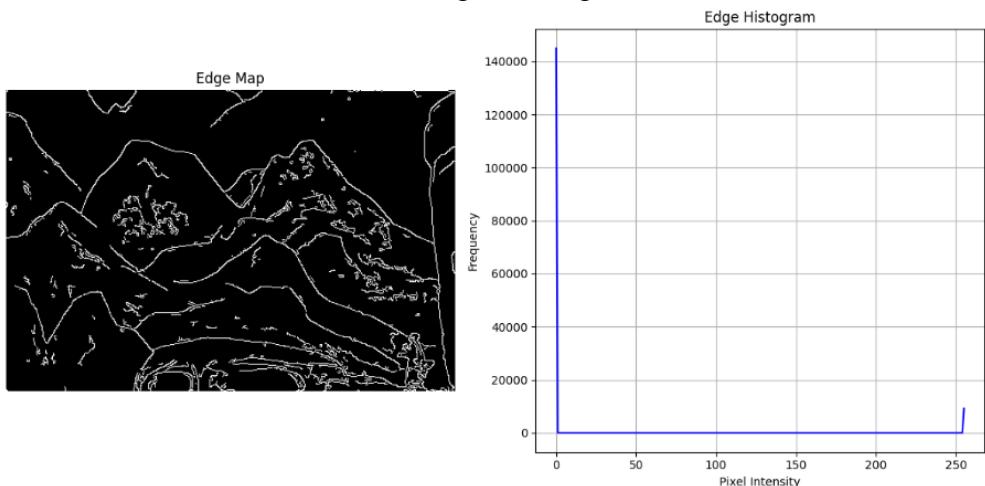


AI Generated Image

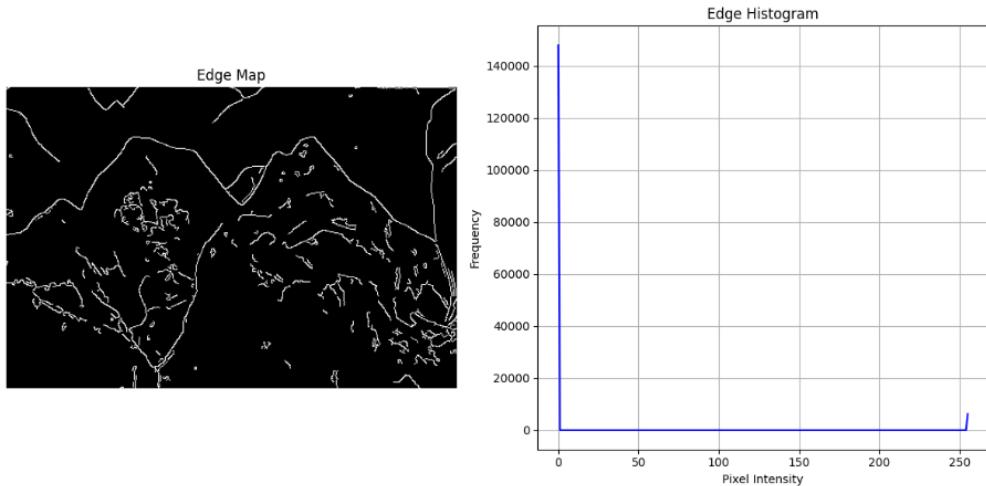


Kemudian kami juga menggunakan feature extraction edge detection dan berikut adalah contoh fitur yang diambil dari salah satu gambar menggunakan feature extraction edge detection:

Original Image

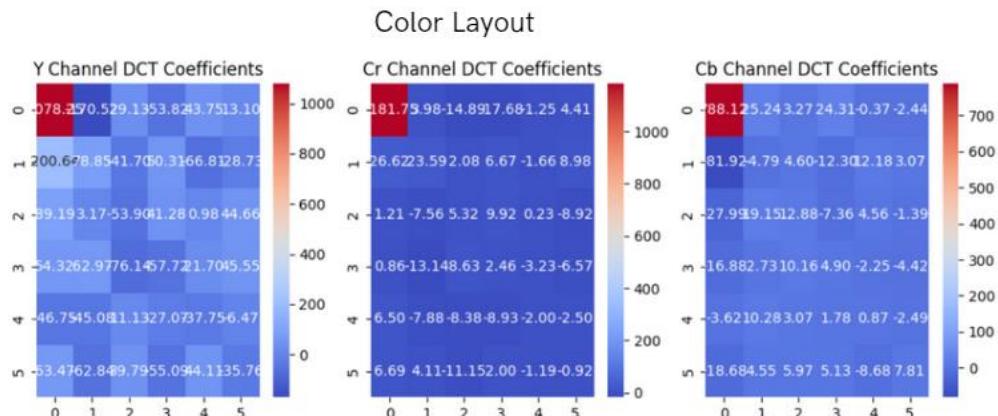


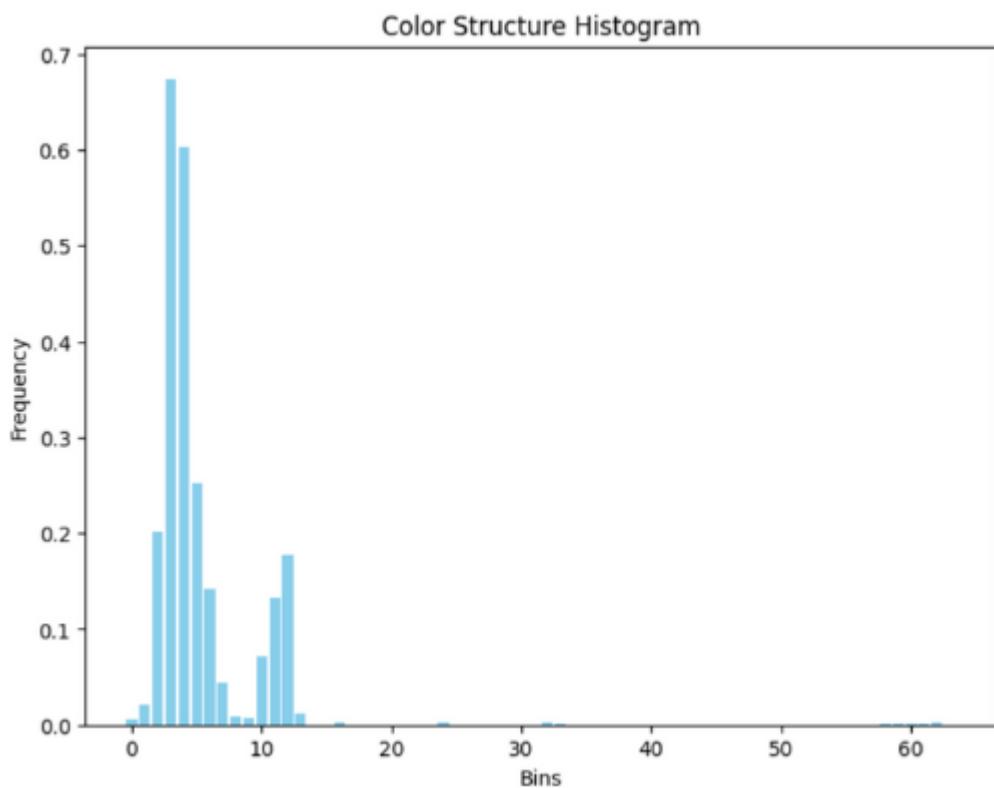
AI Generated Image

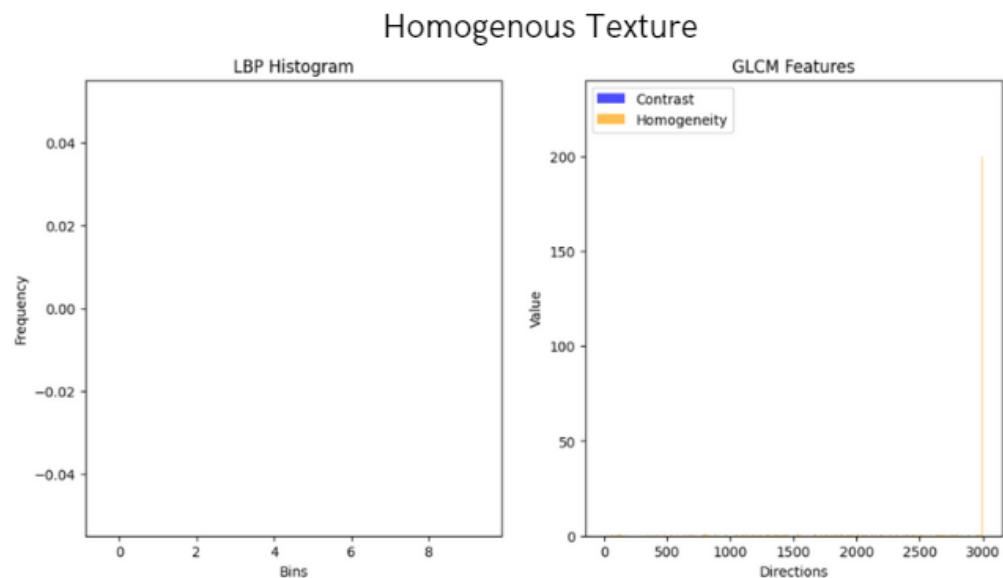
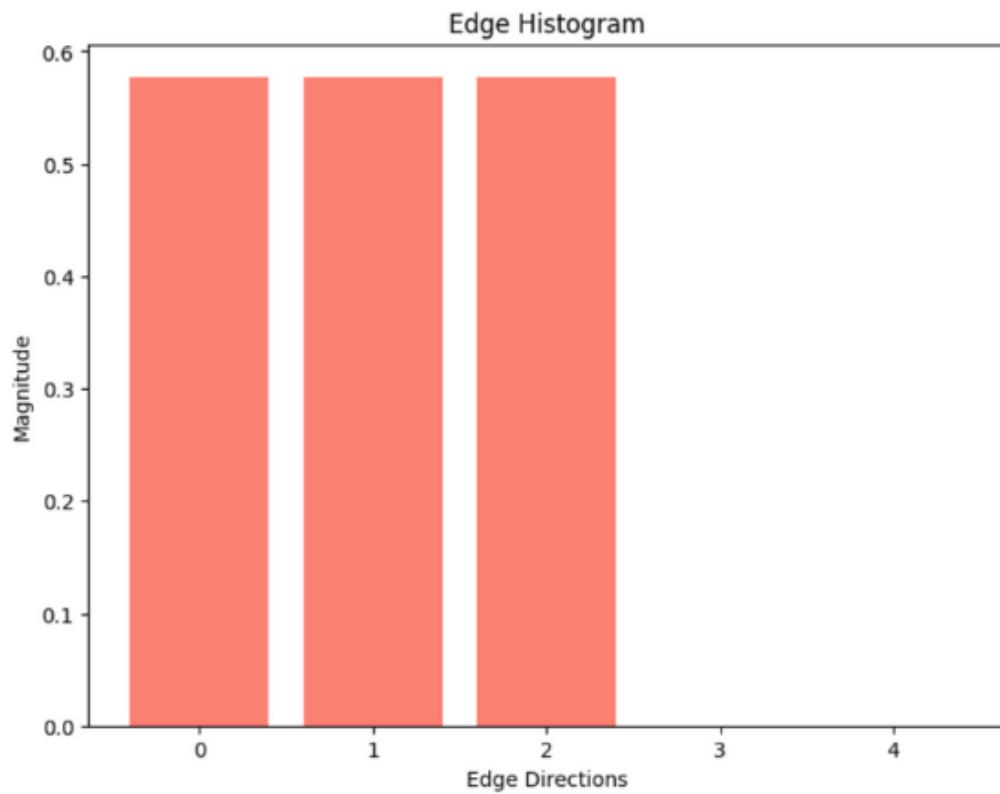


Terakhir, kami menggunakan feature extraction MPEG7, kami menggunakan 4 deskriptor saja yaitu color layout, color structure histogram, homogenous texture dan edge histogram. Berikut adalah contoh fitur yang diambil dari salah satu gambar menggunakan feature extraction MPEG7:

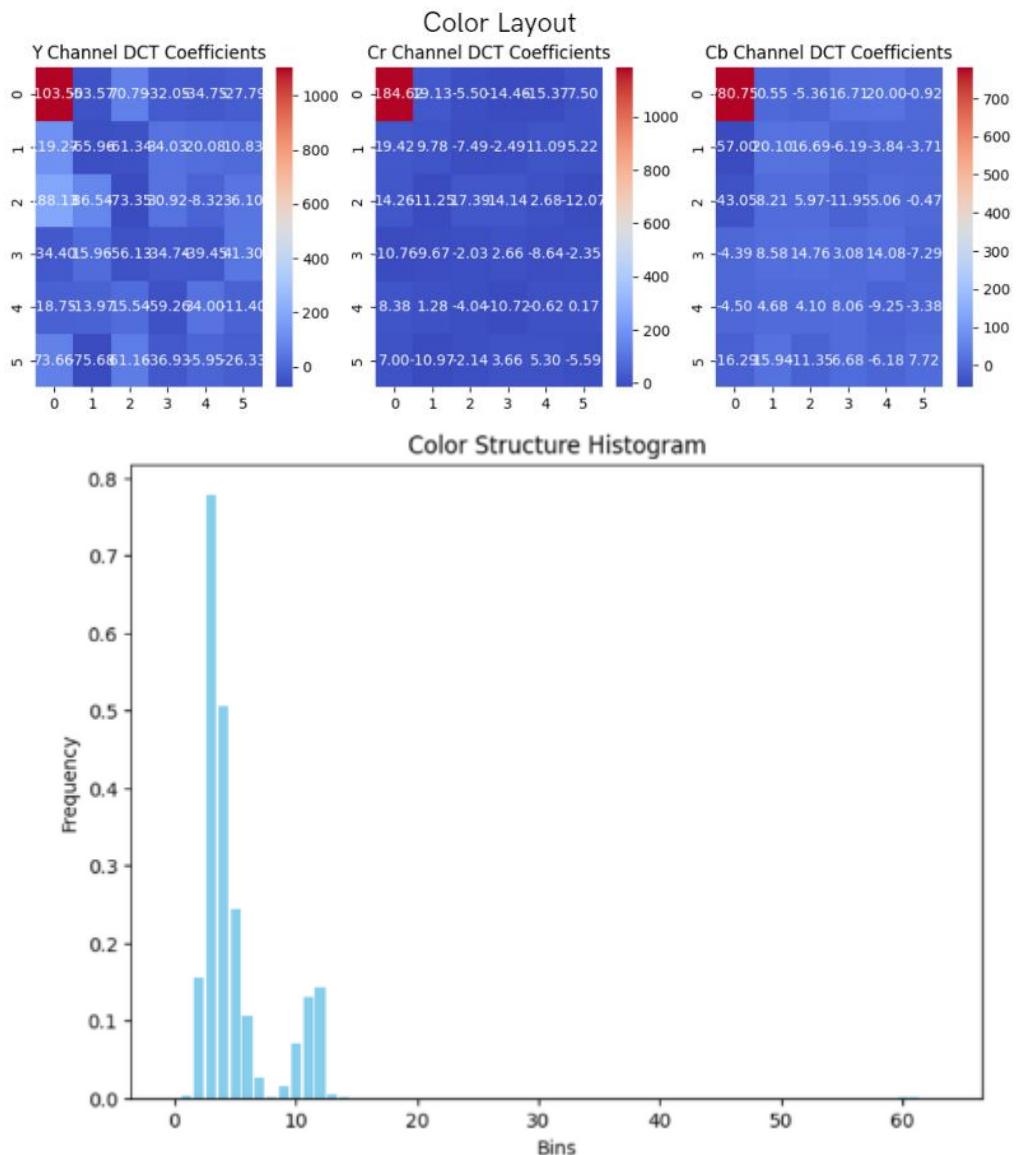
Original Image:

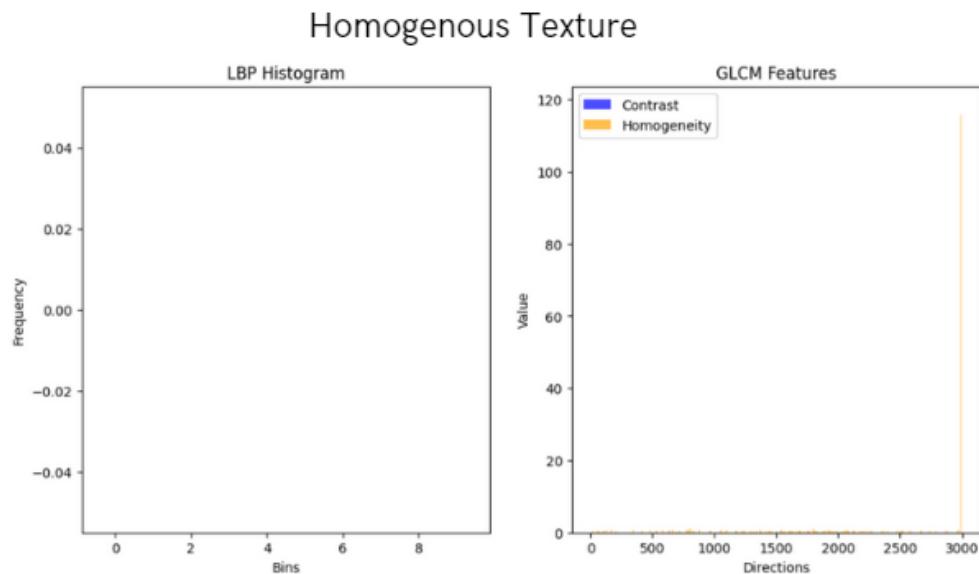
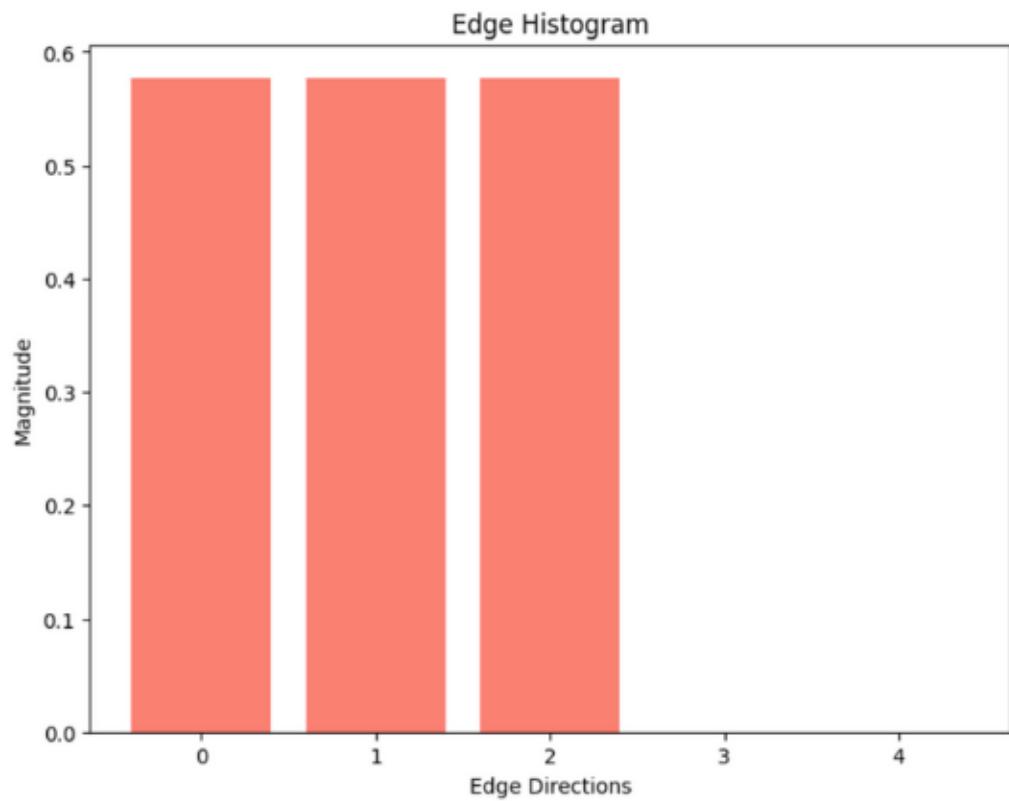




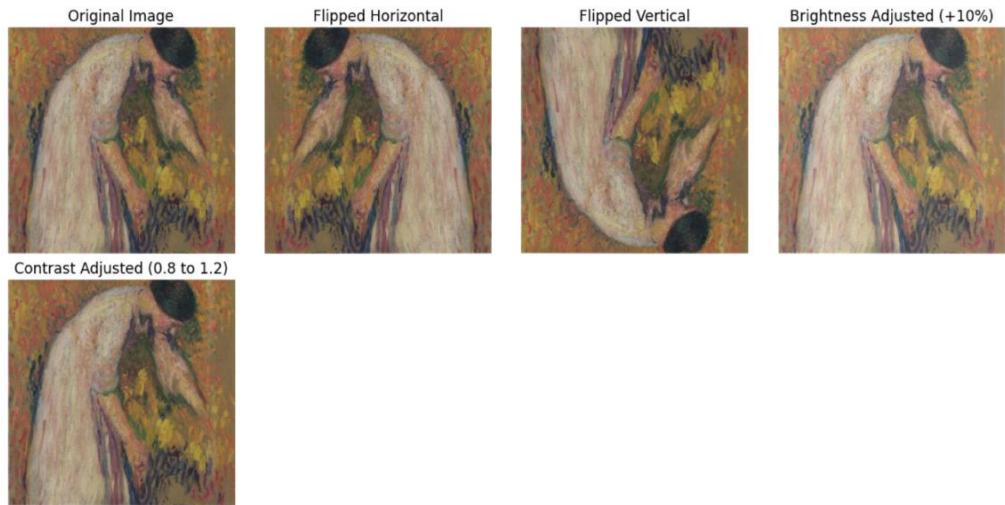


AI Generated Image:





Kami juga menggunakan beberapa augmentasi untuk training pada model. Augmentasi data yang kami gunakan diantaranya adalah flipped horizontal, flipped vertical, brightness adjust, dan contrast adjust. Berikut adalah contohnya:

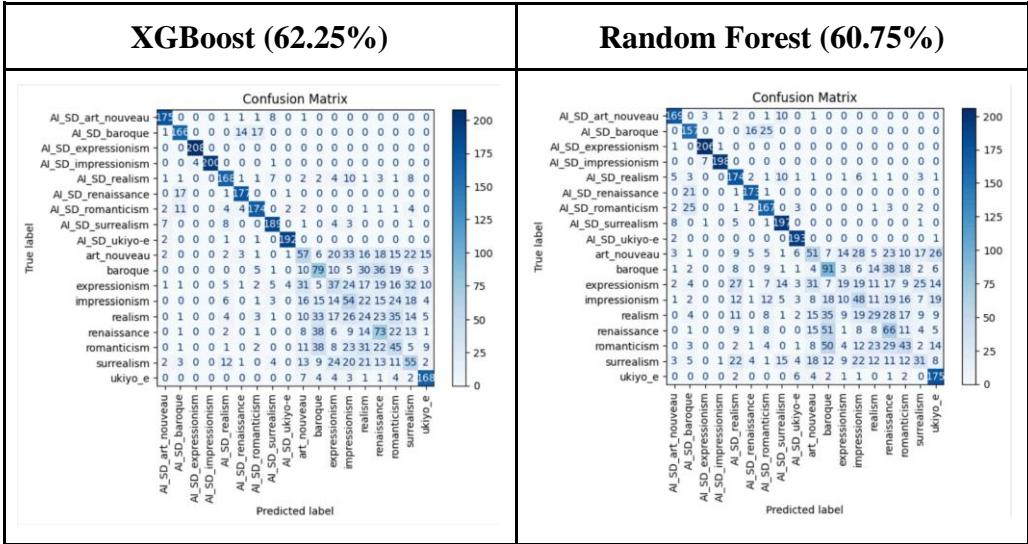


Experiment Results

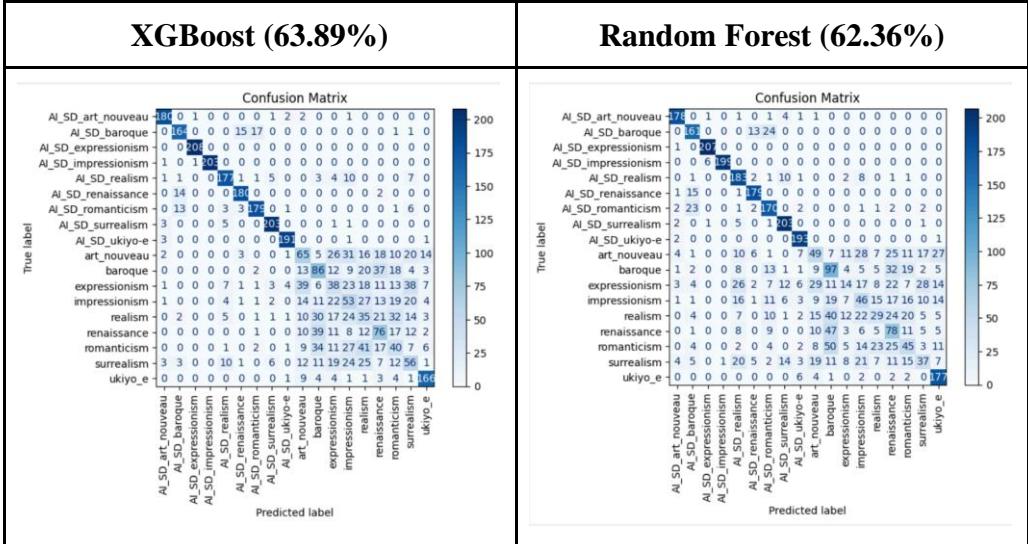
Berikut adalah hasil dari hasil eksperimen pertama yang kami lakukan

Model	Total Images	Accuracy
XGB + HSV	18000 images	62,25%
XGB + HSV + Edge	18000 images	63,89%
XGB + MPEG7	3600 images	64,17%
RF + HSV	18000 images	60,75%
RF + HSV + Edge	18000 images	62,36%
RF+ MPEG7	3600 images	58,47%
Xception	18000 images	72.16%
ResNet	18000 images	72.80%
ViT (Vision Transformer)	18000 images	80,8%

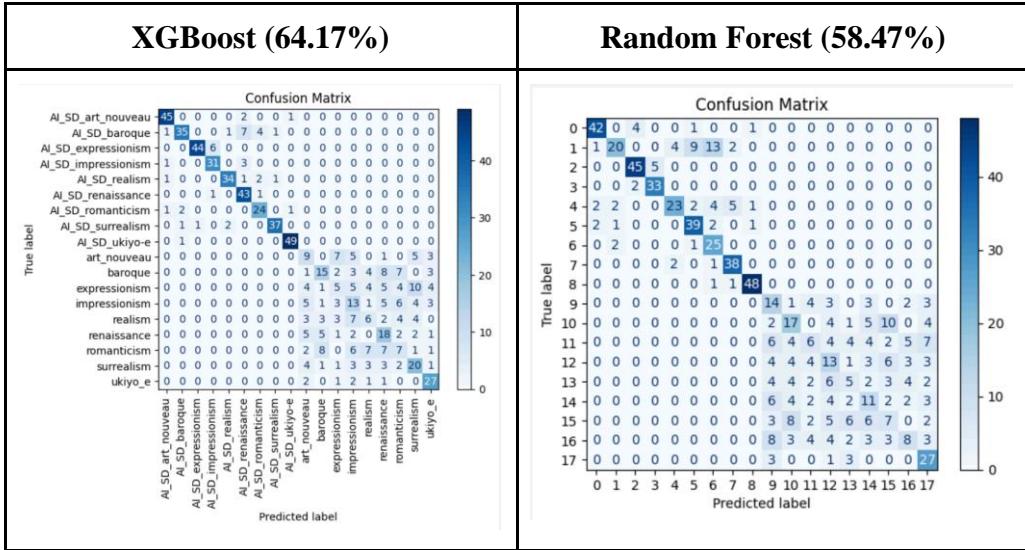
Methods: HSV



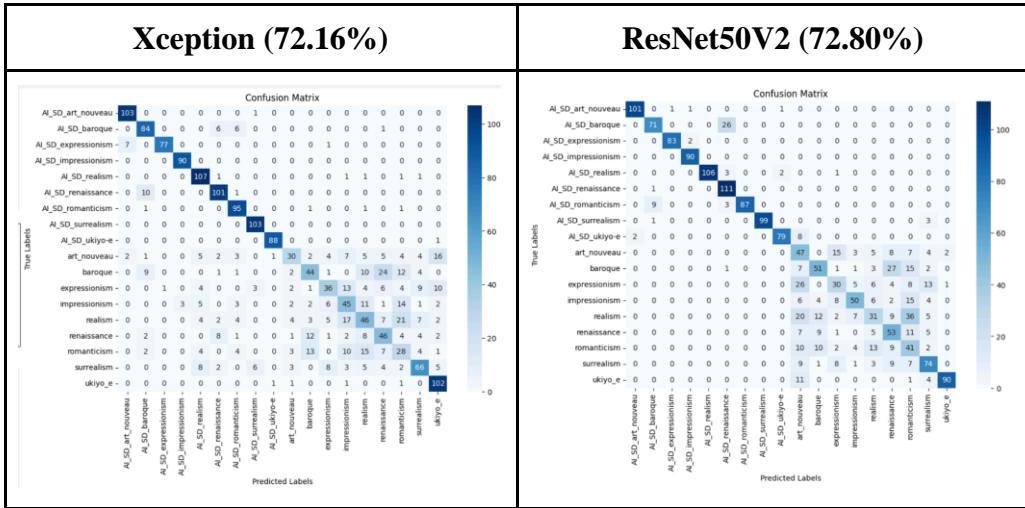
Methods: HSV + Edge



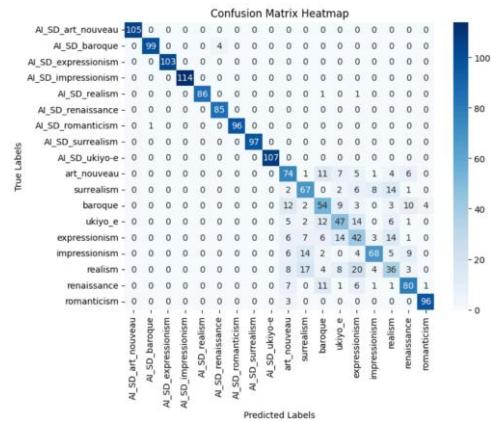
Methods: MPEG - 7 (Color Layout + Color Structure + Edge Histogram + Homogenous Texture)



Methods: CNN (Xception & ResNet50V2)



Methods: ViT Google / ViT - base - patch 16 - 224 (80.8%)

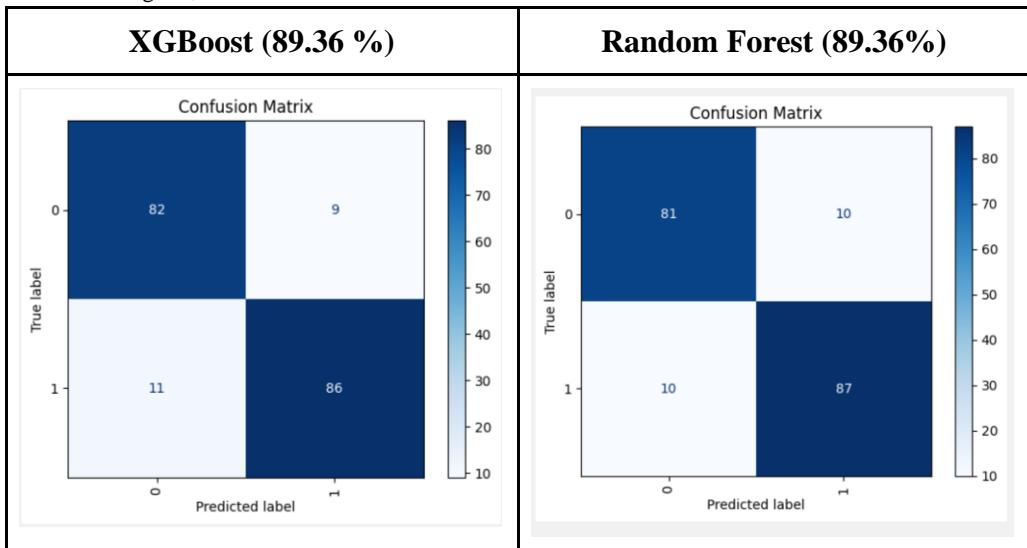


Berikut adalah hasil dari eksperimen kedua yang kami lakukan:

Model	Total Images	Accuracy
XGB + HSV	940 images	89,36%
XGB + HSV + Edge	940 images	88,30%
XGB + MPEG7	940 images	87,23%
RF + HSV	940 images	89,36%
RF + HSV + Edge	940 images	88,30%
RF+ MPEG7	940 images	74,47%
Xception	3660 images	81%
ResNet	3660 images	75%
ViT (Vision Transformer)	3660 images	81,14%

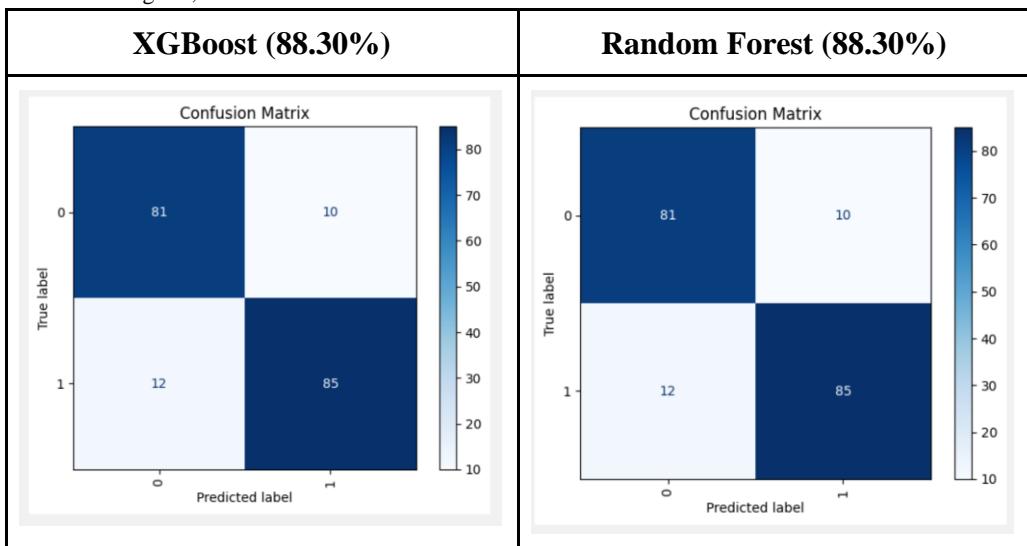
Methods: HSV

Notes: 0 = Original, 1 = AI



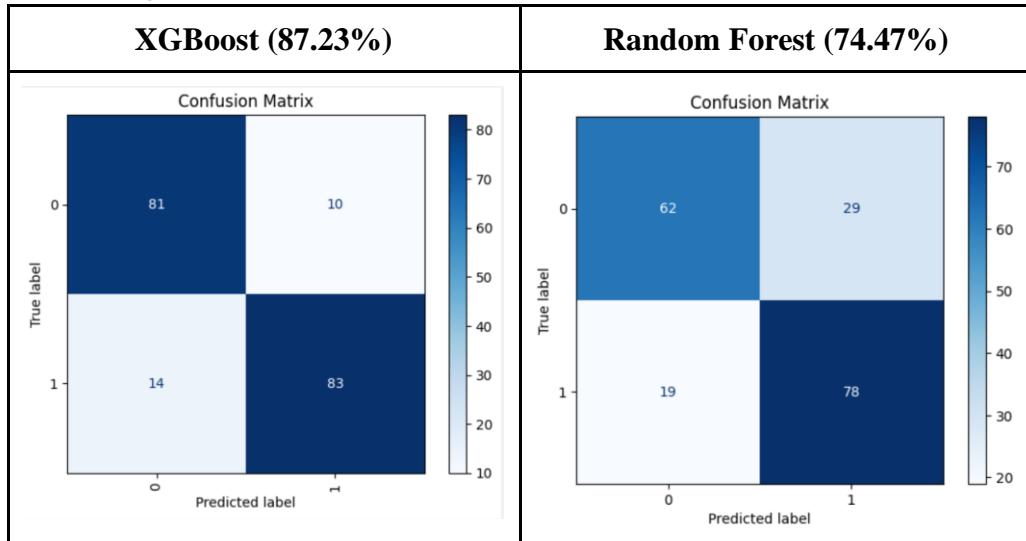
Methods: HSV + Edge

Notes: 0 = Original, 1 = AI

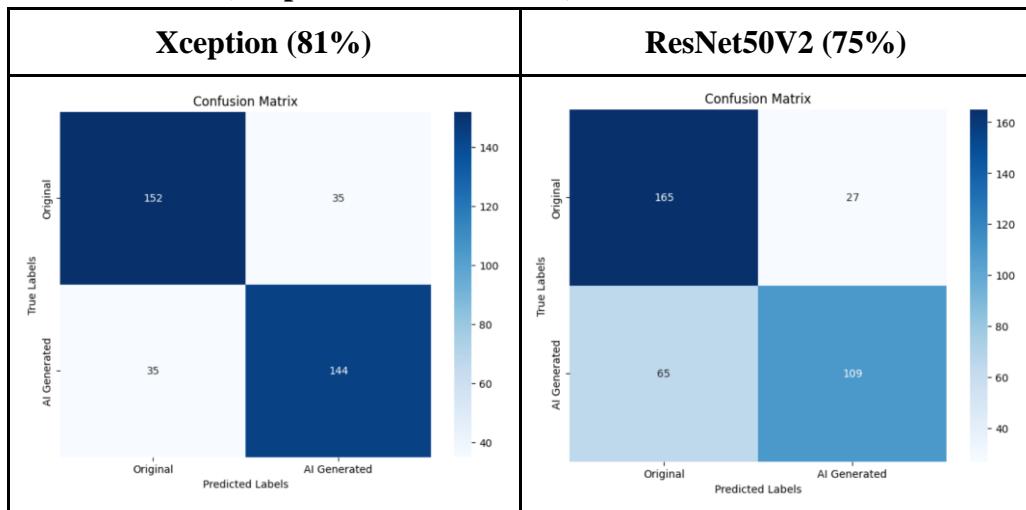


Methods: MPEG - 7 (Color Layout + Color Structure + Edge Histogram + Homogenous Texture)

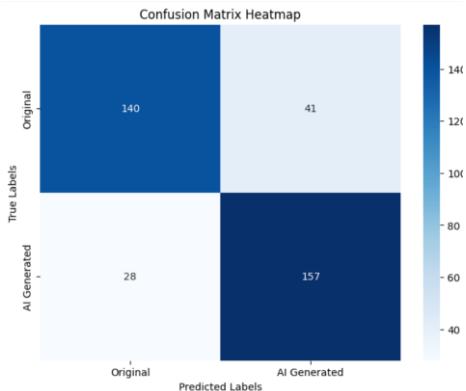
Notes: 0 = Original, 1 = AI



Methods: CNN (Xception & ResNet50V2)



Methods: ViT Google / ViT - base - patch 16 - 224 (81.4%)

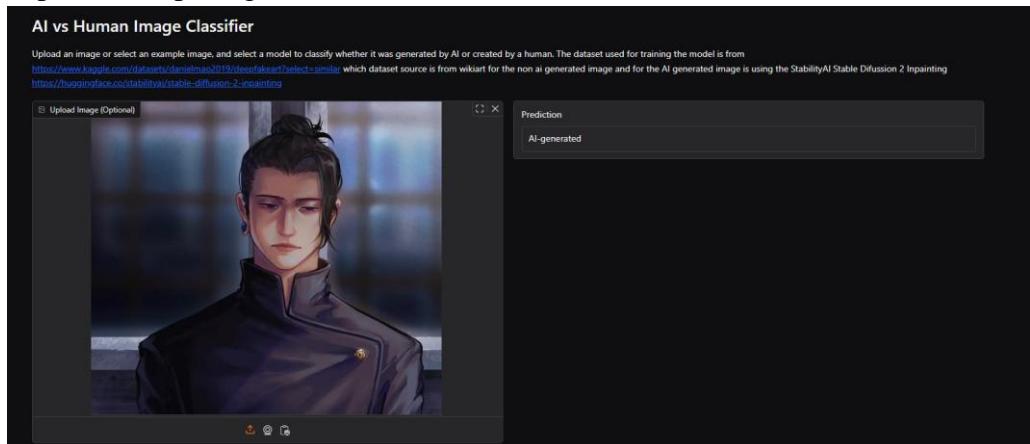


Pada hasil yang didapatkan, untuk model machine learning model terbaiknya adalah pada model XGB dan RF (XGBoost dan Random Forest) dengan feature extraction HSV saja. Model machine learning yang kami buat memiliki accuracy yang lebih baik dibandingkan dengan deep learning karena dataset yang digunakan untuk model machine learning lebih sedikit. Alasan lainnya adalah data yang didapat setelah proses feature extraction bekerja dengan baik pada *Tree - Based model* sehingga didapat accuracy yang cukup bagus.

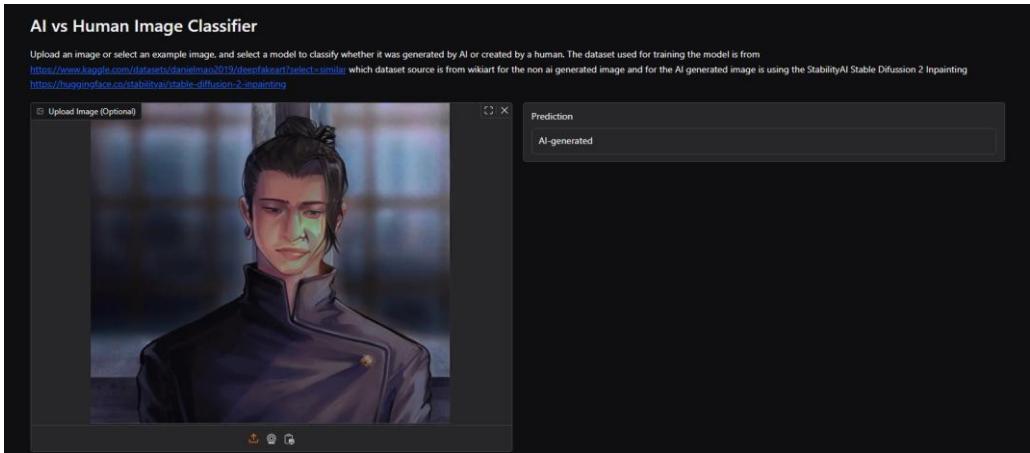
Kami menggunakan dataset yang lebih kecil untuk training model machine learning karena pada saat training jika menggunakan dataset yang jumlahnya sama dengan dataset yang kami gunakan pada model deep learning, jupyter notebook pada kaggle sering restart atau berhenti di tengah-tengah proses training, terutama ketika proses augmentasi sedang berjalan. Kami menggabungkan dataset deepfakes dan dataset art bench, hal ini kami lakukan agar model yang kami buat bisa bekerja lebih baik, terutama karena perbedaan versi pada model Stable Diffusion yang digunakan untuk membuat dataset deepfakes dan art bench. Stable Diffusion yang digunakan untuk dataset deepfakes adalah Stable Diffusion 2 Inpainting, sedangkan untuk dataset artbench adalah Stable Diffusion 1.

Untuk model deep learning yang mendapatkan accuracy terbaik adalah pada model ViT (Vision Transformer). Perbedaan cara kerja ViT dengan Xception dan Resnet adalah ViT bekerja lebih fokus kepada patch-patch kecil, sedangkan untuk model yang berbasis CNN lebih fokus kepada konvolusi.

Kemudian, kami juga sudah melakukan test pada dataset yang belum pernah kami coba saat training, untuk jenis dataset yang sama model yang kami buat sudah dapat memprediksi gambar mana yang merupakan AI generated dan mana yang bukan. Namun, ketika kami mencoba untuk memprediksi gambar menggunakan jenis gambar yang belum pernah ada sebelumnya, model kami akan kesulitan untuk memprediksi gambar tersebut. Misalnya saja kami menggunakan gambar anime atau gambar fanart kartun jepang (karya seni yang dibuat oleh penggemar dari sebuah karya fiksi), pada contoh gambar yang kami gunakan model kami kesulitan untuk mendeteksi gambar manusia, namun jika menggunakan gambar yang AI generated sudah benar. Hal tersebut dapat terjadi karena perbedaan gambar yang digunakan model pada saat training. Untuk jenis gambar fanart kartun jepang belum ada pada dataset yang kami gunakan, sehingga hal tersebut membuat model kami kurang mampu untuk mendeteksi apakah gambar tersebut merupakan AI generated atau bukan. Sebagai contoh dapat dilihat pada gambar dibawah ini



Original Image Prediction



Generated Image Prediction

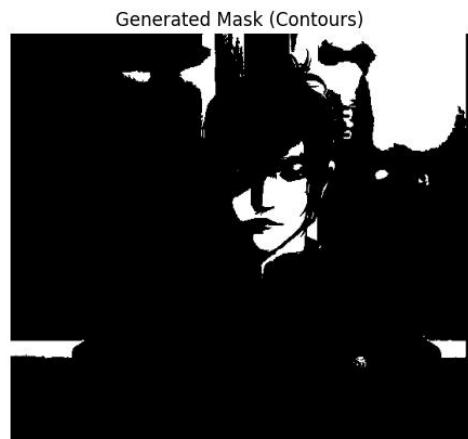
Sumber gambar:

https://www.instagram.com/crybanana7/p/DDo9SgWSDDt/?img_index=1

Untuk membuat generated imagennya, kami menggunakan model AI yang digunakan untuk generate dataset yang kami gunakan yaitu Stable Diffusion 2 Inpainting, dan dengan prompt yang sama juga. Gambar dibawah ini adalah cara kami membuat generated imagennya:

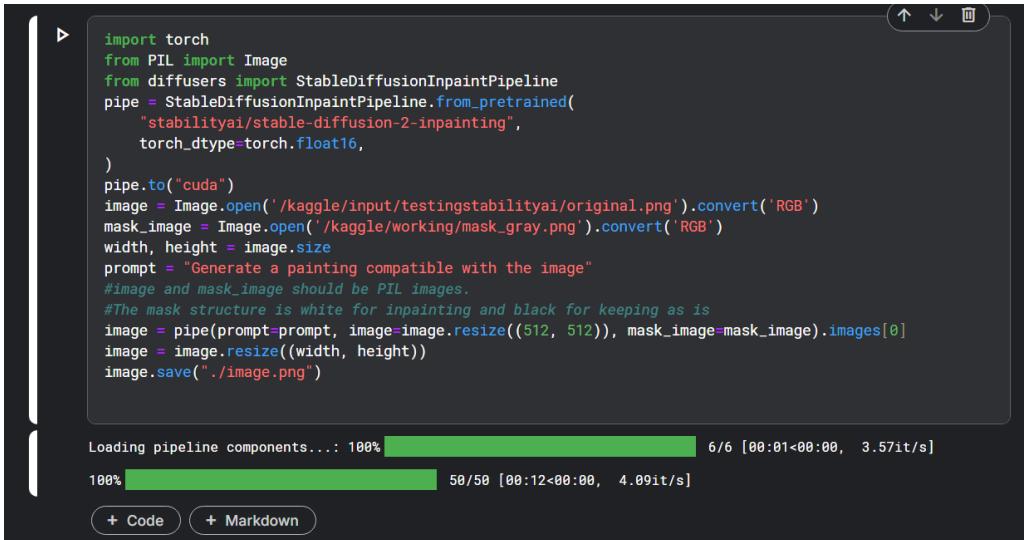


Original Image



Generated Mask (Contours)

Pertama kami membuat masknya terlebih dahulu dari original imagennya, kemudian baru dibuat kembali menggunakan Stable Diffusion 2 Inpainting



```
import torch
from PIL import Image
from diffusers import StableDiffusionInpaintPipeline
pipe = StableDiffusionInpaintPipeline.from_pretrained(
    "stabilityai/stable-diffusion-2-inpainting",
    torch_dtype=torch.float16,
)
pipe.to("cuda")
image = Image.open('/kaggle/input/testingstabilityai/original.png').convert('RGB')
mask_image = Image.open('/kaggle/working/mask_gray.png').convert('RGB')
width, height = image.size
prompt = "Generate a painting compatible with the image"
#image and mask_image should be PIL images.
#The mask structure is white for inpainting and black for keeping as is
image = pipe(prompt=prompt, image=image.resize((512, 512)), mask_image=mask_image).images[0]
image = image.resize((width, height))
image.save("./image.png")
```

Loading pipeline components...: 100% 6/6 [00:01<00:00, 3.57it/s]

100% 50/50 [00:12<00:00, 4.09it/s]

+ Code + Markdown

Keduanya menghasilkan prediksi yang sama yaitu AI-Generated. Terakhir terdapat kendala resource, dimana untuk membuat model deep learning atau machine learning yang bisa memprediksi berbagai jenis art dibutuhkan resource dari GPU, RAM, CPU yang besar, sehingga jenis dataset yang kami gunakan tidak bisa terlalu banyak dan terbatas.

Conclusion

Kesimpulan dari percobaan pertama dan percobaan kedua yang kami lakukan, kami dapat menyimpulkan dua hal krusial dalam penelitian ini:

Dataset

Dataset yang kami gunakan untuk training model terbatas, jika kami dapat menggunakan dataset dengan jumlah yang lebih besar dan bervariasi model yang kami buat dapat lebih bagus lagi, terdapat kemungkinan juga bahwa model yang kami buat dapat mendeteksi AI Generated Art yang dibuat oleh jenis AI selain Stable Diffusion yang sudah digunakan sekarang.

Model

Dari model sendiri sudah cukup baik jika menggunakan dilakukan testing menggunakan art atau gambar dengan jenis yang sama yang digunakan untuk training model.

Keterbatasan

Kami memiliki keterbatasan pada bagian resource untuk training model, baik itu CPU, RAM, dan GPU. Hal ini menyebabkan model yang kami buat juga tidak dapat dilatih dengan dataset yang sangat banyak, dan karena kami training menggunakan kaggle notebook terdapat juga batasan seperti hanya bisa training sekitar 30 jam per minggunya.

References

- [1] L. P. F. Guenther, "Homo Allegoris: How Art Perception and Allegory Analysis Reveal the Life Script Ideology," *Hu Arenas*, vol. 4, pp. 471–486, 2021, doi: 10.1007/s42087-020-00117-7.
- [2] C. Then, E. J. Soewandi, M. F. Danial, S. Achmad, and R. Sutoyo, "The Impact of Artificial Intelligence on Art - A Systematic Literature Review," in *Proceeding - IEEE 9th Information Technology International Seminar, ITIS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ITIS59651.2023.10420208.
- [3] H. H. Jiang et al., "AI Art and its Impact on Artists," in *AIES 2023 - Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, Association for Computing Machinery, Inc, Aug. 2023, pp. 363–374. doi: 10.1145/3600211.3604681.
- [4] The Guardian. (2024, May 8). *Fake Monet and Renoir on eBay among counterfeits identified using AI.* The Guardian. <https://www.theguardian.com/artanddesign/article/2024/may/08/fake-monet-and-renoir-on-ebay-among-counterfeits-identified-using-ai>
- [5] Aboutalebi, H., Mao, D., Fan, R., Xu, C., He, C., & Wong, A. (2024). *DeepfakeArt Challenge: A benchmark dataset for generative AI art forgery and data poisoning detection.* arXiv. <https://arxiv.org/abs/2306.01272>
- [6] Kusuma, S. W., Natalia, F., Ko, C. S., & Sudirman, S. (2024). Detection of AI-generated anime images using deep learning. *ICIC Express Letters Part B: Applications*, 15(3), 295–301. <https://doi.org/10.24507/icicelb.15.03.295>
- [7] Bianco, Tommaso & Castellano, Giovanna & Scaringi, Raffaele & Vessio, Gennaro. (2023). Identifying AI-Generated Art with Deep Learning.
- [8] Huang, Yin-Fu & Wang, Chang-Tai. (2014). Classification of Painting Genres Based on Feature Selection. 10.1007/978-3-642-54900-7_23.
- [9] Ivanova, Krassimira & Stanchev, Peter & Velikova, Evgenia & Vanhoof, Koen & Depaire, Benoît & Mitov, Iliya & Markov, Krassimir. (2010). Features for Art Painting Classification Based on Vector Quantization of MPEG-7 Descriptors. *Lecture Notes in Computer Science*. 10.1007/978-3-642-27872-3_22.
- [10] Maurício, J., Domingues, I., & Bernardino, J. (2023). Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review. *Applied Sciences*, 13(9), 5521. <https://doi.org/10.3390/app13095521>
- [11] Giannakas, Filippos & Troussas, Christos & Krouskas, Akrivi & Sgouropoulou, C. & Voyatzis, Ioannis. (2021). XGBoost and Deep Neural Network Comparison: The Case of Teams' Performance. 10.1007/978-3-030-80421-3_37.

Code dan Model bisa diakses pada link berikut:

<https://github.com/JonathanSuryaS/AI-Art-Detection->

Gradio dapat diakses pada link berikut:

https://huggingface.co/spaces/jovanliem/ai_generated_art_detector

https://huggingface.co/spaces/jovanliem/ai_generated_art_detector_ViT