Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.2, April 2026, pp. 2057~2065
e-ISSN: 2548-6861

AI Image Detection Using EfficientNetB0 Architecture on Generative Adversarial Network

Rudi Hartadi Setiawan [1*], Sindhu Rakasiwi [2*]
[*] Teknik Informatika, Universitas Dian Nuswantoro
111202214081@mhs.dinus.ac.id [1], sindhu.rakasiwi@dsn.dinus.ac.id [2]

Article Info

Article history:

Received 2026-02-11
Revised 2026-04-13
Accepted 2026-04-26

Keyword:

GAN Detection,
EfficientNetB0,
Cross-Generator Generalization,
Transfer Learning,
Fake Images.

ABSTRACT

The development of Generative Adversarial Networks (GANs) has produced synthetic images that are increasingly difficult to distinguish from real photographs, driving the need for reliable automated detection systems. The core problem is that detection models frequently fail when tested on generators different from those used during training, a phenomenon known as the generalization gap. This study evaluates EfficientNetB0 in detecting AI-generated images through a cross-generator approach across five GAN architectures, namely ProGAN, StyleGAN, StyleGAN2, StyleGAN2-ADA, and StyleGAN3. The model was trained on a StyleGAN2 dataset using transfer learning from ImageNet, then evaluated on the four other generators without retraining. In-domain results showed an accuracy of 99.30%, an F1-Score of 99.30%, and an AUC of 99.99%. However, out-of-domain testing revealed an average accuracy drop of 16.64%. StyleGAN2-ADA achieved 99.33% due to its architectural similarity to StyleGAN2, suggesting that generator architecture is a more decisive factor than training strategy. In contrast, StyleGAN3 dropped to an accuracy of 63.05% because its alias-free architecture eliminates the visual patterns that the model typically relies on for detection. The model tends to recognize real images well with high specificity, but its ability to detect synthetic images declines with low sensitivity, and the false negative rate rising from 0.15% to 72.65% on StyleGAN3. These findings highlight the limitations of single-generator training and the need to explore multi-generator strategies or frequency-based feature methods.

I. PENDAHULUAN

Generative Adversarial Networks (GAN) yang diperkenalkan Goodfellow dkk. [1] telah mengubah lanskap gambar sintesis. Evolusi dari ProGAN [2] hingga StyleGAN3 [3] menunjukkan meningkatnya kualitas hasil pembuatan gambar AI, yang membuat sulitnya membedakan gambar asli dengan gambar AI. Penelitian Nightingale dkk. [4] di MIT Media Lab menunjukkan akurasi deteksi manusia hanya 50-70%, tidak jauh berbeda dari tebakan acak. Di sisi lain, penyalahgunaan untuk *deepfake*, manipulasi identitas, dan disinformasi terus meningkat [5], [6]. Platform media sosial dan organisasi fact-checking telah mengidentifikasi kebutuhan mendesak akan sistem deteksi otomatis yang kuat dan fleksibel.

Deteksi gambar GAN menghadapi kompleksitas yang berbeda dari domain deteksi lain seperti spam atau malware. Generator terbaru seperti StyleGAN3 menerapkan alias-free architecture yang secara fundamental mengurangi pola visual seperti tekstur terkunci, pola visual fase, dan pola aliasing yang menjadi ciri khas generator sebelumnya [3]. Bahkan pengembang GAN mengakui kesulitan mendeteksi output karya mereka sendiri seiring evolusi teknologi. Tantangan ini diperburuk oleh sifat dinamis generator dimana pola visual yang dapat dideteksi hari ini mungkin sudah tidak ada di model generasi berikutnya.

Riset deteksi gambar AI telah mengeksplorasi berbagai pendekatan dengan tingkat keberhasilan yang bervariasi. Wang dkk. [7] menunjukkan CNN sederhana dapat mencapai akurasi tinggi dalam mendeteksi gambar GAN [8], [9]. Namun, dengan catatan hasil penelitian berlaku saat

penelitian tersebut dilakukan. Model yang dilatih pada satu generator mengalami penurunan performa ketika diuji pada generator berbeda, fenomena yang dikenal sebagai kesenjangan generalisasi [7], [8], [10]. Penelitian mereka mengevaluasi 3 arsitektur GAN dan mengidentifikasi masalah lintas-generator sebagai hambatan utama untuk deployment praktis.

Ancaman deepfake terus berkembang seiring kemampuan generator yang semakin sulit dideteksi secara visual [5], [6]. Gambar sintetis yang telah mengalami kompresi JPEG atau disebarluaskan melalui media sosial semakin menantang untuk diidentifikasi karena sebagian pola visual deteksi hilang dalam proses tersebut. Sebagai alternatif, Pendekatan berbasis frekuensi [11] menawarkan alternatif dengan menggunakan sidik jari GAN dalam domain frekuensi. Proses upsampling pada generator GAN meninggalkan pola visual periodis pada spektrum Fourier, terutama pada frekuensi yang berkorelasi dengan faktor upsampling yang digunakan. Pola visual ini bersifat karakteristik per-generator dan lebih persisten dibanding pola spasial yang mudah dihilangkan kompresi. Pendekatan forensik berbasis derau juga digunakan, seperti filter SRM (Steganalysis Rich Model) yang memanfaatkan perbedaan statistik antara derau kamera asli dan derau generator GAN. Biaya komputasi yang tinggi dan kompleksitas implementasi membatasi aplikasi real-time, menciptakan trade-off antara kedalaman analisis dan efisiensi yang belum terpecahkan.

Transfer learning dari *pre-trained* vision models memberikan keseimbangan praktis. Model yang telah belajar mengenali fitur visual dari dataset besar seperti ImageNet [12] dapat di *fine-tune* untuk tugas spesifik deteksi gambar AI. Fitur tingkat rendah seperti *edge detectors*, *texture patterns*, dan *color distributions* yang berguna untuk klasifikasi objek ternyata juga relevan untuk mendeteksi pola visual GAN. Pendekatan ini telah berhasil dalam berbagai tugas *computer vision* [13], [14], namun evaluasi yang mendalam dalam konteks lintas-generator deteksi masih terbatas.

EfficientNet [15] menghadirkan pendekatan baru melalui *compound scaling* dengan menskalakan *depth*, *width*, dan resolusi secara bersamaan dengan proporsi teroptimasi. EfficientNetB0 sebagai baseline memiliki karakteristik yang sesuai untuk tugas ini yaitu dengan 5.3 juta parameter, model ini jauh lebih efisien dibanding ResNet-50 [13] (25 juta parameter) atau VGG-16 [16] (138 juta parameter), namun mampu mencapai performa seimbang. Ukuran model yang kompak ini tidak hanya menguntungkan dari sisi efisiensi komputasi, tetapi juga mengurangi risiko *overfitting* pada pola visual spesifik dari satu generator, yang merupakan tantangan utama dalam deteksi lintas-generator. Selain itu, compound scaling pada EfficientNetB0 menghasilkan representasi fitur yang lebih seimbang dibanding arsitektur yang hanya menskalakan satu dimensi saja, sehingga lebih sesuai untuk mendeteksi pola visual GAN yang bervariasi antar generator.

Keberagaman ini menciptakan pertimbangan yang sulit [10], [17] model yang terlalu mempelajari pada satu generator akan gagal pada generator lain, sementara model yang terlalu general akan kehilangan kemampuan mendeteksi pola visual yang halus namun penting. Pertanyaannya adalah bagaimana kemampuan model yang dilatih pada satu generator dapat digunakan untuk generator berbeda, terutama jika generator tersebut memiliki arsitektur yang berbeda dari yang dipelajari.

Penelitian ini melakukan evaluasi terhadap EfficientNetB0 untuk deteksi lintas-generator. Berbeda dengan penelitian sebelumnya yang menggunakan 2-3 generator, penelitian ini mengevaluasi lima arsitektur GAN yang merepresentasikan evolusi teknologi dari 2018 hingga 2021. Setiap generator dipilih berdasarkan pertimbangan tertentu, ProGAN mewakili era awal GAN beresolusi tinggi, seri StyleGAN hingga StyleGAN2 menunjukkan perkembangan progresif dalam kualitas dan kontrol, StyleGAN2-ADA memperkenalkan teknik augmentasi adaptif, sementara StyleGAN3 merepresentasikan state-of-the-art terkini dengan arsitektur yang fundamentalnya berbeda.

Kontribusi utama penelitian ini meliputi:

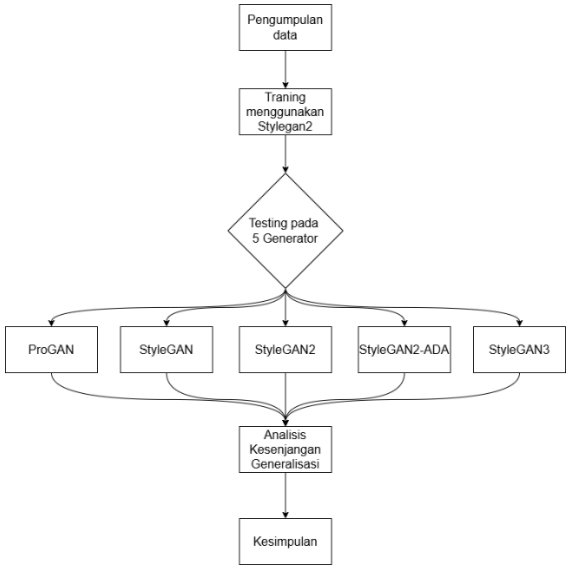
1) Benchmark performa EfficientNetB0 pada lima generator berbeda (ProGAN, StyleGAN, StyleGAN2, StyleGAN2-ADA, StyleGAN3) yang dapat menjadi baseline untuk penelitian lanjutan.
2) Analisis pola kesalahan dan karakteristik setiap generator yang memberikan wawasan untuk desain sistem deteksi yang lebih kuat dan generalisasi yang lebih baik.
3) Kuantifikasi kesenjangan generalisasi antara deteksi in-domain dan out-of-domain yang dapat menjadi acuan pengembangan model yang lebih kuat terhadap evolusi teknologi GAN.

II. METODE

A. Desain Penellitian

Penelitian ini mengevaluasi kemampuan generalisasi EfficientNetB0 melalui pendekatan single-generator training dengan multi-generator testing. Model dilatih secara eksklusif pada StyleGAN2, kemudian dievaluasi pada lima generator berbeda tanpa fine-tuning atau retraining tambahan untuk mengukur kemampuan generalisasi murni dari *transfer learning*.

Gambar 1 alur penelitian dimulai dari pengumpulan data gambar asli (FFHQ) dan gambar sintetis dari lima generator GAN. Model EfficientNetB0 kemudian dilatih menggunakan dataset StyleGAN2 saja. Setelah pelatihan, model yang sama diuji pada kelima generator tanpa pelatihan ulang, yaitu StyleGAN2 sebagai in-domain, serta ProGAN, StyleGAN, StyleGAN2-ADA, dan StyleGAN3 sebagai out-of-domain. Hasil dari kelima pengujian tersebut dianalisis untuk mengukur kesenjangan generalisasi dan menghasilkan kesimpulan penelitian.



Gambar 1.Alur Penelitian

Pendekatan single-generator training dipilih untuk mengisolasi dan mengukur seberapa baik fitur yang dipelajari dapat diterapkan ke generator lain, yang merupakan pertanyaan inti dalam deteksi lintas-generator. Protokol eksperimen menggunakan random seed (42) agar penelitian dapat direplikasi.

B. Dataset

1) Struktur dan Komposisi: Dataset penelitian terdiri dari gambar wajah beresolusi 1024×1024 piksel dengan struktur seimbang. Data training dan validasi menggunakan gambar asli dari FFHQ yang dipasangkan dengan gambar sintetis dari StyleGAN2, sementara data testing menggunakan strategi berbagi gambar asli untuk lima generator berbeda. Distribusi lengkap dataset ditunjukkan pada Tabel 1.

TABEL 1
DISTRIBUSI DATASET

Generator	Training	Validation	Testing	Total
FFHQ	10,000	2,000	2,000	14,000
StyleGAN2	10,000	2,000	2,000	14,000
ProGAN	-	-	2,000	2,000
StyleGAN	-	-	2,000	2,000
StyleGAN2-ADA	-	-	2,000	2,000
StyleGAN3	-	-	2,000	2,000

Tabel 1 menunjukkan distribusi dataset dengan total 36.000 gambar (14.000 asli + 22.000 sintetis). Data training terdiri dari 10.000 gambar asli dan 10.000 gambar StyleGAN2, sedangkan data validasi menggunakan 2.000 gambar dari masing-masing kategori.

Untuk pengujian, diterapkan strategi berbagi gambar asli di mana 2.000 gambar asli yang sama dipasangkan dengan 2.000 gambar sintetis dari masing-masing lima generator (StyleGAN2, ProGAN, StyleGAN, StyleGAN2-ADA, StyleGAN3), menghasilkan lima set pengujian tersendiri. Strategi ini mengikuti protokol evaluasi lintas-generator dari Wang dkk. [7] dan memastikan bahwa perbedaan performa model disebabkan secara murni oleh karakteristik generator, bukan oleh variasi gambar asli.

Pembagian data dilakukan secara random dengan *stratified sampling* untuk menjaga distribusi karakteristik wajah yang seimbang. Penting untuk dicatat bahwa 2.000 gambar asli untuk testing sepenuhnya terpisah dari training set dan validation set untuk mencegah data leakage.

2) Sumber Data: Gambar asli berasal dari Flickr-Faces-HQ (FFHQ) dataset [18], koleksi 70.000 gambar beresolusi 1024×1024 dengan keberagaman tinggi dalam pose, ekspresi, usia, dan etnis. Dipilih subset 14.000 gambar secara acak.

StyleGAN3 [3] dihasilkan menggunakan model pra-latih resmi dengan vektor laten acak tanpa pemangkasan atau pemilihan untuk representasi yang tidak memihak. ProGAN [2], StyleGAN [18], StyleGAN2 [19], dan StyleGAN2-ADA [20] diperoleh dari sampel gambar yang dibagikan di repositori GitHub resmi masing-masing generator. Gambar-gambar ini merupakan output representatif dari model pra-latih yang telah divalidasi oleh pengembang asli.

Pendekatan ini memastikan bahwa gambar sintetis mencerminkan kualitas output standar dari setiap generator tanpa kecendurungan seleksi.

C. Arsitektur Model dan Transfer Learning

Model menggunakan EfficientNetB0 [15] dengan metode *compound scaling* yang menyeimbangkan kedalaman, lebar, dan resolusi. Arsitektur terdiri dari 16 blok MBConv dengan optimasi squeeze-and-excitation. Detail konfigurasi arsitektur ditunjukkan pada Tabel 2.

TABEL 2
KONFIGURASI ARSITEKTUR EFFICIENTNETB0

Komponen	Konfigurasi	Parameter
Input	224 × 224 × 3 (RGB)	-
Stem + MBConv Blocks	16 blocks, 7 stages	~4.8M
Head Conv + Pooling	Conv 1×1, 1280 channels	~320K
Classifier (Modified)	Linear (1280 - 2) + Softmax	2,562

Pada Tabel 2 distribusi parameter EfficientNetB0 terdiri dari empat komponen utama seperti yang terlihat. Parameter kurang lebih 4.8M berada pada backbone yang terdiri dari stem dan 16 blok MBConv tersebar di 7 stages. Head network dengan convolutional layer 1×1 (1280 channels) dan pooling menggunakan kurang lebih 320K parameter, sementara classifier layer yang dimodifikasi hanya memerlukan 2.562 parameter. Total keseluruhan sekitar 5.3M parameter, jauh lebih efisien dibandingkan ResNet-50 (25.6M) atau VGG-16 (138M).

Modifikasi dilakukan pada classifier layer dengan mengubah dimensi output dari 1000 (kelas ImageNet) menjadi 2 untuk klasifikasi biner antara gambar asli dan palsu.

Layer model classifier diganti dengan Linear(1280 - 2) yang diinisialisasi secara acak menggunakan Kaiming uniform, sementara layer lain mempertahankan bobot pra-latih dari ImageNet [12]. Full fine-tuning dilakukan pada semua layer karena tugas deteksi gambar GAN memerlukan adaptasi dari fitur tingkat rendah seperti tepi dan tekstur hingga fitur tingkat tinggi seperti pola semantik. Berbeda dengan pendekatan yang hanya melatih classifier head, full fine-tuning memungkinkan model menyesuaikan seluruh representasi visual terhadap karakteristik gambar GAN yang tidak ada pada data ImageNet.

D. Konfigurasi Pelatihan

1) Hardware dan Setup: Pelatihan dilakukan pada Google Colab (GPU Tesla T4, 16 GB VRAM) menggunakan PyTorch 2.0+ dan CUDA 11.8. Dataset disimpan di local disk untuk meminimalkan I/O latency.

2) Hyperparameter dan Optimizer:

TABEL 3
KONFIGURASI HYPERPARAMETER PELATIHAN

Parameter	Nilai
Optimizer	Adam
Learning Rate	0.001-0.00025
Batch Size	16
Max Epochs	30
Early Stopping	Patience = 5

Tabel 3 menunjukkan konfigurasi hyperparameter yang digunakan dalam pelatihan model. Optimizer Adam dipilih karena kemampuannya dalam kecepatan belajar adaptif dan stabilisasi yang cepat. *Learning rate* dimulai dari 0.001 dan secara bertahap diturunkan hingga 0.00025 untuk stabilisasi training di fase akhir. Batch size 16 dipilih sebagai keseimbangan antara kestabilan gradient dan keterbatasan memori GPU. Model dilatih maksimal 30 epoch dengan mekanisme *early stopping* (patience = 5) sebagai pengaman untuk mencegah *overfitting*.

E. Augmentasi Data

TABEL 4
AUGMENTASI DATA

Transformasi	Parameter
Resize	256 × 256
Random Crop	224 × 224
Random Horizontal Flip	p = 0.5
Random Rotation	±10°
Color Jitter	Brightness/Contrast/Saturation = 0.2, Hue = 0.1
Normalize	ImageNet μ, σ

Tabel 4 konfigurasi augmentasi data yang diterapkan pada training pipeline. Resize ke 256×256 dilakukan untuk standardisasi dimensi awal, diikuti random crop ke 224×224 untuk memberikan variasi spasial dan meningkatkan kemampuan generalisasi model. Random horizontal flip dengan probabilitas 0.5 diterapkan untuk meningkatkan invariansi terhadap orientasi wajah, sementara random rotation (±10°) memberikan robustness terhadap variasi pose minor tanpa mengubah struktur fundamental wajah.

Setiap augmentasi dipilih untuk meningkatkan robustness tanpa mengubah karakteristik wajah yang penting. Random crop memberikan variasi posisi, flip memanfaatkan simetri wajah, dan rotasi ±10° menangani variasi pose minor. Parameter ColorJitter dipilih berdasarkan best practice [21] dalam face detection, cukup untuk meningkatkan robustness terhadap variasi pencahayaan tanpa mengubah pola visual yang menjadi ciri khas gambar GAN. Normalisasi menggunakan statistik ImageNet penting untuk kesesuaian distribusi input dengan bobot pra-latih dari *transfer learning*.

Validation dan testing menggunakan pipeline minimal tanpa operasi acak untuk evaluasi konsisten. Resize langsung ke 224×224 memastikan model melihat seluruh konten gambar. Tidak adanya augmentasi pada testing bertujuan untuk mengukur performa pada gambar standar, memastikan perbedaan performa antar generator dapat dikaitkan secara murni pada karakteristik generator.

F. Protokol Evaluasi Lintas-Generator

Evaluasi lintas-generator dilakukan dengan menguji model yang sama pada lima test set tanpa retraining atau fine-tuning. Model akurasi validasi pada StyleGAN2 digunakan untuk prediksi pada semua generator. Protokol evaluasi untuk menjaga konsistensi dengan:

1) Preprocessing dilakukan dengan Resize 224×224 + Normalization pada semua generator.
2) Batch size 16 images per batch pada semua generator.
3) Metrik evaluasi yang konsisten pada semua generator.

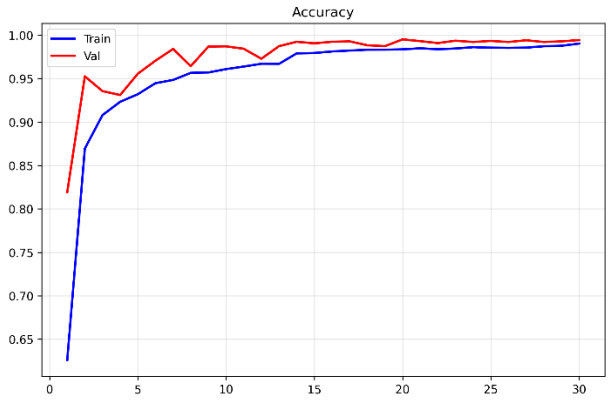
Setiap set pengujian dievaluasi secara tersendiri dengan menghitung confusion matrix (TN, FP, FN, TP) dan menggunakan metrik Akurasi, Presisi, Recall, F1-Score, AUC-ROC, dan False Positive/Negative Rate. Tidak ada penyesuaian threshold tuning, menggunakan ambang batas default 0.5 pada probabilitas softmax untuk semua generator agar evaluasi tidak berpihak. Metrik-metrik ini dipilih untuk memberikan gambaran mendalam tentang performa model dari berbagai perspektif seperti akurasi keseluruhan, kemampuan deteksi (recall), keandalan prediksi (precision), dan keseimbangan antara false positive vs false negative. Kesenjangan generalisasi dihitung dengan mengurangi hasil akurasi in-domain dengan rata-rata akurasi out-of-domain.

III. HASIL DAN PEMBAHASAN

A. Performa Model

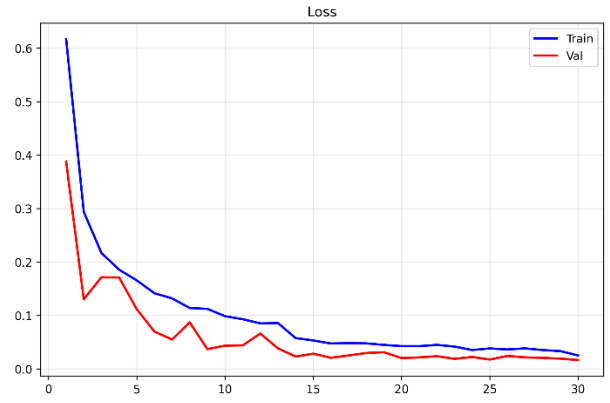
Model EfficientNetB0 dilatih selama 30 epoch dengan penurunan loss yang stabil. Model mencapai performa optimal di epoch 30 dengan akurasi validasi 99.42% dan AUC 1.0000, artinya pemisahan berhasil antara kelas asli dan sintetis. *Learning rate* diturunkan dari 0.0005 ke 0.00025 pada epoch 29, memungkinkan optimasi yang lebih halus di fase akhir pelatihan. Gap minim antara akurasi training

(99.02%) dan akurasi validation (99.42%) artinya model tidak mengalami *overfitting* meskipun menggunakan full fine-tuning pada semua layer.



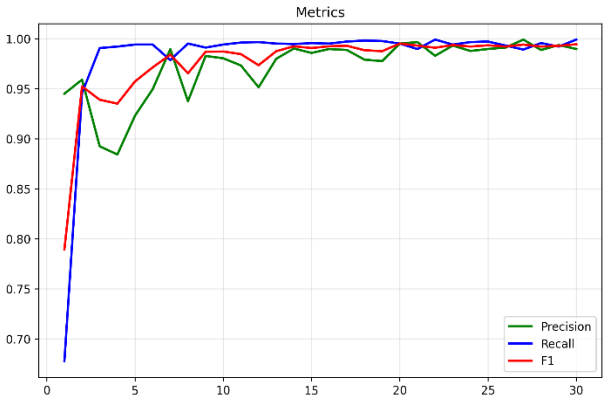
Gambar 2. Kurva Akurasi

Pada Gambar 2 akurasi mengalami peningkatan cepat di 5 epoch pertama, mencapai di atas 95%, dan stabil di atas 99% pada epoch akhir dengan jarak yang tidak jauh antara training dan validation, sehingga tidak terjadi *overfitting* pada saat pelatihan.



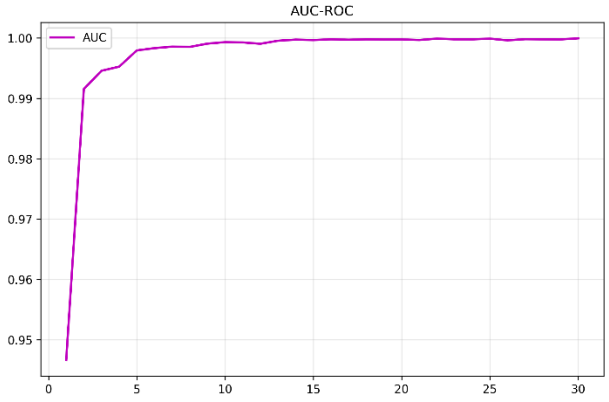
Gambar 3. Kurva Loss

Gambar 3 Kurva loss mengalami penurunan konsisten dan stabil, dengan training dan validation loss yang menurun secara stabil dari nilai awal sekitar 0.6 hingga mendekati 0.



Gambar 4. Kurva metrik klasifikasi

Gambar 4 Kurva metrik klasifikasi menunjukkan precision, recall, dan F1-score yang ketiganya stabil di atas 99% setelah epoch 15, meskipun terdapat variasi kecil di fase tengah training.



Gambar 5. Kurva AUC-ROC

Gambar 5 Kurva AUC-ROC peningkatan tajam di epoch awal dan stabil sekitar 0.999 setelah epoch 13, artinya performa klasifikasi yang mendekati optimal.

Transfer learning dari ImageNet bekerja dengan baik, terlihat dari pola pelatihan ini. Model tidak perlu mempelajari fitur-fitur dasar gambar dari nol, tetapi langsung dapat disesuaikan untuk mengenali karakteristik khusus gambar StyleGAN2.

Evaluasi menggunakan test set StyleGAN2, model mencapai akurasi 99.30%, sedikit turun 0.12% dari validation set. Selisih kecil ini normal dan model dapat bekerja konsisten pada data baru. Model berhasil mendeteksi 1997 dari 2000 gambar sintetis (recall 99.85%), dengan hanya melewatkan 3 gambar. Dari 2022 gambar yang diprediksi sintetis, 1997 benar dan 25 salah (precision 98.76%). Nilai AUC 99.99% artinya model sudah baik dalam membedakan kedua kelas berdasarkan skor probabilitas.

B. Evaluasi Lintas-Generator

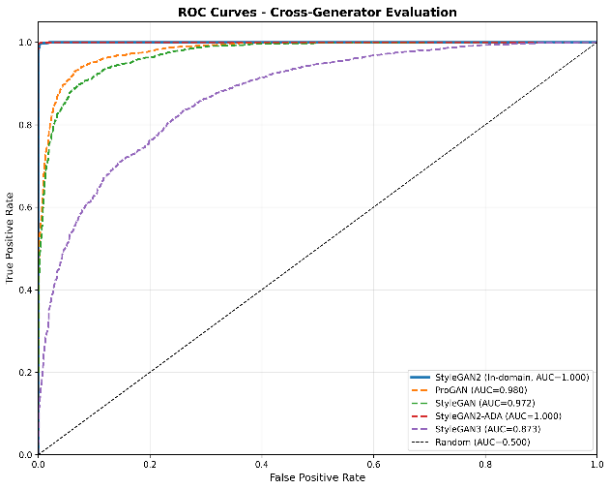
Ketika model diuji pada generator yang berbeda-beda, performanya bervariasi. Hasil evaluasi secara menyeluruh untuk kelima generator tersebut ditampilkan pada Tabel 5.

Tabel 5 performa StyleGAN2-ADA sedikit lebih tinggi dari baseline in-domain (akurasi 99.33% vs 99.30%). Fenomena ini dapat dijelaskan melalui kemiripan arsitektur StyleGAN2-ADA pada dasarnya adalah StyleGAN2 dengan modifikasi pada strategi pelatihan melalui adaptive discriminator augmentation [20], bukan perubahan fundamental pada arsitektur generator. Akibatnya, pola visual yang dihasilkan StyleGAN2-ADA mirip dengan StyleGAN2. Oleh karena itu, arsitektur generator jauh lebih menentukan karakteristik pola visual dibandingkan training strategy yang digunakan.

TABEL 5
HASIL EVALUASI LINTAS-GENERATOR

Generator	Akurasi	Presisi	Recall	F1	AUC	FPR	FNR
StyleGAN2	99.30%	98.76%	99.85%	99.30%	99.99%	1.25%	0.15%
StyleGAN2-ADA	99.33%	98.76%	99.90%	99.33%	100.00%	1.25%	0.10%
ProGAN	85.28%	98.29%	71.80%	82.98%	98.02%	1.25%	28.20%
StyleGAN	83.00%	98.18%	67.25%	79.82%	97.23%	1.25%	32.75%
StyleGAN3	63.05%	95.63%	27.35%	42.54%	87.27%	1.25%	72.65%

Penurunan performa terjadi pada ProGAN, StyleGAN dan StyleGAN3. ProGAN dengan akurasi 85.28% dan recall yang turun menjadi 71.80%, artinya 564 dari 2000 gambar sintetis (28.20%) tidak terdeteksi model. Performa StyleGAN lebih rendah lagi dengan akurasi 83.00% dan recall 67.25%, di mana 655 gambar sintetis (32.75%) lolos dari deteksi. Meski demikian, precision kedua generator masih tinggi (>98%). Artinya ketika model memprediksi gambar sintetis, prediksi tersebut cenderung benar. Namun, masalahnya terletak pada banyaknya gambar sintetis yang diprediksi sebagai asli. Pada StyleGAN3 akurasi hanya 63.05% hampir seperti tebakan acak pada dataset seimbang. Ini disebabkan oleh recall yang rendah (27.35%), di mana 1453 dari 2000 gambar sintetis (72.65%) tidak terdeteksi. Dengan kata lain 7 dari 10 gambar sintetis StyleGAN3 lolos dari deteksi model. Meski precision masih relatif tinggi (95.63%), hal ini lebih mencerminkan model yang aman, hanya memprediksi gambar sintetis ketika yakin saja, menghasilkan false positive rendah namun false negative tinggi.

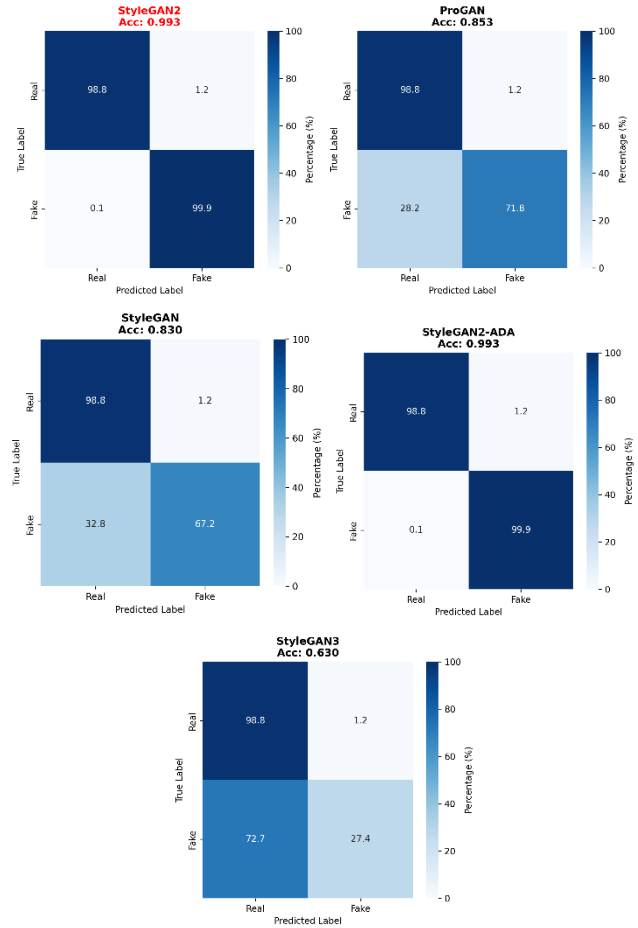


Gambar 6. Kurva ROC

Gambar 6 Kurva ROC memvisualisasikan pola dimana StyleGAN2 dan StyleGAN2-ADA terlihat kurva yang hampir sempurna dengan AUC sekitar 1.000, menanadakan separasi probabilitas yang baik antara asli dan sintetis. ProGAN dan StyleGAN masih mempertahankan kurva yang baik dengan AUC 0.980 dan 0.972, yang berarti model masih memiliki kemampuan pemisahan yang wajar meskipun threshold default 0.5 menghasilkan akurasi turun. StyleGAN3 mengalami penurunan signifikan dengan AUC 0.873. Kurva yang jauh dari model kesulitan dalam memisahkan distribusi probabilitas asli dengan sintetis pada generator StyleGAN3.

C. Analisis Confusion Matrix dan Pola Eror

Analisis confusion matrix memperlihatkan pola yang konsisten dan informatif tentang karakteristik model (Tabel 6 dan Gambar 7).



Gambar 7. Confusion Matrix semua Generator

Gambar 7 menunjukkan visualisasi confusion matrix untuk kelima generator yang diuji. Visualisasi ini memperjelas perbedaan performa model, dalam hal distribusi kesalahan prediksi. Nilai TN dan FP konsisten di semua generator (kotak kiri atas dan kanan atas), sementara nilai TP dan FN bervariasi (kotak kanan bawah dan kiri bawah), yang mengindikasikan bahwa variabilitas performa model terletak pada kemampuan

mendeteksi gambar sintetis, bukan pada kemampuan mengenali gambar asli.

TABEL 6
CONFUSION MATRIX PER-GENERATOR

Generator	TP	TN	FP	FN	Benar	Salah
StyleGAN2	1997	1975	25	3	3972 (99.30%)	28 (0.70%)
ProGAN	1436	1975	25	564	3411 (85.28%)	589 (14.72%)
StyleGAN	1345	1975	25	655	3320 (83.00%)	680 (17.00%)
StyleGAN2-ADA	1998	1975	25	2	3973 (99.33%)	27 (0.68%)
StyleGAN3	547	1975	25	1453	2522 (63.05%)	1478 (36.95%)

Pada tabel 6 pada nilai *true negative* (TN) dan *false positive* (FP) identik di semua generator TN=1975 dan FP=25. Sebelum menganalisis pola ini, penting untuk dicatat bahwa konsistensi ini bukan merupakan anomali, melainkan konsekuensi dari strategi berbagi gambar asli yang telah dijelaskan pada bagian Metodologi. Karena kelima *test set* menggunakan 2.000 gambar asli yang identik, performa model dalam mendeteksi gambar asli secara alami konsisten di semua generator. Variasi performa murni berasal dari perbedaan dalam deteksi gambar sintetis, yang tercermin dari nilai *true positive* (TP) dan *false negative* (FN) yang bervariasi.

Konsistensi dalam deteksi gambar asli artinya fitur yang dipelajari model untuk mengenali keaslian bersifat stabil dan kuat. Model tidak keliru membedakan gambar asli dengan gambar sintetis dari generator lain, dimana deteksi gambar asli bersifat independen dari konteks generator yang digunakan untuk gambar sintetis dalam *test set* yang sama. Dari perspektif praktis, ini merupakan karakteristik yang bernilai dalam skenario deployment karena model dapat untuk tidak menghasilkan *false alarm* pada gambar asli, terlepas dari jenis gambar sintetis yang sedang beredar. Sebaliknya, nilai FN dan TP bervariasi antar generator, masalah ini bukan terletak pada salah mengidentifikasi gambar asli sebagai sintetis, melainkan pada gagal mendeteksi gambar sintetis. FN meningkat dari hanya 3 pada StyleGAN2 menjadi 1.453 pada StyleGAN3, peningkatan sebesar 484 kali lipat. Model terspesialisasi pada pola visual spesifik StyleGAN2, dan semakin berbeda generator dari StyleGAN2, semakin banyak gambar sintetis yang terlewat dari deteksi.

Kecenderungan model untuk melakukan prediksi yang aman dapat dijelaskan melalui proses pelatihan. Selama pelatihan pada StyleGAN2, model belajar bahwa jika tidak ada pola visual yang familiar, maka kemungkinan besar gambar tersebut asli. Strategi ini bekerja dengan baik pada validation set StyleGAN2, namun berbalik merugikan pada generator out-of-domain yang menghasilkan pola visual berbeda. Model menghadapi pola visual yang tidak dikenali dan secara default memprediksi gambar asli untuk menghindari false positive. Pendekatan ini menghasilkan specificity yang tinggi namun sensitivity yang rendah pada out-of-domain. Dari perspektif deployment, karakteristik ini memiliki keuntungan dan kerugian, di mana false alarm rate yang rendah (1,25%) membangun kepercayaan pengguna, namun miss rate yang tinggi pada generator yang lebih baru (72,65% untuk StyleGAN3) menciptakan rasa aman yang palsu.

D. Kuantifikasi Kesenjangan Generalisasi

Untuk mengukur kemampuan generalisasi model secara kuantitatif, dihitung kesenjangan generalisasi sebagai perbedaan performa antara in-domain dan rata-rata out-of-domain. Tabel 7 menunjukkan analisis kesenjangan generalisasi untuk metrik utama.

TABEL 7
ANALISIS KESENJAGAN GENERALISASI

Metrik	In-Domain	Out-of-Domain (Rata-rata)	Gap
Accuracy	99.30%	82.66%	+16.64%
AUC	99.99%	95.63%	+4.36%
Recall	99.85%	58.33%	+41.52%
F1-Score	99.30%	76.17%	+23.13%

Pada table 7 kesenjangan generalisasi untuk akurasi mencapai 16,64% penurunan yang besar mengingat performa baseline tinggi. Gap pada AUC relatif lebih kecil (4,36%), meskipun akurasi turun, kemampuan model dalam menghasilkan skor probabilitas yang membedakan masih tergolong baik. Dengan penyesuaian threshold per-generator, dimungkinkan untuk meningkatkan akurasi pada out-of-domain.

Gap pada recall (41.52%) dan F1-score (23.13%). Penurunan recall yang masif mengkonfirmasi bahwa masalah utama adalah missed detection, bukan false alarm. Model masih mampu menghasilkan precision tinggi pada out-of-domain (rata-rata >98%), namun kehilangan hampir setengah sensitivitas-nya. Ini mencerminkan kelebihan dan kekurangan fundamental dalam model yaitu mempertahankan ketepatan dalam mendeteksi gambar sintetis (presisi tinggi) mengorbankan jangkauan pendeteksian (recall rendah).

Analisis penurunan per-generator memberikan pemahaman lebih detail tentang karakteristik masing-masing generator (Tabel 8).

TABEL 8
PENURUNAN PER-GENERATOR

Generator	Akurasi	Penurunan dengan In-Domain
StyleGAN2-ADA	99.33%	-0.03%
ProGAN	85.28%	+14.02%
StyleGAN	83.00%	+16.30%
StyleGAN3	63.05%	+36.25%

Dari Tabel 8 untuk StyleGAN2-ADA mengalami peningkatan (0.03%), karena kemiripan arsitektur dengan StyleGAN2. ProGAN dan StyleGAN mengalami penurunan wajar di rentang 14-16%, yang masih dalam rentang dapat diterima untuk banyak aplikasi praktis dengan proper calibration atau ensemble strategies [17], [22]. Namun, StyleGAN3 mengalami penurunan yang tinggi (36.25%) hal

ini bukan perburukan biasa, melainkan perubahan fundamental dalam kemampuan deteksi.

Penurunan yang tinggi pada StyleGAN3 bukan karena lebih sulit dari StyleGAN2, melainkan fundamental yang berbeda dalam cara menghasilkan gambar sintetis. Arsitektur alias-free yang digunakan StyleGAN3 menghilangkan banyak pola visual yang menjadi basis deteksi model. tekstur terkunci, pola visual fase di boundaries, dan pola aliasing yang khas dari generator sebelumnya sebagian besar tidak ada di StyleGAN3, memaksa model untuk bergantung pada petunjuk yang lebih halus, petunjuk yang tidak dipelajari secara memadai karena tidak ada dalam data pelatihan StyleGAN2.

E. Pembahasan

1) Transfer Learning dan Keterbatasan Single Generator Traning: Hasil dari *transfer learning* dari ImageNet efektif untuk deteksi gambar AI pada generator yang sama dengan data pelatihan. EfficientNetB0 dengan hanya 5.3 juta parameter mampu mencapai performa hampir sempurna (99.42% validation, 99.30% testing) pada StyleGAN2, mendemonstrasikan bahwa fitur pra-latih dari ImageNet [12] relevan untuk mendeteksi pola visual GAN. Fine-tuning memungkinkan model menyesuaikan filter-filter ini untuk lebih sensitif terhadap pola visual spesifik GAN sambil mempertahankan kemampuan mengenali struktur visual dasar [14]. Namun, kemampuan generalisasi out-of-domain, memiliki limitasi fundamental dari pendekatan single-generator training [7], [8]. Masalahnya dataset training hanya berisi satu jenis pola visual, menyebabkan model menghafal karakteristik spesifik tersebut daripada mempelajari prinsip kepalsuan yang generalizable. Fitur yang berguna untuk mendeteksi StyleGAN2 seperti pola visual peningkatan resolusi tertentu atau pola AdaIN normalization tidak selalu berguna atau bahkan muncul dalam output dari generator lain. Solusi yang jelas adalah multi-generator training [17], [22] di mana model dilatih dari berbagai generator secara bersamaan. Namun, pendekatan ini memiliki tantangan tersendiri yaitu ketidakseimbangan kelas antar generator, potensi konflik antar pola visual, dan kebutuhan dataset yang lebih besar. Miss rate 72.65% pada StyleGAN3 menunjukkan risiko nyata ketika sistem ini diterapkan secara praktis. Pada platform pendeteksi gambar ai, 7 dari 10 gambar sintetis StyleGAN3 akan lolos dari deteksi otomatis. Untuk verifikasi keaslian gambar di media jurnalistik, false negative rate setinggi ini berpotensi merusak kepercayaan publik. Perlu dipertimbangkan threshold adaptif per-generator dan mekanisme verifikasi berlapis, bukan mengandalkan keputusan biner tunggal dari satu model.

2) Kemiripan Arsitektur sebagai Penentu Generalisasi: Pada StyleGAN2-ADA memberikan pengetahuan tentang faktor yang mempengaruhi generalisasi. Meskipun dilatih dengan strategi sepenuhnya berbeda melalui adaptive discriminator augmentation [20], StyleGAN2-ADA menghasilkan performa deteksi yang identik dengan StyleGAN2 (99.33% vs 99.30%). Artinya arsitektur generator jauh lebih menentukan karakteristik pola visual dibanding cara generator dilatih. Model yang dilatih pada StyleGAN2 dapat generalize dengan mulus ke StyleGAN2-ADA karena struktur generator yang identik. Untuk membangun detector yang kuat, lebih penting untuk melatih model pada arsitektur generator yang beragam dari pada berbagai strategi training dari arsitektur yang sama [10]. Melatih pada StyleGAN2 dengan berbagai variasi pelatihan kemungkinan tidak akan secara signifikan meningkatkan generalisasis dari ProGAN, StyleGAN atau StyleGAN3 yang dibutuhkan adalah penelusuran ke generator yang arsitekturnya berbeda.

3) Evolusi GAN dan Tantangan Deteksi Masa Depan: Perkembangan akurasi deteksi menuju tren yang jelas dimana generator lama (ProGAN, StyleGAN) lebih mudah dideteksi dibanding yang lebih baru (StyleGAN3), bahkan ketika model tidak pernah melihat generator yang lebih baru selama pelatihan. ProGAN dengan akurasi 85.28% masih bisa dideteksi, StyleGAN dengan 83.00% sedikit lebih susah. Namun, kesulitan dari StyleGAN2 (99.30%) ke StyleGAN3 (63.05%) bukan kemajuan bertahap melainkan perubahan mendasar. Alias-free arsitektur dalam StyleGAN3 [3] dirancang khusus untuk menghilangkan tekstur terkunci dan pola visual pergeseran yang mengganggu versi sebelumnya. Perbedaan ini tidak hanya soal kualitas gambar, melainkan perubahan fundamental pada distribusi fitur. Gambar ProGAN dan StyleGAN memiliki pola aktivasi yang terstruktur pada layer konvolusi awal EfficientNetB0, karena progressive growing (ProGAN) dan AdaIN normalization (StyleGAN/StyleGAN2) meninggalkan jejak statistik yang khas pada distribusi piksel. StyleGAN3 berbeda. Translation equivariance yang sempurna membuat distribusi fiturnya mendekati gambar asli, sehingga AUC turun ke 0.873 berbeda dari ProGAN (0.980) dan StyleGAN (0.972) yang distribusinya masih cukup terpisah. Pendekatan berbasis spektrum frekuensi fourier dan analisis gangguan residual berpotensi lebih efektif untuk kasus ini, karena pola visual periodis dari proses filtering sinyal masih dapat terdeteksi meskipun sinyalnya jauh lebih lemah dibanding generator sebelumnya. Jika generasi baru GAN melanjutkan tren ini, metode yang bergantung semata pada pola visual spasial tidak akan memadai. Pendekatan hibrida yang menggabungkan analisis frekuensi dengan pemahaman semantik gambar seperti ketidaklogisan fisik objek atau inkonsistensi bayangan kemungkinan lebih kuat dalam menghadapi perbaikan kualitas generator [17].

IV. KESIMPULAN

Penelitian ini mengevaluasi kemampuan EfficientNetB0 dalam mendeteksi gambar buatan AI melalui pendekatan lintas-generator pada lima arsitektur GAN yang merepresentasikan evolusi teknologi 2018-2021. *Transfer learning* dari ImageNet efektif untuk deteksi in-domain dengan akurasi 99.30% pada StyleGAN2, namun mengalami kesenjangan generalisasi dengan rata-rata penurunan akurasi

16.64% dan recall 41.52% pada generator out-of-domain. Kemampuan generalisasi terbukti bergantung pada kemiripan arsitektur, bukan strategi pelatihan. StyleGAN2-ADA mencapai 99.33% karena arsitekturnya identik dengan StyleGAN2, sedangkan StyleGAN3 turun ke 63.05% dengan false negative rate meningkat dari 0.15% menjadi 72.65%, artinya arsitektur alias-free menghasilkan gambar yang secara visual berbeda fundamental sehingga sulit dideteksi. Model konsisten mengenali gambar asli di semua generator, namun kemampuan mendeteksi gambar sintetis menurun seiring perbedaan arsitektur generator dari data pelatihan, menciptakan rasa aman yang palsu dalam aplikasi praktis. Perlu dicatat bahwa seluruh dataset yang digunakan berasal dari lingkungan terkontrol, yaitu gambar FFHQ beresolusi tinggi tanpa kompresi dan gambar sintetis dari model resmi masing-masing generator. Validitas hasil pada gambar dunia nyata yang telah mengalami kompresi JPEG, cropping, atau penyebaran melalui media sosial belum dapat dipastikan. Tren evolusi GAN menunjukkan generator baru semakin sulit dideteksi. Penelitian selanjutnya perlu mengeksplorasi multi-generator training, metode deteksi berbasis analisis frekuensi dan gangguan residual, evaluasi pada dataset dunia nyata, dan perluasan ke model generatif non-GAN seperti diffusion models untuk membangun sistem deteksi yang lebih adaptif.

DAFTAR PUSTAKA

[1] I. Goodfellow *dkk.*, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, hlm. 2672–2680, 2014.

[2] T. Karras, T. Aila, S. Laine, dan J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," 26 Februari 2018, *arXiv*: arXiv:1710.10196. doi: 10.48550/arXiv.1710.10196.

[3] T. Karras *dkk.*, "Alias-Free Generative Adversarial Networks," dalam *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, hlm. 852–863. Diakses: 17 November 2025. https://proceedings.neurips.cc/paper/2021/hash/076ccd93ad68be51f23707988e934906-Abstract.html

[4] S. J. Nightingale dan H. Farid, "AI-synthesized faces are indistinguishable from real faces and more trustworthy," *Proc. Natl. Acad. Sci.*, vol. 119, no. 8, 2022, doi: 10.1073/pnas.2120481119.

[5] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, dan M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," hlm. 1–11, Okt 2019, doi: 10.1109/iccv.2019.00009.

[6] N. Jain, P. Majumdar, M. Singh, dan M. Vatsa, "Detecting deepfakes with self-blended images," *IEEE Trans. Biom. Behav. Identity Sci.*, vol. 5, no. 3, hlm. 326–337, 2023, doi: 10.1109/TBIOM.2023.3245089.

[7] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, dan A. A. Efros, "CNN-Generated Images Are Surprisingly Easy to Spot… for Now," dalam *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun 2020, hlm. 8692–8701. doi: 10.1109/CVPR42600.2020.00872.

[8] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, dan L. Verdoliva, "Are GAN generated images easy to detect? A critical analysis of the state-of-the-art," dipresentasikan pada IEEE International Conference on Multimedia and Expo, 2021, hlm. 1–6. doi: 10.1109/ICME51207.2021.9428429.

[9] L. Nataraj *dkk.*, "Detecting GAN generated fake images using co-occurrence matrices," *Electron. Imaging*, vol. 2019, no. 5, hlm. 532–1, 2019, doi: 10.2352/ISSN.2470-1173.2019.5.MWSF-532.

[10] M. Albright dan S. McCloskey, "Source generator attribution via inversion," dipresentasikan pada IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2021, hlm. 2701–2710. doi: 10.1109/CVPRW53098.2021.00305.

[11] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, dan T. Holz, "Leveraging frequency analysis for deep fake image recognition," dipresentasikan pada International Conference on Machine Learning, 2020, hlm. 3247–3258.

[12] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, dan L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," dipresentasikan pada IEEE Conference on Computer Vision and Pattern Recognition, 2009, hlm. 248–255. doi: 10.1109/CVPR.2009.5206848.

[13] K. He, X. Zhang, S. Ren, dan J. Sun, "Deep residual learning for image recognition," dipresentasikan pada IEEE Conference on Computer Vision and Pattern Recognition, 2016, hlm. 770–778. doi: 10.1109/CVPR.2016.90.

[14] J. Yosinski, J. Clune, Y. Bengio, dan H. Lipson, "How transferable are features in deep neural networks?," *Adv. Neural Inf. Process. Syst.*, vol. 27, hlm. 3320–3328, 2014.

[15] M. Tan dan Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," dalam *Proceedings of the 36th International Conference on Machine Learning*, PMLR, Mei 2019, hlm. 6105–6114. Diakses: 18 November 2025. https://proceedings.mlr.press/v97/tan19a.html

[16] K. Simonyan dan A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv Prepr. ArXiv14091556*, 2014, doi: 10.48550/arXiv.1409.1556.

[17] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, dan X. Yang, "End-to-end reconstruction-classification learning for face forgery detection," dipresentasikan pada IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, hlm. 4113–4122. doi: 10.1109/CVPR52688.2022.00409.

[18] T. Karras, S. Laine, dan T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," dipresentasikan pada IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, hlm. 4401–4410. doi: 10.1109/CVPR.2019.00453.

[19] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, dan T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," dalam *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun 2020, hlm. 8107–8116. doi: 10.1109/CVPR42600.2020.00813.

[20] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, dan T. Aila, "Training Generative Adversarial Networks with Limited Data," dalam *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, hlm. 12104–12114. https://proceedings.neurips.cc/paper/2020/hash/8d30aa96e72440759f74bd2306c1fa3d-Abstract.html

[21] C. Shorten dan T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, hlm. 1–48, 2019, doi: 10.1186/s40537-019-0197-0.

[22] F. Marra, C. Saltori, G. Boato, dan L. Verdoliva, "Incremental learning for the detection and classification of GAN-generated images," dipresentasikan pada IEEE International Workshop on Information Forensics and Security, 2019, hlm. 1–6. doi: 10.1109/WIFS47025.2019.9035107.