

Website-Based Implementation of Convolutional Neural Networks for Real Face and Deepfake Image Detection

Dimas Aditya Nugraha ^{1*)}, Syariful Alam ²⁾, Chandra Dewi Lestari ³⁾

¹⁾²⁾³⁾Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana

^{*)}Correspondence author: dimasadyangrh50@gmail.com, Purwakarta, Indonesia

DOI: <https://doi.org/10.37012/jtik.v12i1.3630>

Abstract

The development of artificial intelligence increases the risk of facial image manipulation through deepfake technology that is difficult to distinguish visually. This study aims to implement a Convolutional Neural Network (CNN) based on EfficientNetB0 to classify real and deepfake static facial images and integrate the model into a website application. The study follows the CRISP-DM framework using the FaceForencis++ extracted frames dataset from Kaggle, consisting of 60,000 images with 30,000 real images and 30,000 fake images. The data were divided into 70:15:15 ratios for training, validation, and testing, then processed through 224 x 224 pixel resizing, data augmentation, and facial area cropping using OpenCV Haar Cascade during inference. The model achieved 82.50% validation accuracy and 0.3874 validation loss at the 10th epoch. Decision threshold optimization at 0.78 produced 80.99% accuracy, 80.98% macro F1-score, and 78.56% recall for the fake class on the test data. The model was deployed in a Streamlit application, allowing users to upload facial images and receive detection results directly. The results indicate that the combination of EfficientNetB0 and threshold optimization can support preliminary verification of digital facial image authenticity.

Keywords: Deepfake, CNN, EfficientNetB0, Streamlit, Facial Image

Abstrak

Perkembangan kecerdasan buatan meningkatkan risiko manipulasi citra wajah melalui teknologi *deepfake* yang sulit dibedakan secara visual. Penelitian ini bertujuan mengimplementasikan Convolutional Neural Network (CNN) berbasis EfficientNetB0 untuk mengklasifikasikan citra wajah statis asli dan *deepfake* serta mengintegrasikannya ke aplikasi *website*. Penelitian mengikuti kerangka CRISP-DM dengan *dataset* FaceForencis++ *extracted frames* dari Kaggle sebanyak 60.000 citra, terdiri atas 30.000 citra *real* dan 30.000 citra *fake*. Data dibagi 70:15:15 untuk latih, validasi, dan uji, kemudian diproses melalui penyeragaman ukuran 224 x 224 piksel, augmentasi data, dan pemotongan area wajah menggunakan OpenCV Haar Cascade pada tahap inferensi. Model mencapai *validation accuracy* 82,50% dan *validation loss* 0,3874 pada epoch ke-10. Optimasi *decision threshold* 0,78 menghasilkan *accuracy* 80,99%, *macro F1-score* 80,98%, dan *recall* kelas *fake* 78,56% pada data uji. Model diterapkan dalam aplikasi Streamlit sehingga pengguna dapat mengunggah citra wajah dan memperoleh hasil deteksi secara langsung. Hasil penelitian menunjukkan bahwa kombinasi EfficientNetB0 dan optimasi *threshold* dapat digunakan sebagai alat bantu pemeriksaan awal keaslian citra wajah digital.

Kata Kunci: Deepfake, CNN, EfficientNetB0, Streamlit, Citra Wajah

PENDAHULUAN

Kemajuan kecerdasan buatan memberikan manfaat besar bagi pengolahan data dan citra digital, tetapi juga menghadirkan risiko baru melalui teknologi *deepfake*. *Deepfake* memanfaatkan teknik kecerdasan buatan untuk menghasilkan atau memanipulasi citra, suara, dan video sehingga objek yang ditampilkan tampak meyakinkan meskipun tidak merepresentasikan kejadian sebenarnya (Seveney et al., 2025). Pada konteks citra wajah, manipulasi tersebut dapat menimbulkan risiko terhadap keamanan data biometrik, reputasi individu, serta penyebaran disinformasi digital (Palindria et al., 2024).

Deteksi *deepfake* secara manual menjadi semakin sulit karena kualitas citra sintesis terus meningkat. Kondisi tersebut mendorong kebutuhan sistem yang mampu mengenali pola visual dan artefak manipulasi secara otomatis. Convolutional Neural Network (CNN) merupakan pendekatan deep learning yang banyak digunakan dalam klasifikasi citra karena mampu mengekstraksi fitur visual secara otomatis dari data gambar (Setiyono et al., 2023). Pada kasus identifikasi wajah asli dan *deepfake*, CNN telah digunakan untuk membedakan citra asli dan citra buatan berdasarkan pola fitur yang tidak selalu terlihat oleh pengamat manusia (Mu et al., 2024).

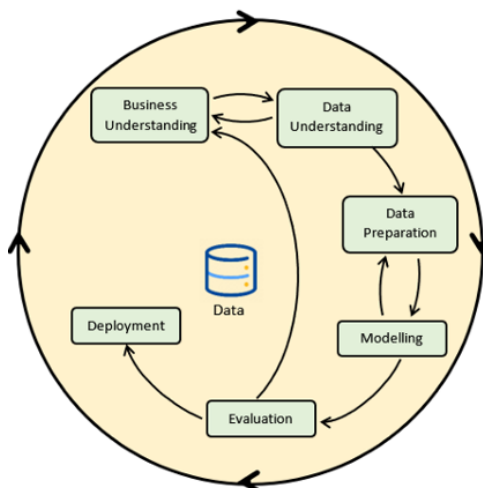
Sejumlah penelitian sebelumnya telah membahas pemanfaatan CNN untuk deteksi *deepfake*. Raharjo et al. (2025) menekankan pentingnya analisis tekstur dan pola wajah dalam forensik *deepfake* berbasis CNN. Salim et al., (2025) menggunakan MTCNN dan MobileNetV2 untuk mitigasi risiko disinformasi digital berbasis *deepfake* di Indonesia, sedangkan Gulo et al., (2025) menerapkan CNN untuk membedakan citra asli dan citra *deepfake*. Taufiqurrahman et al., (2025) juga menunjukkan bahwa pengembangan arsitektur CNN-Dilated dapat digunakan untuk mendeteksi manipulasi citra *deepfake*. Sementara itu, EfficientNetB0 dipandang relevan karena memiliki efisiensi komputasi yang baik melalui kombinasi *Mobile Inverted Bottleneck Convolution* dan *Squeeze-and-Excitation Network* sehingga mampu mengekstraksi fitur citra secara efektif (Nastitie et al., 2025). Namun, sistem deteksi tidak hanya memerlukan model klasifikasi, tetapi juga alur penggunaan yang dapat diakses oleh pengguna umum melalui aplikasi berbasis *website*.

Berdasarkan permasalahan tersebut, penelitian ini mengimplementasikan CNN berbasis EfficientNetB0 untuk mendeteksi citra wajah statis asli dan *deepfake*. Kontribusi

penelitian terletak pada penggunaan *dataset* citra seimbang, penerapan *preprocessing* dan augmentasi citra, evaluasi *threshold* keputusan pada data uji, serta integrasi model ke aplikasi Streamlit. Tujuan penelitian ini adalah membangun model klasifikasi citra wajah asli dan *deepfake*, mengevaluasi performanya menggunakan metrik akurasi, presisi, *recall*, dan F1-score, serta menerapkan model dalam *website* agar dapat digunakan sebagai alat bantu pemeriksaan awal keaslian citra digital.

METODE PENELITIAN

Penelitian menggunakan kerangka CRISP-DM karena alurnya sesuai untuk proyek berbasis data dan *machine learning*. Tahapan yang digunakan meliputi *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment* (Santiastry & Apriandari, 2024). Alur penelitian ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian Berbasis CRISP-DM

Pada tahap *business understanding*, masalah utama yang diidentifikasi adalah sulitnya membedakan citra wajah asli dan citra *deepfake* secara visual. Output tahap ini adalah kebutuhan sistem deteksi berbasis *website*. Pada tahap *data understanding*, penelitian menggunakan dataset Kaggle berjudul FaceForensics++ extracted frames, yaitu kumpulan frame hasil ekstraksi dari dataset FaceForensics++ untuk penelitian deteksi *deepfake* (7elmy, n.d.). *Dataset* yang digunakan dalam eksperimen berjumlah 60.000 citra dan dibagi seimbang ke dalam kelas *real* dan *fake*.

Tabel 1. Pembagian *Dataset* Citra Wajah

Kelas	Latih	Validasi	Uji	Total
Asli / <i>real</i>	21.000	4.500	4.500	30.000
Manipulasi / <i>fake</i>	21.000	4.500	4.500	30.000
Total	42.000	9.000	9.000	60.000

Tahap *data preparation* dilakukan dengan menyeragamkan ukuran citra menjadi 224 x 224 piksel sesuai kebutuhan masukan EfficientNetB0. Augmentasi diterapkan pada data latih untuk meningkatkan variasi visual dan membantu model mengenali pola wajah pada kondisi yang lebih beragam. Parameter augmentasi yang digunakan disajikan pada Tabel 2.

Tabel 2. Parameter Augmentasi Data

Augmentasi	Nilai	Keterangan
Random Flip	Horizontal	Membalik citra secara acak dari kiri ke kanan.
Random Rotation	0,05	Memutar citra secara acak hingga sekitar 5%.
Random Zoom	0,05	Mengubah skala pembesaran secara acak sebesar 5%.
Random Contrast	0,08	Mengubah kontras pencahayaan secara acak sebesar 8%.

Tahap *modeling* menggunakan EfficientNetB0 sebagai jaringan dasar. Lapisan klasifikasi asli diganti dengan Global Average Pooling 2D, dua lapisan *dropout* 0,5, satu lapisan *dense* 256 neuron dengan aktivasi ReLU, dan satu neuron keluaran dengan aktivasi sigmoid. Struktur tersebut digunakan untuk klasifikasi biner antara citra *real* dan *fake*.

Tabel 3. Modifikasi Lapisan Klasifikasi EfficientNetB0

Lapisan	Konfigurasi	Fungsi
Global Average Pooling 2D	Standar	Mereduksi peta fitur menjadi vektor satu dimensi.
<i>Dropout</i> 1	0,5	Mengurangi risiko <i>overfitting</i> pada representasi fitur awal.
<i>Dense</i>	256 neuron, ReLU	Mempelajari hubungan nonlinier antarfitur.
<i>Dropout</i> 2	0,5	Mengurangi ketergantungan antarneuron.
<i>Dense</i> keluaran	1 neuron, sigmoid	Menghasilkan probabilitas klasifikasi biner.

Pelatihan dilakukan dalam dua tahap. Pada tiga epoch pertama, lapisan dasar EfficientNetB0 dikunci agar lapisan klasifikasi baru dapat menyesuaikan bobot. Setelah itu, dilakukan *fine-tuning* dengan membuka lapisan ke-100 dan seterusnya untuk menyesuaikan

representasi fitur terhadap karakteristik artefak *deepfake*. Evaluasi akhir dilakukan menggunakan 9.000 data uji melalui *confusion matrix*, *accuracy*, *precision*, *recall*, *macro F1-score*, serta perbandingan *threshold* 0,50 dan 0,78. Model akhir kemudian diintegrasikan ke aplikasi *website* menggunakan Streamlit, sedangkan OpenCV Haar Cascade digunakan pada tahap inferensi untuk mendeteksi dan memotong area wajah sebelum citra diklasifikasikan.

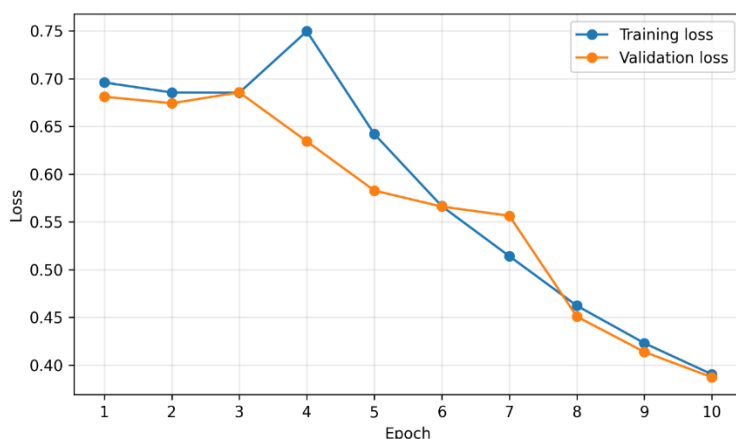
HASIL DAN PEMBAHASAN

Performa pelatihan model selama sepuluh epoch disajikan pada Tabel 4. Fase *warm-up* menunjukkan peningkatan yang terbatas karena lapisan dasar masih dikunci. Setelah masuk ke fase *fine-tuning*, *training accuracy* dan *validation accuracy* meningkat secara bertahap hingga mencapai 82,31% dan 82,50% pada epoch ke-10.

Tabel 4. Hasil Pelatihan Model EfficientNetB0

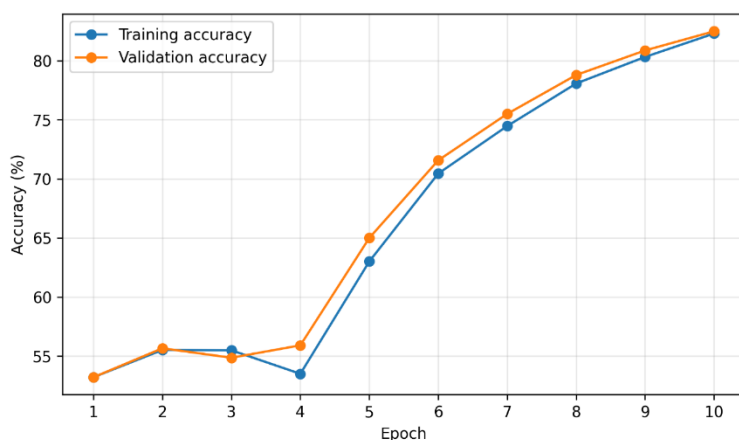
Epoch	Fase	Train Loss	Train Acc.	Val. Loss	Val. Acc.
1	Warm-up 1	0,6961	53,22%	0,6812	53,22%
2	Warm-up 2	0,6856	55,53%	0,6743	55,67%
3	Warm-up 3	0,6854	55,50%	0,6857	54,87%
4	Fine-Tune 1	0,7497	53,51%	0,6344	55,92%
5	Fine-Tune 2	0,6421	63,02%	0,5829	65,01%
6	Fine-Tune 3	0,5665	70,46%	0,5661	71,58%
7	Fine-Tune 4	0,5142	74,48%	0,5565	75,51%
8	Fine-Tune 5	0,4623	78,08%	0,4509	78,80%
9	Fine-Tune 6	0,4231	80,33%	0,4139	80,88%
10	Fine-Tune 7	0,3905	82,31%	0,3874	82,50%

Tren nilai *loss* ditampilkan pada Gambar 2 dibawah. *Validation loss* menurun dari 0,6812 menjadi 0,3874. Penurunan ini menunjukkan bahwa proses *fine-tuning* membantu model menyesuaikan fitur terhadap karakteristik data validasi. Walaupun terdapat fluktuasi pada transisi dari *warm-up* ke *fine-tuning*, arah umum kurva menunjukkan perbaikan performa.



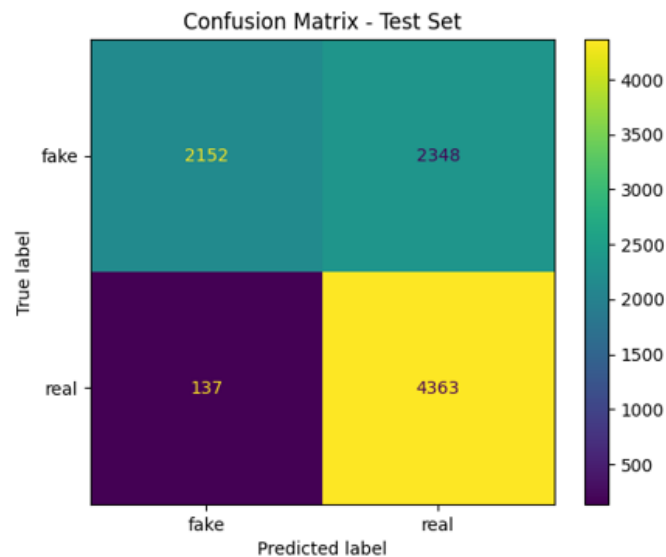
Gambar 2. Kurva Training Loss dan Validation Loss

Gambar 3 dibawah menunjukkan peningkatan *training accuracy* dan *validation accuracy*. Selisih akhir antara akurasi latih dan validasi hanya 0,19 poin persentase, sehingga tidak tampak kesenjangan generalisasi yang besar pada data validasi. Meskipun demikian, evaluasi pada data uji tetap diperlukan karena data berasal dari *frame dataset* yang karakteristiknya dapat memiliki kemiripan visual.



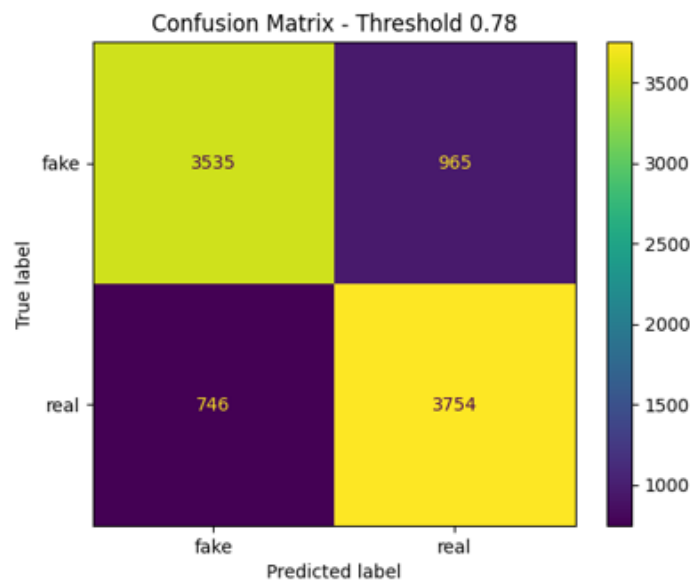
Gambar 3. Kurva Training Accuracy dan Validation Accuracy

Pengujian *threshold* dilakukan untuk melihat pengaruh ambang keputusan terhadap kemampuan model mengenali kelas *fake*. Pada *threshold* standar 0,50, model memperoleh akurasi 72,39% dan *macro F1-score* 70,62%. *Confusion matrix* pada Gambar 4 dibawah menunjukkan bahwa 2.152 citra *fake* berhasil dikenali sebagai *fake*, sedangkan 2.348 citra *fake* salah diklasifikasikan sebagai *real*.



Gambar 4. *Confucion Matrix* pada *Threshold* 0,50

Pada *threshold* 0,78, kemampuan model mendeteksi citra *fake* meningkat. Gambar 5 dibawah menunjukkan bahwa jumlah citra *fake* yang terdeteksi benar naik menjadi 3.535, sedangkan kesalahan *fake* sebagai *real* turun menjadi 965. Konsekuensinya, jumlah citra *real* yang salah diklasifikasikan sebagai *fake* meningkat menjadi 746. Perubahan ini menunjukkan adanya pertukaran antara sensitivitas terhadap citra manipulasi dan risiko alarm palsu pada citra asli.



Gambar 5. *Confucion Matrix* pada *Threshold* 0,78

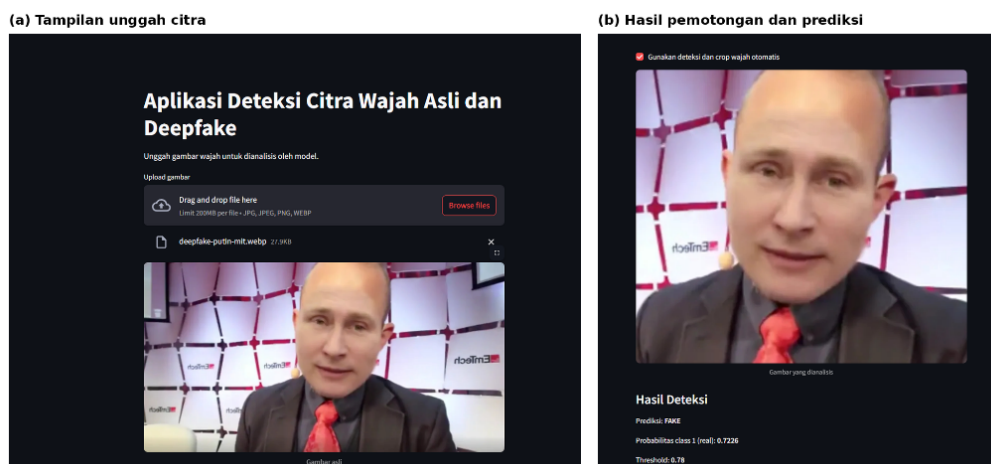
Tabel 5. Perbandingan Metrik Berdasarkan *Threshold*

<i>Threshold</i>	Akurasi	Presisi Makro	Recall Makro	Macro F1	Recall Fake
0,50	72,39%	79,51%	72,39%	70,62%	47,82%
0,78	80,99%	81,06%	80,99%	80,98%	78,56%

Berdasarkan Tabel 5, *threshold* 0,78 menghasilkan peningkatan akurasi sebesar 8,60 poin persentase dan *macro F1-score* sebesar 10,36 poin persentase dibandingkan *threshold* 0,50. *Recall* kelas *fake* juga meningkat dari 47,82% menjadi 78,56%. Untuk konteks alat bantu pemeriksaan awal, *threshold* 0,78 lebih sesuai karena menurunkan kemungkinan citra manipulasi lolos sebagai citra asli. Namun, pemilihan *threshold* tetap perlu mempertimbangkan risiko penggunaan karena peningkatan sensitivitas dapat memunculkan lebih banyak alarm palsu pada citra *real*.

Hasil tersebut menunjukkan bahwa performa model berada pada kisaran yang masih sejalan dengan penelitian *deepfake* berbasis CNN lainnya. Hutagalung et al., (2025) melaporkan penggunaan MobileViT-CBAM untuk deteksi *deepfake* secara *real-time* pada perangkat *mobile*, sedangkan Suharyanto et al., (2024) membandingkan CNN dengan *Capsule Networks* dalam mendeteksi *deepfake*. Tobing et al., (2025) juga menunjukkan bahwa pendekatan CNN dapat digunakan untuk mendukung keamanan digital melalui deteksi citra palsu. Perbedaan utama penelitian ini terletak pada fokus citra wajah statis, penggunaan EfficientNetB0, optimasi *threshold* keputusan, serta integrasi model ke aplikasi *website* berbasis Streamlit.

Model akhir diintegrasikan ke aplikasi *website* menggunakan Streamlit. Streamlit dipilih karena mendukung pembangunan antarmuka interaktif berbasis Python untuk kebutuhan *machine learning* dan visualisasi data (Febrian et al., 2025). Alur aplikasi dimulai dari unggah citra, pendeteksian area wajah menggunakan Haar Cascade, *square crop*, penyeragaman ukuran 224 x 224 piksel, prediksi model, dan penampilan label hasil deteksi.



Gambar 6. Antarmuka Unggah Citra dan Hasil Deteksi pada Streamlit

Gambar 6 memperlihatkan antarmuka unggah citra dan hasil deteksi. Pemotongan area wajah membantu mengurangi pengaruh latar belakang sehingga masukan model lebih berfokus pada wilayah wajah. Walaupun demikian, penggunaan Haar Cascade masih memiliki keterbatasan pada wajah yang menoleh, tertutup, atau berada dalam pencahayaan rendah. Selain itu, model dilatih menggunakan *frame* dari FaceForensics++, sehingga performanya pada citra dari generator baru, kompresi ekstrem, atau demografi yang tidak terwakili belum dapat dipastikan. Oleh karena itu, aplikasi lebih tepat diposisikan sebagai alat bantu pemeriksaan awal, bukan bukti forensik tunggal.

KESIMPULAN DAN REKOMENDASI

Penelitian ini berhasil mengimplementasikan CNN berbasis EfficientNetB0 untuk klasifikasi citra wajah statis asli dan *deepfake* serta mengintegrasikannya ke aplikasi *website*. Model mencapai *validation accuracy* 82,50% dan *validation loss* 0,3874 pada epoch ke-10. Pada 9.000 data uji, *threshold* 0,78 memberikan performa lebih seimbang dibandingkan *threshold* 0,50, dengan *accuracy* 80,99%, *macro F1-score* 80,98%, dan *recall fake* 78,56%. Integrasi OpenCV Haar Cascade dan Streamlit menghasilkan alur penggunaan yang praktis mulai dari unggah citra, pemotongan wajah, hingga penyajian prediksi. Penelitian selanjutnya disarankan menggunakan pemisahan data berdasarkan identitas atau video asal, memperluas *dataset* dengan generator dan demografi yang lebih beragam, membandingkan

metode pelokalan wajah seperti MTCNN atau MediaPipe, serta mengembangkan deteksi video yang mempertimbangkan informasi temporal.

REFERENSI

- Febrian, R. D., Heri Irawan, R., & Helilintar, R. (2025). Implementasi Algoritma K-Nearest Neighbor Penentuan Takaran Pakan Ikan. *Jurnal Nusantara Of Engineering*, 8(2). <https://ojs.unpkediri.ac.id/index.php/noe>
- Gulo, S. A., Pertiwi, A. A., Putri, S., Nasution, S., & Syahputra, H. (2025). Deteksi Deepfake Dalam Citra Menggunakan Convolutional Neural Network (CNN). *Jurnal Mahasiswa Teknik Informatika*, 9(5).
- Hutagalung, C. A., Akhmad, D. M., Resita, E., & Rahmi, T. (2025). Deteksi Deepfake Real-Time pada Perangkat Mobile Menggunakan Arsitektur MobileViT-CBAM Teroptimasi. *RIGGS: Journal of Artificial Intelligence and Digital Business*, 4(4), 1543–1550. <https://doi.org/10.31004/riggs.v4i4.3634>
- Mu, J., Adrezo, M., & Haikal, A. N. (2024). Identifikasi Wajah Asli dan Buatan Deepfake Menggunakan Metode Convolutional Neural Network. *Teknika*, 13(1), 45–50. <https://doi.org/10.34148/teknika.v13i1.705>
- Nastitie, C., Handayani, I., & Yogyakarta, U. T. (2025). Perbandingan Metode Mobilenetv2 Dan Efficientnetb0 Untuk Klasifikasi Jenis Pisang Comparison Of Mobilenetv2 And Efficientnetb0 Methods For Banana Type Classification. *Journal of Information Technology and Computer Science (INTECOMS)*, 8(6).
- Palindria, E. A., Thufail, S. M., & Febrian, R. M. (2024). Ancaman Deepfake Buatan AI Dan Implikasinya Terhadap Keamanan Data Biometrik Di Indonesia. *Jurnal Spectrum Hukum*.
- Raharjo, T., Irfan Adristi, F., Yusroni Romadhona, F., Rahmadi Syahputra, R., Yusuf Halim, M., Ashshidiqie Rachman, M., Nur Marjianto, R., Santicho, D., Kusuma, P., & Ramadhani, E. (2025). Analisis Forensik Deepfake Berbasis Convolutional Neural Network (Cnn) Untuk Deteksi Inkonsistensi Tekstur Dan Pola Pada Citra Wajah. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 9, Number 2).

- Salim, H. A., Muliawan, A., & Fauziah, D. A. (2025). Deteksi Deepfake dalam Mitigasi Risiko Disinformasi Digital menggunakan Metode Multi-Task Cascaded Convolutional Network dan Mobilenetv2 di Indonesia. *SEMANTIK*, 11. <https://doi.org/https://doi.org/10.55679/semantik.v11i2.239>
- Santiastry, S., & Apriandari, W. (2024). Penerapan Algoritma Naive Bayes Dan Metode Crisp-Dm Dalam Prediksi Hasil Tes Kemampuan Bahasa Inggris Mahasiswa. *Jurnal Mahasiswa Teknik Informatika*, 8(5).
- Setiyono, B., Riv'an Arif, M., Aini, Q. Q., Soegianto, T. H., Ohanna, J., Andrean, R., Gunawan, F., & Rizkia, A. P. (2023). Identifikasi Tanaman Obat Indonesia Melalui Citra Daun Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*. <https://doi.org/10.25126/jtiik.2023106809>
- Suharyanto, D., Lubis, C., & Agus, D. B. (2024). Penerapan Convolutional Neural Network dan Capsule Networks Dalam Mendeteksi Deepfake. *Jurnal Ilmu Komputer Dan Sistem Informasi*.
- Seveney, M. C., Wicaksono, D. B., & Soetijono, I. K. (2025). Urgensi Regulasi Terhadap Penyalahgunaan Deepfake Berbasis Ai (Artificial Intelligence) Pada Konten Pornografi. *Jurnal Disiplin: Majalah Civitas Akademika Sekolah Tinggi Ilmu Hukum Sumpah Pemuda*, 31.
- Taufiqurrahman, R. F., Anggraeny, F. T., & Sihananto, A. N. (2025). Penerapan Arsitektur CNN-Dilated untuk Deteksi Manipulasi Citra Deepfake. *ILKOMNIKA: Journal of Computer Science and Applied Informatics E*, 7(3), 495–506. <https://doi.org/10.28926/ilkomnika.v7i3.821>
- Tobing, F. A. T., Kusnadi, A., Pane, I. Z., & Winantyo, R. (2025). Deepfake detection using convolutional neural networks: a deep learning approach for digital security. *Indonesian Journal of Electrical Engineering and Computer Science*, 39(2), 1092. <https://doi.org/10.11591/ijeecs.v39.i2.pp1092-1099>
- 7elmy, A. (n.d.). *FaceForencis++ extracted frames*. Kaggle. <https://www.kaggle.com/datasets/adham7elmy/faceforencispp-extracted-frames>