

Implementasi Arsitektur MTCNN pada Kelas Dimensi Piksel Berbeda dan *Plotting* Multi-Wajah pada Hasil Deteksi

Implementation of MTCNN Architecture on Different Pixel Dimension Classes and Plotting Multi-Face on Detection Results

Merry Anggraeni¹⁾, Hillman Akhyar Damanik²⁾

Fakultas Teknologi Informasi Universitas Budi Luhur

Jl. Ciledug Raya, Petukangan Utara, Jakarta Selatan, DKI Jakarta, Indonesia 1226012
merry.anggraeni@budiluhur.ac.id

Diterima: 23 September 2024 || Direvisi: 30 November 2024 || Disetujui: 15 Maret 2025

Abstrak – Deteksi wajah merupakan tugas *computer vision* untuk mengidentifikasi dan memverifikasi seseorang berdasarkan foto wajah mereka. Deteksi dan penyelarasan wajah di lingkungan yang tidak dibatasi sangat menantang karena berbagai pose, iluminasi, dan oklusi. Wajah manusia sulit untuk dimodelkan karena ada banyak variabel yang dapat berubah, misalnya ekspresi wajah, orientasi, kondisi pencahayaan, dan oklusi parsial, seperti kacamata hitam, syal, topeng, dan lainnya.. Studi terbaru menunjukkan bahwa pendekatan *deep learning* (pembelajaran yang mendalam) dapat mencapai kinerja yang mengesankan pada dua tugas ini. Pada penelitian ini akan dilakukan pendeteksian wajah pada multi-wajah sekaligus memetakan satu persatu hasil deteksi wajah yang didapat (*face crop*) untuk kebutuhan berbagai sistem yang berkaitan dengan pendeteksian wajah dengan menggunakan pendekatan Multi-Task Cascaded Convolutional Neural Network (MTCNN). Penelitian ini bertujuan untuk mengimplementasikan arsitektur MTCNN menggunakan TensorFlow dan OpenCV, dengan dua manfaat utama. Pertama, penelitian ini diharapkan dapat menyediakan model pra-pelatihan yang berkinerja optimal serta memperkuat bukti dari penelitian-penelitian sebelumnya yang telah meneliti model ini. Kedua, model ini dapat digunakan sebagai *input* bagi sistem lain. Variabel *input* berupa gambar foto wajah yang berisi satu atau lebih untuk diproses. Gambar foto wajah ini akan memiliki berbagai dimensi piksel untuk mewakili resolusi yang berbeda. Variabel *output* yang dihasilkan berupa koordinat lokasi wajah terdeteksi ataupun berupa landmark titik-titik kunci wajah, seperti posisi mata, sudut hidung, dan mulut. Hasil penelitian menunjukkan skor rata-rata pada berbagai dimensi piksel dalam *dataset*, dengan akurasi sebesar 93%, presisi 95%, *recall* 96%, *F1-score* 95%, dan ROC-AUC 90,89%.

Kata Kunci: *Convolutional Neural Networks*, deteksi wajah, MTCNN, *deep learning*, *face crop*

Abstract – Face detection is a computer vision task to identify and verify a person based on a photo of their face. Face detection and alignment in unconstrained environments are very challenging due to various poses, illumination, and occlusions. The human face is difficult to model because there are many variables that can change, such as facial expression, orientation, lighting conditions, and partial occlusions, such as sunglasses, scarves, masks, and others. Recent studies have shown that deep learning approaches can achieve impressive performance on these two tasks. In this paper, face detection on multi-faces will be carried out as well as mapping one by one the results of the face detection obtained (*face crop*) for the needs of various systems related to face detection using the Multi-Task Cascaded Convolutional Neural Network (MTCNN) approach. This study aims to implement the MTCNN architecture using TensorFlow and OpenCV, with two main benefits. First, this study is expected to provide a pre-training model that performs optimally and strengthens evidence from previous studies that have examined this model. Second, this model can be used as input for other systems. The input variable is a photo image of a face containing one or more to be processed. This photo image will have various pixel dimensions to represent different resolutions. The output variable produced is in the form of coordinates of the detected face location or in the form of landmarks of key facial points, such as the position of the eyes, the corner of the nose, and the mouth. The results of the study showed an average score on various pixel dimensions in the dataset, with an accuracy of 93%, a precision of 95%, a recall of 96%, an *F1-score* of 95%, and an ROC-AUC of 90.89%.

Keywords: *Convolutional Neural Networks*, face detection, MTCNN, deep learning, face crop

PENDAHULUAN

Algoritma Viola-Jones, yang dikembangkan oleh Paul Viola dan Michael Jones pada tahun 2001, merupakan metode deteksi wajah berbasis fitur yang populer di masanya. Algoritma ini menggunakan fitur sederhana dalam rangkaian *cascade* yang memungkinkan deteksi wajah secara *real-time*, meskipun lambat dalam proses pelatihan. Namun, seiring perkembangan teknologi, pendekatan Viola-Jones mulai dianggap ketinggalan zaman karena performanya yang terbatas dalam lingkungan yang lebih kompleks, seperti kondisi pencahayaan atau pose yang bervariasi, dan tingkat akurasi yang menurun dalam mendeteksi wajah yang mengalami oklusi (misalnya tertutup kacamata atau masker). (Viola & Jones, 2001). Deteksi wajah merupakan subjek yang sangat populer dengan berbagai macam aplikasi (Kumari Sirivarshitha et al., 2023) (Xie et al., 2019) (Salih et al., 2022) (Hofer et al., 2021) (Raman, 2023). *Smartphone* dan laptop modern hadir dengan perangkat lunak pendeteksi wajah bawaan, yang dapat mengotentikasi identitas pengguna. Ada banyak aplikasi yang dapat menangkap, mendeteksi, dan memproses wajah secara *real time*, dapat mengidentifikasi usia dan jenis kelamin pengguna, dan juga dapat menerapkan beberapa filter yang sangat keren. Daftar ini tidak terbatas pada aplikasi seluler ini karena deteksi wajah juga memiliki berbagai aplikasi dalam pengawasan, keamanan, dan biometrik. Tetapi asal mula kisah suksesnya berawal dari tahun 2001, ketika Viola dan Jones mengusulkan Kerangka Deteksi Objek pertama untuk Deteksi Wajah Waktu Nyata dalam Rekaman Video (Chen et al., 2019) (Xiang & Zhu, 2017) (Zhao et al., 2020a). Wajah manusia sulit untuk dimodelkan karena ada banyak variabel yang dapat berubah, misalnya ekspresi wajah, orientasi, kondisi pencahayaan dan oklusi parsial seperti kacamata hitam, syal dan topeng (Zhang et al., 2016).

Penelitian menggunakan MTCNN digunakan untuk mendeteksi dan memotong wajah yang terdeteksi masker untuk mengekstrak fitur wajah yang terlihat, sistem ini dapat mengenali individu dengan akurasi tinggi. Data yang digunakan mencakup berbagai kondisi, seperti pencahayaan, sudut wajah, jenis masker, dan keberadaan kacamata (Galindo et al., 2022). Penelitian mendeteksi kelelahan pengemudi untuk mencegah kecelakaan lalu lintas dengan memonitor frekuensi kedipan mata menggunakan Multi-Task Cascaded Convolutional Networks

(MTCNN) dan OpenCV diterapkan dengan tiga lapisan jaringan MTCNN mendeteksi wajah, kemudian menghitung rasio kedipan mata untuk mengidentifikasi tingkat kelelahan pengemudi (Suhas et al., 2023).

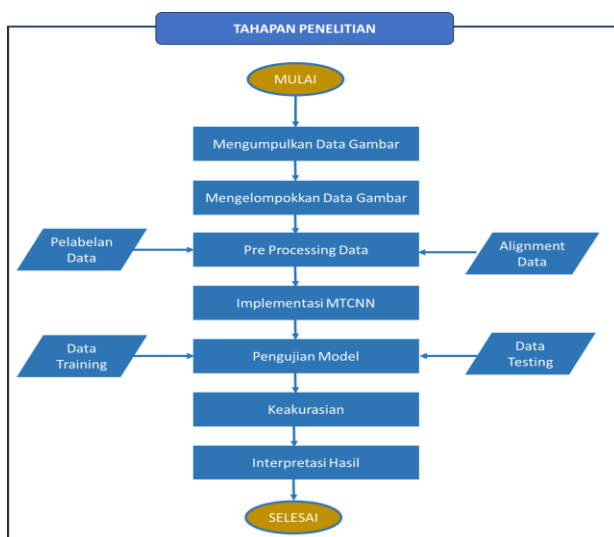
Deteksi wajah dilakukan dengan teknik pengenalan wajah, dengan tujuan melokalisasi dan mengekstrak wilayah wajah dari latar belakang (Zhao et al., 2020b) (Shih et al., 2023). Teknik ini juga memiliki beberapa aplikasi di berbagai bidang seperti pengambilan gambar berbasis konten, pengkodean video, konferensi video, pengawasan kerumunan, dan antarmuka manusia-komputer yang cerdas (Suhas et al., 2023) (Ma & Wang, 2018). Deteksi Wajah dapat digunakan dalam berbagai skenario, misalnya untuk mendeteksi apakah seseorang hadir di suatu ruangan atau tidak. Penulis mengusulkan struktur *cascade convolutional neural network* (CNN) untuk menyelidiki deteksi wajah pada citra dunia nyata yang memiliki variasi visual besar, seperti pose, ekspresi, dan pencahayaan. Implementasi, deteksi, dan pengambilan wajah akan diperoleh dengan bantuan teknologi pencocokan visual langsung. Eksperimen mendeteksi wajah dengan rotasi keluar bidang ± 90 derajat (Farfade et al., 2015) (Ravidas & Ansari, 2018) (Xu et al., 2021). Di pada penelitiannya memperkenalkan algoritma MTCNN dan menerapkannya pada penilaian kesamaan dengan dua masalah nyata dengan metode praktis yang efektif untuk meningkatkan presisi klasifikasi yang diusulkan untuk mendapatkan efek yang lebih baik wajah (Du, 2020). Metode CNN sebagai salah satu model dari *deep learning* yang memiliki kemampuan yang baik dalam klasifikasi data dengan struktur dua dimensi seperti citra untuk klasifikasi pembalap MotoGP, serta menggunakan performa GPU untuk mempercepat waktu pelatihan. Berdasarkan hasil pengujian menggunakan epoch sejumlah 150 terhadap citra pembalap MotoGP, menunjukkan akurasi sebesar 96,67% terhadap data test dengan waktu pelatihan mencapai 473 detik, sedangkan penggunaan GPU dapat mempercepat proses komputasi hingga 11 kali lipat dibanding jika menggunakan CPU (Thohari & Hertantyo, 2018). Metode CNN disajikan untuk mendeteksi wajah simultan, pelokalan *landmark*, estimasi pose, dan pengenalan *gender* menggunakan CNN. Metode yang diusulkan disebut, HyperFace, menggabungkan lapisan menengah dari CNN yang dalam menggunakan CNN terpisah diikuti oleh algoritma pembelajaran multi-tugas yang beroperasi pada fitur yang menyatu. Teknik mengeksplorasi sinergi di antara tugas-tugas yang meningkatkan kinerja individu mereka. Eksperimen ekstensif menunjukkan bahwa model yang diusulkan mampu menangkap informasi global dan lokal di wajah dan berkinerja jauh lebih baik (Ranjan et al., 2019). Penelitian menggunakan *compact CNN* menyajikan

solusi baru dari masalah deteksi wajah frontal berdasarkan *compact convolutional neural network*.

Pada penelitian selanjutnya penulis dengan teknik MTCNN akan menerapkan sistem deteksi wajah dengan menggunakan pendekatan berbasis gambar. Dengan mengidentifikasi wajah manusia untuk dimodelkan dapat dideteksi dengan akurasi yang baik dan diplot sehingga dapat dipergunakan sebagai masukan untuk berbagai aplikasi yang memerlukannya. Pada penelitian ini akan dilakukan pendeteksian wajah pada multi-wajah sekaligus memetakan satu persatu hasil deteksi wajah yang didapat (*face crop*) untuk kebutuhan berbagai sistem yang berkaitan dengan pendeteksian wajah dengan menggunakan pendekatan MTCNN. Tujuan dari penelitian ini adalah untuk menyediakan implementasi arsitektur MTCNN menggunakan TensorFlow dan OpenCV, yang diharapkan memberikan dua manfaat utama. Pertama, penelitian ini bertujuan untuk menghasilkan model pra-pelatihan dengan kinerja optimal, yang dapat berkontribusi dalam memperkuat pembuktian terhadap penelitian-penelitian sebelumnya yang telah mengeksplorasi model MTCNN. Kedua, model yang dihasilkan dari penelitian ini dapat dimanfaatkan sebagai *input* bagi sistem lain, sehingga meningkatkan fleksibilitas dan kegunaannya dalam berbagai aplikasi berbasis deteksi wajah.

METODOLOGI PENELITIAN

Tahapan kegiatan metodologi pada penelitian ini dapat dilihat pada Gambar 1. Pada tahap 1 dilakukan identifikasi masalah, studi pustaka, dan perumusan masalah. Setelah itu, dilakukan pengumpulan data, *preprocessing* data, pelatihan dan representasi hasil, pengujian model, dan analisis dan pembahasan. Berikut adalah penjelasan setiap tahapan.



Gambar 1 Tahapan Penelitian

Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah berupa citra yang mengandung wajah manusia dalam suatu grup, perkumpulan atau banyak orang untuk dideteksi wajahnya ke dalam set pelatihan. inilah yang akan membantu dalam meningkatkan prediksi dari deteksi wajah, sehingga metode yang diusulkan dan diimplementasikan dapat mencocokkan bentuk wajah dengan set pelatihan yang dibuat. Semakin baik data yang dikumpulkan, semakin baik akurasi lokasi titik-titik wajah di setiap citra foto yang dapat terdeteksi, dengan langkah sebagai berikut:

- 1) Dataset yang dikumpulkan pada penelitian ini ditulis dalam format JPG, berisi citra yang berjumlah 104 data citra wajah dengan berbagai jumlah dan posisi lokasi wajah. Variabel *input* berupa gambar foto wajah yang berisi satu atau lebih untuk diproses. Gambar foto wajah ini akan memiliki berbagai dimensi piksel untuk mewakili resolusi yang berbeda. Variabel *input* citra yang dilabelkan berdasarkan dimensi *shape* (*Width : Height*).
- 2) Dataset diklasifikasikan berdasarkan dimensi citra, menjadi 3 klasifikasi yaitu, citra dengan dimensi di bawah 1,000 x 1,000 piksel, citra dengan dimensi antara 1,000 x 1,000 sampai 2,000 x 2,000 piksel, citra dengan dimensi di atas 2,000 x 2,000 piksel dan didapati total 104 citra pada seluruh kelasnya.

Visualisasi Data

Memvisualisasikan data artinya memahami jenis data apa yang dimiliki dalam penelitian ini. Maka karena penelitian ini jenis data yang digunakan adalah gambar, diperlukan pengetahuan tentang cara cara mengekstrak wajah dari keseluruhan gambar.

Preprocessing

Pre-processing adalah proses awal dilakukannya perbaikan suatu citra untuk menyesuaikan *dataset* sebagai *input* yang berkaitan dengan persiapan dan transformasi dataset, karena detektor wajah mengidentifikasi wajah dalam area berbentuk persegi, sehingga wajah yang terdeteksi seringkali mengandung beberapa *noise*, seperti warna latar belakang (Saraswat & Raj, 2022) (Sukumar et al., 2016). Tahap ini merupakan pemrosesan awal agar dataset siap

digunakan pada tahap selanjutnya. *Preprocessing* gambar dilakukan sebelum memberi *input* pada jaringan (*net*). *Preprocessing* dilakukan dengan menyiapkan beberapa parameter:

- 1) *min_face_size* menunjukkan gambar *input* yang diubah ukurannya minimum yang dimasukkan ke P-Net. Semakin rendah nilai ini, semakin akurat modelnya, tetapi mengorbankan kecepatan (Fortuna & Khaeruzzaman, 2022).
- 2) *Threshold* (ambang) mewakili ambang batas skor deteksi wajah yang meningkat sebagai kaskade jaringan.
Kemudian Argumen: *img*: array bentuk numpy mengambang [h, w, c].
Mengembalikan: array numpy berbentuk [1, c, h, w].

Teknik Analisis

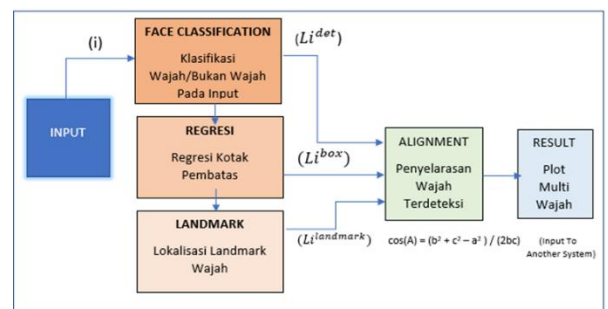
Pada penelitian ini, sebelum melakukan *sub-plot* terhadap masing-masing wajah dalam gambar multi-wajah yang terdeteksi, hasil deteksi wajah menggunakan MTCNN dianalisis berdasarkan nilai skor yang diperoleh. Analisis ini mencakup pengukuran keakuratan model serta identifikasi hambatan seperti oklusi yang dapat memengaruhi hasil deteksi. Selain itu, dilakukan evaluasi terhadap pengaruh dimensi gambar dan jumlah wajah dalam frame citra yang telah diklasifikasikan pada *dataset*. Langkah ini dilakukan untuk memastikan hasil deteksi wajah yang optimal sebelum digunakan sebagai *input* bagi sistem.

Implementasi Model Deteksi Wajah

Implementasi CNN (*Convolutional Neural Network*) yang digunakan dalam penelitian ini adalah metode MTCNN, yang berfungsi sebagai kerangka kerja untuk mengintegrasikan dua tugas utama, yaitu deteksi wajah dan penyalarsan wajah, menggunakan pendekatan pembelajaran multi-tugas. Pada tahap pertama, digunakan CNN yang sederhana untuk secara cepat menghasilkan *region proposal* atau wilayah kandidat wajah. Tahap kedua menyempurnakan wilayah kandidat tersebut menggunakan CNN yang lebih kompleks. Pada tahap ketiga, MTCNN menerapkan jaringan yang lebih mendalam untuk memperbaiki hasil deteksi dan menyalarskan landmark wajah dengan tingkat presisi yang lebih tinggi. MTCNN dipilih dalam penelitian ini karena keunggulannya dalam menangani tugas tersebut secara bersamaan dengan efisien, menggunakan pendekatan pembelajaran multi-tugas yang memungkinkan hasil deteksi dan penyalarsan wajah yang lebih akurat.

Setiap tahap jaringan MTCNN diimplementasikan untuk mengintegrasikan tugas deteksi dan penyalarsan wajah secara bersamaan. Gambar 2 merupakan *flow* implementasi model deteksi wajah menggunakan metode (MTCNN).

1. *Input* – merupakan gambar wajah (atau sekumpulan gambar) yang akan diproses.
2. *Face Classification* - Pada tahap pertama ini, model akan melakukan klasifikasi wajah pada gambar *input* untuk menentukan apakah terdapat wajah yang terdeteksi di dalam gambar.
3. *Regression* (Regresi Kotak Pembatas) – Tahap kedua ini akan digunakan menentukan lokasi wajah dalam gambar. Pada tahap ini, model akan menghasilkan kotak pembatas (*bounding box*) untuk menunjukkan posisi wajah dalam gambar, berdasarkan hasil dari klasifikasi wajah.
4. *Landmark* (Lokalisasi Landmark Wajah) – pada tahap ketiga ini dilakukan lokalisasi *landmark* wajah dilakukan untuk menentukan titik-titik (*landmark*) (posisi mata, hidung, mulut, dan sudut wajah lainnya), bertujuan untuk mendapatkan informasi lebih rinci mengenai posisi fitur wajah.
5. *Alignment* (Penyalarsan Wajah Terdeteksi) - Tahap keempat proses penyalarsan wajah, akan dilakukan untuk memastikan bahwa wajah terdeteksi dengan posisi yang konsisten.
6. *Result* – tahap akhir merupakan wajah yang terdeteksi dengan *landmark* yang akurat.



Gambar 2 Blok Diagram Kerja Sistem

Penilaian wajah (*face judgement*): Klasifikasi biner yang menggunakan fungsi lintas entropi (*cross-entropy loss*):

$$Li^{det} = -(y_i^{det} \log(pi) + (1 - y_i^{det})(1 - \log(pi))) \quad (1)$$

pi adalah probabilitas bahwa sampel wajah *xi* diprediksi oleh MTCNN adalah wajah yang sebenarnya. *yi^{det}* adalah singkatan dari *ground-truth*, dimana *yi^{det}* ∈ {0,1}.

Regresi kotak pembatas (Bounding-Box Regression): Untuk setiap jendela kandidat, *offset* antara kandidat dan ground-truth terdekat dihitung. *Square loss function* digunakan untuk tugas ini

$$L_i^{box} = ||y_i^{box} - y_i^{gt}||_2^2 \quad (2)$$

y_i^{box} adalah target regresi dari jaringan. y_i^{gt} adalah *ground-truth* empat dimensi koordinat, termasuk koordinat kiri atas, tinggi dan lebarnya. Properti dari *bounding-box* berisi banyak jenis informasi berlabel yang relevan, seperti blur, ekspresi, iluminasi, tidak valid, oklusi, pose.

Lokalisasi Landmark Wajah (Feature Location): lokalisasi *landmark* wajah dirumuskan sebagai masalah regresi, di mana fungsi *loss*nya adalah jarak Euclidean:

$$L_i^{landmark} = ||y_i^{landmark} - y_i^{gt}||_2^2 \quad (3)$$

Demikian juga, $y_i^{landmark}$ adalah koordinat fitur yang diregresi dari jaringan. y_i^{gt} adalah *ground-truth* yang mengandung lima koordinat, yaitu dua mata, dua sudut mulut, dan hidung.

Langkah terakhir adalah mengemas semua informasi ke dalam kamus dengan tiga kunci: 'box', 'confidence', and 'keypoints'. 'box'. 'box' berisi koordinat kotak pembatas, 'confidence' berisi tingkat kepercayaan jaringan untuk setiap kotak, dan 'key' berisi koordinat setiap *landmark* wajah (mata, hidung, dan titik akhir mulut).

Pembagian *dataset* pada penelitian ini dilakukan berdasarkan kelas *dataset*. Setiap kelas memiliki folder tersendiri yang berisi beberapa citra dengan dimensi sesuai kelas tersebut.. Setelah itu, sebagian besar citra ini diuji menggunakan MTCNN dan menghasilkan beberapa skor, seperti skor *confidence* dan *threshold*, yang digunakan untuk menghitung menghitung akurasi, presisi, *recall*, *True Positive Rate* (TPR), dan *False Positive Rate* (FPR) dalam perhitungan ROC-AUC, di mana semua skor ini digunakan untuk evaluasi. Tujuan dilakukannya ini adalah untuk menyeimbangkan data dan melihat perbedaan skor hasil performa algoritma ini.

Pengujian Sistem

Salah satu aspek penting dari Machine Learning adalah evaluasi model. Dalam penelitian ini perlu memiliki beberapa mekanisme untuk mengevaluasi model yang diujikan. Di sinilah metrik kinerja akan memberi gambaran tentang seberapa bagus sebuah model. Beberapa metrik seperti menghitung akurasi, presisi, *recall*, *F1-score*, ROC-AUC, akan digunakan untuk mengevaluasi tugas klasifikasi deteksi wajah ini. Deteksi wajah adalah tugas mendeteksi semua wajah

manusia pada gambar yang diberikan. Teknik ini biasanya dilakukan dengan mengekstraksi daftar kotak pembatas, yaitu koordinat persegi panjang terkecil di sekitar wajah (Arora et al., 2022). Detektor yang sempurna harus:

- cepat — idealnya *real-time*, minimal di atas 1 *Frames Per Second* (FPS).
- akurat — seharusnya hanya mendeteksi wajah (tidak ada positif palsu) dan mendeteksi semua wajah (tidak ada negatif palsu)
- kokoh — wajah harus dideteksi dalam berbagai pose, rotasi, kondisi pencahayaanmemanfaatkan semua sumber daya yang
- tersedia — menggunakan GPU jika memungkinkan, menggunakan *input* warna (RGB).

Tolak Ukur (Benchmark)

Benchmark dalam penelitian ini dirancang untuk mengevaluasi performa algoritma MTCNN dalam mendeteksi wajah, dengan fokus pada efisiensi jumlah wajah pada *dataset* dengan karakteristik yang bervariasi, dari *dataset* berisi citra yang berjumlah 104 data citra wajah dengan berbagai jumlah dan posisi lokasi wajah. Relevansi dari tolak ukur tertentu yang dipilih, seperti dimensi citra, jumlah wajah dalam satu *frame*, dan FPS. Dengan *benchmark* untuk evaluasi ini akan membandingkan sesuatu yang sejenis melalui tolak ukur yang sudah ditentukan, dengan tujuan mendapatkan hasil perbandingan sehingga bisa melakukan perbaikan. Uji *Benchmark* membantu menguji performa utilitas. *Benchmark* memfasilitasi perbandingan yang sederhana antara perangkat CPU dengan menilai kinerjanya melalui serangkaian pengujian standar, dan ini bermanfaat dalam berbagai aspek. Evaluasi dengan *benchmark* ini akan menghasilkan hasil akhir berupa angka (skor). Semakin tinggi skor hasil *benchmark*, maka dapat diartikan bahwa perangkat tersebut mempunyai performa yang lebih baik. Dalam penelitian ini, dilakukan serangkaian uji algoritma Pembelajaran Mendalam (Deep Learning) ini pada perangkat keras yang digunakan. Tolak ukur yang diujikan merupakan Internal:

1. Data uji merupakan *dataset* yang telah dibagi menjadi 3 klasifikasi data:
 - a) Data citra berdimensi di bawah 1,000 x 1,000 piksel, citra dengan dimensi antara 1,000 x 1,000 sampai 2,000 x 2,000

piksel, citra dengan dimensi di atas 2,000 x 2,000 piksel

- b) Data citra memiliki jumlah *face* 2 atau 3 dan data citra memiliki 10 jumlah wajah dalam setiap *frame*, dengan wajah frontal menghadap kamera.
2. Pengujian tersebut dilakukan dengan menjalankan program pada perangkat penelitian dan kemudian mengukur waktu yang diperlukan untuk menyelesaikannya.
3. Mengukur FPS (*Frame Per Second*)

FPS digunakan untuk mengukur berapa banyak *frame* yang ditangkap dalam satu detik, dimana idealnya minimal diatas 1 FPS.

Performance Evaluation Metrics

Performance Evaluation Metric merupakan satuan yang digunakan untuk menguji kualitas model *machine learning* yang telah dilatih. Setelah proses pengujian terdapat nilai-nilai yang dapat dicari dengan membandingkan data latih dan data validasi sehingga menghasilkan tabel *confusion matrix* (Bao & Dang, 2021) (Wang, 2008) (Yang et al., 2020).

a. Akurasi

Akurasi digunakan untuk mengukur kinerja model, metrik merupakan yang mengukur proporsi prediksi benar yang dibuat oleh model terhadap jumlah total prediksi yang dibuat. teknik akurasi merupakan salah satu metrik yang paling banyak digunakan untuk mengevaluasi kinerja model klasifikasi.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

b. Presisi (Mengukur proporsi prediksi positif yang benar)

Presisi merupakan penggambaran tingkat keakuratan data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, nilai *precision* dapat didefinisikan sebagai rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif yang dibuat oleh model.

Secara matematis, ini didefinisikan sebagai berikut.

$$\text{Presisi} = \frac{TP}{(TP+FP)} \quad (5)$$

Dari semua kelas positif yang telah di prediksi dengan benar, berapa banyak data yang benar benar positif. Nilai dari *precision* didapat dari Persamaan 3. Pada persamaan tersebut terdapat TP sebagai *True Positive* dibagi dengan TP ditambah FP sebagai *False Positive*. Nilainya berkisar dari 0 hingga 1. Nilainya berkisar dari 0 hingga 1.

c. Recall

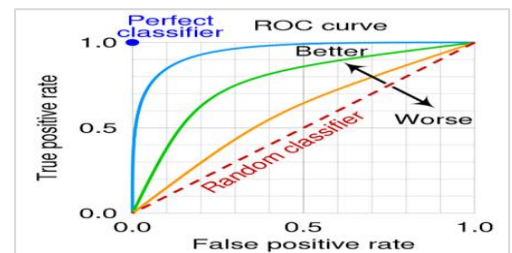
Recall mengukur keefektifan model klasifikasi dalam mengidentifikasi semua contoh yang relevan dari kumpulan data. *Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Persamaan 4 dilakukan untuk mencari nilai *recall*. Maka dari itu, data yang positif merupakan hasil prediksi dari model yang dibuat dan divalidasi dengan data uji coba. Nilai *recall* dapat dihitung dengan Persamaan 4.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (6)$$

Dengan TP merupakan *True Positive* dan FN merupakan *False Negative* [19].

d. AUC-ROC curve

Kurva ROC (*Receiver Operating Characteristic*) dan AUC (*Area Under the Curve*) adalah metrik *Machine Learning* yang digunakan untuk mengevaluasi performa model klasifikasi biner. Kurva ROC adalah plot tingkat positif sejati (TPR) terhadap tingkat positif palsu (FPR) di berbagai pengaturan ambang batas, dan dibuat dengan memvariasikan ambang batas untuk memprediksi hasil positif atau negatif dan memplot TPR terhadap FPR untuk setiap ambang batas. Pada gambar 3 di bawah, setiap garis berwarna mewakili kurva ROC dari sistem klasifikasi biner yang berbeda. Sumbu mewakili FPR dan TPR. Garis diagonal mewakili pengklasifikasi acak, sedangkan sudut kiri atas mewakili pengklasifikasi sempurna dengan TPR=1 dan FPR=0.



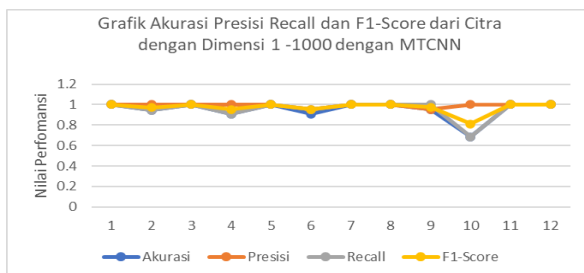
Gambar 3 Kurva ROC

Pada saat yang sama, AUC mewakili kinerja model secara keseluruhan. AUC adalah area di bawah kurva ROC, yang mewakili probabilitas bahwa sampel positif yang dipilih secara acak akan diberi peringkat lebih tinggi oleh model daripada sampel negatif yang dipilih secara acak. Model sempurna akan memiliki AUC 1, sedangkan model acak akan memiliki AUC 0,5. AUC memberikan nilai tunggal yang meringkas performa keseluruhan model dan sangat berguna saat membandingkan performa beberapa model.

Tingkat positif benar dan salah pada ambang yang berbeda sangat berguna ketika kelas tidak seimbang, yang berarti ada lebih banyak sampel negatif daripada yang positif. Dalam kasus seperti itu, kurva ROC dan AUC dapat memberikan penilaian performa model yang lebih akurat daripada metrik seperti akurasi atau skor F1, yang mungkin bias terhadap kelas mayoritas.

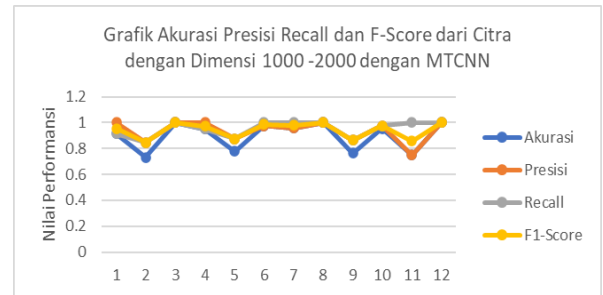
HASIL DAN PEMBAHASAN

Berdasarkan dataset yang telah dipisahkan kelas dimensinya, dilakukan implementasi CNN menggunakan MTCNN untuk mendapatkan hasil pembuktian algoritma *deep learning* ini. Gambar 4 merupakan performansi hasil uji dari kelas dimensi dataset dari model MTCNN pada dataset dengan dimensi citra antara 1 hingga 1,000 piksel. Nilai performansi yang dianalisis adalah menghitung akurasi, presisi, *recall*, *F1-score*, ROC-AUC.



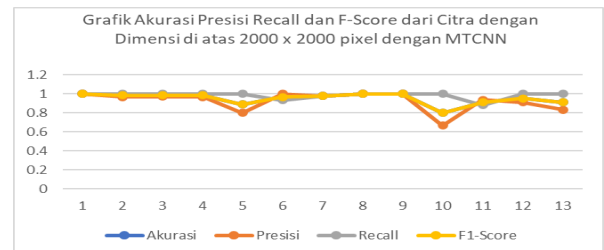
Gambar 4 Nilai Performansi dari Citra dengan Dimensi di bawah 1000 x 1000 piksel

Gambar 5 memperlihatkan performansi model dengan citra menggunakan dimensi (1000-2000 piksel). Pengujian pada rentang dimensi 1000–2000 piksel memberikan gambaran performansi model MTCNN dalam kondisi yang realistis. Hasil pengujian ini diharapkan menunjukkan bahwa algoritma mampu mendeteksi wajah secara akurat, cepat, dan efisien.



Gambar 5 Nilai Performansi dari Citra Dimensi Antara 1000 x 1000 Sampai 2000 x 2000

Gambar 6 memperlihatkan performansi model dengan citra menggunakan dimensi diatas (2000 piksel).



Gambar 6 Nilai Performansi dari Citra dengan Dimensi di Atas 2000 x 2000 Piksel

Tabel 1 Tabel Rata-rata Performa Tiga Kelas Dimensi Citra

Dimensi (D)	Recall	Precision	F1-score	Accuracy
D < 1,000 x 1,000	95,75%	99,20%	97,20%	94,99%
D > 1,000 x 1,000				
D < 2,000 x 2,000	95,26%	93,76%	94,38%	89,86%
D > 2,000 x 2,000	98,40%	92,50%	95,01%	95,02%

Analisis Oklusi Citra

Gambar 7 merupakan hasil pengujian citra dalam rancangan arsitektur MTCNN yang digunakan dalam penelitian ini, berhasil mengekstrak wajah dari keseluruhan gambar yang memiliki tantangan dalam deteksi wajah yang dapat mengurangi akurasi dan tingkat deteksi pengenalan wajah.



Gambar 7 Citra Hasil Uji Dataset Terhadap Beberapa Hambatan/Oklusi yang Dapat Diatasi dengan Baik oleh MTCNN

Dari hasil pengujian citra dalam rancangan arsitektur MTCNN yang digunakan dalam penelitian ini, berhasil mengekstrak wajah dari keseluruhan gambar dan memiliki tantangan dalam deteksi wajah yang dapat mengurangi akurasi dan tingkat deteksi pengenalan wajah, beberapa diantaranya adalah:

- 1) Okultasi. Wajah mungkin sebagian tersembunyi oleh objek, seperti kacamata, syal, tangan, rambut, topi, dan objek lainnya, yang berdampak pada tingkat deteksi.
- 2) Orientasi. Orientasi wajah dan sudut ke arah kamera berdampak pada tingkat deteksi wajah.
- 3) Scale/Banyak wajah dalam satu gambar. Gambar dengan jumlah wajah manusia yang tinggi sangat menantang untuk tingkat deteksi yang akurat.
- 4) Pencahayaan. Setiap perubahan dalam kondisi pencahayaan subjek menimbulkan masalah untuk deteksi wajah resolusi rendah. Gambar beresolusi rendah atau *noise* gambar berdampak negatif pada tingkat deteksi.

Analisis Pengaruh Dimensi dan Jumlah Wajah

Pada okultasi dalam hal wajah yang berjumlah besar pada 1 *frame* citra, ditemukan ternyata ukuran dimensi sangat berpengaruh pada pendeteksian jumlah wajah yang dapat ditemukan oleh *detector* MTCNN ini. Pada penelitian ini didapati dengan menguji citra yang merupakan citra yang berisi wajah *selfie* dengan jumlah terbanyak di dunia, melibatkan kurang lebih 1,150 wajah manusia. Semakin besar jumlah piksel dari suatu citra, maka jumlah wajah yang ditemukan meningkat keakurasiannya. Namun waktu proses mengalami penambahan waktu dengan penjelasan seperti pada gambar 8.

Scale Dimensi	Jumlah Wajah	Waktu Proses	Citra Hasil	Nilai Confidence
326 x 580	57	0.399		<pre> 48 0.939668 ('left_eye': (492, 182), 49 0.932171 ('left_eye': (81, 258), 50 0.930404 ('left_eye': (283, 187), 51 0.915710 ('left_eye': (359, 193), 52 0.881253 ('left_eye': (371, 200), 53 0.823224 ('left_eye': (48, 195), 54 0.820862 ('left_eye': (510, 171), 55 0.766010 ('left_eye': (513, 226), 56 0.712585 ('left_eye': (127, 182), Number of face : 57 </pre>
2000 x 3558	252	45 menit		<pre> confidence 0 1.000000 ('left_eye': (1242, 1063), 1 1.000000 ('left_eye': (3364, 1494), 2 1.000000 ('left_eye': (513, 1420), 3 1.000000 ('left_eye': (3143, 1259), 4 0.999998 ('left_eye': (657, 1412), ... 247 0.714713 ('left_eye': (3104, 1941), 248 0.715795 ('left_eye': (384, 715), 249 0.711942 ('left_eye': (3042, 529), 250 0.708775 ('left_eye': (2462, 1132), 251 0.707422 ('left_eye': (1419, 1075), [252 rows x 2 columns] Number of face : 252 </pre>

Gambar 8 Citra dengan Lebih dari 1150 Wajah dalam Satu Frame

Dari *dataset* yang telah diklasifikasikan berdasarkan kelas dimensi, maka didapati hasil uji tolok ukur terhadap kemampuan perangkat keras dalam melaksanakan tugas algoritma penelitian ini adalah seperti tertulis dalam tabel *benchmark*

Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
2.20 GHz, 64-bit operating system, x64-based processor. Python 3.10.6, with a **CPU-based** Tensorflow 2.9.1

Tabel 2 Citra dengan 2 Wajah Frontal

Ukuran Citra	Total Piksel	Waktu Proses	FPS
640 x 640	409,600	0.238 sec	4.2
653 x 723	472,119	0.298 sec	3.4
960 x 1280	1,228,800	0.322 sec	3.1
1040 x 780	811,200	0.334 sec	3.0
1280 x 960	1,228,800	0.397 sec	2.5
1280 x 1280	1,638,400	0.354 sec	2.8

Ukuran Citra	Total Piksel	Waktu Proses	FPS
2719 x 2193	5,962,767	0.460 sec	2.2

FPS pada tabel 2 merupakan *frame per second*. Parameter ini digunakan untuk mengukur berapa banyak *frame* yang ditangkap dalam satu detik.

Tabel 3 merupakan citra dengan 10 wajah frontal.

Tabel 3 Citra dengan 10 Wajah Frontal

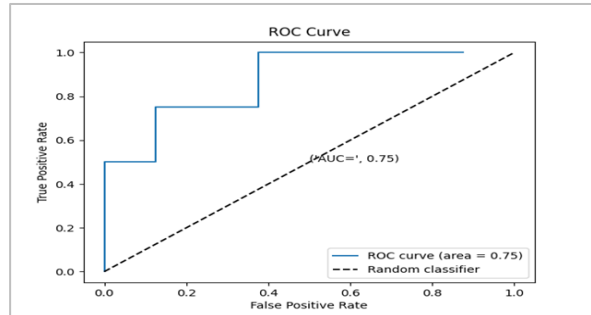
Ukuran Citra	Total Piksel	Waktu Proses	FPS
720 x 720	518400	0.261 sec	3.8
738 x 854	630252	0.259 sec	3.9
592 x 1052	622784	0.254 sec	3.9
1238 x 1070	1324660	0.534 sec	1.9
2309 x 2447	5650123	1.810 sec	0.6
2322 x 4128	9585216	1.671 sec	0.6
3456 x 5184	17915904	2.000 sec	0.5

Berdasarkan hasil uji *dataset* pada nilai *confidence* dan klasifikasi deteksi ada atau tidaknya wajah pada citra, pada penelitian ini ditentukan ambang batas *confidence* levelnya untuk dinyatakan sebagai wajah manusia adalah $> 95\%$, namun ditemukan di beberapa *False Negatif* dengan nilai *confidence* di bawah 95% ternyata adalah merupakan tampilan wajah dan harus diberi nilai sebagai TP (*True Positif*), hal ini bisa terjadi ketika ada ketidakseimbangan dalam data. Maka nilai-nilai akurasi, presisi dan recall tidak cukup untuk mengevaluasi implementasi dr algoritma mesin pembelajaran ini. Dibutuhkan metrik lain yaitu dengan kurva ROC-AUC.

AUC adalah area di bawah kurva ROC, mulai dari 0,0 hingga 1,0, dengan nilai yang lebih tinggi menunjukkan kinerja pengklasifikasi yang lebih baik. Kurva ini mewakili probabilitas dengan model penelitian ini mampu membedakan antara dua kelas yang ada di dalam target. Semua dilakukan dengan *threshold value*: [1,0.99, 0.90, 0.85, 0.80, 0.77, 0.75, 0.70, 0.60, 0.40, 0.20, 0]

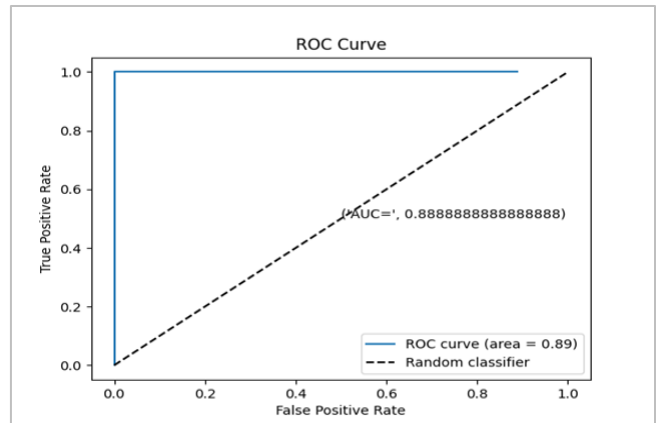
- 1) Dimensi Piksel di bawah 1000 x 1000

Gambar 9. ROC-AUC dengan skor terendah: 0.75



Gambar 9 ROC-AUC skor 0.75

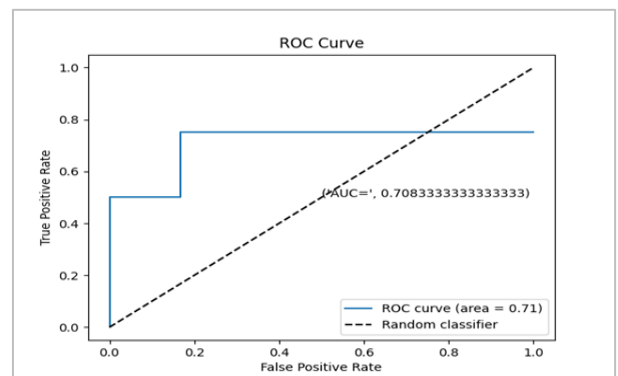
Gambar 10. Kurva ROC-AUC skor tertinggi: 0.89



Gambar 10 Kurva ROC-AUC Skor 0.89

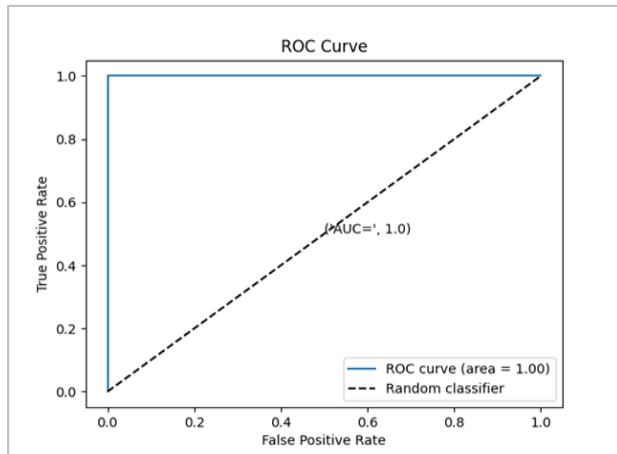
- 2) Dimensi Piksel Antara 1000 x 1000 sampai 2000 x 2000

Gambar 11 Kurva ROC-AUC dengan skor terendah 0.71



Gambar 11 Kurva ROC-AUC Skor 0.71

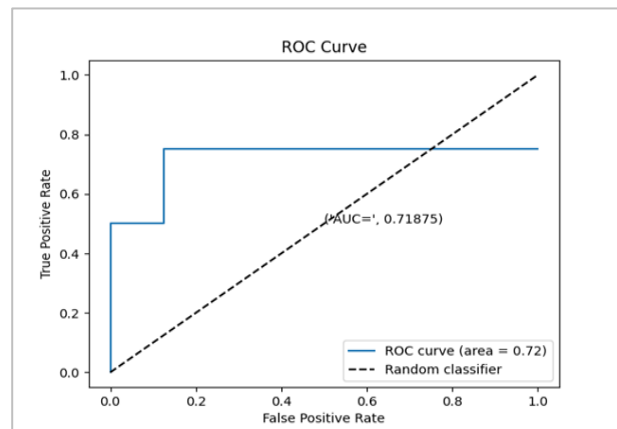
Gambar 12 Kurva ROC-AUC dengan skor tertinggi 1.0



Gambar 12 ROC-AUC Skor 1.0

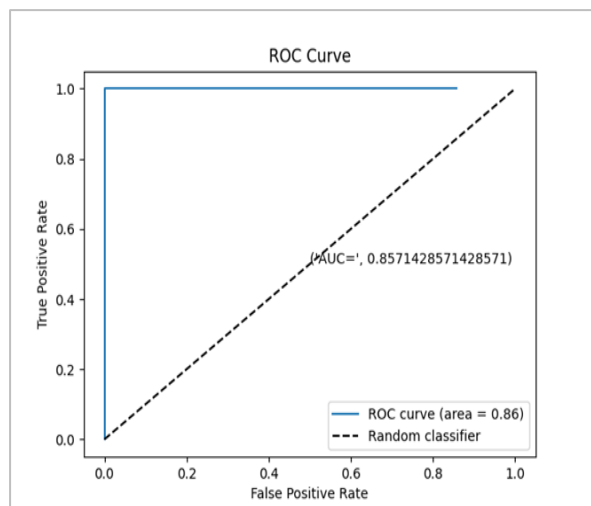
3) Dimensi Piksel di atas 2000 x 2000

Gambar 13 Kurva ROC-AUC dengan skor terendah: 0.71



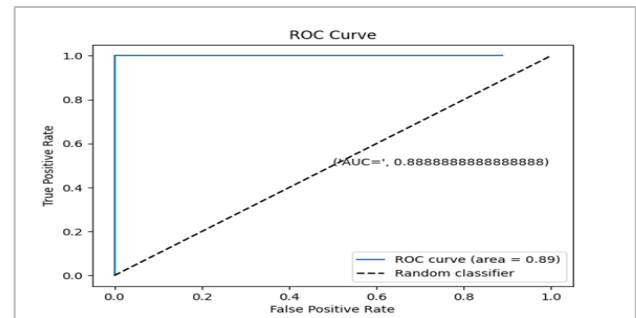
Gambar 13 Kurva ROC-AUC Skor 0.71

Gambar 14 Kurva ROC-AUC dengan skor tertinggi: 0.86



Gambar 14 Kurva ROC-AUC Skor 0.86

Dari hasil ROC-AUC di atas, maka untuk semua kelas dimensi, didapati nilai rata-rata AUC-ROC kurva nya adalah sebesar 0.89, seperti dijelaskan pada gambar 15.



Gambar 15 Kurva AUC-ROC untuk Nilai Skor ROC-AUC Rata-rata dari Semua Kelas Dimensi Data Model

Nilai skor yang didapat ini menggambarkan semakin tinggi AUC, semakin baik performa model dalam membedakan antara kelas positif dan negatif. Ketika $AUC = 1$, pengklasifikasi dapat dengan benar membedakan antara semua titik kelas Positif dan Negatif. Namun, jika AUC adalah 0, maka classifier akan memprediksi semua Negatif sebagai Positif dan semua Positif sebagai Negatif.

Ketika $0,5 < AUC < 1$, ada kemungkinan besar pengklasifikasi dapat membedakan nilai kelas positif dari nilai negatif. Ini karena pengklasifikasi mampu mendeteksi lebih banyak jumlah *True positive* dan *True negative* daripada *False negative* dan *False positive*.

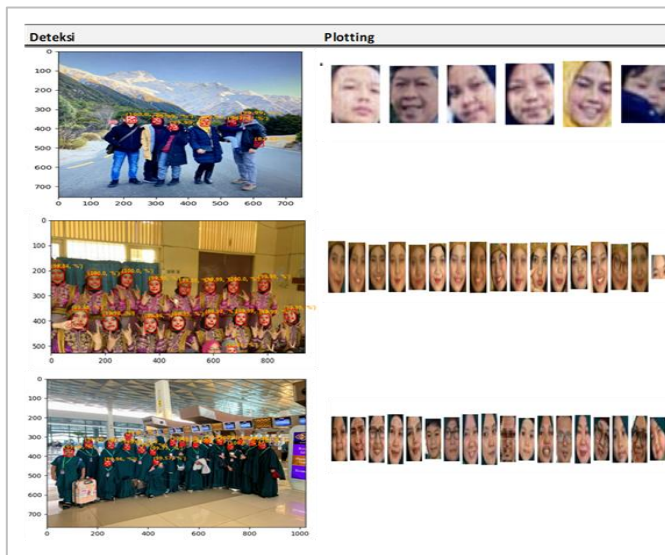
Ketika $AUC = 0,5$, maka pengklasifikasi tidak dapat membedakan antara poin kelas Positif dan Negatif. Artinya pengklasifikasi memprediksi kelas acak atau kelas konstan untuk semua titik data. Jadi, semakin tinggi nilai AUC sebuah *classifier*, maka semakin baik kemampuannya untuk membedakan antara kelas positif dan negatif.

Pada hasil keseluruhan dan rata-rata yang didapat dalam penelitian ini dapat disimpulkan dari skor nilai rata-rata ROC-AUC = 0.88, membuktikan bahwa model ini memiliki kemampuan yang sangat baik untuk membedakan kelas positif maupun negatif, sehingga akurasi hasilnya sangat baik.

Hasil Plot Wajah

Tujuan penelitian untuk mengimplementasikan arsitektur MTCNN sebagai tambahan penguat paper-paper pembuktian yang telah dilakukan sebelumnya, pembuktian terhadap akurasi dan pengaruh lainnya dalam mendapatkan deteksi wajah yang kuat, juga sebagai *inputan* ekstraksi wajah ke sistem yang lain

(plotting wajah dalam citra). Setelah mendapati nilai akurasi yang tinggi, maka hasil keluaran (*output*) dapat menjadi masukan/*input* sistem lain. Hal ini dilakukan dengan mengekstraksi setiap wajah dan memplotnya sebagai *subplot* terpisah, maka dengan mudah hasilnya dapat disimpan ke *file*. Dengan *draw_faces()* dipakai pada pengkodean maka akan mengekstrak dan memplot setiap wajah yang terdeteksi dalam sebuah foto. Gambar 16 merupakan beberapa gambar hasil dari *plot* wajah yang dapat menjadi *input* bagi sistem lain.



Gambar 16 *Plotting* Wajah Terdeteksi Sebagai *Input* Sistem

KESIMPULAN

Penerapan MTCNN berhasil menggunakan TensorFlow dan OpenCV untuk mendeteksi wajah dari berbagai kondisi citra, termasuk citra dengan dimensi kecil hingga besar dan jumlah wajah yang bervariasi. Hasil yang dilakukan dengan pengujian menunjukkan bahwa model mencapai tingkat akurasi, presisi, recall, *F1-score*, dan ROC-AUC yang tinggi. Secara spesifik dari hasil, performa rata-rata pada dimensi citra kurang dari 1.000 x 1.000 piksel mencatat akurasi 94,99%, presisi 99,20%, recall 95,75%, *F1-score* 97,20%, dan ROC-AUC 90,89%. Pada dimensi antara 1.000 x 1.000 hingga 2.000 x 2.000 piksel, performa rata-rata adalah akurasi 89,86%, presisi 93,76%, recall 95,26%, dan *F1-score* 94,38%. Sementara itu, citra dengan dimensi lebih besar dari 2.000 x 2.000 piksel menunjukkan akurasi 95,02%, presisi 92,50%, recall 98,40%, dan *F1-score* 95,01%. Hasil ini menunjukkan bahwa dimensi citra memiliki pengaruh signifikan terhadap performa dari deteksi wajah.

Teknik yang digunakan berhasil mendeteksi wajah pada citra dengan jumlah wajah yang besar, seperti pada citra yang mengandung lebih dari 1.150 wajah. Semakin tinggi resolusi dari sebuah resolusi citra, jumlah wajah yang terdeteksi dengan akurasi tinggi meningkat, namun waktu proses juga bertambah. Citra dengan dimensi 720 x 720 piksel, waktu prosesnya adalah 0,261 detik dengan kecepatan 3,8 FPS, sedangkan untuk citra dengan dimensi 3456 x 5184 piksel, waktu prosesnya meningkat menjadi 2,000 detik dengan kecepatan hanya 0,5 FPS. Parameter seperti pencahayaan, orientasi wajah, dan oklusi juga diuji, menunjukkan bahwa model tetap efektif meskipun menghadapi tantangan-tantangan tersebut.

Penelitian ini membuktikan keakuratan dan efisiensi model MTCNN tetapi juga memberikan hasil deteksi yang dapat dimanfaatkan sebagai *input* untuk sistem lain. Wajah-wajah yang terdeteksi diproses lebih lanjut melalui metode *face cropping* untuk menghasilkan data wajah individual yang dapat digunakan untuk berbagai aplikasi seperti pengenalan wajah, keamanan, atau analisis lanjutan. Dengan rata-rata ROC-AUC sebesar 0,88, model ini terbukti sangat baik dalam membedakan kelas positif dan negatif.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada DRPPM Universitas Budi Luhur atas dukungan dan sumber pendanaan dalam penelitian ini.

DAFTAR PUSTAKA

- Arora, M., Naithani, S., & Areeckal, A. S. (2022). A web-based application for face detection in real-time images and videos. *Journal of Physics: Conference Series*, 2161(1), 012071. <https://doi.org/10.1088/1742-6596/2161/1/012071>
- Bao, Y., & Dang, R. (2021). Face Detection under Non-uniform Low Light Based on Improved MTCNN. *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, 704–707. <https://doi.org/10.1109/ICAICE54393.2021.00138>
- Chen, X., Luo, X., Liu, X., & Fang, J. (2019). *Eyes Localization Algorithm Based on Prior MTCNN Face Detection*. *Itaic*, 1763–1767.
- Du, J. (2020). High-Precision Portrait Classification Based on MTCNN and Its Application on Similarity Judgement. *Journal of Physics: Conference Series*, 1518(1). <https://doi.org/10.1088/1742-6596/1518/1/012066>

- Farfade, S. S., Saberian, M. J., & Li, L.-J. (2015). Multi-view Face Detection Using Deep Convolutional Neural Networks. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 643–650.
<https://doi.org/10.1145/2671188.2749408>
- Fortuna, I., & Khaeruzzaman, Y. (2022). Implementation of OCR and Face Recognition on Mobile Based Voting System Application in Indonesia. *IJNMT (International Journal of New Media Technology)*, 9(1), 20–27.
<https://doi.org/10.31937/ijnmt.v9i1.2658>
- Galindo, A. M. O., Baldovino, R. G., Billones, R. K. C., Dadios, E. J. P., & Valenzuela, I. C. (2022). Masked Face Recognition Using Multi-Task Cascaded Convolutional Networks and FaceNet. *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–6.
<https://doi.org/10.1109/HNICEM57413.2022.10109391>
- Hofer, P., Roland, M., Schwarz, P., Schwaighofer, M., & Mayrhofer, R. (2021). Importance of different facial parts for face detection networks. *2021 IEEE International Workshop on Biometrics and Forensics (IWBF)*, 1–6.
<https://doi.org/10.1109/IWBF50991.2021.9465087>
- Kumari Sirivarshitha, A., Sravani, K., Priya, K. S., & Bhavani, V. (2023). An approach for Face Detection and Face Recognition using OpenCV and Face Recognition Libraries in Python. *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1, 1274–1278.
<https://doi.org/10.1109/ICACCS57279.2023.10113066>
- Ma, M., & Wang, J. (2018). Multi-View Face Detection and Landmark Localization Based on MTCNN. *2018 Chinese Automation Congress (CAC)*, 4200–4205.
<https://doi.org/10.1109/CAC.2018.8623535>
- Raman. (2023). Multiscale Facial Expression Based Face Recognition using Normalized Segment Classification Detection in Face Images. *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1, 2381–2386.
<https://doi.org/10.1109/ICACCS57279.2023.10112676>
- Ranjan, R., Patel, V. M., & Chellappa, R. (2019). HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1), 121–135.
<https://doi.org/10.1109/TPAMI.2017.2781233>
- Ravidas, S., & Ansari, M. A. (2018). *Multi View Face Detection using Deep Learning*.
- Salih, Z. A., Thabit, R., Zidan, K. A., & Khoo, B. E. (2022). A new face image manipulation reveal scheme based on face detection and image watermarking. *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAET)*, 1–6.
<https://doi.org/10.1109/IICAET55139.2022.9936838>
- Saraswat, P., & Raj, S. (2022). DATA PRE-PROCESSING TECHNIQUES IN DATA MINING: A REVIEW. *International Journal of Innovative Research in Computer Science & Technology*, 122–125.
<https://doi.org/10.55524/ijircst.2022.10.1.22>
- Shih, H.-C., Tai, S.-K., Hu, C.-Y., Lee, W.-S., & Liu, H.-Y. (2023). PhotoSaver: Group Photographing Guidance System Using Multi-Task Cascaded Convolutional Networks. *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–4.
<https://doi.org/10.1109/ISCAS46773.2023.10181414>
- Suhas, K. K., Prakash, M. S., Venkatesh, Y., & Latha, P. M. (2023). Multi-Task Cascaded Convolutional Neural Networks (MTCNN) based Driver Drowsiness Detection. *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, 790–793.
<https://doi.org/10.1109/ICCMC56507.2023.10084076>
- Sukumar, P., Robert, L., & Yuvaraj, S. (2016). Review on modern Data Preprocessing techniques in Web usage mining (WUM). *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, 64–69.
<https://doi.org/10.1109/CSITSS.2016.7779441>
- Thohari, A., & Hertantyo, G. B. (2018). Implementasi Convolutional Neural Network untuk Klasifikasi Pembalap MotoGP Berbasis GPU. *Proceedings on Conference on Electrical Engineering, Telematics, Industrial Technology, and Creative Media*, 50–55.
- Viola, P., & Jones, M. (2001). Managing work role performance: Challenges for twenty-first century organizations and their employees. *Rapid Object Detection Using a Boosted Cascade of Simple Features*, 511–518.
- Wang, K. (2008). Improving Engineering Data Management with Semantic Web Techniques. *Journal of Service Science and Management*, 01(03), 199–205.
<https://doi.org/10.4236/jssm.2008.13021>

- Xiang, J., & Zhu, G. (2017). *Joint Face detection and Facial Expression Recognition with MTCNN*. 424–427. <https://doi.org/10.1109/ICISCE.2017.95>
- Xie, R., Zhang, Q., Yang, E., & Zhu, Q. (2019). A Method of Small Face Detection Based on CNN. *2019 4th International Conference on Computational Intelligence and Applications (ICCIA)*, 78–82. <https://doi.org/10.1109/ICCIA.2019.00022>
- Xu, B., Wang, W., Guo, L., Chen, G., Wang, Y., Zhang, W., & Li, Y. (2021). Evaluation of Deep Learning for Automatic Multi-View Face Detection in Cattle. *Agriculture*, 11(11), Article 11. <https://doi.org/10.3390/agriculture11111062>
- Yang, Z., Ge, W., & Zhang, Z. (2020). Face Recognition Based on MTCNN and Integrated Application of FaceNet and LBP Method. *2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*, 95–98. <https://doi.org/10.1109/AIAM50918.2020.00024>
- Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503. <https://doi.org/10.1109/LSP.2016.2603342>
- Zhao, F., Li, J., Zhang, L., Li, Z., & Na, S. G. (2020a). Multi-view face recognition using deep neural networks. *Future Generation Computer Systems*, 111, 375–380. <https://doi.org/10.1016/j.future.2020.05.002>
- Zhao, F., Li, J., Zhang, L., Li, Z., & Na, S.-G. (2020b). Multi-view face recognition using deep neural networks. *Future Generation Computer Systems*, 111, 375–380. <https://doi.org/10.1016/j.future.2020.05.002>

Halaman ini sengaja dikosongkan