

ABSTRACT

Facial recognition is a technology that is widely used in various fields such as education, security, and even health. At this time where the corona virus is still quite high and the use of masks is still mandatory in some places, especially closed places and many people. this causes the face recognition to be less effective because most of the faces are covered.

The face, which is mostly covered, becomes an obstacle when facial recognition works, because many facial landmarks are covered. An example is the mouth and nose, often the shape of the face changes according to the shape of the mask. With this problem the author tries to modify the facial recognition method. where the face is usually recognized by all its parts, in this study only used the upper half of the face, namely the eyes, eyebrows, and a small part of the nose.

In this study also used MTCNN as a face detector and VGGFace for facial landmark extraction. From the research conducted, it was found that this method has an accuracy rate of 95.6%. this result is an increase of about 2.3% compared to using the whole face.

Keywords : MTCNN, VGGFace, Face Recognition, Covid-19

INTISARI

Pengenalan wajah merupakan sebuah teknologi yang banyak digunakan diberbagai bidang contohnya pendidikan, keamanan, dan bahkan kesehatan. Pada saat ini virus corona masih cukup tinggi dan penggunaan masker masih diwajibkan dibeberapa tempat, khususnya tempat tertutup dan banyak orang. hal ini menyebabkan berkurang efektifnya pengenalan wajah karena sebagian besar wajah tertutupi.

Wajah yang sebagian besar tertutupi ini menjadi kendala saat face recognition bekerja, karena banyak facial landmark yang tertutupi. Contohnya adalah mulut dan hidung, seringkali bentuk wajah pun berubah mengikuti bentuk masker. Dengan masalah ini penulis mencoba memodifikasi metode pengenalan wajah. Biasanya wajah dikenali dengan seluruh bagiannya, pada penelitian ini hanya menggunakan separuh bagian atas wajah yaitu mata, alis, dan sebagian kecil hidung.

Pada penelitian ini juga digunakan MTCNN sebagai detektor wajah dan VGGFace untuk ekstraksi landmark wajah. dari penelitian yang dilakukan didapatkan bahwa metode ini memiliki tingkat akurasi sebesar 95,6%. hasil ini meningkat sekitar 2,3% dibanding menggunakan seluruh wajah.

Kata kunci – MTCNN, VGGFace, Pengenalan Wajah, COVID-19

DAFTAR ISI

PERNYATAAN	ii
PRAKATA	iii
ARTI LAMBANG DAN SINGKATAN	v
ABSTRACT	vi
INTISARI	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pertanyaan Penelitian	2
1.3 Rumusan Penelitian	2
1.4 Keaslian penelitian	3
1.5 Tujuan Penelitian	5
1.6 Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	6
2.1 Tinjauan Pustaka	6
2.2 Landasan Teori	8
2.2.1 <i>Convolutional Neural Network (CNN)</i>	9
2.2.2 <i>Multi-task Cascaded Convolutional Neural Network (MTCNN)</i> ..	10
2.2.3 VGG-Face	13
2.3 Hipotesis	14
BAB III METODOLOGI	15
3.1 Alat dan Bahan	15
3.1.1 Alat	15
3.1.2 Dataset	15
3.2 Jalannya Penelitian	16
3.2.1 Studi Literatur	16
3.2.2 Pengumpulan Dataset	17
3.2.4 Implementasi	17
3.2.5 Penulisan Thesis	18
3.3 Perancangan Sistem	19
3.3.1 Blok Deteksi Wajah dengan MTCNN	20
3.3.2 Blok Pemotongan Wajah	21
3.3.3 Blok Ekstraksi Pixel ke dalam Array	23
3.3.4 Blok Ekstraksi Landmark	23

3.3.5	Blok Data Pickle	25
3.3.6	Blok Pembandingan	26
3.4	Cara Analisis	27
BAB IV HASIL DAN PEMBAHASAN.....		29
4.1	Hasil Sistem	30
4.1.1	Hasil Blok Deteksi Wajah	30
4.1.2	Hasil Blok Pemotongan Wajah	31
4.1.3	Hasil Blok Ekstraksi Pixel	33
4.1.4	Hasil Blok Ekstraksi Landmark	33
4.1.5	Hasil Blok Data Pickle	34
4.1.6	Hasil Blok Pembandingan.....	34
4.1.7	Pembandingan VGG Face dan Facenet.....	38
4.1.8	Kekurangan dan Kelebihan	39
BAB V KESIMPULAN DAN SARAN.....		41
5.1	Kesimpulan	41
5.2	Saran	41
DAFTAR PUSTAKA.....		42
LAMPIRAN.....		1

DAFTAR GAMBAR

Gambar 2.1.	Arsitektur CNN [6]	9
Gambar 2.2.	Arsitektur MTCNN	12
Gambar 2.3.	Pipeline MTCNN [9].....	12
Gambar 2.4.	Arsitektur Resnet50.....	14
Gambar 3.1.	Dataset.....	16
Gambar 3.2.	Diagram Jalannya Penelitian.....	16
Gambar 3.3.	Diagram Alir Perancangan Sistem	19
Gambar 3.4.	Masukan dan keluaran Blok Deteksi Wajah.....	21
Gambar 3.5.	Masukan dan keluaran Blok Pemotongan Wajah.....	23
Gambar 3.6.	Masukan dan keluaran Blok Ekstraksi Landmark	25
Gambar 4.1.	Dataset Training	29
Gambar 4.2.	Dataset Test	30
Gambar 4.3.	Hasil Deteksi Wajah	31
Gambar 4.4.	Hasil Potongan	32
Gambar 4.5.	Hasil Ekstraksi Pixel	33
Gambar 4.6.	Hasil Ekstraksi.....	34
Gambar 4.7.	Hasil Pembandingan.....	35
Gambar 4.8.	Perbandingan seluruh wajah dan separuh wajah.....	36
Gambar 4.9.	Persamaan fitur dan penggunaan aksesoris	37
Gambar 4.10.	Prediksi VGGFace (kiri) dan Facenet (kanan).....	39

DAFTAR TABEL

Tabel 4.1 Tabel Hasil Separuh Wajah.....	36
Tabel 4.2 Tabel Sensitifitas dan presisi	37
Tabel 4.3 Tabel Perbandingan VGGFace (Resnet50) dan Facenet	39

BAB I PENDAHULUAN

1.1 Latar Belakang

Pandemi Corona Virus yang telah muncul mulai tahun 2019 telah menyebabkan banyak korban jiwa. Penyakit ini telah dikategorikan sebagai pandemi oleh Organisasi Kesehatan Dunia. Dengan tingkat penularan yang tinggi dan beberapa variasi yang lebih menular lagi membuat virus ini menjadi hal yang paling di khawatirkan oleh banyak orang. Pada puncak penularannya di Indonesia mencatatkan tingkat positif lebih dari 100.000 orang dengan tingkat kematian yang cukup tinggi.

Virus COVID-19 merupakan virus yang menyerang sistem pernapasan yang menyebabkan demam, batuk, dan sesak nafas. Virus ini dapat ditularkan dari droplet atau tetesan liur dari orang yang terinfeksi virus ini. Untuk mengurangi penyebaran virus ini dapat dilakukan penggunaan masker dan menjaga jarak. Seiring perkembangannya untuk lebih mencegah penyebaran virus ini, pemerintah dari berbagai negara mewajibkan penggunaan masker. Masker yang digunakan merupakan masker medis yang terdiri dari beberapa lapisan yang diharapkan dapat menyaring virus COVID-19 dan mengurangi risiko tertular penyakit ini Namun penggunaan masker dapat berdampak pada beberapa bidang salah satunya pengenalan wajah.

Pengenalan wajah memanfaatkan seluruh bagian wajah manusia mulai dari bentuk muka, alis, hidung, mata, mulut, dagu, dan lainnya. Dengan semakin banyak titik yang dapat di deteksi maka diharapkan akan semakin tinggi pula tingkat akurasi pengenalan wajahnya. Penggunaan masker menyebabkan beberapa titik wajah tertutupi terutama pada bagian mulut dan hidung. Hal ini dapat mengurangi tingkat pengenalan wajah karena semakin sedikit titik yang dapat dideteksi. Penggunaan pengenalan wajah yang biasa akan mendeteksi seluruh wajah beserta maskernya sehingga akan mengaburkan atau memberikan titik yang salah.

Saat ini algoritma pengenalan wajah sudah cukup berkembang. Pada

penelitian dengan menggunakan algoritma CNN-PCA, pengenalan wajah dengan seluruh wajah dideteksi mendapatkan akurasi sebesar 90% [1]. Pada penelitian lainnya dengan menggunakan CNN untuk mengidentifikasi wajah didapatkan akurasi sebesar 96.7%[2]. Pada penelitian dengan menggunakan algoritma Viola-Jones dan PCA pengenalan wajah tanpa masker mendapatkan akurasi sebesar 95% namun jika bermasker tingkat akurasi turun menjadi 72%[3]. Adapun dengan menggunakan algoritma MTCNN – Facenet akurasi yang didapat adalah 97% untuk wajah tanpa masker dan 82% untuk wajah bermasker. Dari penelitian-penelitian yang ada MTCNN merupakan algoritma dengan tingkat akurasi pengenalan wajah bermasker tertinggi.

Oleh karena itu, penelitian ini mengambil judul “Pengenalan Wajah Bermasker dan Tanpa Masker dengan MTCNN dan VGGFace” dengan menggunakan wajah bagian atas yang akan digunakan untuk pengenalan wajah.

1.2 Pertanyaan Penelitian

Pertanyaan penelitian dari thesis ini adalah

- Apakah pengenalan wajah dapat digunakan pada wajah bermasker?
- Bagaimana meningkatkan akurasi dalam proses face recognition?
- Bagaimana algoritma untuk pengenalan wajah dengan separuh wajah bagian atas?

1.3 Rumusan Penelitian

Dari penelitian di atas didapatkan rumusan masalah yaitu Proses pengenalan wajah yang konvensional akan mendeteksi seluruh wajah beserta maskernya sehingga akan mengaburkan atau memberikan titik yang salah.

1.4 Keaslian penelitian

N o	Peneliti	Judul	Model Face Detectio n	Facial Recogniti on	Pengguna an Wajah	Pengenal an Orang
1	E. Winarno, I. Husni Al Amin, H. Februariyant i, P. W. Adi, W. Hadikurniaw ati and M. T. Anwar	Attendance System Based on Face Recognition System Using CNN- PCA Method and Real- time Camera	CNN	PCA	Semua bagian	Ya
2	R. K. Shukla, A. K. Tiwari and V. Verma	Identification of with Face Mask and without Face Mask using Face Recognition Model	CNN		Semua bagian	Tidak
3	M. S. Ejaz, M. R. Islam, M. Sifatullah and A. Sarker	Implementati on of Principal Component Analysis on Masked and Non-masked Face Recognition	Viola- Jones	PCA	Semua bagian	Ya

4	M. S. Ejaz and M. R. Islam	Masked Face Recognition Using Convolutional Neural Network	MTCNN	Facenet	Semua bagian	Ya
5	Hanif Naufal	Masked & Unmasked Face Recognition Using MTCNN & VGG Face	MTCNN	VGG	Wajah bagian atas	Ya

Pada penelitian dengan judul *“Attendance System Based on Face Recognition System Using CNN-PCA Method and Real-time Camera”* penulis membuat aplikasi sistem kehadiran menggunakan algoritma CNN yang dikombinasikan dengan algoritma PCA. Pada sistem ini seluruh wajah dideteksi menggunakan CNN kemudian dilakukan pengenalan wajah menggunakan PCA. Dari penelitian yang dilakukan didapatkan tingkat akurasi sebesar 90%. Pada penelitian ini belum diimplementasikan penggunaan masker.

Pada penelitian dengan judul *“Identification of with Face Mask and without Face Mask using Face Recognition Model”* dilakukan pengenalan wajah bermasker dengan menggunakan algoritma CNN. Pada penelitian ini penulis melakukan deteksi terhadap penggunaan masker. Gambar yang ditampilkan dideteksi apakah merupakan wajah atau bukan, kemudian dilakukan deteksi wajah tersebut menggunakan masker atau tidak. Pada penelitian ini didapatkan akurasi sebesar 96.7% untuk mendeteksi wajah. Penelitian ini belum diimplementasikan pengenalan wajah.

Pada penelitian dengan judul *“Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition”* dilakukan deteksi wajah dengan menggunakan Viola-jones kemudian dilakukan pengenalan wajah dengan

menggunakan algoritma PCA. Dari penelitian yang dilakukan didapatkan tingkat akurasi pengenalan wajah tanpa masker sebesar 95% sedangkan jika menggunakan masker tingkat akurasinya turun menjadi 72%.

Pada penelitian dengan judul “Masked Face Recognition Using Convolutional Neural Network” dilakukan deteksi wajah menggunakan MTCNN kemudian dilakukan pengenalan wajah menggunakan Facenet. Dari penelitian yang dilakukan didapatkan tingkat akurasi pengenalan wajah tanpa masker sebesar 97% dan jika menggunakan masker tingkat akurasi menjadi 82%.

Dari penelitian-penelitian tersebut tingkat akurasi pengenalan wajah bermasker cukup rendah jika dibanding tanpa menggunakan masker. MTCNN merupakan algoritma dengan tingkat akurasi yang paling tinggi untuk pengenalan wajah bermasker. Pengenalan dengan seluruh wajah jika bermasker akan memberikan beberapa titik yang salah.

1.5 Tujuan Penelitian

Tujuan dari thesis ini adalah untuk

- Meningkatkan tingkat akurasi, presisi dan sensitivitas pengenalan wajah menggunakan MTCNN dan VGG Face dengan mendeteksi separuh wajah bagian atas.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Dapat dijadikan sarana informasi bagi masyarakat
2. Memberikan pengetahuan tentang pengenalan wajah

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang terdahulu dapat dijadikan salah satu acuan penulis dalam melakukan penelitian, sehingga peneliti dapat menambah wawasan dan teori terkait dengan penelitian yang sedang dilakukan. Dari penelitian yang terdahulu, penulis belum menemukan judul yang sama seperti judul yang diambil oleh penulis. Namun penulis menemukan beberapa penelitian yang dapat menjadi bahan acuan untuk penelitian ini.

Pada penelitian yang telah dilakukan oleh [1] dengan judul “Attendance System Based on Face Recognition System Using CNN-PCA Method and Real-time Camera” tahun 2019 tentang pengenalan wajah untuk daftar hadir, peneliti menggunakan CNN-PCA sebagai metode ekstraksi landmark wajah. Gambar wajah diambil menggunakan kamera secara realtime kemudian dilakukan preprocessing pada gambar untuk menghasilkan gambar wajah yang lebih tajam. Pada penelitian ini gambar wajah 2 dimensi akan direkonstruksi menggunakan CNN menjadi 3 dimensi. Setelah itu gambar wajah akan dilakukan ekstraksi menggunakan PCA untuk kemudian dilakukan perbandingan kesamaan dengan menggunakan metode mahalanobis. Dari penelitian ini didapatkan hasil akurasi yang cukup tinggi hingga 98% untuk pengenalan wajah.

Pada penelitian yang dilakukan oleh [2] dengan judul “Identification of with Face Mask and without Face Mask using Face Recognition Model” tahun 2021

tentang identifikasi wajah bermasker atau tidak bermasker, peneliti menggunakan CNN sebagai metode pengenalan wajah. Pada penelitian ini dilakukan deteksi penggunaan masker pada wajah. Penggunaan CNN untuk mengambil gambar wajah kemudian melakukan deteksi landmark wajah, untuk selanjutnya wajah yang bermasker dan tidak bermasker akan dibedakan kelasnya. Penelitian ini dilakukan karena wajah yang menggunakan masker akan mengurangi jumlah titik landmark yang dapat dideteksi. Dari penelitian ini didapatkan hasil 96,70% untuk membedakan wajah bermasker atau tanpa masker.

Pada penelitian yang dilakukan oleh [3] dengan judul “Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition” tahun 2019 tentang pengenalan wajah menggunakan PCA sebagai metode ekstraksi landmark wajah. Pada penelitian ini dilakukan pengenalan wajah yang menggunakan masker atau tidak menggunakan masker. Penggunaan dataset campuran antara orang bermasker dan tanpa masker yang didapatkan dari tahun 1992-1994 sedangkan untuk wajah menggunakan masker peneliti membuat dataset tersendiri dari dataset ORL tersebut. Pada penelitian tersebut menggunakan 300 foto sebagai dataset training dan 80 foto sebagai dataset test. Pengenalan wajah menggunakan PCA pada penelitian ini didapatkan hasil 95% untuk pengenalan wajah tanpa masker dan 72% untuk pengenalan wajah dengan masker. Penelitian ini membuktikan bahwa PCA memberikan hasil pengenalan yang rendah untuk wajah bermasker.

Pada penelitian yang dilakukan oleh [4] dengan judul “Masked Face Recognition Using Convolutional Neural Network” tahun 2019 tentang pengenalan

wajah menggunakan MTCNN sebagai metode deteksi wajah dan Facenet untuk ekstraksi landmark wajah. Pada penelitian ini MTCNN digunakan untuk mendeteksi wajah dan melakukan pemotongan untuk wajah yang terdeteksi. Kemudian dilakukan image postprocessing dan ekstraksi menggunakan embedding facenet. Dataset yang digunakan merupakan gambar orang yang tidak hanya menggunakan masker namun juga menggunakan kacamata hitam yang terdiri dari 990 gambar wajah dari 45 orang berbeda. Gambar-gambar tersebut kemudian dibagi untuk menjadi dataset training dan dataset test. Dari test yang dilakukan didapatkan hasil rata-rata 82% untuk pengenalan wajah bermasker.

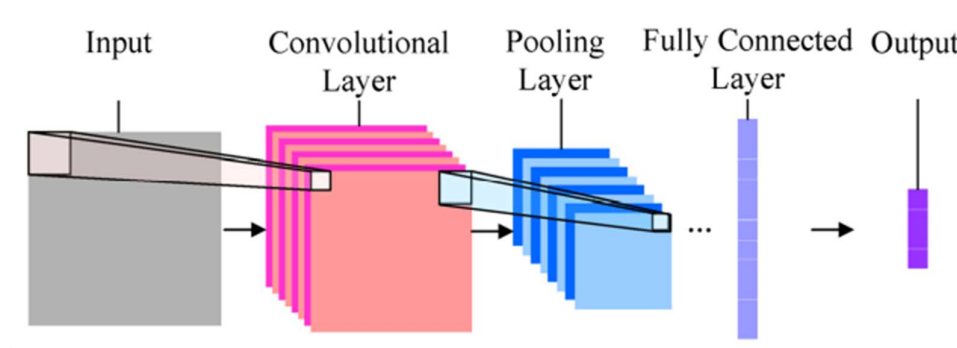
Dari penelitian-penelitian yang telah penulis sebutkan belum ada metode atau algoritma untuk pengenalan wajah bermasker dengan tingkat akurasi yang cukup tinggi. Penelitian yang ada menggunakan seluruh wajah untuk pengenalan wajah, hal ini akan memberikan titik-titik landmark wajah yang salah. Oleh karena itu penulis melakukan penelitian dengan judul “PENGENALAN WAJAH BERMASKER DAN TANPA MASKER DENGAN MENGGUNAKAN MTCNN DAN VGG FACE UNTUK WAJAH BAGIAN ATAS”. Metode yang digunakan adalah MTCNN dan VGG Face. Sedangkan wajah yang dideteksi untuk di ekstrak landmarknya adalah wajah bagian atas. Dataset yang digunakan merupakan campuran wajah bermasker dan tidak bermasker.

2.2 Landasan Teori

Landasan teori dituliskan berdasar tinjauan pustaka, sebagai bentuk yang lebih spesifik sesuai dengan arah penelitian Anda. Landasan teori ini didapat dari paper maupun buku, yang mendasari metodologi penelitian yang dibahas di Bab III.

2.2.1 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan pengembangan Multilayer Perceptron (MLP) yang merupakan salah satu neural network bertipe feed forward (bukan berulang). CNN merupakan sebuah neural network yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra [5]. CNN digunakan untuk menganalisis gambar visual, mendeteksi dan mengenali objek pada image, yang merupakan vektor berdimensi tinggi yang akan melibatkan banyak parameter untuk mencirikan jaringan. Secara garis besar, CNN tidak terlalu jauh berbeda dengan neural network biasanya. CNN terdiri dari neuron yang memiliki weight, bias dan activation function. Arsitektur CNN dapat dilihat pada gambar 2.1.



Gambar 2.1. Arsitektur CNN [6]

Dari gambar 2.1 dapat dilihat bahwa CNN memiliki tiga layer yaitu :

1. Convolution Layer Convolution. Layer ini melakukan proses konvolusi menggunakan sebuah filter. Filter ini diinisialisasi dengan menggunakan nilai tertentu, dan nilai dari filter inilah yang menjadi parameter yang akan di-update dalam proses learning.

2. Pooling layer Pooling layer. Pada layer ini dimensi data akan dikurangi.

Prinsipnya pooling layer terdiri dari filter dengan ukuran tertentu. Tujuan dari penggunaan pooling layer adalah mengurangi dimensi dari feature map, sehingga mempercepat komputasi.

3. Fully connected layer. Pada fully connected layer mengambil input dari hasil output pooling layer yang berupa feature map. Feature map tersebut masih berbentuk multidimensional array. Oleh karena itu, lapisan ini akan melakukan reshape feature map dan menghasilkan vektor sebanyak n -dimensi, n adalah jumlah kelas output yang harus dipilih program.

Jaringan saraf convolutional juga berkontribusi pada pengenalan wajah dengan menyediakan pengklasifikasi yang kuat [7]. Selain dapat digunakan pada citra, CNN juga dapat digunakan untuk pengenalan suara. Pada penelitian ini akan digunakan salah satu model convolutional neural network (CNN) yaitu VGG Face karena memiliki tingkat presisi yang tinggi [8]. Dan untuk pengenalan wajah akan digunakan MultiTask Cascade Convolutional Neural Network (MTCNN).

2.2.2 *Multi-task Cascaded Convolutional Neural Network (MTCNN)*

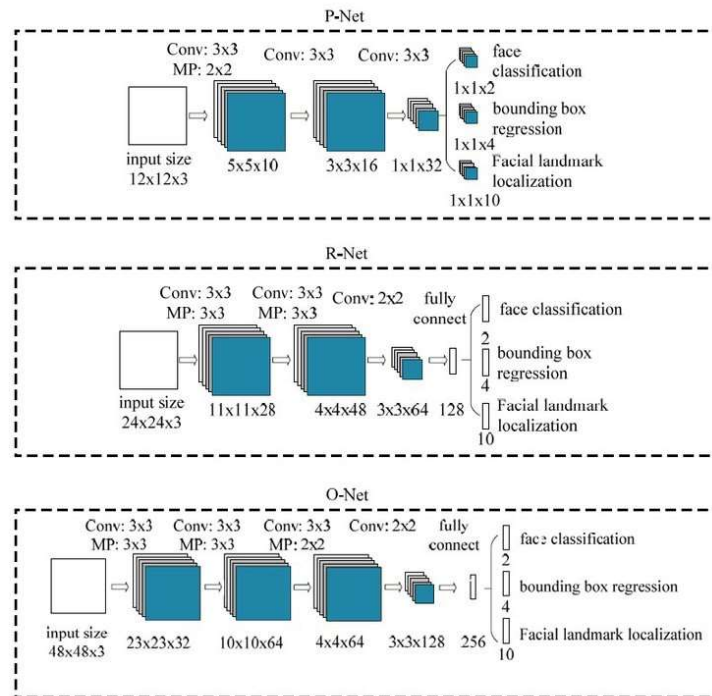
Deteksi wajah penting dilakukan saat melakukan pengenalan wajah. Jika tidak ada wajah yang terdeteksi, sistem pengenalan wajah tidak dapat melanjutkan pengenalannya. Pada penelitian ini digunakan Multi-task Cascaded Convolutional Neural Network (MTCNN) untuk melakukan pendeteksian wajah. MTCNN adalah model untuk mendeteksi wajah. MTCNN digunakan untuk mendeteksi wajah dari gambar yang diberikan dan kemudian menghasilkan deskriptor wajah berdimensi tinggi. Ada 3 tahapan pendeteksi jaringan saraf pada MTCNN. Sebelum tahap

pertama dimulai, gambar yang diberikan diubah ukurannya berkali-kali sehingga wajah dengan ukuran berbeda dapat dideteksi.

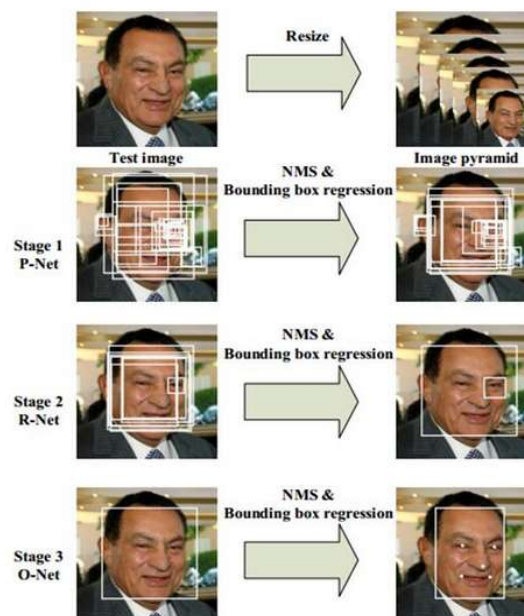
Pada tahap pertama yang disebut P-Net, dilakukan pendeteksian pertama. deteksi pertama ini memiliki tingkat akurasi yang rendah sehingga masih banyak kotak pembatas yang salah. Hasil propagasi dari P-Net adalah vector 32 dimensi yang digunakan untuk mengetahui apakah sel 12 x 12 terdapat wajah atau tidak. Jika terdeteksi wajah maka akan dibuat kotak pembatas. Kotak dengan area yang tertutupi terbesar akan dihapus untuk tahap selanjutnya.

Pada tahap kedua, yang disebut R-Net, dilakukan seleksi terhadap hasil dari tahap pertama, untuk mendapatkan kotak pembatas yang lebih akurat. Dari tahap ini sel 24 x 24 akan di tingkatkan menjadi 48 x 48 untuk di gunakan pada tahap selanjutnya.

O-Net, hasil dari tahap sebelumnya disempurnakan kembali sehingga mendapatkan kotak pembatas yang lebih akurat dari sebelumnya. Ketiga tahapan tersebut dapat melakukan klasifikasi wajah, regresi bounding box dan mendeteksi landmark wajah seperti mata, hidung, dan mulut. Arsitektur dari MTCNN dapat dilihat pada Gambar 2.2 dan visualisasi langkah dalam MTCNN dapat dilihat pada Gambar 2.3



Gambar 2.2. Arsitektur MTCNN



Gambar 2.3. Pipeline MTCNN [9]

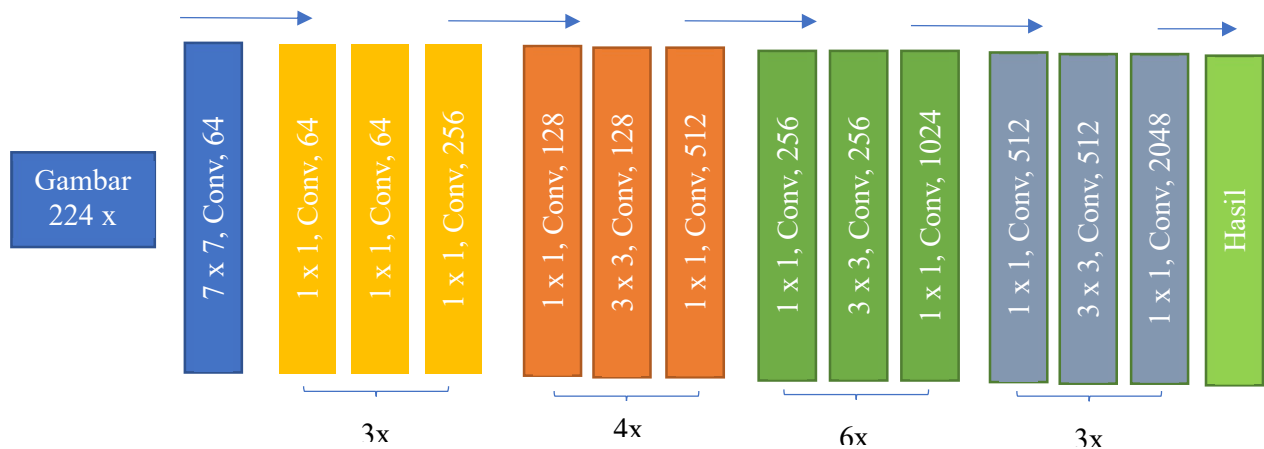
MTCNN dipilih karena memiliki tingkat akurasi yang tinggi untuk pendeteksian wajah [10] dan efisiensi dalam pengenalan citra berbasis pipeline [11]. Selain itu, MTCNN juga dapat mendeteksi wajah meskipun sebagian wajah tertutup tangan atau benda lain [12]. Hal ini sangat berguna untuk mendeteksi dan mengenali wajah menggunakan masker. MTCNN akan mendeteksi kedua mata, mulut pada kedua ujung dan hidung.

Setelah deteksi wajah selesai, langkah selanjutnya adalah memotong dan mengatur ukuran gambar dari hasil MTCNN. Kotak pembatas yang dihasilkan dari MTCNN digunakan untuk memotong wajah gambar dari keseluruhan gambar. Sebelum gambar dapat digunakan untuk langkah selanjutnya yaitu gambar Face Recognition perlu diubah ukurannya terlebih dahulu. Untuk dapat digunakan pada VGGFace, diperlukan gambar dengan ukuran 224 x 224 piksel. Untuk alasan ini, gambar dipotong kemudian diubah ukurannya sesuai dengan kebutuhan ini.

2.2.3 VGG-Face

VGGFace adalah model yang dikembangkan untuk pengenalan wajah. Ada beberapa model di VGGFace, yaitu VGG16, Resnet50, dan Senet50. Dalam penelitian ini, Resnet50 (Residual Network) akan digunakan untuk melakukan ekstraksi landmark. Resnet50 dipilih karena dibandingkan dengan model lainnya, Resnet50 memiliki 3 layer bottleneck block yang dapat meningkatkan akurasi dan meminimalkan waktu untuk training data [13]. Resnet50 akan melakukan iterasi dengan 50 layer. Hasil dari iterasi ini berupa average pooling 1x1x2048. Average pooling inilah yang digunakan sebagai face embedding. Kemudian face embedding diprediksi dari model yang diberikan dalam bentuk vektor. Kemudian panjang

vektor tersebut dinormalisasi menggunakan normalisasi vektor L2. Ini disebut sebagai 'deskriptor wajah'. Jarak antara deskriptor wajah dihitung menggunakan persamaan cosinus [5]. Kemudian face embedding disimpan untuk digunakan sebagai pembanding. Face embedding merupakan hasil dari arsitektur Resnet50 yang dapat dilihat pada Gambar 2.4.



Gambar 2.4. Arsitektur Resnet50

2.3 Hipotesis

Hipotesis dari penelitian ini adalah

1. Adanya peningkatan akurasi, presisi, dan sensitifitas pada pengenalan wajah menggunakan algoritma MTCNN dan VGG Face dengan hanya menggunakan wajah bagian atas.

BAB III METODOLOGI

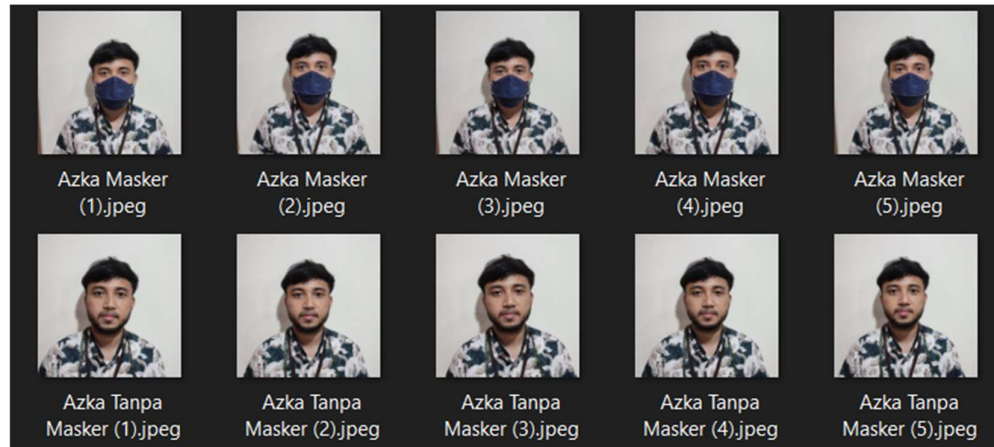
3.1 Alat dan Bahan

3.1.1 Alat

- Laptop Lenovo Ideapad Slim 5i, Intel Core i5, MX450, dan ram 8 Gb sebagai alat utama pengembangan system
- Windows 10 Education dan Ubuntu
- Logitech Webcam 720p
- Xiaomi Redmi Note 10 sebagai alat pengambilan dataset

3.1.2 Dataset

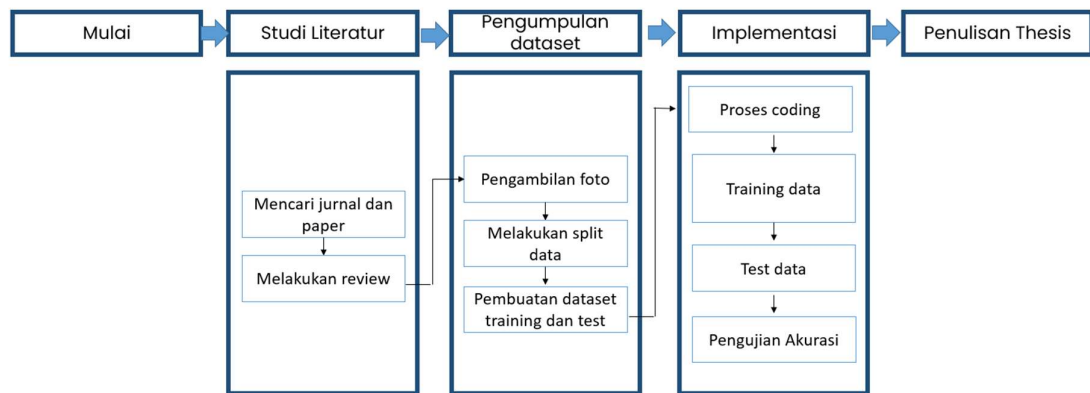
Pada Penelitian ini menggunakan dataset yang didapatkan dengan cara mengambil foto beberapa orang yang menghadap kedepan. Foto diambil dari bagian pundak hingga bagian atas kepala. Foto yang diambil merupakan foto wajah dengan menggunakan masker dan tanpa masker. Foto diambil dari 15 orang berbeda, masing-masing dengan 13 foto dengan masker dan 13 foto tanpa masker, dengan total 390 foto. Selanjutnya foto-foto tersebut dibagi untuk membuat test dataset dan training dataset. Test dataset berisi 3 foto tanpa masker dan 3 foto dengan masker untuk setiap orang. Sedangkan dataset training menggunakan 5 foto tanpa masker dan 5 foto dengan masker untuk setiap orang. Contoh dataset yang digunakan dapat dilihat pada Gambar 3.1



Gambar 3.1. Dataset

3.2 Jalannya Penelitian

Jalannya penelitian telah dilakukan sesuai diagram alir berikut



Gambar 3.2. Diagram Jalannya Penelitian

3.2.1 Studi Literatur

Pada tahap ini penulis melakukan pembelajaran melalui jurnal, paper, dan buku terkait thesis yang akan diambil. Jurnal dan paper yang digunakan merupakan jurnal terbaru dengan maksimal tahun pada 2018 untuk mendapatkan referensi yang

terkini. Pada tahap ini penulis mencatat hal-hal penting sesuai dengan topik yaitu face recognition. Mempelajari tentang Neural Network dan pengaplikasiannya pada face recognition. Selain itu penulis juga membuat catatan terkait penelitian-penelitian yang telah dilakukan sebelumnya. Penelitian yang menggunakan metode-metode berbeda satu sama lain untuk pengenalan wajah.

3.2.2 Pengumpulan Dataset

Pada tahap ini penulis melakukan pengumpulan dataset. Dataset yang digunakan merupakan dataset yang diambil sendiri oleh penulis. Dataset berisi wajah orang menghadap depan dengan menggunakan masker dan tanpa masker yang diambil dari 15 orang berbeda. Setiap orang diambil 13 foto dengan masker dan 13 foto tanpa masker sehingga mendapatkan total 390 foto wajah. Foto yang diambil merupakan foto dari pundak hingga bagian atas kepala menyisakan sedikit ruang di atas rambut. Foto-foto tersebut berbentuk kota atau dengan perbandingan sisi 1:1 dengan orientasi portrait. Foto-foto tersebut kemudian dimasukkan ke dalam folder dengan nama folder sesuai dengan nama orang yang difoto. Foto-foto tersebut diambil dengan izin pemilik dan digunakan sesuai izin pemilik untuk pengerjaan thesis dan penelitian penulis. Setelah dilakukan pengumpulan dataset dan foto-foto telah dikelompokkan maka penulis selanjutnya melakukan split data untuk membuat dataset training dan dataset test.

3.2.4 Implementasi

Pada tahap ini penulis melakukan implementasi dengan caramelakukan coding untuk training dan test, melakukan pengujian dengan menggunakan test dataset, dan terakhir melakukan analisis terhadap hasil yang didapatkan. Pada tahap

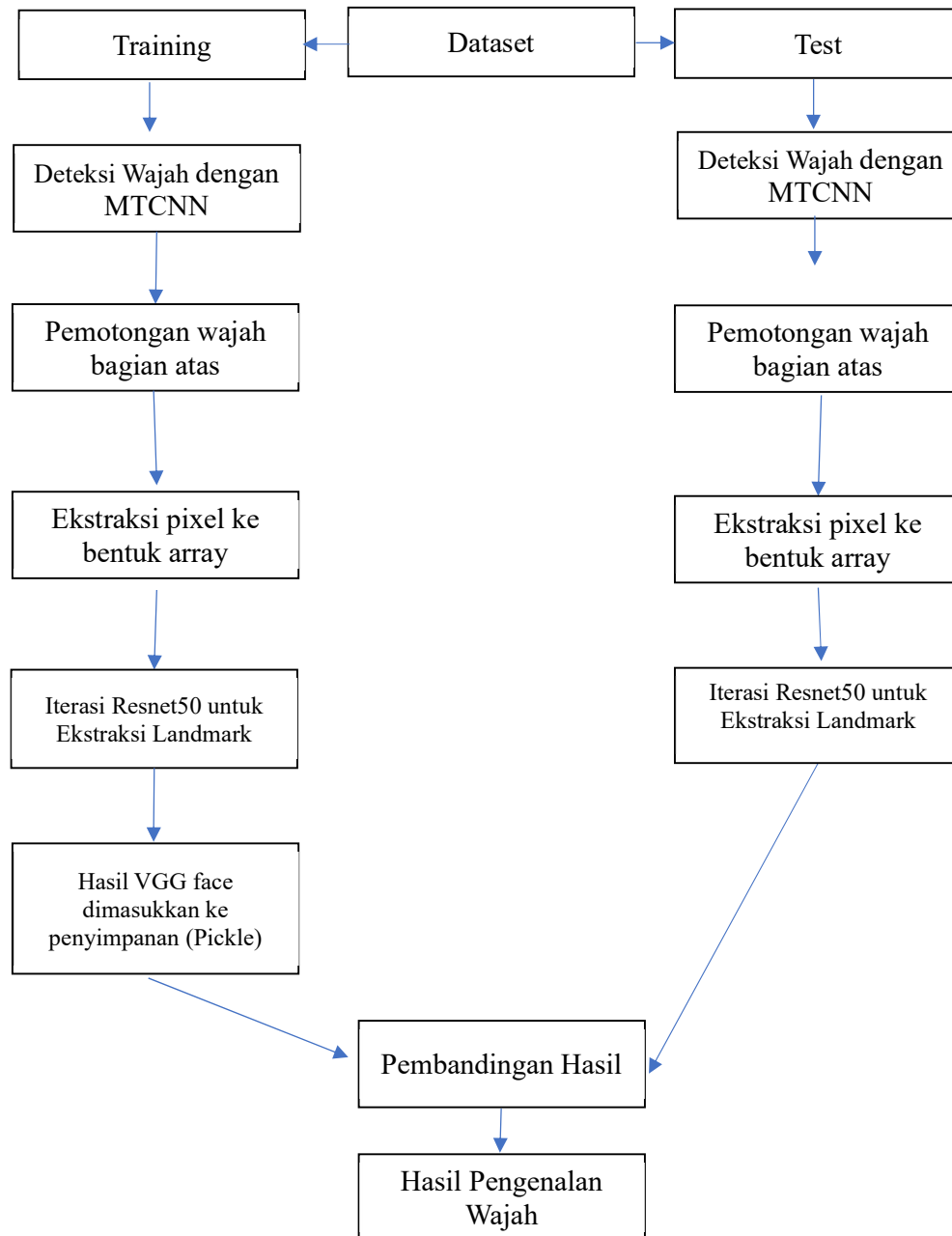
ini penulis membuat kode pertama untuk melakukan training dataset. Kode dibuat dengan Bahasa pemrograman python. Kode ini digunakan untuk mengolah dataset training menjadi embedding yang akan digunakan pada penelitian ini. Pada kode ini terdapat beberapa blok yaitu blok deteksi dan pemotongan wajah dan blok ekstraksi landmark. Hasil akhir dari kode training ini adalah berupa data pickle. Proses training data ini akan memasukkan embbeding-embedding menghasilkan data pickle yang akan digunakan untuk proses selanjutnya. Pembuatan kode selanjutnya adalah untuk melakukan perbandingan antara dataset test dan dataset training. Pada kode ini terdapat blok-blok yang sama yaitu blok deteksi dan pemotongan wajah dan blok ekstraksi landmark namun dengan tambahan blok perbandingan. Pada implementasi akan dilakukan perbandingan antara dataset training yang ada di data pickle dan dataset test. Setelah itu akan dilakukan analisis dan pengujian akurasi dengan menggunakan convolutional matrix. Sehingga didapatkan hasil penelitian yang berupa akurasi, presisi, dan sensitivity. Selain itu pada tahap ini penulis juga melakukan percobaan menggunakan algoritma lain sebagai perbandingan.

3.2.5 Penulisan Thesis

Setelah melaksanakan penelitian dan mendapatkan hasil. Pada tahap ini penulis membuat thesis dari hasil tahap-tahap sebelumnya. Thesis berisi latar belakang, rumusan masalah, tinjauan Pustaka, Metodologi, Hasil Pembahasan, dan kesimpulan. Penulisan thesis ini juga termasuk masukan dari pembimbing dan penguji saat dilakukan sidang.

3.3 Perancangan Sistem

Blok penjelasan perancangan sistem ada pada grafik berikut



Gambar 3.3. Diagram Alir Perancangan Sistem

3.3.1 Blok Deteksi Wajah dengan MTCNN

Pada blok sistem ini terdiri dari proses deteksi wajah, pemberian bounding box. Pada blok sistem ini, proses pertama yang dilakukan adalah foto yang ada di dalam dataset akan dideteksi menggunakan MTCNN. Gambar yang digunakan merupakan gambar dengan ukuran skala 1:1. Ukuran gambar yang akan dideteksi adalah 1080x1080 pixel yang mana diambil menggunakan smartphone. Pada proses ini MTCNN akan menentukan gambar yang diberikan terdapat gambar wajah atau tidak di dalamnya. Proses ini merupakan proses *face classification* yang menggunakan binary classification dengan cross entropy loss. Pada proses ini akan dideteksi bagian mata, hidung, dan mulut untuk menentukan apakah gambar tersebut merupakan gambar wajah atau bukan. Setelah dilakukan face classification maka proses selanjutnya adalah pemberian bounding box dengan bounding box regression. Kedua proses ini akan dilakukan berulang hingga 3 kali dengan perbedaan input di ketiga stage tersebut. Stage yang pertama dengan menggunakan input berupa 12x12x3, stage pertama inilah yang disebut dengan P-Net. Keluaran dari stage ini berupa jumlah bounding box yang ada yang akan digunakan untuk stage berikutnya. Pada stage kedua dengan menggunakan input 24x24x3, stage kedua inilah yang disebut dengan R-Net. Keluaran dari stage ini berupa jumlah bounding box yang ada yang akan digunakan untuk stage berikutnya. Pada stage ketiga atau terakhir ini menggunakan input 48x48x3, stage ketiga inilah yang disebut dengan O-Net. Keluaran dari stage ini adalah gambar wajah yang sudah terdeteksi dan sudah diberikan kotak pembatas. Contoh masukan dan keluaran pada blok ini dapat dilihat pada Gambar 3.4.



Gambar 3.4. Masukan dan keluaran Blok Deteksi Wajah

3.3.2 Blok Pemotongan Wajah

Setelah dilakukan deteksi wajah dan didapatkan gambar wajah yang telah dideteksi bagian mata, hidung, dan mulutnya serta telah diberikan kotak pembatas maka selanjutnya adalah memotong gambar tersebut hingga didapatkan wajah bagian atas saja. Pemotongan wajah ini dilakukan dengan masukan berupa gambar yang dimana wajah telah terdeteksi. Kemudian gambar wajah diambil sesuai dengan kotak pembatas yang telah dibuat sehingga yang digunakan untuk proses ini hanyalah bagian wajah saja tidak termasuk dengan latar belakang foto dimana wajah tidak terdeteksi. Setelah diambil wajah sesuai dengan kotak pembatas maka gambar tersebut dipotong untuk mendapatkan wajah bagian atas saja. Wajah yang dipotong mulai dari bagian atas hidung hingga bagian dahi. Sehingga yang didapatkan adalah bagian mata, alis, dan dahi. Sistem akan melakukan pemotongan dengan menggunakan persamaan 3-1.

$$h_1 = h_0 \times 0,5 \quad (3-1)$$

Dari persamaan di atas yaitu h_0 yang merupakan tinggi awal atau tinggi seluruh wajah yang didapatkan dari proses sebelumnya yaitu proses deteksi wajah. Tinggi awal ini merupakan tinggi kotak pembatas bukan tinggi gambar keseluruhan. Sehingga tinggi yang digunakan merupakan tinggi wajah yang terdeteksi. Tinggi wajah atau tinggi awal yang digunakan dengan besaran pixel. Tinggi awal antara gambar satu dan lainnya akan berbeda tergantung dengan wajah yang terdeteksi. Untuk mendapatkan setengah wajah bagian atas (h_1) maka tinggi awal tersebut harus dibagi dengan nilai yang sesuai, Dari penelitian yang telah dilakukan nilai terbaik yang digunakan adalah 0,5. Tinggi awal tersebut akan dikalikan dengan 0,5 untuk mendapatkan separuh wajah bagian atas. Pemotongan ini dimulai dari titik bawah bagian dari kotak pembatas yang merupakan masukan dan bukan dari bagian atas. Hal ini untuk memastikan bahwa yang terpotong adalah bagian bawah bukan bagian atas. Dan juga untuk memastikan hanya sedikit bagian masker atau sama sekali tidak ada yang masuk ke dalam hasil pemotongan ini. Dari proses pemotongan ini akan menghasilkan separuh wajah bagian atas yang terpilih. Tepatnya adalah mulai dari bagian atas hidung hingga batas antara jidat dan rambut. Keluaran dari proses ini adalah gambar separuh wajah bagian atas yaitu bagian atas hidung hingga batas antara jidat dan rambut dari wajah yang terdeteksi sebelumnya. Hasil dari pemotongan ini ukurannya masih tidak menentu karena perbedaan masukan dari setiap gambar wajah yang terdeteksi. Ukuran-ukuran ini harus diseragamkan agar dapat digunakan pada proses selanjutnya. Hasil keluaran dari proses sebelumnya dimana ukurannya masih tidak sesuai dengan kebutuhan maka

akan disesuaikan dengan ukuran yang sesuai yaitu ukuran 224x224 pixel. Pada proses perubahan ukuran ini hasil dari proses pemotongan akan diubah ukurannya menjadi menjadi 224x224 pixel. Contoh masukan dan keluaran pada Blok pemotongan wajah dapat dilihat pada Gambar 3.5.



Gambar 3.5. Masukan dan keluaran Blok Pemotongan Wajah

3.3.3 Blok Ekstraksi Pixel ke dalam Array

Setelah gambar dideteksi, dipotong, dan disesuaikan ukurannya maka proses terakhir untuk mendapatkan keluaran adalah merupah gambar tersebut menjadi array yang akan digunakan untuk proses pada blok selanjutnya. Gambar yang masih berupa image akan diubah menjadi array dengan fungsi yang ada di dalam python. Sehingga keluaran akhir dari blok deteksi dan pemotongan wajah ini adalah array yang berisi konversi dari gambar wajah yang dideteksi kemudian dipotong untuk wajah bagian atas saja. Keluaran akhir ini yang kemudian akan digunakan untuk proses pada blok selanjutnya yaitu blok ekstraksi landmark.

3.3.4 Blok Ekstraksi Landmark

Pada blok sistem ini, hasil keluaran dari blok sebelumnya yaitu blok deteksi dan pemotongan wajah yang berupa array yang berisi konversi dari gambar wajah yang dideteksi kemudian dipotong untuk wajah bagian atas akan dilakukan

ekstraksi landmark. Pada blok ekstraksi landmark ini menggunakan VGG Face. Karena hasil ekstrak wajah hanya menghasilkan setengah dari wajah, maka bagian yang diekstrak oleh VGG Face terutamanya pada bagian mata, alis, dahi, dan Sebagian hidung. Pada VGG Face ini digunakan model Resnet50 yang dinilai cukup baik dalam tingkat pengenalan wajah. Resnet50 akan melakukan iterasi dengan menggunakan pretrained model. Masukan untuk proses ini berupa array berisi konversi dari gambar wajah yang dideteksi kemudian dipotong untuk wajah bagian atas dengan ukuran 224x224. Untuk mendapatkan hasil terbaik maka ukuran yang digunakan merupakan ukuran maksimal dari Resnet-50 yaitu 224x224. Resnet-50 akan melakukan iterasi sebanyak 50 layer.

Pada iterasi pertama Resnet-50 menggunakan masukan dengan merubah image menjadi 7x7 conv 64 dimana akan digunakan untuk proses iterasi selanjutnya. Pada iterasi kedua terdapat satu paket iterasi yaitu dengan masukan berupa kernel ukuran 64, 64, dan 256. Iterasi kedua ini akan diulang sebanyak tiga kali. Pada iterasi selanjutnya dengan masukan berupa kernel dengan ukuran 128, 128, dan 512 yang akan dilakukan iterasi sebanyak empat kali. Pada iterasi selanjutnya digunakan masukan berupa kernel dengan ukuran 256, 256, dan 1024 dan dilakukan iterasi sebanyak enam kali. Untuk paket iterasi terakhir dengan masukan berupa kernel 512, 512 dan 2048 yang akan dilakukan iterasi sebanyak tiga kali. Setelah Resnet50 melakukan seluruh iterasinya sebanyak 50 layer maka keluarannya berupa array yang berisi embedding dari gambar separuh wajah bagian atas yang telah dideteksi dan lakukan pemotongan pada blok sebelumnya yaitu blok deteksi dan pemotongan wajah. Contoh hasil masukan dan keluaran dari blok ekstraksi

landmak dapat dilihat pada Gambar 3.6.



Gambar 3.6. Masukan dan keluaran Blok Ekstraksi Landmark

3.3.5 Blok Data Pickle

Setelah didapatkan array berisi embedding tersebut maka ada perbedaan sesuai dengan dataset yang digunakan. Pada dataset training, array yang berisi embedding dari gambar separuh wajah bagian atas yang telah dideteksi dan lakukan pemotongan tersebut akan dikelompokkan sesuai dengan kelasnya. Kelas dalam hal ini adalah nama dari pemilik wajah yang terdeteksi pada gambar yang diberikan. Masing-masing embedding dari wajah-wajah yang diberikan akan dikelompokkan berdasarkan kelas ke dalam list pickle. Sehingga keluaran dari alur dataset training pada blok ekstraksi landmark ini adalah list pickle yang berisi kelas-kelas nama pemilik wajah yang didalamnya terdapat embedding hasil dari iterasi VGG Face model Resnet50 yang telah dilakukan. List pickle inilah yang akan digunakan sebagai pembandingan pada blok selanjutnya yaitu blok pembandingan.

Sedangkan pada alur dataset test pada blok ekstraksi landmark ini, hasil dari iterasi Resnet50 dari masukan dataset test yang diberikan tidak dimasukkan ke

dalam list pickle. Keluaran iterasi yang berupa array berisi embedding separuh wajah bagian atas yang telah dideteksi dan lakukan pemotongan akan langsung digunakan pada blok selanjutnya yaitu blok pembandingan. Kedua keluaran ini akan digunakan dan dibandingkan pada blok selanjutnya yaitu blok pembandingan. Sehingga keluaran dari blok ekstraksi landmark ini adalah berupa list pickle dataset training dan embedding dari dataset test.

3.3.6 Blok Pembandingan

Pada blok pembandingan ini, hasil dari blok sebelumnya yaitu blok ekstraksi landmark yang berupa list pickle yang berisi embedding separuh wajah bagian atas yang telah dideteksi dan dilakukan pemotongan dari dataset training dan embbeding dari dataset test akan dibandingkan. Proses pertama pada blok pembandingan ini adalah mengambil masukan berupa embbeding gambar dari dataset test yang telah melalui blok deteksi dan pemotongan wajah dan blok ekstraksi landmark. Embedding yang berupa array dari separuh wajah bagian atas yang telah dideteksi dan dilakukan pemotongan dari sebuah gambar dataset test akan dibandingkan dengan embedding-embedding yang ada dalam list pickle dari dataset training. Akan dilakukan iterasi sebanyak embbeding yang ada di dalam list data pickle. Hal ini untuk memastikan bahwa pembandingan dilakukan untuk setiap embbeding hasil dari dataset training.

Iterasi dilakukan secara berulang untuk mendapatkan nilai pembandingan dari hasil pembandingan embbeding yang ada dalam list pickle dan embedding dari dataset test. Pada blok ini digunakan batas atau threshold yang digunakan untuk membatasi hasil pembandingan yang kurang sesuai atau dengan nilai yang kecil.

Batas atau threshold ini juga digunakan untuk mendapatkan nilai pembandingan embedding yang ada di list pickle dan embedding dari dataset test dengan nilai yang tinggi atau tingkat persamaan yang tinggi. Batas yang digunakan pada blok ini dengan menggunakan nilai 0.5 sehingga nilai hasil pembandingan di bawah 0.5 tidak akan digunakan. Sehingga keluaran dari blok pembandingan ini adalah nilai hasil dari pembandingan antara embedding yang ada di list pickle yang merupakan hasil dari dataset training dan embedding hasil dari dataset test. Selain itu keluaran dari blok ini juga berupa kelas atau nama dari pemilik wajah dengan nilai hasil pembandingan yang paling tinggi.

3.4 Cara Analisis

Hasil yang didapatkan dari tahap-tahap di atas, kemudian akan dianalisis menggunakan metode confusion matriks. Poin-poin yang akan didapatkan adalah tingkat akurasi, tingkat presisi, dan tingkat sensitifitasnya. Poin-poin tersebut didapatkan dengan menggunakan tabel confusin matriks yang berisi beberapa kelas sesuai dengan jumlah orang yang ada dalam dataset. Tingkat akurasi, tingkat presisi dan tingkat sensitifitas akan didapatkan dengan menggunakan persamaan (3-1)(3-2) dan (3-3)

$$Accuracy = \frac{TP}{Total\ Test\ Dataset} \quad (3-1)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (3-2)$$

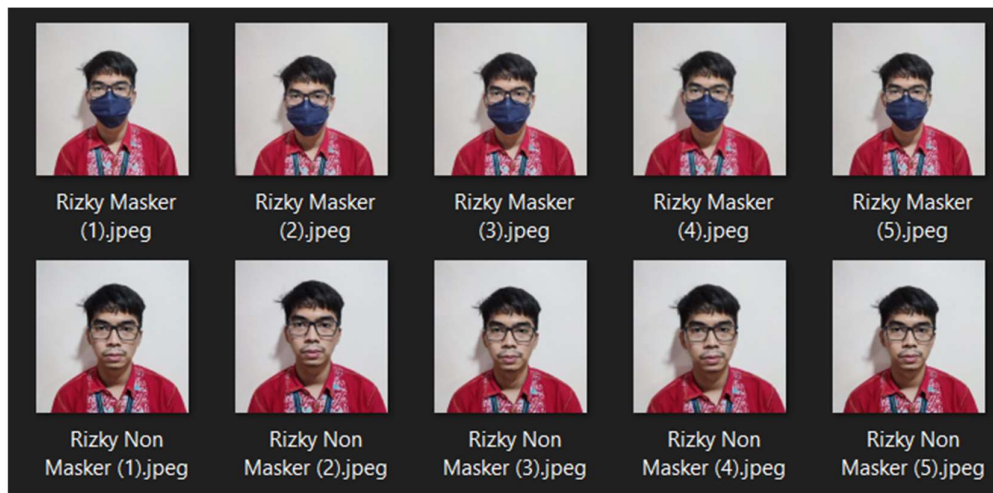
$$Sensitivity = \frac{TP}{(TP+FN)} \quad (3-3)$$

TP (True Positive) adalah gambar yang diprediksi benar. FN (False Negatif) adalah jumlah gambar orang yang diberikan prediksi oleh orang lain. Sedangkan FP (False Positif) adalah jumlah gambar orang yang diprediksi sebagai orang lain. Tingkat akurasi didapatkan dengan cara membagi nilai true positive dengan total test dataset. Tingkat presisi didapat dengan cara membagi true positive dengan hasil tambah antara true positive dan false positive. Selanjutnya tingkat sensitifitas didapatkan dengan cara membagi true positive dengan hasil tambah antara true positive dan false negative. Kemudian hasil tersebut akan dibandingkan dengan tingkat akurasi, presisi dan sensitifitas dari pengenalan wajah yang menggunakan seluruh bagian wajah.

BAB IV

HASIL DAN PEMBAHASAN

Dari pengumpulan dataset oleh penulis didapatkan foto wajah dari 15 orang berbeda. Masing-masing orang diambil foto menghadap depan sebanyak 13 foto bermasker dan 13 foto tidak bermasker. Dengan total foto sejumlah 390 foto wajah. Dari 390 foto wajah diambil 90 foto wajah untuk dijadikan test dataset. Masing-masing orang diambil 3 foto bermasker dan 3 foto tanpa masker. 300 foto yang bukan test dataset dimasukkan ke dalam folder yang diberinama sesuai pemilik foto dengan rincian masing-masing orang 5 foto bermasker dan 5 foto tanpa masker. Sama halnya dengan 90 foto test dataset dimasukkan pula ke dalam folder sesuai dengan nama. Maka didapatkan 2 folder berbeda yang masing-masing berisi 15 folder nama orang. Contoh isi dari dataset training dan dataset test training dapat dilihat pada gambar 4.1 dan 4.2



Gambar 4.1. Dataset Training



Gambar 4.2. Dataset Test

4.1 Hasil Sistem

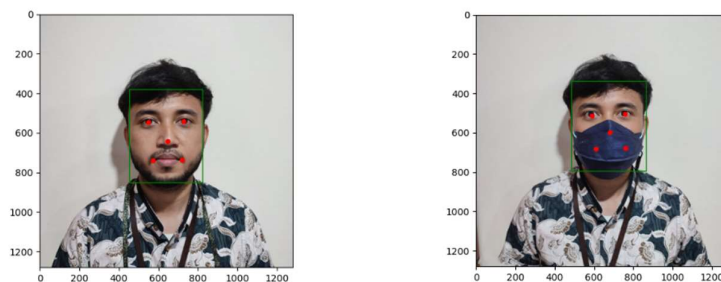
Dari sistem yang sudah dibuat menjadi beberapa blok maka didapatkan hasil sebagai berikut :

4.1.1 Hasil Blok Deteksi Wajah

Pada blok deteksi dan pemotongan wajah yang dilakukan menggunakan MTCNN. Langkah pertama adalah melakukan deteksi wajah pada gambar yang diberikan. MTCNN melakukan proses stage P-Net, R-Net, dan O-Net untuk mendapatkan hasil deteksi wajah dengan titik muka berupa mata, hidung, dan mulut. Pada langkah ini wajah orang dengan masker tetap akan terdeteksi karena MTCNN akan memberikan prediksi letak titik mulut dan hidung. Prediksi titik mulut dan hidung ini cukup berbeda dari wajah yang tanpa masker. MTCNN cukup baik untuk mendeteksi wajah karena dari semua gambar yang diberikan pada dataset training maupun dataset test ke semuanya dapat terdeteksi wajah. Penulis juga telah mencoba untuk memberikan gambar masukan berupa gambar wajah yang dipotong, wajah tidak terdeteksi dan terjadi error. Hal ini terjadi karena sistem MTCNN tidak

dapat memberikan titik di luar batas gambar yang diberikan. Penulis juga mengalami kendala jika foto diambil dalam keadaan vertical atau kamera miring. Hal ini terjadi karena ukuran gambar yang tidak sesuai dimana lebar lebih besar dibanding tinggi sehingga menimbulkan error pada sistem.

Setelah wajah terdeteksi maka sistem akan memberikan bounding box atau kotak pembatas pada bagian wajah yang terdeteksi. Kotak pembatas ini yang akan digunakan pada langkah berikutnya. Gambar wajah di dalam kotak pembatas akan diambil untuk dilakukan pemotongan. Hasil dari deteksi wajah dapat dilihat pada Gambar 4.3.



Gambar 4.3. Hasil Deteksi Wajah

4.1.2 Hasil Blok Pemotongan Wajah

Dari hasil deteksi wajah pada gambar yang diberikan dihasilkan bahwa seluruh gambar dalam dataset training maupun dataset test berhasil untuk dideteksi. Selanjutnya gambar dengan bounding box atau kotak pembatas tersebut akan dilakukan pemotongan. Pemotongan dilakukan terhadap gambar wajah yang ada di dalam kotak pembatas. Dari seluruh gambar wajah dengan menggunakan masker yang dilakukan pemotongan, didapatkan bahwa hasil dari pemotongan dengan nilai

potong 0,5 akan termasuk masker di dalamnya. Rasio masker ini berkisar 10%-15% dari ukuran gambar setelah dilakukan pemotongan. Rasio ini didapatkan karena perbedaan dalam penggunaan masker. Ada orang yang menggunakan masker hingga pangkal hidung, ada juga yang menggunakan hanya hingga pertengahan hidung.

Setelah dilakukan pemotongan, gambar separuh wajah bagian atas ini ukurannya masih berbeda satu sama lain, hal ini terjadi karena ukuran wajah tiap orang berbeda sehingga hasil potongnya pun berbeda ukurannya. Gambar yang sudah dipotong kemudian disesuaikan ukurannya menjadi 224 x 224 pixel. Hasil dari pemotongan dan penyesuaian ukuran gambar ini dapat dilihat pada Gambar 4.4.



Gambar 4.4. Hasil Potongan

4.1.3 Hasil Blok Ekstraksi Pixel

Setelah gambar dipotong dan disesuaikan ukurannya, gambar kemudian diubah atau dikonversi menjadi array dengan menggunakan fitur yang ada di dalam python. Hasil dari konversi ini yang akan digunakan pada blok selanjutnya. Contoh hasil dari ekstraksi landmark ada pada Gambar 4.5

```
108]
[[[208 208 208]
 [199 199 199]
 [210 210 210]
 ...
 [ 72  74  73]
 [136 138 137]
 [177 179 177]]

[[[207 207 207]
 [207 207 207]
 [206 206 206]
 ...
 [ 69  71  70]
 [ 83  85  84]
 [119 121 118]]
```

Gambar 4.5. Hasil Ekstraksi Pixel

4.1.4 Hasil Blok Ekstraksi Landmark

Pada blok ekstraksi landmark wajah ini langkah pertama yang dilakukan adalah melakukan ekstraksi landmark menggunakan VGG Face dalam hal ini menggunakan model Resnet50. Dari array yang merupakan hasil dari konversi gambar wajah yang telah dideteksi, dipotong, dan disesuaikan ukurannya tersebut akan dilakukan iterasi sebanyak 50 layer. Pada langkah ini akan dideteksi vector-vektor landmark wajah bagian atas saja. Dari iterasi Resnet50 sebanyak 50 layer ini akan dihasilkan matrik embedding sepanjang 2048. Matriks ini berisi vector dari landmark wajah gambar yang diberikan sebelumnya. Pada matriks embedding ini beberapa bernilai 0 karena landmark tidak terdeteksi. Hal ini terjadi karena bagian wajah bagian bawah terpotong sehingga sistem akan memberikan nilai 0 untuk

vector ini. Contoh embedding yang didapatkan dapat dilihat pada Gambar 4.6.



Gambar 4.6. Hasil Ekstraksi

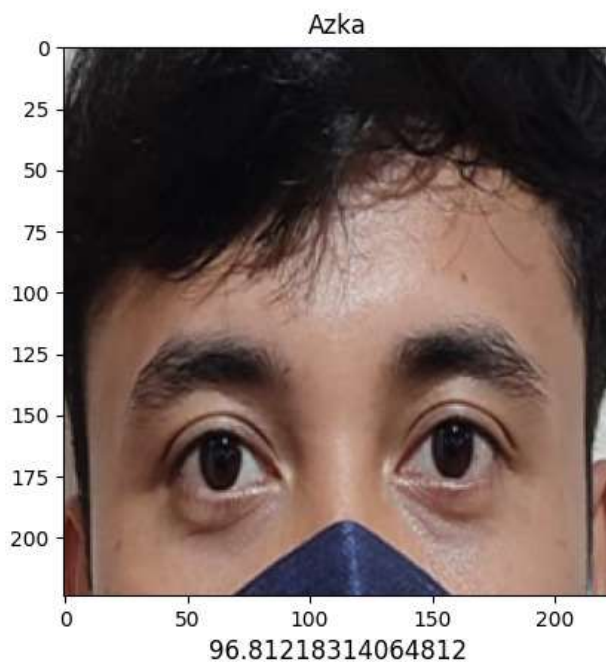
4.1.5 Hasil Blok Data Pickle

Setelah embedding didapatkan maka dari gambar dataset training embedding ini akan dimasukkan satu persatu ke dalam list pickle. Sedangkan jika dari alur dataset test, embedding akan langsung digunakan untuk pembandingan berikutnya. Hasil dari blok ini adalah sebuah file data pickle.

4.1.6 Hasil Blok Pembandingan

Pada blok ini dilakukan pembandingan dari embedding gambar yang dites dengan embedding gambar dari dataset training yang sudah ada pada list pickle. Embedding yang didapat akan dihitung nilai kesamaannya dengan embedding yang ada. Hasil yang didapat pada blok ini merupakan nilai tingkat kesamaan antara gambar wajah yang dites dengan yang sudah ada dan juga kelas dari embedding dengan nilai paling tinggi. Dari embbeding sebelumnya disebutkan bahwa beberapa

vector akan bernilai 0, hal ini akan membuat tingkat kesamaan yang lebih tinggi saat dibandingkan karena titik-titik vector ini tidak diberikan nilai yang kabur atau nilai yang salah. Hal ini terjadi karena semakin sedikit perbedaan antara matriks embedding dari dataset training yang ada di list pickle dengan embedding hasil dari dataset test. Contoh hasil pembandingan dapat dilihat pada Gambar 4.7.

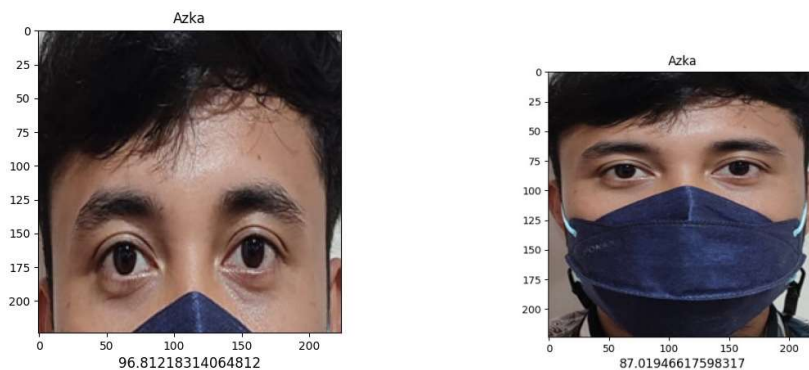


Gambar 4.7. Hasil Pembandingan

Dari langkah-langkah yang telah dilaksanakan di atas, didapatkan bahwa nilai untuk pemotongan terbaik ada di 0.5, jika nilai dibawah 0,5 maka wajah akan susah dikenali dengan banyak prediksi yang salah. Sedangkan untuk nilai threshold terbaik ada di 0,5 untuk memastikan prediksi yang bernilai rendah tidak tercetak atau tidak digunakan.

Telah dilakukan juga percobaan jika data yang digunakan tidak dipotong

terlebih dahulu. Hasil yang didapatkan dari percobaan ini menunjukkan dengan seluruh wajah yang dideteksi akan memberikan nilai titik-titik vector pada embedding yang kabur sehingga menurunkan tingkat kesamaan dataset yang sudah ditraining dengan gambar test. Perbandingan tingkat kesamaan antara wajah yang dipotong dan tidak dipotong dapat dilihat pada gambar 4.8.



Gambar 4.8. Perbandingan seluruh wajah dan separuh wajah

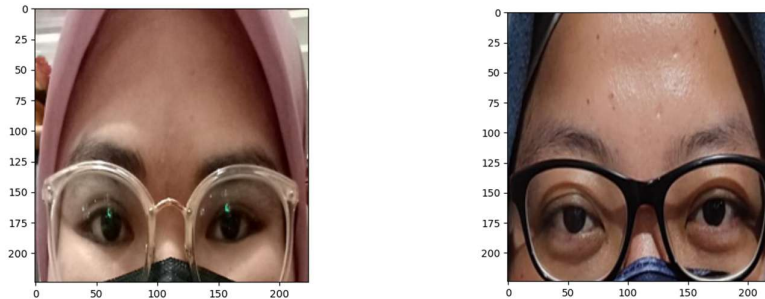
Selanjutnya dari hasil pengenalan wajah dengan menggunakan bagian wajah atas dapat dilihat pada tabel confusion matriks 4.1

Tabel 4.1 Tabel Hasil Separuh Wajah

		Actual Class														
Prediction		Hanif	Azka	Kresna	Rizky	Taufik	Idos	Friska	Ulfa	Yusak	Grace	Yuda	Rimba	Adhi	Siska	Hesa
	Hanif	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	Azka	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
	Kresna	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
	Rizky	1	0	0	5	0	0	0	0	0	0	0	0	0	0	0
	Taufik	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0
	Idos	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
	Friska	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0
	Ulfa	0	0	0	0	0	0	0	5	0	0	0	0	0	0	1
	Yusak	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0
	Grace	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
	Yuda	0	0	0	0	0	1	0	0	0	0	5	0	0	0	0

Rimba	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0
Adhi	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0
Siska	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0
Hesa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6

Dari table 4.1 didapatkan hasil bahwa dari 90 foto test dataset ada 4 foto yang salah diprediksi atau salah pengenalan wajahnya. Sedangkan 86 foto dapat diprediksi dengan benar pengenalan wajahnya. Beberapa gambar yang diprediksi salah ini terjadi karena beberapa kesamaan fitur pada wajah dan juga penggunaan aksesoris seperti jilbab dan kacamata yang masih memberikan titik kabur ke beberapa vector. Contoh kesamaan fitur dan penggunaan aksesoris dapat dilihat pada Gambar 4.9.



Gambar 4.9. Persamaan fitur dan penggunaan aksesoris

Selanjutnya kita dapat menghitung tingkat akurasi, tingkat presisi dan tingkat sensitifitas dengan menggunakan rumus yang sudah disebutkan pada Bab III sebelumnya. Hasil unuk tingkat sensitifitas dan presisi masing-masing kelas dapat dilihat pada table 4.2

Tabel 4.2 Tabel Sensitifitas dan presisi

Kelas	Sensitivity	Precision
Hanif	0,83	0,71
Azka	1,00	1,00
Kresna	1,00	1,00

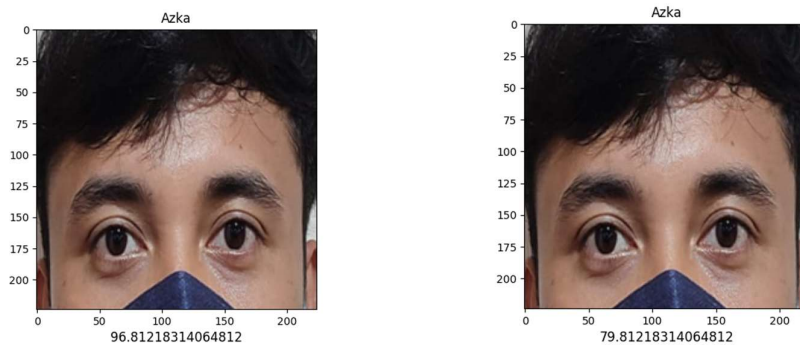
Rizky	0,83	0,80
Taufik	1,00	1,00
Idos	1,00	0,75
Friska	1,00	1,00
Ulfa	0,83	0,83
Yusak	1,00	1,00
Grace	1,00	1,00
Yuda	0,83	1,00
Rimba	1,00	0,86
Adhi	1,00	1,00
Siska	1,00	1,00
Hesa	1,00	0,86

Dari table 4.1 dapat dihasilkan rata-rata tingkat sensitifitas untuk pengenalan wajah dengan separuh wajah bagian atas adalah 96%. Sedangkan untuk tingkat presisi didapatkan senilai 94%. Untuk nilai tingkat akurasi pengenalan wajah dengan separuh wajah bagian atas adalah sebesar 95,6%.

Jika dibandingkan dengan pengenalan wajah dengan seluruh bagian wajah maka didapatkan bahwa terjadi peningkatan tingkat akurasi sebesar 2,3% dari 93,3% menjadi 95,6%. Dengan kelasahan deteksi berkurang dari 6 foto menjadi 4 foto wajah.

4.1.7 Pembandingan VGG Face dan Facenet

Penulis juga melakukan percobaan dengan menggunakan algorima MTCNN – Facenet. Dengan menggunakan alur yang sama dan dataset yang sama yaitu dataset training dan dataset test. Dan dilakukan pemotongan pada wajah dengan nilai yang sama. Didapatkan bahwa VGGFace dengan model Resnet50 memiliki tingkat nilai prediksi yang lebih tinggi dibandingkan dengan Facenet. Contoh prediksi dengan VGGFace dan Facenet dengan gambar test yang sama dapat dilihat pada Gambar 4.10.



Gambar 4.10. Prediksi VGGFace (kiri) dan Facenet (kanan)

Hal ini disebabkan karena VGGFace dengan menggunakan Resnet 50 memiliki 50 layer iterasi dengan output embedding 2048 dimensi sedangkan jika Facenet memiliki 20 layer iterasi dengan output embedding 128 dimensi. Sehingga nilai perbandingan akan lebih banyak pada VGGFace sehingga jika ada titik yang salah tidak akan mengurangi banyak nilai prediksi. Hal ini menghasilkan nilai akurasi VGG Face yang lebih tinggi dengan akurasi 95,6% dibanding Facenet dengan akurasi 88,9%. Dengan kesalahan prediksi VGG Face ada 4 dibandingkan Facenet ada 10 gambar yang salah prediksi. Perbandingan antara VGGFace dan Facenet dapat dilihat pada Table 4.3.

Tabel 4.3 Tabel Perbandingan VGGFace (Resnet50) dan Facenet

Algoritma	Layer iterasi	Dimensi Output	Akurasi	Sensitivity	Presisi	Kesalahan Prediksi
Resnet50	50	2048	95,6%	95,6%	93,8%	4
Facenet	22	128	88,9%	90,0%	89,3%	10

4.1.8 Kekurangan dan Kelebihan

Pada penelitian yang dilakukan memiliki beberapa kelebihan yaitu

1. Sistem dapat mengenali wajah bermasker ataupun tidak bermasker.
2. Sistem dapat memberikan prediksi dengan cepat

Sedangkan kekurangan pada penelitian yang dilakukan adalah

1. Sistem mengalami kesulitan jika orang menggunakan aksesoris pada wajah bagian atas contohnya kacamata.
2. Dataset yang digunakan pada penelitian ini masih terbatas.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang dilakukan berupa penelitian pengenalan wajah menggunakan MTCNN dan VGGFace dengan separuh wajah bagian atas didapatkan kesimpulan sebagai berikut

1. Tingkat akurasi pengenalan wajah menggunakan MTCNN dan VGGFace dengan separuh wajah bagian atas meningkat sebesar 2,3% dari 93,3% dengan seluruh bagian wajah yaitu 95,6%.
2. Tingkat sensitifitas pengenalan wajah menggunakan MTCNN dan VGGFace dengan separuh wajah bagian atas meningkat sebesar 2,7% dari 93,3% dengan seluruh bagian wajah yaitu 96%.
3. Tingkat presisi pengenalan wajah menggunakan MTCNN dan VGGFace dengan separuh wajah bagian atas meningkat sebesar 0,7% dari 93,3% dengan seluruh bagian wajah yaitu 94%.

Dengan menggunakan MTCNN dan VGGFace dengan separuh wajah bagian atas dapat meningkatkan tingkat akurasi, sensitifitas, dan presisi terhadap pengenalan wajah bermasker. Sehingga pengenalan wajah pada wajah bermasker dapat dilakukan dengan menggunakan algoritma MTCNN dan VGGFace dengan tingkat akurasi 95,6%.

5.2 Saran

Hasil penelitian ini menunjukkan bahwa MTCNN dan VGGFace dapat digunakan untuk pengenalan wajah, namun masih memiliki beberapa kekurangan. Kesalahan deteksi karena penggunaan kacamata dan jilbab hal ini dipengaruhi oleh jumlah dataset yang kurang bervariasi. Disarankan untuk menggunakan dataset dimana orang menggunakan kacamata dan tidak menggunakan kacamata atau aksesoris lainnya.

DAFTAR PUSTAKA

- [1] E. Winarno, I. Husni Al Amin, H. Februariyanti, P. W. Adi, W. Hadikurniawati and M. T. Anwar, "Attendance System Based on Face Recognition System Using CNN-PCA Method and Real-time Camera," 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2019.
- [2] R. K. Shukla, A. K. Tiwari and V. Verma, "Identification of with Face Mask and without Face Mask using Face Recognition Model," 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021.
- [3] M. S. Ejaz, M. R. Islam, M. Sifatullah and A. Sarker, "Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019.
- [4] M. S. Ejaz and M. R. Islam, "Masked Face Recognition Using Convolutional Neural Network," 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019.
- [5] Cao, Qiong, et al. "Vggface2: A dataset for recognising faces across pose and age." 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). IEEE, 2018.
- [6] M. Peng, C. Wang, T. Chen, G. Liu, and X. Fu, "Dual temporal scale convolutional neural network for micro-expression recognition," *Front. Psychol.*, vol. 8, p. 1745, 2017
- [7] Puthea, Khem, Rudy Hartanto, and Risanuri Hidayat. "A review paper on attendance marking system based on face recognition." 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE). IEEE, 2017.
- [8] Y. B. Chandra and G. K. Reddy, "A Comparative Analysis Of Face Recognition Models On Masked Faces" *International Journal of Scientific & Technology Research* Vol 9, 2020
- [9] K. Zhang, Z. Zhang., Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," in *IEEE Signal Processing Letters* 23, no.10, 2016,
- [10] Mool, Akshay, J. Panda, and Kapil Sharma. "Optimizable face detection and tracking model with occlusion resolution for high quality videos." *Multimedia Tools and Applications* 81.8 (2022): 10391-10406.

- [11] An, Xiangjing, Wensen Chang, and Xiangdong Chen. "Multi-layer template correlation neural network for recognition of lane mark based on pipelined image processing structure." Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C). Vol. 3. IEEE, 1999.
- [12] Lindner, Tymoteusz, et al. "Face recognition system based on a single-board computer." 2020 International Conference Mechatronic Systems and Materials (MSM). IEEE, 2020.
- [13] S. Petluru and P. Singh, "Transfer Learning-based Facial Expression Recognition with modified ResNet50," 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 2022

LAMPIRAN

```
# import

import cv2

import os

from keras_vggface import VGGFace

from keras_vggface.utils import decode_predictions

from mtcnn.mtcnn import MTCNN

import json

import pickle

from numpy import asarray

from keras_vggface.utils import preprocess_input

from PIL import Image

from matplotlib import pyplot

import tensorflow as tf

from scipy.spatial.distance import cosine

import timeit

tf.config.experimental.set_visible_devices([], 'GPU')

# ekstrak dr foto

def extract_face(filename, required_size=(224, 224)):

    # load foto / image

    pixels = pyplot.imread(filename)

    # detector

    detector = MTCNN()

    # detect wajah dari gambar
```

```
results = detector.detect_faces(pixels)

# membuat bounding box, dan memotong
x1, y1, width, height = results[0]['box']

x2, y2 = x1 + width, y1 + round(height*0.50)

# extract wajah
face = pixels[y1:y2, x1:x2]

# resize ukuran jadi 224x224
image = Image.fromarray(face)

image = image.resize(required_size)

face_array = asarray(image)

return face_array


# extract wajah dan embedding
def get_embeddings(filenamees):

    # ekstrak wajah
    faces = [extract_face(f) for f in filenamees]

    # jadi array
    samples = asarray(faces, 'float32')

    samples = preprocess_input(samples, version=2)

    # vgg face
    model = VGGFace(model='resnet50', include_top=False,
input_shape=(224, 224, 3), pooling='avg')

    # prediksi
    yhat = model.predict(samples)

    print (yhat.shape)

    print (yhat)

    return yhat
```

```
# pembandingan dari embedding yang diketahui

def banding(trained_embedding, test_embedding, thresh=0.50):

    score = cosine(trained_embedding, test_embedding)

    if score <= thresh:

        return score

    else:

        return -1

name_faces = {}

with open('data.pickle', 'rb') as f:

    name_faces = pickle.load(f)

# buka wajah

filenames = ['az.jpeg']

# ambil embedding

start=timeit.timeit()

embeddings = get_embeddings(filenames)

name = "Unknown"

curthres = 0.35

for i in name_faces:

    k = banding(embeddings, name_faces[i], curthres)

    if k > 0:

        name = i

        curthres = k

        akurasi = 100-(k*100)

        print(akurasi)

        print(name)
```

```
end=timeit.timeit()

print(end - start)

pixels = extract_face('az.jpeg')

pyplot.imshow(pixels)

pyplot.title(name, fontsize=12)

pyplot.xlabel(akurasi, fontsize=12)

pyplot.show()
```

