

**TESIS**

**IDENTIFIKASI WAJAH BERMASKER MENGGUNAKAN ALGORITMA  
MULTI-TASK CASCADED CONVOLUTIONAL NEURAL NETWORK**



Disusun oleh:

**Nama** : Stryati  
**NIM** : 18.52.1117  
**Konsentrasi** : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2021**

**TESIS**

**IDENTIFIKASI WAJAH BERMASKER MENGGUNAKAN ALGORITMA  
MULTI-TASK CASCADED CONVOLUTIONAL NEURAL NETWORK**

**MASKED FACIAL IDENTIFICATION USING MULTI-TASK CASCADED  
CONVOLUTIONAL NEURAL NETWORK ALGORITHM**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Sriyati  
NIM : 18.52.1117  
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA  
2021**

**HALAMAN PENGESAHAN**

**IDENTIFIKASI WAJAH BERMASKER MENGGUNAKAN ALGORITMA  
MULTI-TASK CASCADED CONVOLUTIONAL NEURAL NETWORK**

**MASKED FACIAL IDENTIFICATION USING MULTI-TASK CASCADED  
CONVOLUTIONAL NEURAL NETWORK ALGORITHM**

Dipersiapkan dan Disusun oleh

**Sriyati**

**18.52.1117**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Selasa, 5 January 2021

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 5 January 2021

**Rektor**

**Prof. Dr. M. Suyanto, M.M.**  
**NIK. 190302001**

## HALAMAN PERSETUJUAN

**IDENTIFIKASI WAJAH BERMASKER MENGGUNAKAN ALGORITMA  
MULTI-TASK CASCADED CONVOLUTIONAL NEURAL NETWORK  
MASKED FACIAL IDENTIFICATION USING MULTI-TASK CASCADED  
CONVOLUTIONAL NEURAL NETWORK ALGORITHM**

Dipersiapkan dan Disusun oleh

**Sriyati**

**18.52.1117**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Selasa, 5 January 2021

**Pembimbing Utama**

**Dr. Arief Setvanto, S.Si, M.T.**  
**NIK. 190302036**

**Pembimbing Pendamping**

**Emha Taufik Luthfi, S.T., M.Kom**  
**NIK. 90302125**

**Anggota Tim Penguji**

**Dr. Andi Sunyoto, M.Kom.**  
**NIK. 190302052**

**Dr. Suwanto Raharjo, S.Si.**  
**NIK. 999106**

**Dr. Arief Setvanto, S.Si, M.T.**  
**NIK. 190302036**

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer.

Yogyakarta, 5 January 2021  
**Direktur Program Pascasarjana**

**Dr. Kusriani, M.Kom.**  
**NIK. 190302106**

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Sriyati  
NIM : 18.52.1117  
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:  
**Identifikasi Wajah Bermasker Menggunakan Algoritma Multi-Task  
Cascaded Convolutional Neural Network**

Dosen Pembimbing Utama : Dr. Arief Setyanto, S.Si, M.T.  
Dosen Pembimbing Pendamping : Emha Taufik Luthfi, S.T., M.Kom.

- c. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
- d. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
- e. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
- f. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
- g. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 5 January 2020

Yang Menyatakan,

Meterai

Sriyati

Sriyati

Sriyati

Sriyati

Sriyati



## HALAMAN PERSEMBAHAN

Alhamdulillah penulis panjat kan puji syukur kepada Allah SWT atas segala rahmat dan hidayahnya, sehingga berkesempatan untuk menyelesaikan laporan tesis ini dengan segala kekurangan penulis. Segala syukur penulis ucapkan kepadaMu karena telah menghadirkan mereka yang memberikan semangat dan doa di saat menjalani proses pembuatan skripsi ini. Dengan segala kerendahan hati, saya persembahkan laporan tesis ini kepada:

- a. Bapak Dr. Arief Setyanto, S.Si, M.T. selaku pembimbing utama yang selalu memberikan bimbingan, motivasi dan arahan dalam proses pengerjaan tesis.
- b. Bapak Emha Taufik Luthfi, S.T., M.Kom, selaku pembimbing pendamping yang telah memberikan bimbingan dan masukan-masukan dalam proses pengerjaan.
- c. Keluarga Besar saya yang mensupport dan memberikan doanya selama ini.
- d. Teman-teman seangkatan kelas B21 yang tidak mungkin untuk disebutkan satu persatu. Terimakasih atas semuanya yang melengkapi keseharian dalam menimba ilmu.
- e. Teman yang selalu mensupport dan memberi dukungan serta menjadi dataset dalam penelitian ini.

Saya ucapkan terima kasih yang sebesar-besarnya, mohon maaf jika ada salah kata baik sengaja atau tidak selama ini. Sukses buat kalian semua dilancarkan segala urusannya, semoga Allah SWT memberikan rahmat dan hidayahnya kepada kita.

semua, Amin.....

## HALAMAN MOTTO

"Dimanapun kita berada jangan pernah melupakan  
darimana kita berasal"

"Jalanku memang lambat tapi aku bisa menikmati  
disetiap jalan yang aku lalui"

"Harta yang paling indah di dunia ini  
adalah orangtua"

"Ada 3 kunci keberhasilan yaitu keyakinan,  
kesungguhan dan kesabaran"

"Kesempurnaan hanya milik Allah SWT"



## KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan tesis ini dengan baik. Meskipun dalam penyelesaian ini banyak ditemui kendala. Tesis ini disusun Guna memenuhi salah satu syarat untuk menyelesaikan Program Studi S2 Teknik Informatika di Universitas Amikom Yogyakarta.

Berbagai pihak telah banyak membantu penulis dalam penyelesaian tesis ini, untuk itu penulis ucapkan terima kasih kepada:

1. Bapak Dr. Arief Setyanto, S.Si, M.T. selaku pembimbing utama yang selalu memberikan bimbingan, motivasi dan arahan dalam proses pengerjaan tesis.
2. Bapak Emha Taufik Luthfi, S.T., M.Kom, selaku pembimbing pendamping yang telah memberikan bimbingan dan masukan-masukan dalam proses pengerjaan.
3. Keluarga Besar saya yang mensupport dan memberikan doanya selama ini
4. Teman-teman seangkatan kelas B21 yang tidak mungkin untuk disebutkan satu persatu. Terimakasih atas semuanya yang melengkapi keseharian dalam menimba ilmu.
5. Teman yang selalu mensupport dan memberi dukungan serta menjadi dataset dalam penelitian ini.

Penulis menyadari bahwa penulisan tesis ini masih banyak kekurangan, oleh karenanya kritik dan saran sangat penulis harapkan guna menyempurnakan viii tesis ini. Akhir kata penulis mengucapkan banyak terima kasih dan semoga tesis ini bisa memberi manfaat bagi kita semua.

Yogyakarta, 5 January 2020

Penulis

## DAFTAR ISI

|   |       |
|---|-------|
| HALAMAN PENGESAHAN .....                | iii   |
| HALAMAN PERSETUJUAN .....               | iv    |
| HALAMAN PERNYATAAN KEASLIAN TESIS ..... | v     |
| HALAMAN PERSEMBAHAN .....               | vi    |
| HALAMAN MOTTO .....                     | vii   |
| KATA PENGANTAR .....                    | viii  |
| DAFTAR ISI .....                        | ix    |
| DAFTAR tabel .....                      | xii   |
| DAFTAR GAMBAR .....                     | xiv   |
| INTISARI .....                          | xvii  |
| <i>ABSTRACT</i> .....                   | xviii |
| BAB I PENDAHULUAN .....                 | 1     |
| 1.1. Latar Belakang Masalah .....       | 1     |
| 1.2. Rumusan Masalah .....              | 8     |
| 1.3. Batasan Masalah .....              | 8     |
| 1.4. Tujuan Penelitian .....            | 9     |
| 1.5. Manfaat Penelitian .....           | 10    |
| BAB II TINJAUAN PUSTAKA .....           | 11    |
| 2.1. Tinjauan Pustaka .....             | 11    |
| 2.2. Keaslian Penelitian .....          | 18    |
| 2.3. Landasan Teori .....               | 31    |
| BAB III METODE PENELITIAN .....         | 49    |

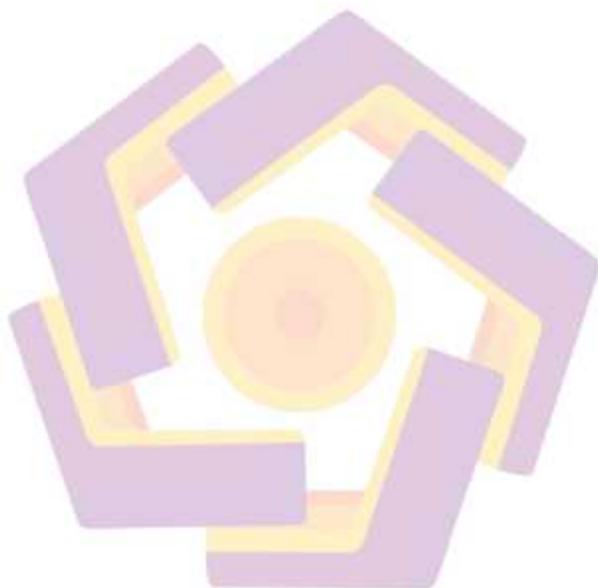
|   |           |
|---|-----------|
| 3.1. Jenis, Sifat, dan Pendekatan Penelitian .....  | 49        |
| 3.2. Metode Pengumpulan Data.....   | 49        |
| 3.3 Methodology .....   | 51        |
| 3.3.1 Deteksi wajah menggunakan OpenCV DNN dan TensorFlow .....                                   | 51        |
| 3.3.2 Identifikasi wajah menggunakan <i>Multi-task Cascade Convolutional Neural Network</i> ..... | 53        |
| 3.3.1 Penyematan wajah menggunakan FaceNet .....  | 59        |
| 3.3.4 Klasifikasi wajah menggunakan Linear Support Vector Machine .....                           | 61        |
| 3.3. Alur Penelitian .....  | 62        |
| <b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>  | <b>65</b> |
| 4.1. Dataset and Experimental Setup .....   | 65        |
| 4.1.1 Dataset .....   | 65        |
| 4.1.2 Experimental Setup .....  | 68        |
| 4.2 Deteksi wajah bermasker menggunakan OpenCV DNN dan TensorFlow .....                           | 69        |
| 4.2.1. Training data .....  | 69        |
| 4.2.2. Testing mask .....   | 75        |
| 4.3 Proses deteksi wajah menggunakan <i>Multi Task Cascade Convolutional Neural Network</i> ..... | 80        |
| 4.3.1 Tahap 1: Image Pyramid .....  | 80        |
| 4.3.2 Tahap 2: P-Net.....   | 83        |
| 4.3.3 Tahap 3: R-Net.....   | 84        |

|   |     |
|---|-----|
| 4.3.4 Tahap 4: O-Net .....  | 85  |
| 4.3.5 Tiga tugas pelatihan CNN detectors .....  | 86  |
| 4.4 Identifikasi Wajah Tanpa Masker Pada Dataset2 menggunakan Multi Task<br>Cascade Convolutional Neural Network .....            | 89  |
| 4.4.1 Deteksi Wajah Untuk <i>Face Recognition</i> .....   | 89  |
| 4.4.2. Membuat Penyematan Wajah ( <i>Face Embeddings</i> ) .....  | 93  |
| 4.4.3 Klasifikasi Wajah ( <i>Face Classification</i> ) .....  | 94  |
| 4.4.3.1. Memplot wajah asli dan Prediksi .....  | 95  |
| 4.5 Identifikasi wajah bermasker pada dataset3 menggunakan <i>Multi Task</i><br><i>Cascade Convolutional Neural Network</i> ..... | 99  |
| 4.5.1 Deteksi Wajah Bermasker ( <i>Faces Mask Detection</i> ) .....   | 100 |
| 4.6 Pengujian pada dataset real wajah dengan beragam jenis masker .....   | 106 |
| 4.7 Pengujian diluar dataset latih pada pengujian dataset 3 .....   | 109 |
| BAB V PENUTUP .....   | 110 |
| 5.1. Kesimpulan .....   | 110 |
| 5.2. Saran .....  | 111 |
| Terdapat beberapa saran untuk penelitian selanjutnya, diantaranya: .....  | 111 |
| DAFTAR PUSTAKA .....  | 112 |

## DAFTAR TABEL

|  |     |
|--|-----|
| Tabel 4.1 Training accuracy/loss.....  | 71  |
| Tabel 4.2 Hasil Akurasi, Recall, Precision, F1-Score, Support Dari Training Face Mask Detection..... | 74  |
| Tabel 4.3 Dataset2 .....   | 90  |
| Tabel 4.4 Array NumPy yang berisi data piksel wajah dari direktori Wawan riyanto .....               | 91  |
| Tabel 4.5 Hasil proses dari load data training dan validasi.....                                     | 92  |
| Tabel 4.6 Terdeteksi wajah yang diidentifikasi dengan benar oleh SVM Classifier .....                | 96  |
| Tabel 4.7 Analisis foto Aldi alkulakat dari data testing.....  | 97  |
| Tabel 4.8 Analisis foto Elfrida dari data testing.....   | 97  |
| Tabel 4.9 Analisis foto Yohan dari data testing .....  | 98  |
| Tabel 4.10 Analisis foto Wawan Riyanto dari data testing.....  | 98  |
| Tabel 4.11 Terdeteksi wajah dengan masker yang diidentifikasi dengan benar oleh SVM Classifier.....  | 103 |
| Tabel 4.12 Analisis data wajah Aldi alkulakat menggunakan masker .....                               | 104 |
| Tabel 4.13 Analisis data wajah Elfrida menggunakan masker .....                                      | 104 |
| Tabel 4.14 Analisis data wajah Yohan menggunakan masker .....  | 105 |
| Tabel 4.15 Analisis data wajah Wawan riyanto menggunakan masker.....                                 | 105 |
| Tabel 4.16 Hasil deteksi wajah dengan MTCNN pada data real (data train).....                         | 106 |
| Tabel 4.17 Hasil prediksi, ekspektasi dan confidence dari Elfrida pada dataset real .....            | 107 |

|   |     |
|---|-----|
| Tabel 4.18 Hasil prediksi, ekspektasi dan confidence dari Rofik pada dataset real         | 107 |
| Tabel 4.19 Hasil prediksi, ekspektasi dan confidence dari Wawan riyanto pada dataset real | 108 |
| Tabel 4.20 Hasil prediksi, ekspektasi dan confidence dari Yohan pada dataset real         | 108 |



## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Typical training image for face recognition.....                           | 32 |
| Gambar 2.2 Hubungan antara human vision, computer vision, machine .....               | 34 |
| Gambar 2.3 Features and classifiers .....   | 35 |
| Gambar 2.4 Neural Network Classifier.....   | 39 |
| Gambar 2.5 Contoh filter gambar 2D .....  | 41 |
| Gambar 2.6 The operation of a convolution layer is illustrated.....                   | 42 |
| Gambar 2.7 The operation of a convolution layer is illustrated.....                   | 42 |
| Gambar 2.8 The operation of a convolution layer with a zero padding .....             | 43 |
| Gambar 2.9 The operation of max-pooling layer.....                                    | 44 |
| Gambar 2.10 Pipeline for the Multi-Task Cascaded Convolutional Neural Network .....   | 45 |
| Gambar 2.11 Architectures P-Net, R-Net, and O-Net.....                                | 46 |
| Gambar 2.12 Model structure.....  | 46 |
| Gambar 2.13 Triplet Loss.....   | 47 |
| Gambar 3.1 Alur proses pengumpulan data.....  | 50 |
| Gambar 3.2 Tahapan deteksi wajah bermasker menggunakan OpenCV DNN dan TensorFlow..... | 52 |
| Gambar 3.3 Pipeline of MTCNN method.....  | 54 |
| Gambar 3.4 Architectures of P-Net .....   | 56 |
| Gambar 3.5 Architectures of R-Net.....  | 57 |
| Gambar 3.6 Architectures of O-Net.....  | 58 |
| Gambar 3. 7 Triplet Loss .....  | 60 |

|  |    |
|--|----|
| Gambar 3.8 Alur Penelitian.....  | 63 |
| Gambar 4.1 Sampel dataset wajah yang di develop oleh .....                                       | 65 |
| Gambar 4.2 Salah satu dataset2 yang dikumpulkan .....  | 66 |
| Gambar 4.3 Deteksi wajah tanpa masker dengan MTCNN.....  | 66 |
| Gambar 4.4 Masker COVID-19 developed by Pyimagesearch.....                                       | 67 |
| Gambar 4.5 Salah satu dataset3 yang dikumpulkan .....  | 67 |
| Gambar 4.6 deteksi wajah dengan masker dengan MTCNN .....  | 68 |
| Gambar 4.7 Sampel gambar dataset real dengan berbagai macam masker .....                         | 68 |
| Gambar 4.8 Deteksi wajah bermasker pada dataset real.....  | 68 |
| Gambar 4.9 Pre-process konversi gambar ke format array.....                                      | 70 |
| Gambar 4.10 Grafik Training Loss Dan Validation Loss .....                                       | 71 |
| Gambar 4.11 Training Accuracy dan Validation Accuracy.....                                       | 72 |
| Gambar 4.12 Confusion matrix .....   | 74 |
| Gambar 4.13 Pengurangan rata-rata .....  | 76 |
| Gambar 4.14 Dimensi blob yang dihasilkan dari blob From Image.....                               | 77 |
| Gambar 4.15 Output gambar deteksi masker wajah COVID-19 dengan OpenCV<br>DNN dan Tensorflow..... | 79 |
| Gambar 4.16 Output gambar deteksi No mask dengan OpenCV DNN dan<br>Tensorflow.....               | 79 |
| Gambar 4. 17 Image pyramid.....  | 80 |
| Gambar 4.18 Piramida Gaussian level.....   | 82 |
| Gambar 4.19 Level Laplacian pyramid.....   | 82 |
| Gambar 4.20 P-Net (Proposal network).....  | 83 |

|   |     |
|---|-----|
| Gambar 4.21 R-Net.....  | 84  |
| Gambar 4.22 O-Net.....  | 85  |
| Gambar 4.23 Non-Maximum Suppression.....  | 87  |
| Gambar 4.24 Bounding box regression output network .....  | 88  |
| Gambar 4.25 Face detect dengan Combined loss function (dot, box, landmark) .                      | 88  |
| Gambar 4. 26 Proses identifikasi wajah salah satu dataset 2 (tanpa masker) .....                  | 89  |
| Gambar 4. 27 wajah yang terdeteksi dalam direktori training .....                                 | 91  |
| Gambar 4.28 Contoh pada anchor, positif, dan negatif dengan menerapkan kerugian Triplet Loss..... | 93  |
| Gambar 4.29 Hasil Face Embedding .....  | 94  |
| Gambar 4.30 identifikasi wajah salah satu dataset 3 (dengan masker).....                          | 99  |
| Gambar 4.31 Deteksi wajah bermasker .....   | 100 |
| Gambar 4. 32 Face detection dengan MTCNN .....  | 101 |
| Gambar 4.33 Face mask deteksi dari Aldi alkulakat .....   | 101 |
| Gambar 4.34 Face mask deteksi dari Elfrid .....   | 102 |
| Gambar 4.35 Face mask deteksi dari Yohan.....   | 102 |
| Gambar 4.36 Face mask deteksi dari Wawan riyanto .....  | 102 |
| Gambar 4. 37 Hasil pengujian diluar data latih.....   | 109 |

## INTISARI

Covid-19 telah ditetapkan sebagai pandemi global, menggunakan masker wajah telah menjadi bagian penting dalam kehidupan kita. Masyarakat dihimbau untuk menutupi wajahnya ketika berada di area publik untuk menghindari penyebaran virus. Penggunaan masker wajah ini telah menimbulkan pertanyaan serius mengenai sistem pengenalan wajah untuk mengidentifikasi dan memverifikasi seseorang dikarenakan oleh masalah wajah yang sebagian tertutup masker atau penutup wajah lainnya.

Oleh karena itu penelitian ini menyajikan deteksi wajah dengan masker dan tanpa masker dan melakukan identifikasi dengan landmark wajah minimum untuk mengetahui siapa orang di bawah wajah dengan masker tersebut. Penelitian ini menggunakan *Deep Neural network* untuk deteksi masker wajah dan metode *Multi-Task Cascaded Convolutional Neural Network*, *FaceNet* dan *SVM classifier* untuk proses identifikasi. Penelitian ini menggunakan tiga dataset yaitu dataset kumpulan wajah (dataset 1), dataset wajah tanpa masker (dataset 2) dataset wajah dengan masker virtual dan real (dataset 3).

Hasil dari deteksi wajah dengan masker dilatih pada dataset 1 mencapai akurasi 98% pada pelatihan dan untuk pengujian mendapatkan 100% akurasi pada wajah tanpa masker, 99,99% akurasi untuk wajah dengan masker, identifikasi dengan metode *Multi-Task Cascaded Convolutional Neural Network* mendapatkan ketepatan dan rata-rata keyakinan identitas dari data pengujian 78.69 untuk data wajah tanpa masker, 81.18 untuk data wajah dengan masker virtual dan dataset real dengan beragam masker mendapatkan keyakinan 83.659.

Kata kunci: deteksi wajah dengan masker, identifikasi wajah dengan masker, *Multi-Task Cascaded Convolutional Neural Network*, *FaceNet*, *SVM classifier*.

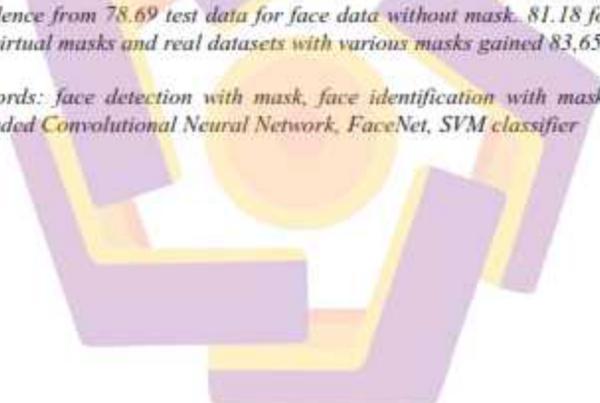
## **ABSTRACT**

*Covid-19 has been designated as a global pandemic, using face masks has become an important part of our lives. People are encouraged to cover their faces when in public areas to avoid the spread of the virus. The use of these face masks has raised serious questions about facial recognition systems to identify and verify a person due to facial problems partially covered by masks or other face coverings.*

*Therefore, this study presents face detection with masks and without masks and performs identification with minimum facial landmarks to find out who the person under the face with the mask is. The study used the Deep Neural network for face mask detection and the Multi-Task Cascaded Convolutional Neural Network, FaceNet and SVM classifier methods for the identification process. This study used three datasets, namely face set dataset (dataset1), face dataset without mask (dataset 2) face dataset with virtual and real mask (dataset 3).*

*The results of face detection with masks trained on dataset 1 achieved 98% accuracy at training and for testing gained 100% accuracy on faces without masks. 99.99% accuracy for face with mask, identification with Multi-Task Cascaded Convolutional Neural Network method obtains accuracy and average identity confidence from 78.69 test data for face data without mask. 81.18 for facial data with virtual masks and real datasets with various masks gained 83.659 confidence*

*Keywords: face detection with mask, face identification with mask, Multi-Task Cascaded Convolutional Neural Network, FaceNet, SVM classifier*



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Pada bulan Desember 2019, serangkaian kasus pneumonia virus penyebab tidak diketahui muncul di Wuhan, Hubei, Cina, yang dinamai 2019 novel coronavirus (2019-nCoV) atau COVID-19 (Yu, Ping, 2020). Pada tanggal 2 Maret 2020 Indonesia mengumumkan kasus pertama positif COVID-19 yang disampaikan oleh Presiden RI Joko Widodo (Ihsanuddin, 2020). Dan World Health Organization (WHO) pada tanggal 11 Maret 2020, menyatakan Covid-19 sebagai Pandemi Global (Putri, 2020).

Dalam gelombang situasi pandemi covid-19, pejabat kesehatan masyarakat mewajibkan semua orang untuk mengenakan masker di tempat umum. Menurut WHO kasus yang terinfeksi di seluruh dunia menjadi 22 juta (Chowdary et al., 2020). Mayoritas kasus positif ditemukan di tempat ramai dan over-area keramaian. Oleh karena itu, diresepkan oleh para ilmuwan bahwa mengenakan masker di tempat umum dapat mencegah penularan penyakit (Feng et al., 2020). Dari kondisi ini memunculkan inisiatif untuk melakukan identifikasi wajah bermasker dengan melakukan pengenalan wajah (*face recognition*). Bagaimana jika inisiatif ini dikembangkan *face recognition* yang tidak hanya mengidentifikasi gambar yang tidak bermasker tetapi juga bisa melakukan identifikasi siapa orang yang mengenakan masker tersebut. Dalam kasus ini *COVID-19 face mask detection* pada sistem pengenalan wajah diharapkan dapat mengidentifikasi wajah yang

ada dalam gambar. Ini dapat beroperasi dalam salah satu atau kedua mode: (1) verifikasi wajah (atau otentikasi), dan (2) identifikasi wajah atau pengakuan (Jain & Li, 2011). Melalui teknik kecerdasan buatan (AI) seperti *Machine Learning (ML)* dan pembelajaran (*Deep Learning*) yang dapat digunakan dalam banyak hal seperti halnya mencegah penularan COVID-19 (Agarwal et al., 2020).

Mendeteksi wajah bermasker dapat menggunakan *OpenCV*, *Keras* atau *TensorFlow*, dan *Deep Learning*. Sejumlah metode *Deep Learning* telah dikembangkan dan ditunjukkan untuk deteksi wajah salah satunya adalah *Multi-Task Cascaded Convolutional* dimana metode ini mampu mengenali fitur wajah lainnya seperti mata, hidung dan mulut, yang disebut deteksi tengara (Zhang et al., 2016). Dari sini MTCNN akan digunakan untuk deteksi wajah dengan model *FaceNet* digunakan untuk membuat *embedding* wajah untuk setiap wajah bermasker yang terdeteksi, dimana *FaceNet* adalah sistem pengenalan wajah *Deep Convolutional Neural Network* yang dilatih melalui fungsi yang mendorong vektor untuk identitas yang sama menjadi lebih mirip (jarak yang lebih kecil), sedangkan vektor untuk identitas yang berbeda diharapkan menjadi kurang mirip (jarak yang lebih besar). *FaceNet* model, dilatih untuk secara langsung mengoptimalkan penyematan (*embedding*) itu sendiri, daripada lapisan *bottleneck* menengah seperti dalam pendekatan *deep learning* sebelumnya (Schroff & Philbin, 2015). Maka selanjutnya mengembangkan model pengklasifikasi *Linear Support Vector Machine (SVM)* untuk memprediksi identitas wajah bermasker tertentu. SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization (SRM)* dengan

tujuan menemukan hyperplane terbaik yang memisahkan dua buah class pada input space (Nugroho et al., 2003).

Penelitian tentang deteksi wajah bermasker dilakukan oleh (Chowdary et al., 2020), penelitian ini menggunakan *Transfer Learning of Inception V3*, pembelajaran transfer yang diusulkan mencapai akurasi sebesar 99,92%, 99,9% selama pelatihan, dan 100%, 100% selama pengujian untuk deteksi wajah *no mask* dan *with mask*. Selanjutnya oleh (Militante et al., 2020), penelitian ini menggunakan sistem *Deep Learning* dan model VGG-16 CNN, untuk pengenalan wajah bermasker secara *Real-Time* dan dilengkapi dengan alarm, didapatkan hasil 96% untuk akurasi kinerja. Kemudian penelitian yang dilakukan oleh (Anwar & Raychowdhury, 2020), menyajikan alat *open-source*, *Mask The Face* yang digunakan untuk menutupi wajah. Penelitian menghasilkan penciptaan kumpulan data besar wajah bermasker dan melaporkan peningkatan ~ 38% untuk sistem *Facenet* yang ada untuk wajah bermasker maupun tanpa masker. Penelitian terkait oleh (Loey et al., 2021), penelitian ini menggunakan dua komponen. Komponen pertama dirancang untuk ekstraksi fitur menggunakan *Resnet50*. Sementara komponen kedua dirancang untuk proses klasifikasi masker wajah menggunakan pohon keputusan, *Support Vector Mesin (SVM)*, dan algoritma ansambel. Pengklasifikasi SVM mencapai akurasi pengujian 99,64% dalam RMFD. Dalam SMFD, mencapai 99,49%, sementara di LFW, mencapai akurasi pengujian 100%. Penelitian terkait deteksi wajah bermasker, mereka tidak membahas mengenai identifikasi siapa yang berada di balik wajah bermasker yang menjadi topik dalam penelitian yang akan dilakukan.

(Guillermo et al., 2020), penelitian ini menggunakan studi Prajna Bhandary tentang penambahan wajah masker untuk datasetnya. Pelaksanaan studi ini menghasilkan 99% dalam semua pelatihan dan pengujian parameter utama. Dalam penelitian, studi Prajna Bhandary digunakan untuk membuat dataset sendiri sedangkan yang akan dilakukan tidak membuat dataset melainkan menggunakan dataset Prajna Bhandary.

Selanjutnya penelitian yang berkaitan dengan pengenalan wajah oleh (Damale et al., 2018.), penelitian menggunakan DNN sebagai deteksi wajah, dengan tiga metode yaitu SVM, MLP dan CNN, yang digunakan sebagai pengklasifikasi untuk mengevaluasi pengenalan wajah, dan diperoleh akurasi pengujian sekitar 87%, 86,5% dan 98%. Penelitian serupa oleh (Sun et al., 2014), penelitian yang dilakukan menggunakan fitur verifikasi *Deep IDentification (Deep ID2)* dan mendapatkan akurasi verifikasi wajah 99,15% pada LFW. Penelitian lain dari (Zhan et al., 2016), penelitian ini menggunakan kombinasi CNN dan SVM sebagai sistem deteksi wajah, dalam sistem yang diusulkan, fitur yang diekstrak oleh CNN dapat menjadi representasi yang baik untuk deteksi wajah, ditambah dengan keuntungan SVM untuk klasifikasi. Penelitian serupa oleh (Arsal et al., 2020), penelitian ini membahas pengenalan wajah untuk pintu akses ruangan perkantoran oleh pegawai bank. Dalam penelitian menggunakan algoritma *Convolutional Neural Network (CNN)* dan 350 data wajah dari 5 orang dataset wajah pegawai bank sebagai dataset. Hasil yang diperoleh dari mengidentifikasi wajah yang ditangkap oleh kamera dengan persentase keakuratan 95%. Penelitian mengenai pengenalan wajah yang

telah dilakukan tidak membahas pengenalan wajah terhadap kondisi wajah tertutupi oleh masker yang menjadi topik dalam penelitian yang akan dilakukan.

Penelitian terkait penggunaan FaceNet model sebagai pengenalan wajah dilakukan oleh (Cahyono et al., 2020), dalam penelitiannya menggunakan Support Vector Mesin (SVM) sebagai klasifikasi dan memperoleh akurasi sempurna yaitu 100%. Penelitian lain oleh (Jose et al., 2019), menggunakan FaceNet dan MTCNN di Jetson TX2 sebagai sistem pengawasan berbasis pengenalan wajah didapatkan deteksi akurasi 97%. Kemudian penelitian yang dilakukan oleh (Nyein et al., 2019), menggunakan kombinasi FaceNet dan Support Vector Machine (SVM) untuk pengenalan multi wajah dan diperoleh akurasi 99,6%. Penelitian terkait oleh (Rosal et al., n.d.), dalam penelitiannya membahas tentang penggunaan dua jenis model pre-trained yaitu CASIA-WebFace dan VGGFace2. Tes dilakukan pada beberapa set data publik, seperti YALE, JAFFE, AT & T, Georgia Tech, dan Essex. Hasil dari penelitian ini, FaceNet mampu menyentuh akurasi 100% pada beberapa dataset yang diujikan.

Penelitian Lain terkait penggunaan metode *Multi-task Cascaded Convolutional Neural Network* oleh (Chen et al., n.d.), dalam penelitian penggunaan metode untuk algoritma lokalisasi mata yang didasarkan pada deteksi wajah MTCNN. Metode terkait digunakan untuk menentukan area alis dan kesesuaian dengan area wajah dan posisi pupil. Hasil eksperimental menunjukkan bahwa algoritma yang diusulkan memiliki akurasi deteksi sebesar 95,02%, untuk deteksi gambar mata dengan kualitas berbeda-beda. Selanjutnya oleh (Yin & Liu, 2018), penelitian ini membahas tentang pengenalan *Face Multi-Task Learning (MTL)* dengan

*Convolutional Neural Network (CNN)*. *Multi-task* untuk pengenalan wajah dimana diklasifikasikan identitas adalah tugas utama, dan estimasi Pose, Illumination, and *Expression (PIE)* adalah tugas sampingan.

Penelitian lain mengenai deteksi oklusi pada gambar wajah oleh (L. Song et al., 2019), penelitian ini menggunakan *Pairwise Differential Siamese Network (PDSN)* untuk mempelajari korespondensi antara blok wajah yang teroklusi dan elemen fitur yang rusak, pada gambar wajah dengan random partial occlusions. Kemudian menurut (Khadatkar et al., n.d.). Faktor utama yang mempengaruhi sistem pengenalan wajah adalah pose, iluminasi, penuaan, oklusi serta ekspresi dan diantaranya oklusi paling masalah yang mempengaruhi dalam pengenalan wajah. Penelitian serupa oleh (Kim et al., n.d.), membahas deteksi oklusi wajah menggunakan metode deteksi kepala dan bahu, oklusi area wajah ditentukan dengan mengukur rasio area warna kulit (SCAR) dari seluruh area wajah dan wajah area komponen dengan hasil tingkat deteksi 94.8% meskipun ada berbagai jenis oklusi parah di wajah. Kemudian penelitian yang dilakukan oleh (Xia et al., n.d.), menggunakan CNN untuk pre-training model, dataset yang digunakan peneliti memiliki jenis oklusi yang berbeda, yaitu oklusi mata oleh matahari, kaca mata, oklusi wajah dengan topi atau helm, dan mulut atau hidung oklusi oleh masker. deteksi oklusi yang dilakukan dievaluasi pada dataset wajah AR dan FO, dimana masing-masing mencapai akurasi 100% dan 97,24%. Penelitian terkait oleh (Li et al., n.d.), menggunakan Patch-Gated CNN untuk pengenalan ekspresi wajah di bawah oklusi. Dengan RAF-DB dataset dimana data dengan oklusi beraneka warna, bentuk, dan posisi. Dari hasil analisis ablasi menunjukkan PG-CNN mampu

mengalihkan perhatian dari tambalan yang teroklusi ke yang terkait lainnya. Proses oklusi terkait dengan penelitian yang akan dilakukan adalah satu bagian oklusi yaitu pada bagian mulut dan hidung tertutupi dengan masker.

Berdasarkan latar belakang dan beberapa penelitian sebelumnya maka melalui penelitian ini penulis mengusulkan identifikasi wajah bermasker beserta identitas dari setiap wajah yang bermasker menggunakan *Multi-task Cascaded Convolutional Neural Network*, sebagai deteksi wajah bermasker maupun tanpa masker, *FaceNet model* untuk *face embedding* untuk setiap wajah yang terdeteksi, dan pengklasifikasi *Linear Support Vector Machine (SVM)* untuk memprediksi identitas dari wajah tersebut. Dalam penelitian ini penulis akan melakukan face recognition dan face detection dengan *deep learning* (*OpenCV* dan *Tensorflow*) menggunakan tiga dataset, pertama dataset kumpulan wajah dari (Prajna Bhandary, 2020) untuk (dataset1), kedua dataset wajah tanpa masker yang diambil dari data primer (dataset2), ketiga dataset wajah dengan masker yang diambil dari data primer (dataset 3).

Penelitian ini bermaksud mengidentifikasi wajah yang tertutup sebagian (*occluded*) menggunakan masker. Berbeda dengan Xia, Yi Zhang, Bailing Zhang, and Frans Coenen. 2015, Yin, Xi, and XiaoMing Liu. 2017 dan Song, Lingxue, et al. 20119, yang mengidentifikasi wajah yang tertutup sebagian pada posisi acak, pada penelitian ini posisi dan ukuran oklusi tetap yaitu masker. Dibandingkan dengan penelitian Chowdary G.J et al. 2020, yang mendeteksi penggunaan masker atau tidak, penelitian ini tidak hanya mendeteksi adanya masker tetapi mengidentifikasi siapa yang menggunakan masker. Identifikasi wajah tersebut dapat

digunakan pada berbagai bidang seperti untuk melakukan pengenalan wajah dengan masker dan tanpa masker pada mesin, robot dan berbagai bidang dengan identitas wajah.

### 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan penulis, maka rumusan masalahnya adalah sebagai berikut:

- a. Bagaimana identifikasi identitas dari wajah bermasker maupun tanpa masker dengan menggunakan *Multi-task Cascaded Convolutional Neural Network*?
- b. Dari metode *Multi-task Cascaded Convolutional Neural Network* berapa tingkat akurasi yang dihasilkan pada dataset1 (dataset wajah), dataset2 (dataset wajah tanpa masker yang diambil dari data primer) dan dataset3 (dataset wajah dengan masker yang diambil dari data primer)?

### 1.3. Batasan Masalah

Adapun untuk batasan masalahnya adalah:

- a. Masukan citra adalah citra digital kecuali tipe TIF dan GIF, karena jumlah warna di bawah 256 dan terdapat gradien (perubahan warna).
- b. Posisi citra wajah pada bagian mata tidak terhalangi bagian objek lain, misalnya topi, kacamata hitam dan lain sebagainya.
- c. Deteksi bagian oklusi wajah yaitu hidung dan mulut oklusi oleh masker.
- d. Tipe data yang digunakan adalah klasifikasi.
- e. Metode yang digunakan untuk melakukan identifikasi wajah adalah metode *Multi-task Cascaded Convolutional Neural Network*.

- f. Deteksi wajah bermasker pada dataset 1 adalah menggunakan OpenCV DNN dan Tensorflow.
- g. Dataset yang digunakan untuk deteksi wajah bermasker dan tanpa masker sebelum ke dataset primer menggunakan dataset dari Prajna Bhandary (Prajna Bhandary, 2020).
- h. Tidak menggunakan data berskala besar pada identifikasi wajah baik dengan masker maupun tanpa masker.
- i. Tidak melakukan pengujian confusion matrix terhadap dataset 2 dan 3.
- j. Data yang nanti dihasilkan dari deteksi wajah berupa keakuratan dengan melabeli gambar sebagai masker dan tanpa masker. Di sertai nilai akurasi yang didapatkan.
- k. Data yang nanti di hasilkan dari identifikasi wajah akan menunjukkan nama dan probabilitas yang diprediksi dalam judul gambar.
- l. FaceNet model yang digunakan dari Hiroki Taniai (Hiroki Taniai. 2018).
- m. Tidak melakukan pengujian terhadap faktor yang mempengaruhi proses deteksi wajah atas penggunaan metode yang diterapkan.

#### **1.4. Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah:

- a. Deteksi wajah bermasker menggunakan *Deep learning* bertujuan untuk mengetahui apakah gambar terdeteksi sebagai gambar dengan masker wajah atau tanpa masker wajah.
- b. Deteksi wajah bermasker yang diterapkan bertujuan untuk mengetahui nilai akurasi yang dihasilkan dari deteksi masker wajah maupun tanpa masker wajah.

- c. Identifikasi wajah menggunakan *Multi-task Cascaded Convolutional Neural Network* bertujuan untuk mengetahui identitas foto seseorang dengan wajah tanpa masker dan dengan masker.
- d. Identifikasi wajah bermasker bertujuan untuk mengetahui ketepatan prediksi dari identitas foto dan tingkat keyakinan yang di hasilkan.

### 1.5. Manfaat Penelitian

Bagian ini memuat penjelasan tentang:

- a. Manfaat dari deteksi atau detektor wajah bermasker dengan OpenCV DNN dan Tensorflow adalah untuk mengetahui gambar wajah bermasker dan tidak mengenakan masker terdeteksi atau tidak.
- b. Manfaat dari kegunaan atau kemampuan *Multi-task Cascaded Algoritma Convolutional Neural Network* yaitu untuk mengenali identitas foto seseorang dengan masker dan tanpa masker.
- c. Dapat mengetahui tingkat akurasi dari deteksi wajah bermasker atau tidak bermasker yang dihasilkan pada detektor wajah.
- d. Dapat mengetahui nilai *confidence* dari ketepatan prediksi yang dihasilkan pada identifikasi wajah dengan masker.
- e. Sebagai media dalam mengembangkan ilmu pengetahuan teknologi dalam bidang *Deep Learning*.
- f. Sebagai acuan bagi mahasiswa, terutama untuk mahasiswa yang ingin melakukan penelitian yang sama.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Penelitian yang membahas tentang pengenalan wajah menggunakan masker seperti *face mask detection* dilakukan untuk identifikasi pada orang - orang tidak mengenakan masker di masa pandemi covid-19 ini. Penelitian yang dilakukan oleh (Chowdary et al., 2020), penelitian yang melakukan identifikasi pada orang - orang tidak mengenakan masker menggunakan model pembelajaran *transfer learning InceptionV3* untuk proses otomatisasi. Penelitian ini menyempurnakan *pre-trained state-of-the-art deep learning model, InceptionV3* diuji pada Simulasi *Masked Face Dataset (SMFD)* yang terdiri dari 1570 gambar dimana 785 simulasi gambar wajah dengan masker dan 785 gambar wajah yang tidak bermasker. Dari kumpulan data ini, 1099 gambar, kedua kategori digunakan untuk pelatihan dan 470 gambar sisanya digunakan untuk pengujian model, dengan tingkat akurasi sebesar 99,92%, 99,9% selama pelatihan, dan 100%, 100% selama pengujian pada set data SMFD. Penelitian serupa dilakukan oleh (Militante et al., 2020), penelitian ini membahas pengenalan masker wajah dengan sistem alarm melalui teknik *deep learning* dengan cara *Convolutional Neural Networks* secara *real-time*. Dataset yang dikumpulkan berisi gambar pola 25.000 menggunakan resolusi piksel 224x224. Pengumpulan datanya terdiri dari seseorang yang mengenakan masker wajah dan tidak mengenakan masker wajah kemudian klasifikasi menggunakan arsitektur CNN dan model yang digunakan yaitu VGG-16 CNN dengan akurasi validasi 96%

tercapai selama pelatihan model CNN. Penelitian terkait oleh (Anwar & Raychowdhury, 2020), penelitian ini membahas tentang kaitannya dengan banyak organisasi yang menggunakan pengenalan wajah sebagai sarana otentikasi dan telah mengembangkan dataset yang diperlukan secara in-house. Sayangnya, wajah dengan masker menyulitkan untuk dideteksi dan diakui, sehingga mengancam untuk membuat kumpulan data in-house tidak valid dan membuat sistem pengenalan wajah seperti itu tidak dapat dioperasikan. Penelitian ini memanfaatkan Alat *open-source MaskTheFace* untuk menghasilkan dataset wajah dengan masker dari kumpulan data wajah dengan dukungan fitur yang diperluas. Kemudian dataset yang digunakan yaitu *Masked Faces in Real World for Face Recognition (MRF2)* dengan hasil yang diperoleh dari penggunaan *MaskTheFace*, dilaporkan mengalami peningkatan  $\sim 38\%$  dalam sistem Facenet untuk wajah "masker" maupun tidak "tanpa masker". Keakuratan dilatih ulang sistem juga diuji pada kumpulan data MFR2 dan melaporkan keakuratan yang sama, sehingga dapat memperpanjang untuk wajah dengan masker di kehidupan nyata. Penelitian serupa oleh (Loey et al., 2021), menggunakan model untuk deteksi masker wajah pada berbagai dataset. Penelitian terkait tentang hybrid model untuk deteksi masker wajah. dengan menggunakan tiga komponen, pertama untuk ekstraksi fitur menggunakan *Resnet50*, kedua dirancang untuk proses klasifikasi masker wajah menggunakan *decision trees, Support Vector Machine (SVM), and ensemble algorithm*. Tiga dataset yang digunakan dalam penelitian ini yaitu datasets *are the Real-World Masked Face Dataset (RMFD), the Simulated Masked Face Dataset (SMFD), dan the Labeled Faces in the Wild (LFW)*. 70% untuk pelatihan, 10% untuk validasi,

20% untuk tahap pengujian. Uji coba eksperimental dilakukan pada komputer server dilengkapi oleh prosesor Intel Xeon (2 GHz), RAM 96 GB, perangkat lunak MATLAB. Dengan hasil pengklasifikasi SVM di RMFD mencapai akurasi pengujian 99,64%, dalam SMFD, ia memperoleh 99,49%, sementara di LFW, mencapai akurasi pengujian 100%. Penelitian dari (Guillermo et al., 2020), terkait penggunaan studi Prajna Bhandary tentang penambahan wajah masker untuk membuat datasetnya. Metodologi yang digunakan dalam penelitian ini meliputi beberapa fase utama yaitu wajah buatan pembuatan set data masker, pelatihan detektor masker wajah, dan pengujian detektor masker wajah. Untuk mengklasifikasi menggunakan MobileNetV2. Penelitian ini juga menggunakan OpenCV dan TensorFlow. Hasil dari penelitian mendapatkan akurasi 99% dalam semua pelatihan dan pengujian parameter utama.

Penelitian lain mengenai pengenalan wajah oleh (Damale et al., n.d.), menggunakan DNN sebagai deteksi wajah, dengan tiga metode yaitu SVM, MLP dan CNN, peneliti menggunakan dataset dari 11 orang yang dikumpulkan kedalam database dengan menangkap dari video dengan posisi melihat ke arah yang berbeda. Dataset tersebut dibuat melalui kamera 13MP ponsel, setiap kelas berisi 234 gambar, dengan resolusi gambar 244x244, dan diperoleh akurasi pengujian sekitar 87%, 86,5% dan 98%. Penelitian serupa yang dilakukan oleh (Sun et al., n.d.), yaitu *Face Identification and Verification*, dengan menggunakan fitur verifikasi *Deep IDentification (DeepID2)*, ada dua aspek membangun fitur ideal untuk pengenalan wajah, yaitu, meningkatkan variasi interpersonal dan mengurangi variasi intrapersonal, dalam penelitian verifikasi wajah pada set data LFW dan

mendapatkan akurasi verifikasi wajah mencapai 99,15%. Penelitian lain dari (Zhan et al., 2016) menggunakan kombinasi CNN dan SVM sebagai sistem deteksi wajah, dalam sistem yang diusulkan, fitur yang diekstrak oleh CNN dapat menjadi representasi yang baik untuk deteksi wajah, ditambah dengan keuntungan SVM untuk klasifikasi. Penelitian serupa oleh (Arsal et al., n.d.) membahas tentang pengenalan wajah untuk pintu akses ruangan perkantoran oleh pegawai bank. Dalam penelitian ini menggunakan algoritma *Convolutional Neural Network (CNN)* dan 350 data wajah dari 5 orang dataset wajah pegawai bank sebagai dataset. Hasil yang diperoleh dari mengidentifikasi wajah yang ditangkap oleh kamera dengan persentase keakuratan 95%.

Penelitian yang dilakukan (Cahyono et al., n.d.), terkait penggunaan FaceNet model sebagai pengenalan wajah dengan dukungan Support Vector Mesin (SVM) sebagai klasifikasi dan memperoleh akurasi sempurna yaitu 100%. Pada penelitian lain (Jose et al., 2019), dengan menggunakan FaceNet dan MTCNN di Jetson TX2 sebagai sistem pengawasan berbasis pengenalan wajah didapatkan deteksi akurasi 97%. Kemudian penelitian yang dilakukan oleh (Nyein et al., n.d.), menggunakan kombinasi FaceNet dan Support Vector Machine (SVM) untuk pengenalan multi wajah dan diperoleh akurasi 99,6%. Penelitian terkait dilakukan oleh (Rosal et al., n.d.), membahas tentang FaceNet dengan menggunakan dua jenis model pre-trained yaitu CASIA-WebFace dan VGGFace2.

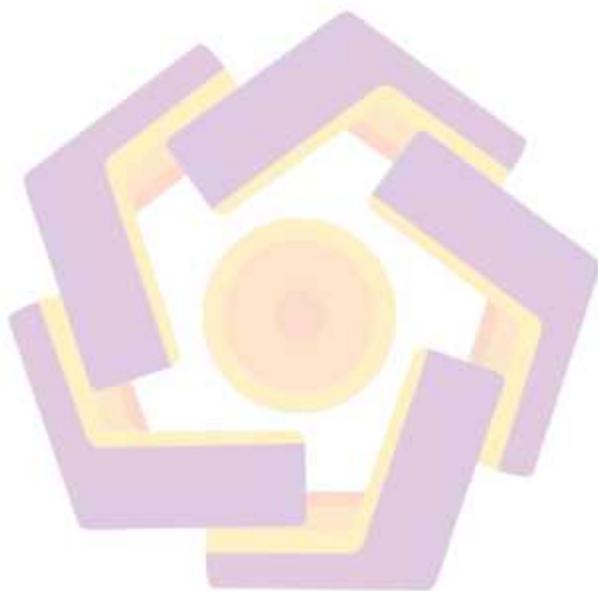
Tes dilakukan pada beberapa set data publik, seperti YALE, JAFFE, AT & T, Georgia Tech, dan Essex. Hasil FaceNet mampu menyentuh akurasi 100% pada beberapa dataset yang diujikan.

Penelitian Lain terkait penggunaan metode *Multi-task Cascaded Convolutional Neural Network*, oleh (Chen et al., 2019), penelitian ini membahas algoritma lokalisasi mata yang didasarkan pada deteksi wajah dengan MTCNN. Dengan dataset 860 BioID sebagai set pelatihan, 300 BioID dataset yang berbeda dari kumpulan pelatihan digunakan sebagai validasi model, dan sisanya digunakan sebagai set pengujian. Metode lokasi titik kunci berdasarkan Wajah MTCNN secara eksperimental divalidasi dalam kondisi kualitas gambar yang berbeda. Hasil dari proyeksi integral digabungkan ke posisi pupil manusia secara akurat. Dan memiliki akurasi deteksi 95.02%, serta memiliki ketahanan yang baik untuk deteksi gambar mata dengan kualitas yang berbeda. Penelitian terkait oleh (Yin & Liu, 2018), penelitian ini membahas tentang pengenalan *face Multi-Task Learning (MTL)* dengan *Convolutional Neural Network (CNN)*, *multi-task* untuk pengenalan wajah dimana diklasifikasikan identitas adalah tugas utama, dan estimasi Pose, Illumination, and Expression (*PIE*) adalah tugas sampingan. Penggunaan metode MTCNN yang dilakukan dengan gambar wajah tanpa penghalang masker (no mask), sedangkan topik penelitian yang akan dilakukan, metode digunakan dengan pengujian pada set data tidak hanya no mask tetapi juga with mask.

Penelitian lain dari mengenai deteksi oklusi pada gambar wajah oleh (L. Song et al., n.d.), peneliti menggunakan *pairwise differential siamese network (PDSN)* untuk mempelajari korespondensi antara blok wajah yang teroklusi dan

elemen fitur yang rusak, pada gambar wajah dengan random partial occlusions. penelitian ini juga menggunakan MTCNN sebagai landmark wajah untuk semua gambar di tahap preprocessing sebelum melakukan deteksi oklusi. Faktor utama yang mempengaruhi sistem pengenalan wajah adalah pose, iluminasi, penuaan, oklusi serta ekspresi dan diantaranya oklusi paling masalah yang mempengaruhi dalam pengenalan wajah (Khadatkar et al., n.d.). Penelitian terkait oleh (Kim et al., n.d.), membahas deteksi oklusi wajah yang digunakan untuk verifikasi oklusi wajah pada aplikasi mesin *teller otomatis* (ATM). Dengan metode deteksi kepala dan bahu, oklusi area wajah ditentukan dengan mengukur rasio area warna kulit (SCAR) dari seluruh area wajah dan wajah area komponen. Evaluasi performance dilakukan dengan menggunakan 120 video urutan 8 subjek (40 urutan video untuk tiga kasus: tidak ada oklusi, oklusi mulut oleh masker dan oklusi mata oleh kacamata hitam). Hasil dari tingkat deteksi mencapai 94.8% meskipun ada berbagai jenis oklusi parah di wajah. Penelitian serupa dilakukan oleh (Xia et al., n.d.), menggunakan *Multi-Task CNN* dengan dataset yang memiliki jenis oklusi yang berbeda, yaitu oklusi mata oleh matahari, kacamata, oklusi wajah dengan topi atau helm, dan mulut atau hidung oklusi oleh masker. Wajah deteksi oklusi yang dilakukan dievaluasi pada dataset wajah AR dan FO, dimana masing-masing mencapai akurasi 100% dan 97,24%. Penelitian dari (Li et al., n.d.), menggunakan *Patch-Gated CNN* untuk pengenalan ekspresi wajah di bawah oklusi. Dengan RAF-DB dataset dimana data dengan oklusi beraneka warna, bentuk, dan posisi. Dari hasil analisis ablasi menunjukkan PG-CNN mampu mengalihkan perhatian dari tambalan yang teroklusi ke yang terkait lainnya. Proses oklusi terkait dengan penelitian yang akan

dilakukan adalah satu bagian oklusi yaitu pada bagian pada mulut dan hidung yang tertutupi oleh masker.



## 2.2. Keaslian Penelitian

Tabel 2.1 Matriks literatur review dan posisi penelitian  
Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network

| No | Judul   | Peneliti, Media Publikasi, dan Tahun  | Tujuan Penelitian   | Kesimpulan  | Saran atau Kelemahan | Perbandingan   |
|----|---|---|---|---|----------------------|--|
| 1  | <i>Face Mask Detection using Transfer Learning of InceptionV3</i> | Chowdary, G.J. et al, Computer Vision and Pattern Recognition(cs.CV) arXiv:2009.08369 [cs.CV], 2020 | Adapun tujuan dari penelitian ini adalah untuk mengotomatiskan proses identifikasi orang – orang yang tidak mengenakan masker | Makalah ini memuat hasil dari Model Transfer Learning of Inception V3 yang diusulkan mencapai akurasi dan kekhususan sebesar 99,92%,99,9% selama pelatihan, dan 100%, 100% selama pengujian pada set data SMFD. | -                    | Perbandingan dengan penelitian yang akan dilakukan adalah dalam penelitian ini dataset yang dilakukan adalah dataset <i>Simulated Masked Face (SMFD)</i> . Sedangkan untuk penelitian yang akan dilakukan menggunakan tiga dataset dimana dataset1 data dengan identitas wajah, dataset2 dataset wajah "tanpa masker", dataset3 data dengan identitas dari seseorang dengan masker. Selain itu dalam penelitian ini menggunakan model Inception V3 sedangkan yang akan dilakukan menggunakan facenet model. Penelitian ini tidak mengidentifikasi siapa yang berada di balik wajah dengan masker tersebut. Sedangkan penelitian yang akan dilakukan mengidentifikasi siapa orang yang berada di balik wajah bermasker. |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul  | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian   | Kesimpulan  | Saran atau Kelemahan | Perbandingan  |
|----|--|--|---|---|----------------------|---|
| 2  | <i>Masked Face Recognition for Secure Authentication</i> | Anwar, Aqeel, and Arijit Raychowdhury, <i>arXiv preprint arXiv:2008.11104</i> , 2020 | Penelitian ini bertujuan untuk mengenali identitas secara akurat baik dengan maupun tanpa masker. | Kesimpulan dari penelitian ini adalah hasil dari penelitian ini melaporkan peningkatan ~38% dalam tingkat positif sejati untuk sistem Facenet yang ada untuk baik wajah bermasker maupun tanpa masker. Keakuratan dilatih ulang sistem juga diuji pada kumpulan data dunia nyata kustom MFR2 dan melaporkan keakuratan yang sama, sehingga dapat memperpanjang untuk wajah bertopeng kehidupan nyata. | -                    | Pada penelitian ini menggunakan alat open-source <i>MaskTheFace</i> untuk menghasilkan wajah dengan dataset wajah dengan masker. Dataset yang peneliti gunakan yaitu MRF2 sedangkan yang akan dilakukan menggunakan dataset yang peneliti dikumpulkan sendiri. Pengenalan wajah yang dilakukan yaitu identifikasi wajah dengan maupun tanpa masker, sedangkan penelitian yang akan dilakukan yaitu pengenalan wajah bermasker beserta siapa orang yang berada di balik masker tersebut. |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul   | Peneliti, Media Publikasi, dan Tahun  | Tujuan Penelitian  | Kesimpulan  | Saran atau Kelemahan   | Perbandingan   |
|----|---|---|--|---|--|--|
| 3  | <i>Real-Time Facemask Recognition with Alarm System using Deep Learning</i> | Militante, Sammy V., and Nanette V, 2020<br><i>11th IEEE Control and System Graduate Research Colloquium (ICSGRC)</i> , IEEE, 2020. | Tujuan mendeteksi orang yang mengenakan masker wajah dan tidak mengenakan masker wajah secara real-time dengan sistem alarm. | Kesimpulan dari makalah ini mempresentasikan studi tentang real-time pengenalan masker wajah dengan sistem alarm melalui teknik pembelajaran dengan cara <i>Convolutional Neural Network</i> . Dan hasil tes menunjukkan tingkat akurasi dalam mendeteksi orang yang mengenakan masker wajah dan tidak mengenakan masker wajah, menggunakan model VGG-16 CNN yang mencapai hasil 96% untuk akurasi kinerja. | Disarankan Integrasi beberapa model CNN dan membandingkan setiap model dengan akurasi kinerja tertinggi selama pelatihan untuk meningkatkan kinerja dalam mendeteksi dan mengenali orang yang mengenakan masker wajah. Dan juga menyarankan para peneliti merekomendasikannya. | Penelitian ini melakukan face mask recognition secara real-time sedangkan yang akan dilakukan tidak melakukan proses secara real-time, peneliti juga menggunakan model VGG-16 CNN sedangkan yang akan dilakukan menggunakan facenet model. Pada penelitian ini tidak menjelaskan mengenai identifikasi siapa orang di balik wajah bermasker. Sedangkan yang akan dilakukan melakukan identifikasi siapa wajah yang berada di balik wajah bermasker tersebut. |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul  | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian  | Kesimpulan  | Saran atau Kelemahan | Perbandingan  |
|----|--|--|--|---|----------------------|---|
| 4  | <i>A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic</i> | Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, Nour Eldeen M. Khalifa, Elsevier, 2020 | Penelitian ini bertujuan untuk mengetahui akurasi dari deteksi masker wajah. | Dalam makalah ini, hibrida model menggunakan deep dan classical machine learning untuk deteksi masker wajah dengan model yang diusulkan terdiri dari dua komponen, tiga dataset yaitu Real-World Masked Face Dataset (RMFD), the Simulated Masked Face Dataset (SMFD), and the Labeled Faces in the Wild (LFW). Dengan pengklasifikasi SVM mencapai akurasi pengujian 99,64% dalam RMFD. In SMFD, mencapai 99,49%, sementara di LFW, mencapai akurasi pengujian 100%. | -                    | Pada penelitian ini menggunakan metode A hybrid deep transfer learning model with machine learning dengan tiga dataset yaitu Real-World Masked Face Dataset (RMFD), the Simulated Masked Face Dataset (SMFD), and the Labeled Faces in the Wild (LFW). sedangkan yang akan dilakukan menggunakan metode Multi-task Cascaded Convolutional Neural Network dengan pengumpulan data lokal. |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul  | Peneliti, Media Publikasi, dan Tahun     | Tujuan Penelitian   | Kesimpulan   | Saran atau Kelemahan   | Perbandingan   |
|----|--|--|---|--|--|--|
| 5  | <i>COVID-19 Risk Assessment through Multiple Face Mask Detection using MobileNetV2 DNN</i> | Guillermo, Marielet, et al, ISCHIA, 2020 | Penelitian ini bertujuan untuk mempromosikan pentingnya pengendalian penyakit dan tindakan pencegahan seperti penggunaan masker wajah di tempat-tempat keramaian. | Penelitian ini menghasilkan 99% dalam semua pelatihan dan pengujian parameter utama. | Peneliti menyarankan bagi peneliti di masa depan untuk memperpanjang studi untuk memiliki analitik real time untuk membantu orang-orang di otoritas merespons jauh lebih cepat terhadap situasi pandemi. | Penelitian ini menggunakan studi Prajna Bhandary untuk menciptakan kumpulan data buatan sedangkan yang akan dilakukan tidak menciptakan tetapi Prajna Bhandary. Penelitian ini sebatas mendeteksi wajah bermasker atau tidak sedangkan yang akan diteliti selain mendeteksi wajah bermasker juga melakukan identifikasi siapa wajah yang menggunakan masker tersebut |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul   | Peneliti, Media Publikasi, dan Tahun                                     | Tujuan Penelitian  | Kesimpulan  | Saran atau Kelemahan   | Perbandingan   |
|----|---|--|--|---|--|--|
| 6  | <i>Face recognition based attendance system using machine learning algorithms</i> | Damale, Radhika C., and Bazeshree V. Pathak. IEEE, 2018.                 | Untuk mengetahui akurasi dari metode SVM, MLP dan CNN dalam melakukan pengenalan wajah | Dari metode SVM, MLP dan CNN menghasilkan akurasi pengujian 87%, 86,5% dan 98% pada database yang dihasilkan sendiri oleh peneliti.   | Peneliti menyarankan di masa depan, menggunakan data wajah yang berbeda lingkungan sebagai klasifikasi untuk meningkatkan akurasi. | Pengenalan wajah yang dilakukan menggunakan metode SVM, MLP dan CNN pada sekumpulan data wajah sedangkan yang akan dilakukan peneliti menggunakan metode MTCNN sebagai identifikasi wajah bermasker. |
| 7  | <i>Deep learning face representation by joint identification-verification</i>     | Sun, Yi, et al. Advances in neural information processing systems. 2014. | Untuk mengurangi variasi intrapersonal sambil memperbesar perbedaan interpersonal      | Efek dari identifikasi wajah dan sinyal pengawas verifikasi representasi fitur mendalam bertepatan dengan dua aspek membangun fitur ideal untuk pengenalan wajah, yaitu, meningkatkan variasi interpersonal dan mengurangi variasi dan intrapersonal hasil akurasi verifikasi wajah 99,15% pada LFW | -  | Penelitian ini melakukan representasi wajah dengan Identification verification features (Deep ID2). Sedangkan penelitian yang akan dilakukan melakukan identifikasi wajah dengan MTCNN               |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul  | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian  | Kesimpulan  | Saran atau Kelemahan   | Perbandingan   |
|----|--|--|--|---|--|--|
| 8  | <i>Face detection using representation learning</i>  | Zhan, Shu, Qin-Qin Tao, and Xiao-Hong Li, <i>Neurocomputing</i> 187, 2016  | Untuk fast face detection berdasarkan representasi convolutional neural network (CNN) sehingga secara eksplisit menangkap berbagai fitur wajah laten | Dalam sistem yang diusulkan, fitur yang diekstrak oleh CNN dapat menjadi representasi yang baik untuk deteksi wajah, ditambah dengan keuntungan SVM untuk klasifikasi, yang membuat pendekatan yang diusulkan semakin meningkatkan tingkat deteksi.   | -  | Penelitian ini melakukan face detection dengan pembelajaran representasi sedangkan yang akan dilakukan identifikasi wajah bermasker menggunakan MTCNN  |
| 9  | <i>Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN</i> | Arsal, Muhammad, Bhetu Agus Wardijono, and Dina Angraini, <i>Jurnal Nasional Teknologi dan Sistem Informasi</i> , 2020 | Untuk sistem keamanan pintu akses pegawai bank dengan menggunakan face recognition Dengan Metode CNN   | Penelitian ini berhasil menggunakan Face Recognition dengan 5 orang dataset wajah pegawai bank yang terdiri dari 70 data wajah pada masing-masing orang. Hasil dari pengujian dataset berhasil mengidentifikasi wajah yang ditangkap oleh kamera dengan persentase keakuratan 95%. Program pada penelitian ini berhasil digunakan oleh sebuah bank untuk pintu akses ruangan perkantoran oleh pegawai bank. | Program Face Recognition dalam penelitian dapat digunakan dalam sistem keamanan yang lebih besar serta ruang lingkup yang lebih luas sesuai dalam perkembangan dunia teknologi | Penelitian yang dilakukan berkaitan dengan face recognition pada kumpulan dataset wajah tanpa masker. Sedangkan yang akan di teliti face recognition juga dilakukan pada wajah dengan masker |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul  | Peneliti, Media Publikasi, dan Tahun  | Tujuan Penelitian   | Kesimpulan   | Saran atau Kelemahan | Perbandingan   |
|----|--|---|---|--|----------------------|--|
| 10 | <i>Face Recognition System using Facenet Algorithm for Employee Presence</i>             | Cahyono, Ferry, Wirawan Wirawan, and Reza Fuad Rachmadi, 2020 4th International Conference on Vocational Education and Training (ICOVET). IEEE, 2020. | Untuk membandingkan kinerja model FaceNet dan Openface untuk sistem pengenalan wajah yang memiliki akurasi tinggi. Hal ini bertujuan untuk menentukan tingkat keakuratan model pada kumpulan data wajah karyawan.             | Pengenalan wajah menggunakan FaceNet dan Support Vector Mesin (SVM) memiliki akurasi sempurna yaitu 100% sedangkan akurasi yang diperoleh dari Openface memiliki nilai akurasi 93,3%. Yang artinya FaceNet memiliki akurasi yang lebih baik ketimbang Openface | -                    | Penelitian ini melakukan perbandingan model arsitektur antara FaceNet dan Openface untuk menemukan akurasi tinggi untuk pengenalan wajah dengan bantuan SVM sebagai klasifikasi sedangkan penelitian yang akan dilakukan menggunakan model facenet untuk feature extraction untuk pengenalan wajah bermasker maupun tanpa masker |
| 11 | <i>Face recognition based surveillance system using facenet and mtcnn on jetson tx2.</i> | Jose, Edwin, et al, 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS). IEEE, 2019.                             | Untuk pengenalan wajah multicamera yang cerdas pada sistem pengawasan, dimaksudkan untuk mengurangi kejahatan dan meningkatkan keselamatan publik dan pribadi dengan menggunakan FaceNet dan MTCNN algoritma pada Jetson TX2. | Sistem pengawasan berbasis pengenalan Wajah menggunakan FaceNet dan MTCNN di Jetson TX2 berhasil diimplementasikan dengan beberapa kamera dan berhasil mendeteksi dengan akurasi 97%.  | -                    | Penelitian ini melakukan face recognition menggunakan FaceNet dan MTCNN yang diterapkan pada Jetson TX2 with multiple cameras. Sedangkan penelitian yang akan dilakukan adalah untuk melakukan identifikasi wajah pada kumpulan data wajah bermasker maupun tanpa masker.  |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No  | Judul   | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian  | Kesimpulan  | Saran atau Kelemahan | Perbandingan   |
|-----|---|--|--|---|----------------------|--|
| 12. | <i>University Classroom Attendance System Using FaceNet and Support Vector Machine.</i> | Nyein, Thida, and Aung Nway Oo, 2019<br><i>International Conference on Advanced Information Technologies (ICAIT)</i> . IEEE, 2019. | Untuk mendapatkan akurasi yang lebih baik untuk pengenalan multi-wajah dengan menggunakan kombinasi FaceNet dan Support Vector Machine (SVM)   | Pengenalan beberapa wajah saat FaceNet digunakan sebagai ekstraktor fitur dan SVM digunakan sebagai pengklasifikasi menunjukkan bahwa pendekatan yang diusulkan cukup baik untuk pengakuan multi-wajah dengan akurasi 99,6%.  | -                    | Penelitian ini melakukan pengenalan multi wajah sedangkan penelitian yang akan dilakukan meliputi identifikasi wajah pada setiap individu baik data wajah bermasker maupun tanpa masker  |
| 13. | <i>Face Recognition using FaceNet (Survey, Performance Test, and Comparison)</i>        | William, Ivan, et al, 2019<br><i>Fourth International Conference on Informatics and Computing (ICIC)</i> . IEEE, 2019.             | Penelitian ini bertujuan untuk melakukan survei, menguji kinerja, dan membandingkan keakuratan hasil mengenali wajah metode FaceNet dengan berbagai metode lain yang telah dikembangkan sebelumnya. (LTDF_MLDN + NNC, PCA and KPCA, sgFKNN, PCA + SVM) | FaceNet menunjukkan hasil yang sangat baik dan lebih unggul daripada metode lain. Dengan menggunakan VGGFace2 pra-terlatih model, FaceNet mampu menyentuh akurasi 100% pada YALE, Kumpulan data JAFFE, AT & T, Essex faces95, Essex grimace, 99,375% untuk essex faces94 dataset dan terburuk 77.67% untuk kumpulan data faces96. | -                    | Penelitian ini melakukan survey dan perbandingan keakuratan hasil kinerja FaceNet dengan LTDF_MLDN + NNC, PCA and KPCA, sgFKNN, PCA + SVM, pada pengenalan wajah, sedangkan penelitian yang akan dilakukan meliputi identifikasi wajah pada data wajah bermasker maupun tanpa masker |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul  | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian  | Kesimpulan   | Saran atau Kelemahan | Perbandingan  |
|----|--|--|--|--|----------------------|---|
| 14 | <i>Eyes Localization Algorithm Based on Prior MTCNN Face Detection</i> | Chen, Xuefeng, et al, 2019 <i>IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)</i> . IEEE, 2019. | Untuk lokalisasi mata dan untuk menemukan pupil dengan tepat pada posisi mata manusia. Dengan kualitas gambar rendah | Posisi pupil diperoleh oleh MTCNN digunakan sebagai nilai kandidat. Priori. Dikombinasikan dengan proyeksi integral gradien horizontal dan proyeksi integral vertikal daerah alis, posisi pupil yang tepat. Poin penting mata manusia adalah diperoleh dengan deteksi tepi dan sudut. Hasil eksperimental menunjukkan bahwa algoritma yang diusulkan memiliki akurasi deteksi 95.02%, dan memiliki ketahanan yang baik untuk deteksi gambar mata dengan kualitas yang berbeda. Penelitian ini juga menyimpulkan Hasil bahwa algoritma yang diusulkan dapat memastikan akurasi tinggi dan membutuhkan lebih sedikit waktu. Karenanya dapat memenuhi persyaratan keandalan dan validitas lokalisasi mata manusia dalam gambar resolusi rendah. | -                    | Perbandingan dengan penelitian yang akan dilakukan yaitu MTCNN yang digunakan untuk lokalisasi posisi mata manusia dengan kualitas rendah sedangkan yang akan dilakukan adalah untuk melakukan lokalisasi untuk pengenalan wajah bermasker. |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul   | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian  | Kesimpulan   | Saran atau Kelemahan | Perbandingan  |
|----|---|--|--|--|----------------------|---|
| 15 | <i>Multi-task convolutional neural network for pose-invariant face recognition.</i>                         | Yin, Xi, and Xiaoming Liu, <i>IEEE Transactions on Image Processing</i> , 2017                           | Dalam makalah ini, peneliti mempelajari pengenalan wajah sebagai masalah multi-tugas di mana klasifikasi identitas adalah tugas dengan estimasi PIE menjadi tugas sampingan. Tujuannya adalah untuk memanfaatkan tugas sampingan untuk meningkatkan kinerja tugas utama, yaitu pengenalan wajah. | Kumpulan data Multi-PIE menunjukkan keefektifan pendekatan yang diusulkan untuk pengenalan wajah yang invarian. Pendekatan yang dilakukan juga berlaku ke set data dalam liar untuk pengenalan wajah pose-invariant dan mencapai kinerja yang sebanding atau lebih baik daripada set data LFW, CFP, dan IJB-A. | -                    | Pada penelitian ini Jaringan saraf konvolusional multitugas digunakan untuk pengenalan wajah pose-invariant. Sedangkan yang akan dilakukan untuk identifikasi wajah bermasker             |
| 16 | <i>Occlusion robust face recognition based on mask learning with pairwise differential siamese network.</i> | Song, Lingxue, et al, <i>Proceedings of the IEEE International Conference on Computer Vision</i> . 2019. | Untuk menemukan dan membuang elemen fitur yang rusak dan pengakuan. Menggunakan jaringan siam diferensial pasangan (PDSN) yang dirancang secara inovatif.  | Metode pengenalan wajah yang kuat oklusi berdasarkan siam diferensial pairwise jaringan (PDSN), menunjukkan keunggulan metode yang diusulkan, terutama kemampuan generalisasi yang hebat pada tugas pengenalan wajah umum.   | -                    | Pada penelitian ini deteksi wajah oklusi pada gambar wajah dengan random-partial occlusions, sedangkan yang akan dilakukan yaitu satu bagian oklusi hidung dan mulut yang tertutup masker |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul   | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian   | Kesimpulan  | Saran atau Kelemahan  | Perbandingan   |
|----|---|--|---|---|---|--|
| 17 | <i>Occlusion invariant face recognition system</i>  | Khadatkar, Ashwin, Roshni Khedgaonkar, and K. S. Patnaik, 2016 <i>World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)</i> . IEEE, 2016. | Untuk mendeteksi oklusi dari wajah gambar dengan menggunakan pengklasifikasi SVM (Support Vector Machine).    | Untuk menyelesaikan masalah oklusi, setiap wajah dibagi menjadi k daerah yang dianalisis dalam isolasi dan membuang bagian oklusi dalam gambar wajah dan berdasarkan bagian yang tidak teroklusi pada gambar wajah untuk mengenali gambar wajah | Saran dari peneliti untuk pekerjaan di masa depan, dapat memberikan solusi untuk pose, iluminasi, masalah pencahayaan dan ekspresi. Kelemahan yang dikatakan pada penelitian ini yaitu Masalah oklusi yang telah diteliti cukup kurang dalam pengenalan wajah daripada pose, iluminasi, pencahayaan dan ekspresi. | Penelitian ini melakukan Sistem pengenalan wajah pada invaransi oklusi, sedangkan yang akan dilakukan yaitu satu bagian oklusi hidung dan mulut yang tertutup masker |
| 18 | <i>Face occlusion detection by using b-spline active contour and skin color information</i> | Kim, Gahyun, et al, 2010 <i>11th International Conference on Control Automation Robotics &amp; Vision</i> . IEEE, 2010.  | Untuk verifikasi oklusi wajah pada aplikasi mesin teller otomatis (ATM) dengan metode deteksi kepala dan bahu | Metode deteksi kepala dan bahu memiliki tingkat deteksi 94,8% meskipun ada berbagai jenis oklusi parah di wajah, dan verifier oklusi yang diusulkan memiliki tingkat verifikasi 86,7%.  | Saran peneliti di masa depan, untuk meningkatkan kinerja verifikasi oklusi dengan menggunakan pose kepala estimator dan detektor pemakai wajah  | Peneliti menggunakan metode deteksi kepala dan bahu untuk verifikasi oklusi wajah sedangkan peneliti menggunakan metode MTCNN  |

Tabel 2.1 Matriks literatur review dan posisi penelitian  
 Identifikasi wajah bermasker menggunakan algoritma Multi-task Cascaded Convolutional Neural Network (Lanjutan)

| No | Judul  | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian   | Kesimpulan  | Saran atau Kelemahan  | Perbandingan  |
|----|--|--|---|---|---|---|
| 19 | <i>Face occlusion detection based on multi-task convolution neural network</i> | Xia, Yizhang, Bailing Zhang, and Frans Coenen, 2015 <i>12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)</i> . IEEE, 2015. | Untuk deteksi oklusi wajah untuk meningkatkan fungsi pengawasan bagi ATM dengan Multi-task Convolution Neural Network | CNN diterapkan dengan strategi pra-pelatihan melalui wajah biasa tugas pengakuan. Model kemudian disempurnakan dengan wajah deteksi oklusi untuk memprediksi bagian wajah yang berbeda dan dievaluasi pada kumpulan data wajah AR dan FO dengan hasil masing-masing mencapai akurasi 100% dan 97,24%.   | -   | Dataset wajah AR dan FO yang digunakan peneliti memiliki jenis oklusi yang berbeda-beda sedangkan yang akan dilakukan yaitu hanya mulut atau hidung, oklusi oleh masker.              |
| 20 | <i>Patch-gated cnn for occlusion-aware facial expression recognition</i>       | Li, Yong, et al, 2018 <i>24th International Conference on Pattern Recognition (ICPR)</i> . IEEE, 2018.   | Untuk pengenalan ekspresi wajah di bawah oklusi dengan Patch-Gated Convolution yang dapat dilatih secara end-to-end   | Karya ini menyajikan CNN Patch-Gated untuk pengenalan ekspresi wajah di bawah oklusi. PG-CNN terdiri dari wilayah dekomposisi, Patch Gated Unit untuk pengenalan ekspresi wajah yang kuat. Eksperimen di bawah intra dan database silang protokol evaluasi menunjukkan PG-CNN mengungguli metode mutakhir. Analisis ablati menunjukkan PG-CNN adalah mampu mengalihkan perhatian dari tambalan yang tidak termasuk ke yang terkait. | Saran dari peneliti untuk pekerjaan di masa depan, mempelajari cara menghasilkan perhatian di wajah tanpa landmark, karena PG-CNN mengandalkan facial landmark localization | Penelitian ini melakukan pengenalan ekspresi wajah di bawah oklusi, sedangkan yang akan dilakukan yaitu identifikasi wajah di bawah oklusi yaitu hidung dan mulut oklusi oleh masker. |

### 2.3. Landasan Teori

#### 1. *Face Recognition*

Pengenalan wajah adalah bidang interdisipliner yang mengintegrasikan teknik dari gambar pengolahan, pengenalan pola, visi komputer, grafik komputer, psikologi, dan pendekatan evaluasi. Secara umum, pengenalan wajah terkomputerisasi mencakup empat Langkah. Pertama, gambar wajah ditingkatkan dan disegmentasi. Kedua, batas wajah dan fitur wajah terdeteksi. Ketiga, fitur wajah yang diekstraksi dicocokkan dengan fitur yang disimpan dalam database. Keempat, klasifikasi gambar wajah menjadi satu atau lebih banyak orang tercapai. Ekstraksi otomatis kepala manusia, batas wajah, dan fitur wajah berperan penting dalam bidang pengendalian akses, identifikasi kriminal, sistem pengawasan, interfacing komputer manusia, dan video berbasis model coding. Untuk mengekstrak fitur wajah, seseorang perlu terlebih dahulu mendeteksi wajah manusia dalam gambar (Frank Shih, 2010).

(Yang et al., n.d.), mengklasifikasikan metode deteksi wajah ke dalam empat kategori: berbasis pengetahuan, fitur invarian, pencocokan template, dan berbasis penampilan. (F. Song et al., 2007), mengusulkan beberapa kriteria diskriminan perbedaan maksimum untuk wajah ekstraksi fitur. (Sung & Poggio, 1994), menggunakan *cluster* Gaussian untuk memodelkan distribusi pola wajah dan non faset. (Rowley et al., n.d.), merancang neural algoritma berbasis jaringan untuk mendeteksi wajah tampilan frontal dalam gambar skala abu-abu. Menurut (Prahariezka Arfienda, 2018). *Facial recognition* adalah salah satu jenis sistem identifikasi “biometrik”. Sistem ini mengidentifikasi seseorang dengan fitur-fitur

khusus pada tubuh maupun DNA yang membedakan satu orang dengan orang lainnya. Contoh identifikasi biometrik lainnya adalah fingerprint atau sidik jari, retina scanning, iris scanning, serta *voice recognition*.

Menurut (Hjelmås & Low, 2001). Deteksi wajah adalah langkah pertama yang diperlukan dalam sistem pengenalan wajah, dengan tujuan melis lokalisasi dan mengekstraksi wilayah wajah dari latar belakang. Mendeteksi wajah dalam sebuah foto mudah dipecahkan oleh manusia, meskipun secara historis telah menantang komputer mengingat sifat wajah yang dinamis. Misalnya, wajah harus dideteksi terlepas dari orientasi atau sudut yang mereka hadapi, tingkat cahaya, pakaian, aksesoris, warna rambut, rambut wajah, riasan, usia, dan sebagainya. ada dua pendekatan utama untuk pengenalan wajah: metode berbasis fitur yang menggunakan filter buatan tangan untuk mencari dan mendeteksi wajah, dan metode berbasis gambar yang belajar secara holistik cara mengekstrak wajah dari seluruh gambar.



Gambar 2.1 Typical training image for face recognition

### 1.1. Machine learning

*Machine learning* adalah jenis kecerdasan buatan (AI) yang memungkinkan komputer untuk belajar dari data tanpa diprogram secara eksplisit. Dengan kata lain, tujuan pembelajaran mesin adalah merancang metode yang secara otomatis melakukan pembelajaran menggunakan pengamatan dunia (disebut "data pelatihan"), tanpa definisi eksplisit tentang aturan atau logika oleh manusia

("pelatih"/"supervisor"). Dalam pengertian itu, pembelajaran mesin dapat dianggap sebagai pemrograman oleh sampel data. Singkatnya, Machine learning adalah tentang belajar untuk melakukan yang lebih baik di masa depan berdasarkan pada apa yang dialami di masa lalu (Khan et al., 2018).

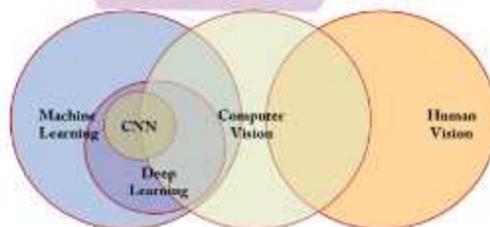
Seperangkat algoritma pembelajaran mesin yang beragam telah diusulkan untuk mencakup data dan jenis masalah. Metode pembelajaran ini terutama dapat dibagi menjadi tiga pendekatan, yaitu diawasi, semi-diawasi, dan tidak diawasi. Namun, sebagian besar metode pembelajaran mesin praktis saat ini diawasi metode pembelajaran, karena kinerjanya yang unggul dibandingkan dengan bagian kontra lainnya. Dalam metode pembelajaran yang diawasi, pelatihan data mengambil bentuk koleksi data:  $x$ ; label:  $y$ / pairs dan tujuannya adalah untuk menghasilkan prediksi  $y$  sebagai respons terhadap sampel kueri  $x$ . Input  $x$  dapat menjadi vektor fitur, atau data yang lebih kompleks seperti gambar, dokumen, atau grafik. Demikian pula, berbagai jenis output  $y$  telah dipelajari. Output  $y$  dapat menjadi label biner yang digunakan dalam masalah klasifikasi biner sederhana (misalnya, "ya" atau "tidak"). Namun, ada juga banyak penelitian yang bekerja pada masalah seperti sebagai klasifikasi multi-kelas di mana  $y$  diberi label oleh salah satu label  $k$ , klasifikasi multi-label di mana  $y$  mengambil secara bersamaan label  $K$ , dan masalah prediksi terstruktur umum di mana  $y$  output dimensi tinggi, yang dibangun dari urutan prediksi (misalnya, semantik segmentasi).

#### a. *Deep learning*

Maneurut (Khan Salman et al. 2018). Ada tiga keuntungan utama yang ditawarkan oleh *deep learning* yaitu:

1. Kesederhanaan: Alih-alih masalah tweak spesifik dan detektor fitur yang disesuaikan, karya menawarkan blok arsitektur dasar, lapisan jaringan, yang diulang beberapa kali untuk menghasilkan jaringan besar.
2. Skalabilitas: Model pembelajaran mendalam mudah diskalakan ke set data besar. Pesaing lainnya metode, misalnya, mesin kernel, mengalami masalah komputasi serius jika set data sangat besar.
3. Transfer domain: Model yang dipelajari pada satu tugas berlaku untuk tugas terkait lainnya dan fitur yang dipelajari cukup umum untuk mengerjakan berbagai tugas yang mungkin langka data yang tersedia.

Karena keberhasilan yang luar biasa dalam mempelajari jaringan saraf yang mendalam ini, niques saat ini canggih untuk deteksi, segmentasi, klasifikasi dan pengakuan (yaitu, identifikasi dan verifikasi) objek dalam gambar. Peneliti sekarang bekerja untuk menerapkan keberhasilan ini dalam pengenalan pola untuk tugas yang lebih kompleks seperti diagnosa medis dan terjemahan bahasa otomatis. Gambar 2.1 menggambarkan hubungan antara visi komputer, pembelajaran mesin, penglihatan manusia, pembelajaran mendalam, dan CNN.

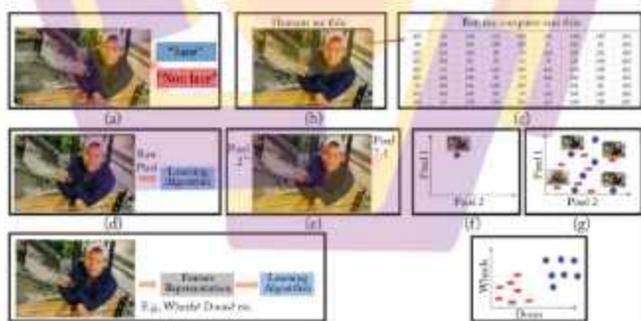


Gambar 2.2 Hubungan antara human vision, computer vision, machine learning, deep learning dan CNN.

### a. Feature extraction and classification

Ekstraksi fitur dan klasifikasi adalah dua tahap kunci dari *computer vision system* (Khan et al., 2018). Metode ekstraksi fitur dapat dibagi menjadi dua kategori yang berbeda, yaitu metode berbasis teknik dan metode berbasis pembelajaran fitur. Sedangkan pengklasifikasi dapat dibagi menjadi dua kelompok, yaitu model dangkal dan dalam. Beberapa pengklasifikasi tradisional terkenal misalnya, SVM (Cortes et al., 1995), RDF (Breiman.2001, Quinlan, 1986).

Pentingnya fitur dan pengklasifikasi adalah akurasi, ketahanan, dan efisiensi sistem penglihatan sangat tergantung pada kualitas fitur gambar dan pengklasifikasi. Ekstraktor fitur yang ideal akan menghasilkan representasi gambar yang membuat pekerjaan pengklasifikasi sepele lihat Gambar 2.3.



Gambar 2.3 Features and classifiers

Gambar 2.3: (a) Tujuannya adalah untuk merancang algoritma yang mengklasifikasikan gambar input ke dalam dua kategori: "face" atau "non-face." (b) Manusia dapat dengan mudah melihat face dan mengkategorikan gambar ini

sebagai "face." Namun, komputer melihat nilai intensitas piksel seperti yang ditunjukkan pada (c) untuk patch kecil di Gambar. Metode visi komputer memproses semua nilai intensitas piksel dan mengklasifikasikan gambar. (d) cara mudah adalah memberi makan nilai intensitas kepada pengklasifikasi dan classifier terpelajar akan kemudian melakukan pekerjaan klasifikasi. Untuk visualisasi yang lebih baik, hanya memilih dua piksel, seperti yang ditunjukkan di (e). Karena pixel 1 relatif cerah dan pixel 2 relatif gelap, gambar itu memiliki posisi ditampilkan dalam tanda plus biru dalam plot yang ditunjukkan di (f). Dengan menambahkan beberapa sampel positif dan negatif, plot dalam (g) menunjukkan bahwa sampel positif dan negatif sangat campur aduk bersama. Jadi jika data ini diumpankan ke pengklasifikasi linear, subdivisi ruang fitur menjadi dua kelas tidak mungkin. (h) Ternyata representasi fitur yang tepat dapat mengatasi masalah ini. Misalnya, menggunakan fitur yang lebih informatif seperti jumlah roda dalam gambar, jumlah pintu dalam gambar, data terlihat seperti (i) dan gambar menjadi jauh lebih mudah untuk mengklasifikasikan (Khan et al., 2018).

*Feature* adalah aspek atau karakteristik khas yang digunakan untuk menyelesaikan tugas komputasi terkait dengan aplikasi tertentu. Misalnya, mengingat gambar wajah, ada berbagai pendekatan untuk mengekstrak fitur, misalnya, rata-rata, varians, gradien, tepi, fitur geometris, fitur warna, dll. Kombinasi fitur  $n$  dapat direpresentasikan sebagai vektor  $n$ -dimensi, yang disebut vektor fitur. Kualitas vektor fitur tergantung pada kemampuannya untuk mendiskriminasi gambar sampel dari kelas yang berbeda. Sampel gambar dari kelas yang sama harus memiliki fitur serupa (Khan et al., 2018).

*Classification* adalah inti dari pengenalan komputer dan pola modern. Tugas klasifikasi adalah menggunakan vektor fitur untuk menampilkan gambar atau wilayah yang diminati (RoI) ke kategori. Tingkat kesulitan tugas klasifikasi tergantung pada variabilitas dalam nilai fitur gambar dari kategori yang sama, relatif terhadap perbedaan antara nilai fitur gambar dari kategori yang berbeda. Namun, kinerja klasifikasi yang sempurna seringkali tidak mungkin. Ini adalah terutama karena adanya kebisingan (dalam bentuk bayangan, oklusi, distorsi perspektif, dll.), outlier (misalnya, gambar dari kategori "bangunan" mungkin berisi orang, hewan, bangunan, atau kategori mobil), ambiguitas (misalnya, bentuk persegi panjang yang sama dapat sesuai dengan tabel atau jendela bangunan), kurangnya label, ketersediaan hanya sampel pelatihan kecil, dan ketidakseimbangan cakupan positif atau negatif dalam sampel data pelatihan (Khan et al., 2018).

### c. *Convolutional Neural Network*

*Convolutional Neural Network* adalah salah satu kategori jaringan saraf, terutama untuk data dimensi tinggi (misalnya, gambar dan video). CNN beroperasi dengan cara yang sangat mirip dengan jaringan saraf standar. Namun, perbedaan utama adalah bahwa masing-masing unit dalam lapisan CNN adalah filter dimensi dua (atau tinggi) yang berbelit-belit dengan input lapisan itu. Ini penting untuk kasus-kasus di mana kita ingin belajar pola dari dimensi tinggi media input, misalnya, gambar atau video. Filter CNN menggabungkan konteks spasial dengan memiliki (tetapi lebih kecil) bentuk spasial sebagai media input, dan gunakan

berbagi parameter untuk mengurangi jumlah variabel yang dapat dipelajari (Khan et al., 2018).

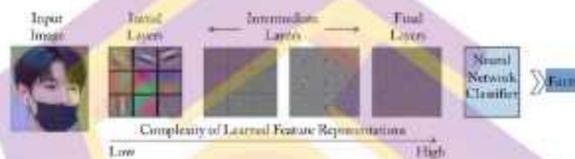
Menurut Suartika et al, 2016). *Convolutional Neural Network (CNN)* adalah pengembangan dari *Multilayer Perceptron (MLP)* yang termasuk dalam neural network bertipe *feed forward*. *Convolutional Neural Network* adalah neural network yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra.

Bentuk awal CNN adalah model Neocognitron yang diusulkan oleh (Fukushima & Miyake, 1982), ini terdiri dari beberapa lapisan yang secara otomatis hierarki abstraksi fitur untuk pengenalan pola. Neocognitron dimotivasi oleh karya mani oleh (Hubel & Wiesel, n.d.), pada korteks visual utama yang menunjukkan bahwa neuron di otak diatur dalam bentuk lapisan. Lapisan-lapisan ini belajar mengenali visual dengan terlebih dahulu mengekstrak fitur lokal dan kemudian menggabungkannya untuk mendapatkan representasi tingkat. Pelatihan jaringan dilakukan menggunakan aturan pembelajaran penguatan. Peningkatan besar atas Neocognitron adalah model LeNet yang diusulkan oleh (LeCun et al., 1989), di mana parameter model dipelajari menggunakan kesalahan back-propagation.

CNN adalah kelas model yang berguna untuk pembelajaran yang diawasi dan tidak diawasi. Paradigma. Mekanisme pembelajaran yang diawasi adalah yang mana input ke sistem dan output yang diinginkan (label sejati) diketahui dan model mempelajari pemetaan antara keduanya. Dalam mekanisme pembelajaran tanpa pengawasan, label sejati untuk sekumpulan input yang diberikan tidak diketahui

dan model ini bertujuan untuk memperkirakan distribusi yang mendasari sampel data input.

Contohnya bisa dilihat pada gambar 2.4. CNN belajar memetakan gambar tertentu ke kategori yang sesuai dengan mendeteksi sejumlah fitur abstrak representasi, mulai dari yang sederhana hingga yang lebih kompleks. Fitur diskriminatif ini kemudian digunakan dalam jaringan untuk memprediksi kategori gambar input yang benar. Pengklasifikasi jaringan saraf identik dengan MLP.



Gambar 2.4 Neural Network Classifier

Gambar 2.4: CNN mempelajari fitur tingkat rendah di lapisan awal, diikuti oleh representasi fitur menengah dan tingkat tinggi yang digunakan untuk tugas klasifikasi. Visualisasi fitur diadopsi dari (Wan et al., 2013), beberapa yang perlu diperhatikan sebagai berikut:

#### 4. Network layer

CNN terdiri dari beberapa blok bangunan dasar, yang disebut lapisan CNN. Di bagian ini, mempelajari blok bangunan dan fungsinya dalam arsitektur CNN. Perhatikan bahwa beberapa lapisan ini menerapkan fungsi dasar seperti *normalization, pooling, convolution, and fully connected layers*.

### 5. *Pre-processing*

Sebelum meneruskan data input ke jaringan, data perlu diproses sebelumnya. Langkah-langkah pra-pemrosesan umum yang digunakan terdiri dari yang berikut ini.

1. *Mean-subtraction*: Patch input (milik train dan set uji) adalah nol berpusat dengan mengurangi rata-rata yang dihitung pada seluruh set pelatihan. Diberikan pelatihan  $N$  gambar, Langkah pengurangan rata-rata sebagai berikut:

$$\mathbf{x}' = \mathbf{x} - \hat{\mathbf{x}}, \quad \text{where } \hat{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i. \quad \dots\dots\dots(1)$$

2. *Normalization*: Data input (milik train dan set uji) dibagi dengan simpangan baku setiap dimensi input (piksel dalam kasus gambar) dihitung pada pelatihan yang ditetapkan untuk menormalkan simpangan baku ke nilai unit. Hal ini dapat diwakili sebagai berikut:

$$\mathbf{x}'' = \frac{\mathbf{x}'}{\sqrt{\frac{\sum_{i=1}^N (\mathbf{x}_i - \hat{\mathbf{x}})^2}{N-1}}}. \quad \dots\dots\dots(2)$$

3. *PCA Whitening*: Tujuan pemutih PCA adalah untuk mengurangi korelasi antara dimensi data feren dengan menormalkannya secara independen. Pendekatan ini dimulai dengan data nol-tengah dan menghitung matriks kovarians yang mengkodekan korelasi dimensi data tween. Matriks kovarians ini kemudian diurai melalui Nilai Singular Value Decomposition (SVD) dan data diklat dengan memproyeksikannya ke eigen vectors ditemukan melalui SVD. Setelah

itu, setiap dimensi dibagi dengan eigen value untuk menormalkan semua dimensi masing-masing dalam ruang data.

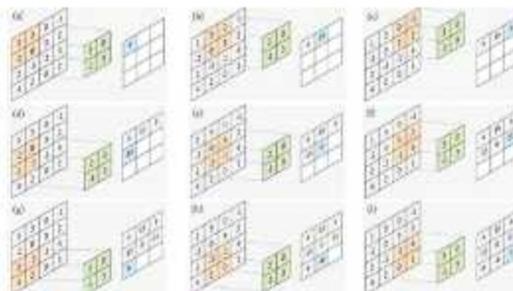
4. Local Contrast Normalization: Skema normalisasi ini mendapatkan motivasinya dari ilmu saraf. Seperti namanya, pendekatan ini menormalkan kontras lokal peta fitur untuk mendapatkan fitur yang lebih menonjol. Ini pertama kali menghasilkan lingkungan lokal untuk setiap piksel, misalnya, untuk radius unit delapan piksel tetangga dipilih.

a. *Convolutional layer*

Lapisan konvolusional adalah komponen terpenting dari CNN. Ini terdiri dari satu set filter (juga disebut kernel konvolusional) yang di konvolusi dengan input tertentu untuk menghasilkan peta output feature. Apa itu Filter? Setiap filter dalam lapisan konvolusional adalah kisi angka diskrit. Sebagai contoh, pertimbangkan filter 2 x 2 diperlihatkan pada gambar 2.5. Bobot setiap filter (angka dalam grid) dipelajari selama pelatihan CNN. Prosedur pembelajaran ini melibatkan inisialisasi bobot filter pada awal pelatihan (pendekatan yang berbeda untuk berat inisialisasi akan dibahas dalam Pasal 5.1). Setelah itu, diberikan pasangan input-output, filter bobot disetel dalam sejumlah iterasi yang berbeda selama prosedur pembelajaran (Khan et al., 2018).

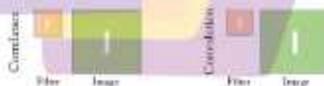
|    |   |
|----|---|
| 2  | 0 |
| -1 | 3 |

Gambar 2.5 Contoh filter gambar 2D



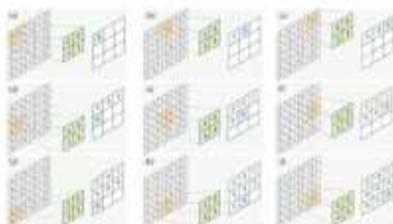
Gambar 2.6 The operation of a convolution layer is illustrated

Gambar 2.6 menjelaskan tentang menunjukkan komputasi yang dilakukan di setiap langkah, karena filter, di slide ke peta fitur input untuk menghitung nilai yang sesuai dalam peta fitur output. Filter 2x2 (ditampilkan dalam warna hijau) dikalikan dengan wilayah berukuran sama (ditunjukkan dalam *orange*) dalam peta fitur input 4x4 dan nilai yang dihasilkan dijumlahkan untuk mendapatkan entri yang sesuai (diperlihatkan dalam warna biru) dalam output peta fitur di setiap langkah konvolusi.



Gambar 2.7 The operation of a convolution layer is illustrated

Gambar 2.7 yaitu Perbedaan antara korelasi dan operasi konvolusi dalam sinyal literatur pengolahan. Dalam pembelajaran mesin, perbedaan ini biasanya tidak penting dan literatur pembelajaran biasanya mengacu pada lapisan yang menerapkan operasi korelasi sebagai operasi konvolusi.



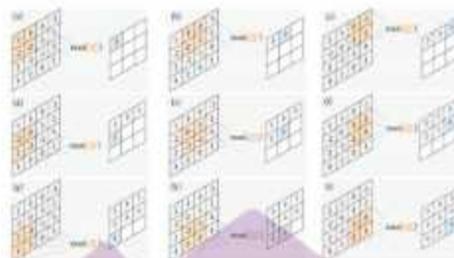
Gambar 2.8 The operation of a convolution layer with a zero padding

Gambar 2.8 memperlihatkan perhitungan yang dilakukan di setiap langkah, sebagaimana filter di slide ke peta fitur input untuk menghitung nilai output yang sesuai peta fitur. Filter  $2 \times 2$  (ditampilkan dalam warna hijau) dikalikan dengan bidang berukuran sama (diperlihatkan dalam *orange*) dalam peta fitur input  $6 \times 6$  (termasuk *zero-padding*) dan nilai yang dihasilkan dijumlahkan untuk mendapatkan entri yang sesuai (ditampilkan dalam warna biru) di peta fitur output di masing-masing langkah konvolusi.

#### 4.7 Pooling layer

Lapisan pooling beroperasi pada blok peta fitur input dan menggabungkan aktivasi fitur. Operasi kombinasi ini didefinisikan oleh fungsi pooling seperti rata-rata atau maks Fungsi. Mirip dengan lapisan konvolusi, kita perlu menentukan ukuran wilayah yang terkumpul dan langkah. Gambar 2.9 menunjukkan operasi pooling maksimum, dimana aktivasi maksimum dipilih dari blok nilai yang dipilih. Jendela ini di slide di seluruh peta fitur input dengan ukuran langkah yang ditentukan oleh langkah (1 dalam kasus Gbr. 2.9). Jika ukuran wilayah yang diberikan oleh  $f \times f$ , dengan langkah  $s$ , ukuran peta fitur output sebagai berikut:

$$h' = \left\lfloor \frac{h - f + s}{s} \right\rfloor, w' = \left\lfloor \frac{w - f + s}{s} \right\rfloor. \dots\dots\dots(3)$$



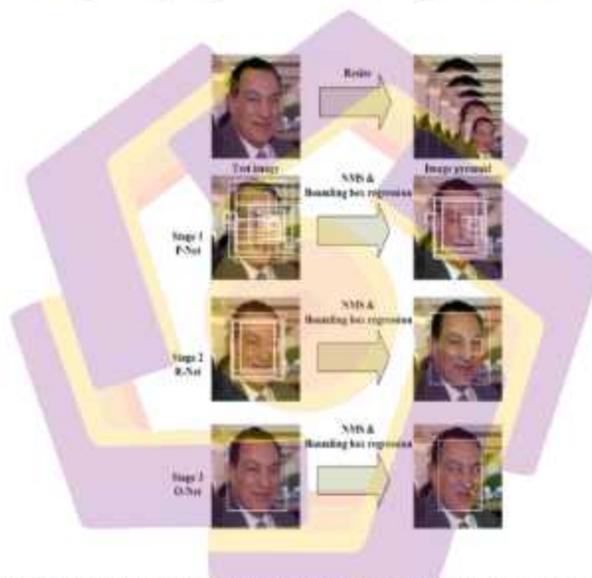
Gambar 2.9 The operation of max-pooling layer

Gambar 2.9 Pengoperasian lapisan *max-pooling* ketika ukuran wilayah pooling adalah  $2 \times 2$  dan langkahnya adalah 1. (a)–(i) memperlihatkan perhitungan yang dilakukan di setiap langkah sebagai wilayah yang dikumpulkan peta fitur input (diperlihatkan dalam oranye) di slide di setiap langkah untuk menghitung nilai yang sesuai di peta fitur output ditampilkan dalam warna biru (Khan et al., 2018).

#### f. *Multi-Task Cascaded Convolutional Neural Network*

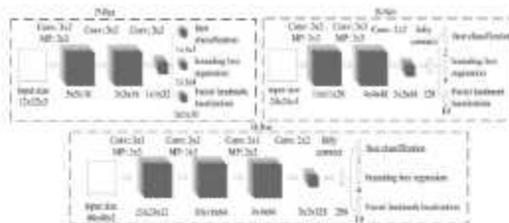
*Multi-Task Cascaded Convolutional Neural Network* (MTCNN) populer karena mencapai hasil canggih pada berbagai dataset patokan, dan karena mampu juga mengenali fitur wajah lainnya seperti mata dan mulut, yang disebut *landmark detection*. Jaringan menggunakan struktur bertingkat dengan tiga jaringan; pertama gambar di-rescaled ke berbagai ukuran yang berbeda (disebut piramida gambar), kemudian model pertama (*Proposal Network or P-Net*) mengusulkan kandidat daerah wajah, model kedua (*Refine Network or R-Net*) memfilter batas kotak, dan model ketiga (*Output Network or O-Net*) mengusulkan *facial landmarks* (Brown & Lowe, 2020). CNN yang diusulkan terdiri dari tiga tahap. Pada tahap pertama, ia

menghasilkan jendela kandidat dengan cepat melalui CNN dangkal. Kemudian, ia menyempurnakan jendela untuk menolak sejumlah besar jendela non-wajah melalui CNN yang lebih kompleks. Akhirnya, ia menggunakan CNN yang lebih kuat untuk memperbaiki hasil dan output *facial landmarks positions*. Gambar di bawah ini memberikan ringkasan yang bermanfaat dari tiga tahap dari atas ke bawah dan output setiap tahap kiri-ke-kanan (Zhang et al., 2016).



Gambar 2.10 Pipeline for the Multi-Task Cascaded Convolutional Neural Network

Model ini disebut *Multi-Task Cascaded Convolutional Neural Network* karena masing-masing dari tiga model dalam kaskade (*P-Net*, *R-Net* dan *O-Net*) dilatih pada tiga tugas, misalnya membuat tiga jenis prediksi; mereka adalah: klasifikasi wajah, *box regression*, dan *facial landmark localization*. Berikut adalah *Architectures* dari *P-Net*, *R-Net*, and *O-Net*.



Gambar 2.11 Architectures P-Net, R-Net, and O-Net

Gambar 2.11 Jenis, Sifat Arsitektur *P-Net*, *R-Net*, dan *O-Net*, di mana "MP" berarti max pooling dan "Conv" berarti konvolusi. Ukuran langkah dalam konvolusi dan pooling adalah 1 dan 2, masing-masing.

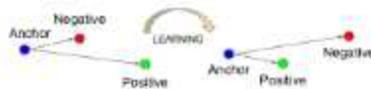
#### g. *FaceNet*

*FaceNet* adalah sistem pengenalan wajah yang dikembangkan pada tahun 2015 oleh para peneliti di Google yang mencapai hasil canggih pada berbagai kumpulan data benchmark pengenalan wajah. Sistem *FaceNet* dapat digunakan secara luas berkat beberapa implementasi *open source* pihak ketiga dari model dan ketersediaan model yang telah dilatih sebelumnya. Sistem *FaceNet* dapat digunakan untuk mengekstrak fitur berkualitas tinggi dari wajah, yang disebut penyematan wajah, yang kemudian dapat digunakan untuk melatih sistem identifikasi wajah. (Brownlee Jason, 2020). Lihat pada gambar 2.12 untuk mengetahui struktur modelnya.



Gambar 2.12 Model structure

Gambar 2.12 Struktur model. Jaringan terdiri dari menempatkan lapisan dan CNN yang dalam diikuti oleh normalisasi L 2, yang menghasilkan penyematan wajah. diikuti oleh *triplet loss* selama pelatihan.



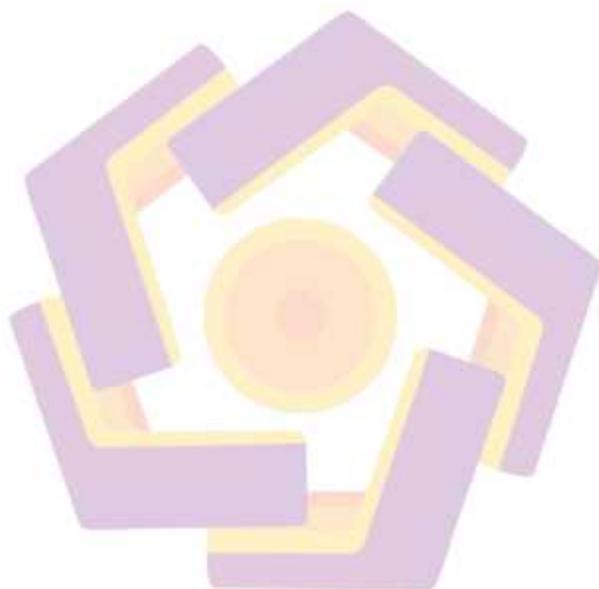
Gambar 2.13 Triplet Loss

Gambar 2.13 *Triplet loss* meminimalkan jarak antara dan positif, yang keduanya memiliki identitas yang sama, dan memaksimalkan jarak antara jangkak dan identitas yang berbeda (Schroff & Philbin, 2015).

#### h. TensorFlow

TensorFlow adalah antarmuka untuk mengekspresikan *Machine Learning Algorithms* dan implementasi untuk mengeksekusi algorithm tersebut. Komputasi yang diungkapkan menggunakan TensorFlow dapat diekspresikan dengan sedikit atau tidak ada perubahan pada berbagai sistem *hetero-geneous*, mulai dari perangkat seluler seperti ponsel dan tablet hingga sistem terdistribusi skala besar dari ratusan mesin dan ribuan perangkat komputasi seperti kartu GPU. Sistem ini fleksibel dan dapat digunakan untuk mengekspresikan berbagai algoritma, termasuk pelatihan dan inferensi algorithms untuk model deep neural network dan telah digunakan untuk melakukan penelitian dan untuk menyebarkan system belajar mesin ke dalam produksi di lebih dari selusin bidang ilmu komputasi dan bidang lainnya, termasuk *recognition, computer vision, robotics, information retrieval, natural language processing, geographic information extraction, and*

*computational drug discovery*. tensorflow API dan reference implementation dirilis sebagai paket open-source di bawah lisensi Apache 2.0 pada bulan November 2015 dan tersedia di [www.tensorflow.org](http://www.tensorflow.org) (Abadi et al., 2016).



## BAB III

### METODE PENELITIAN

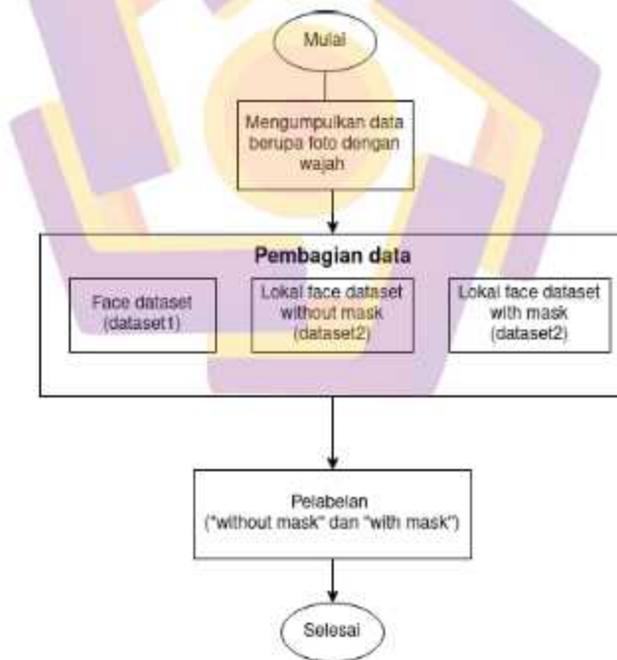
#### 3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini merupakan penelitian eksperimen dengan pendekatan kuantitatif dimana yang pertama dilakukan pelatihan deteksi masker wajah, dan pengujian deteksi masker wajah dengan menggunakan *deep learning* dan selanjutnya melakukan identifikasi dan verifikasi terhadap wajah bermasker serta tanpa masker dengan metode *Multi Task Cascade Convolutional Neural Network* sebagai deteksi wajah serta penggunaan *FaceNet model* untuk membuat penyematan (*embedding*) wajah untuk setiap wajah yang terdeteksi, dan mengembangkan model pengklasifikasi *Linear Support Vector Machine (SVM)* untuk memprediksi identitas wajah tersebut untuk memperoleh tingkat akurasi.

#### 3.2. Metode Pengumpulan Data

Dataset yang digunakan untuk penelitian ini terdiri dari tiga dataset yaitu dimana dataset1 merupakan *dataset wajah* yang diambil dari dataset (Prajna Bhandary, <https://github.com/prajnasb/observations/tree/master/experiments/data>) Dataset ini berupa kumpulan data wajah. Dimana datanya terdiri dari berbagai wajah dengan masker dan tanpa masker berjumlah 1,376 gambar, untuk gambar dengan masker 690 gambar dan tanpa masker 686 gambar (Prajna Bhandary, 2020), kedua dataset wajah tanpa masker(dataset2), dataset ini dikumpulkan sendiri dengan mengambil 62 koleksi gambar dengan kondisi gambar yang berbeda beda, pose, ukuran dan kualitas. Dataset yang dikumpulkan mencakup empat identitas

yang berbeda, yaitu, Aldi alkulakat, Wawan riyanto, Elfrida dan Yohan. Kemudian dari dataset ini dilakukan proses penutupan wajah dengan masker secara virtual dibantu dengan aplikasi *OpenSource GNU Image Manipulation Program* yang kemudian dinamakan sebagai dataset3 yaitu dataset wajah bermasker. Dan juga peneliti mengumpulkan dataset real dimana datanya berisi kumpulan wajah dengan penggunaan berbagai macam jenis masker beserta identitas (Wawan riyanto, Elfrida, Rofik dan Yohan.) Setelah semua dataset terkumpul peneliti melakukan pelabelan dengan label wajah “*without mask*” dan “*with mask*”. Alur proses pengumpulan data set ditunjukkan pada gambar 3.1.



Gambar 3.1 Alur proses pengumpulan data

### 3.3 Methodology

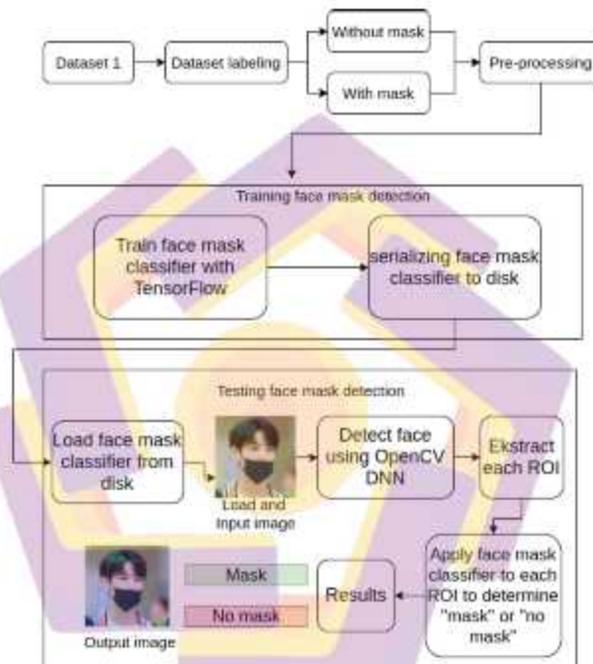
#### 3.3.1 Deteksi wajah menggunakan OpenCV DNN dan TensorFlow

Deteksi wajah adalah masalah dalam visi komputer untuk menemukan dan melokalisasi satu atau lebih wajah dalam sebuah foto. Dimana deteksi wajah merupakan langkah pertama dan terpenting dalam sistem pengenalan wajah apa pun (Dang & Sharma, 2017).

Menemukan wajah dalam foto mengacu pada menemukan koordinat wajah dalam gambar, sedangkan lokalisasi mengacu pada demarkasi tingkat wajah, melalui kotak batas di sekitar wajah. ada dua pendekatan utama untuk pengenalan wajah, metode berbasis fitur yang menggunakan filter buatan tangan untuk mencari dan mendeteksi wajah, dan metode berbasis gambar yang belajar secara holistik cara mengekstrak wajah dari seluruh gambar. Pertama-tama perlu menentukan apakah wajah manusia muncul dalam gambar yang diberikan atau tidak dan di mana wajah yang terdeteksi berada.

Penelitian ini mengimplementasikan OpenCV Deep Neural Network (DNN), TensorFlow dengan model MobileNetV2. Dimana OpenCV untuk membuat pengklasifikasi bertingkat untuk deteksi wajah. Konstruktor yang dapat diambil adalah nama file sebagai argumen yang menentukan file XML untuk model yang sudah dilatih sebelumnya. Manfaat dari implementasi ini adalah menyediakan model deteksi wajah yang telah dilatih sebelumnya, dan menyediakan antarmuka untuk melatih model pada dataset. Model bisa didapatkan pada OpenCV library yang tersedia pada ([opencv.org](http://opencv.org)). Dan TensorFlow sebagai pelatihan dalam kasus ini difokuskan untuk load set data deteksi masker wajah dari disk, melatih model

(menggunakan TensorFlow) di set data ini, lalu membuat serial detektor masker wajah ke disk. Model MobileNetV2 yang digunakan menyediakan prediksi dan akurasi untuk klasifikasi berdasarkan apakah masker dikenakan atau tidak. Adapun tahapan bisa di lihat pada gambar 3.2 berikut.



Gambar 3.2 Tahapan deteksi wajah bermasker menggunakan OpenCV DNN dan TensorFlow

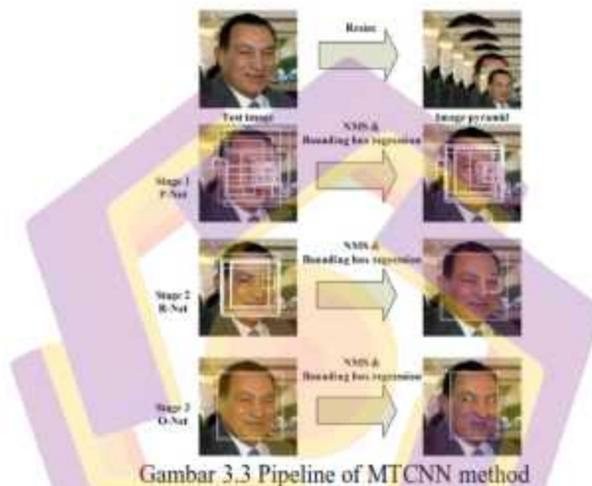
Seperti yang digambarkan dalam gambar 3.2 proses deteksi masker wajah menggunakan dataset 1 yang kemudian dilakukan pelabelan data with mask dan without mask, selanjutnya dilakukan preprocessing (mengubah ukuran menjadi 224×224 piksel, konversi ke format array, dan meningkatkan intensitas piksel dalam gambar input ke rentang [-1, 1]), kemudian dilakukan pelatihan dengan

pengklasifikasian menggunakan Tensorflow yang dimana masalah klasifikasi ditangani oleh MobileNetV2 dengan bobot ImageNet yang sudah dilatih sebelumnya dimuat dari TensorFlow. Kemudian lapisan baru yang dapat dilatih ditambahkan, dan lapisan-lapisan ini dilatih pada set data yang dikumpulkan sehingga dapat menentukan fitur untuk mengklasifikasikan wajah mengenakan masker dari wajah yang tidak mengenakan masker. Kemudian model disetel halus, dan kemudian bobot disimpan. dilanjutkan dengan compile dan latih deteksi masker wajah menggunakan Adam optimizer, serialisasikan dan simpan model klasifikasi masker wajah ke disk untuk digunakan pada langkah berikutnya yaitu testing. Langkah pertama dalam testing adalah memuat masker wajah pengklasifikasi terlatih dari disk. Berikutnya muat dan input image untuk model deteksi wajah dengan OpenCV DNN dilanjutkan dengan ekstrak setiap ROI, langkah selanjutnya adalah menerapkan model pengklasifikasi masker wajah pada ROI yang diekstrak untuk menentukan apakah orang yang ditangkap dalam data pengujian memakai masker atau tidak. Hasil ditunjukkan melalui pemuatan data pengujian asli tetapi, dengan label pada wajah, label termasuk klasifikasi diidentifikasi dengan warna merah untuk "no mask" dan hijau untuk "mask".

### **3.3.2 Identifikasi wajah menggunakan *Multi-task Cascade Convolutional Neural Network***

Multi-task convolutional neural network (Zhang et al., 2016), memiliki Kerangka kerja yang terdiri dari tiga tahap untuk melakukan deteksi wajah dan deteksi tengara secara bersamaan. pertama, pihaknya akan mengusulkan beberapa kandidat jendela dengan cepat melalui CNN dangkal (Liang et al., 2017). Setelah

itu, jaringan kedua akan memperbaiki jendela untuk menolak sejumlah besar jendela non-wajah melalui CNN yang kompleks. Akhirnya, ia menggunakan CNN (Yanagisawa et al., 2018), yang lebih kuat untuk memperbaiki hasilnya dan keluaran lima posisi landmark wajah. Metode ini menggunakan gambar piramida sehingga memiliki gambar dalam skala ganda. Ditunjukkan pada gambar 3.3.



Gambar 3.3 Pipeline of MTCNN method

Pada gambar 3.3 menjelaskan pipeline kerangka kerja bertingkat yang mencakup multitugas tiga tahap jaringan konvolusional yang dalam. Pertama, jendela kandidat diproduksi melalui P-Net cepat. Setelah itu, disempurnakan kandidat ini pada tahap berikutnya melalui R-Net. Pada tahap ketiga, O-Net menghasilkan kotak batas akhir dan wajah posisi landmark.

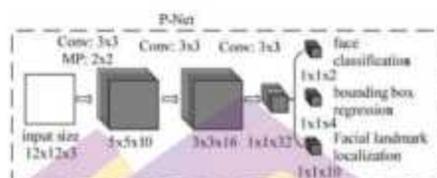
Tahapan pertama mengirimkan gambar sebagai input ke program dalam model ini terlebih dahulu membuat piramida gambar untuk mendeteksi wajah dalam ukuran gambar yang berbeda. Tujuan membuat salinan berbeda dari gambar yang sama adalah untuk mencari yang berbeda ukuran wajah dalam gambar yang

sama. Untuk setiap salinan memiliki  $12 \times 12$  tahap dalam 1 kernel yang akan memeriksa setiap bagian dari seluruh gambar untuk memindai wajah. dimulai dari sudut kiri atas, pada dasarnya adalah bagian gambar dari  $(0,0)$  hingga  $(12,12)$ . Sebagian gambar diumpungkan oleh P-Net lalu mengembalikan koordinat kotak pembatas jika mendeteksi wajah. Maka, itu akan terjadi ulangi proses yang sama dengan bagian  $(0 + 2a, 0 + 2b)$  ke  $(12 + 2a, 12 + 2b)$ , menggeser  $12 \times 12$  kernel, 2 piksel kanan atau bawah sekaligus. Pergeseran 2 piksel dikenal sebagai langkah yang sebenarnya berarti berapa banyak piksel yang dipindahkan kernel setiap saat. 2 langkah membantu mengurangi kompleksitas komputasi tanpa mengorbankan akurasi secara signifikan. Wajah di sebagian besar gambar lebih besar dari 2 piksel. Jadi kemungkinan kernel akan kehilangan wajah karena bergeser 2 piksel sangat rendah.

Manfaat lainnya adalah mesin tempat kode berjalan akan memiliki seperempat jumlah operasi yang harus dihitung, membuat program berjalan lebih cepat dan dengan lebih sedikit memori. Setiap kernel akan lebih kecil dibandingkan dengan gambar yang besar sehingga dapat menemukan wajah yang lebih kecil pada gambar yang lebih besar. Demikian pula dengan kernel akan lebih besar relatif terhadap gambar berukuran lebih kecil sehingga bisa temukan wajah yang lebih besar dalam gambar berukuran lebih kecil.

Setelah mengumpulkan hasil dari P-Net perlu mengurai keluaran P-Net untuk mendapatkan daftar tingkat confidence setiap kotak pembatas dan hapus kotak dengan confidence rendah karena jaringan lebih confidence tentang beberapa kotak dibandingkan dengan yang lain. Setelah itu, pasti ada banyak kotak pembatas

masih tersisa dan banyak yang tumpang tindih. Metode NMS yaitu dikenal sebagai Penekanan Non-Maksimum (Zaytseva et al., 2012), membantu mengurangi jumlah pembatas kotak. Dalam kode khusus ini, sebagian besar jendela disaring. Arsitektur P-Net ditunjukkan pada gambar 3.4.

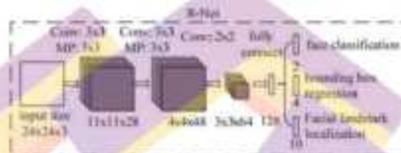


Gambar 3.4 Architectures of P-Net

Pada tahap kedua peneliti mempertimbangkan gambar-gambar yang mungkin hanya berisi sebagian wajah dilihat dari sisi bingkai. Dalam hal ini, jaringan dapat mengembalikan file kotak pembatas yang sebagian keluar dari bingkai. Untuk mengatasi situasi ini setiap kotak pembatas dibuat array dengan ukuran yang sama dan menyalin nilai piksel dari gambar yang ada di kotak pembatas ke array baru. Jika kotak pembatas keluar batas, salin bagian dari gambar di kotak pembatas ke yang baru isi segala sesuatu yang lain dengan nol. Proses mengisi array dengan 0s dikenal sebagai padding. Setelah mengisi kotak pembatas, dilakukan perubahan ukurannya menjadi 24x24 piksel dan menormalkannya menjadi nilai antara -1 hingga 1.

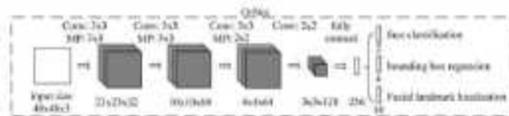
Sekarang dapat kotak pembatas keRNet yang dikenal sebagai Refine Network dan mengumpulkan outputnya. Output R-Net mirip dengan output P-Net. Ini termasuk koordinat baru, kotak pembatas yang lebih akurat dan juga tingkat kepercayaan dari masing-masing batas-kotak ing. Pada tahap ini juga menyinkron

kan kotak dengan konfiden yang lebih rendah, dan melakukan Non-Maximum Suppression pada setiap box untuk menghilangkan kan redundant lebih lanjut kotak. Karena koordinat kotak pembatas baru ini didasarkan pada P-Net kotak pembatas, perlu mengubahnya menjadi koordinat standar dan kemudian membentuk kembali kotak pembatas menjadi persegi untuk diteruskan ke O-Net. Arsitektur R-Net ditunjukkan dalam gambar 3.5.



Gambar 3.5 Architectures of R-Net

Pada tahap terakhir sebelum bisa melewati kotak pembatas dari R-Net, disini memiliki keypad pertama kotak apa pun yang di luar batas. Kemudian harus mengubah ukuran kotak menjadi 48 x 48 piksel sehingga dapat memasukkan kotak pembatas ke dalam O-Net yaitu dikenal sebagai Jaringan Output. Keluaran O-Net berbeda dengan keluaran P-Net dan R-Net. O-Net menyediakan 3 keluaran, koordinat kotak pembatas (out [0]), koordinat dari 5 landmark wajah (out [1]), dan tingkat confidence masing-masing kotak (out [2]). Kemudian sekali lagi harus menyingkirkan kotak dengan confidence level yang lebih rendah, dan standarisasi koordinat kotak pembatas dan koordinat landmark wajah. Terakhir, dengan menerapkan Non-Maximum Suppression lagi untuk mendapatkan wajah fitur orang yang pada dasarnya adalah lima posisi landmark wajah. lima landmark wajah, termasuk kiri mata, mata kanan, hidung, sudut mulut kiri, dan sudut mulut kanan. Arsitektur O-Net ditunjukkan pada gambar 3.6.



Gambar 3.6 Architectures of O-Net

Dalam metode ini disebutkan penggunaan fungsi kehilangan berikut dalam Jaringan:

1. *Cross-Entropy Loss* (Cao et al., 2018): kerugian ini digunakan untuk melakukan klasifikasi wajah untuk daerah yang diajukan. Berikut perhitungannya

$$L_i^{det} = -(y_i^{det} \log(p_i) + (1 - y_i^{det})(1 - \log(p_i))) \dots \dots \dots (3)$$

di mana  $p_i$  adalah probabilitas yang dihasilkan oleh jaringan yang cates sampel  $x_i$  menjadi wajah. Notasi  $y_i \in \{0, 1\}$  menunjukkan *ground-truth label*.

2. *Euclidean Loss* (Malkauthekar, 2013): kerugian ini digunakan untuk melakukan regresi kotak batas dan wajah regresi tengara. Dalam kasus ini digunakan contoh kerugian Euclidean untuk setiap sampel  $x_i$ .

$$L_i^{box} = \|\hat{y}_i^{box} - y_i^{box}\|_2^2 \dots \dots \dots (4)$$

dimana  $\hat{y}_i^{box}$  adalah target regresi yang diperoleh dari jaringan dan  $y_i^{box}$  adalah koordinat kebenaran-dasar. Ada empat koordinat nama, termasuk kiri atas, tinggi dan lebar, dan dengan demikian  $y_i^{box} \in \mathbb{R}^4$ .

3. *Facial Landmark Localization*: mirip dengan tugas bounding box regression deteksi tengara wajah diformulasikan sebagai masalah regresi dan meminimalkan kerugian Euclidean. Berikut formulanya.

$$L_i^{landmark} = \|\hat{y}_i^{landmark} - y_i^{landmark}\|_2^2 \dots \dots \dots (5)$$

dimana  $y_i$  landmark adalah koordinat landmark wajah yang diperoleh dari jaringan dan  $y_i$  adalah koordinat kebenaran dasar untuk sampel ke  $i$ . Ada lima landmark wajah, termasuk kiri mata, mata kanan, hidung, sudut mulut kiri, dan sudut mulut kanan, dan dengan demikian  $y_i \in \mathbb{R}^{10}$ .

4. *Multi Source Training Loss*: Beberapa gambar input tidak hanya berisi gambar wajah tetapi juga beberapa gambar latar belakang seperti objek dll. Dalam hal ini tidak semua kerugian akan digunakan, misalnya ketika gambar pelatihan yang tidak akan berisi wajah, hanya deteksi wajah kerugian akan digunakan dan yang lain akan ditetapkan sebagai 0. Target pembelajaran keseluruhan dirumuskan sebagai berikut.

$$\min \sum_{i=1}^N \sum_{j \in \{\text{det, box, landmark}\}} \alpha_j \beta_j L_j^i \quad (6)$$

di mana  $N$  adalah jumlah sampel pelatihan dan  $\alpha_j$  menunjukkan pentingnya tugas. menggunakan ( $\alpha_{\text{det}} = 1$ ,  $\alpha_{\text{box}} = 0,5$ ,  $\alpha_{\text{landmark}} = 0,5$ ) di P-Net dan R-Net, sedangkan ( $\alpha_{\text{det}} = 1$ ,  $\alpha_{\text{box}} = 0,5$ ,  $\alpha_{\text{landmark}} = 1$ ) dalam jaringan output (O-Net) untuk lokalisasi landmark wajah yang lebih akurat.  $\beta_j \in \{0, 1\}$  adalah indikator tipe the sampel.

### 3.3.1 Penyematan wajah menggunakan FaceNet

Sistem FaceNet dapat digunakan untuk mengekstrak fitur berkualitas tinggi dari wajah, yang disebut penyematan (*embedding*) wajah, yang kemudian dapat digunakan untuk melatih sistem identifikasi wajah (Schroff Florian et al. 2015). Penelitian ini akan menggunakan model Keras FaceNet pre-trained dari (Hiroki Taniai. <https://github.com/nyoki-ntl/keras-facenet>). Dan dilatih pada kumpulan dataset2 (*local face without mask*) dan dataset 3 (*local face with mask*) dengan input

gambar berwarna, untuk memutihkan nilai piksel mereka (distandarisasi di ketiga saluran), dan memiliki bentuk persegi 160x160 piksel. Aspek penting lain dari FaceNet adalah fungsi kerugiannya. Ini menggunakan fungsi kehilangan tiga kali lipat (*Triplet Loss*), ditunjukkan pada gambar 3.7 berikut.



Gambar 3.7 Triplet Loss

Dari gambar 3.7 bahwa gambar anchor (gambar orang tertentu A) lebih dekat ke gambar positif (semua gambar orang A) dibandingkan dengan gambar negatif (semua gambar lainnya). Rumus dari *Triplet Loss* yang digunakan sebagai

berikut (Schroff & Philbin, 2015).

$$\sum_i^N [ \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha ]_+ \tag{7}$$

Keterangan:

$x_i$  : mewakili gambar

$f(x_i)$  : mewakili penyematan gambar

$\alpha$  : margin antara pasangan positif dan negatif

$x_i^a$  : anchor

$x_i^p$  : positive

$x_i^n$  : negative

Kemudian rumus *Triplet Selection* sebagai berikut:

$$\text{Argmax } \|f(x_i^a) - f(x_i^p)\|_2^2 \tag{Eq(1)}$$

$$\text{Argmin } \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{Eq(2)}$$

.....(8)

keterangan:

1. Eq (1): berarti bahwa mengingat gambar jangkar (anchor) orang A, ingin menemukan gambar positif A sedemikian rupa sehingga jarak antara kedua gambar itu terbesar.
2. Eq (2): berarti bahwa mengingat gambar jangkar (anchor) orang A, ingin menemukan gambar negatif sedemikian rupa sehingga jarak antara kedua gambar itu terkecil.

### 3.3.4 Klasifikasi wajah menggunakan Linear Support Vector Machine

Dalam penelitian mengembangkan model pengklasifikasi Linear Support Vector Machine (SVM) untuk memprediksi identitas wajah tertentu. Yang sebelumnya arsitektur *facenet*-nya telah dilatih dengan kumpulan dataset wajah milik empat kelas (Aldi alkulakat, Elfrida, Wawan riyanto dan Yohan). Di sini kelas yang berbeda berarti orang yang berbeda. Terbaik bagian dari *facenet* adalah kita dapat menggunakannya untuk membuat penyematan (*embedding*) untuk set data kustom data sendiri dengan melatih SVM (Lihong et al., 2009).

Di Bagian ini, peneliti mengembangkan model untuk mengklasifikasikan *embeddings* wajah dalam dataset<sup>2</sup>. Pertama *load face embeddings* dataset, normalisasi vektor, ini berarti menskalakan nilai hingga panjang atau besarnya vektor adalah 1 atau panjang satuan, selanjutnya, variabel target string untuk setiap nama dataset perlu dikonversi menjadi bilangan bulat melalui kelas *LabelEncoder* di *scikit-learn*, lalu fit model menggunakan *Linear Support Vector Machine (SVM)* saat bekerja dengan input *embedding* wajah yang dinormalisasi. Ini karena metode ini sangat efektif untuk memisahkan vektor face *embedding*. Dan dapat

memasukkan SVM linear ke data pelatihan menggunakan kelas SVC di scikit-learn dan mengatur atribut 'kernel' ke 'linear'. juga ingin probabilitas nanti ketika membuat prediksi, yang dapat dikonfigurasi dengan mengatur 'probabilitas' ke 'True'. Berikutnya, mengevaluasi model ini dapat dicapai dengan menggunakan model fit untuk membuat prediksi untuk setiap contoh dalam kumpulan data training dan testing, kemudian menghitung akurasi klasifikasi. Berikut rumus dari linear kernel SVM yang digunakan (Jason Brownlee, 2016).

$$K(x, x_i) = \sum (x * x_i) \dots\dots\dots (9)$$

keterangan: input (x) dan setiap vektor dukungan (xi)

Dalam klasifikasi multilabel, fungsi mengembalikan akurasi subset. Jika seluruh set label yang diprediksi untuk sampel sangat cocok dengan kumpulan label yang sebenarnya, maka akurasi subset adalah 1.0; jika tidak, itu adalah 0.0.

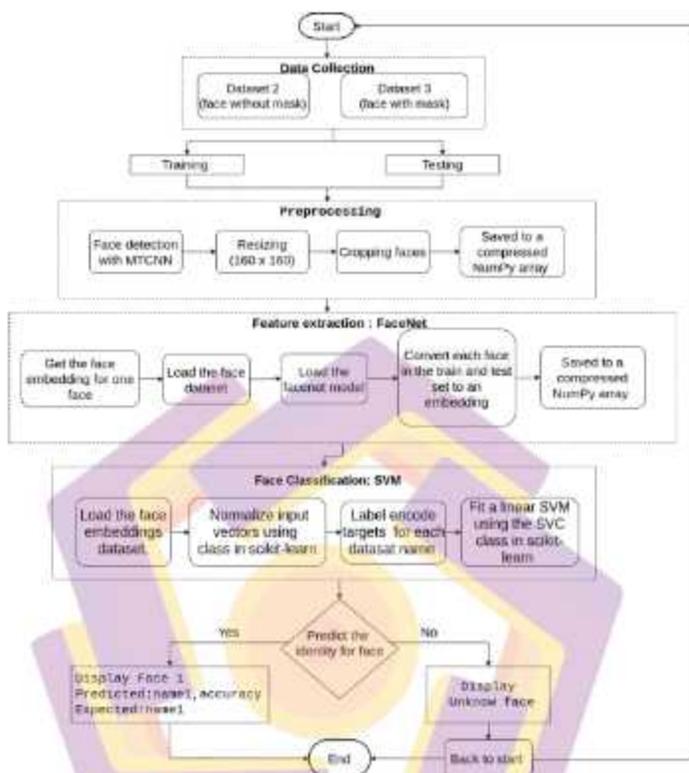
Kinerja model yang diusulkan dievaluasi oleh rumus berikut. Rumus ditentukan oleh *Accuracy score*.

$$accuracy(y, \hat{y}) = \frac{1}{n_{\text{sample}}} \sum_{i=0}^{n_{\text{sample}} - 1} 1(y_i = \hat{y}_i) \dots\dots\dots (10)$$

Keterangan:  $\hat{y}$  mewakili nilai prediksi dari i dan  $y_i$  represents nilai aktual.

### 3.3. Alur Penelitian

Adapun untuk alur penelitiannya, ditunjukkan dalam gambar 3.8 berikut.



Gambar 3.8 Alur Penelitian

Pada gambar 3.8 pertama – tama dataset yang digunakan yaitu dataset 2 (without mask) dan dataset 3 (with mask), yang kemudian data dibagi menjadi dua yaitu data train dan data test, selanjutnya dilakukan preprocessing, dalam tahap preprocessing ada empat proses dilakukan, yaitu deteksi, mengubah ukuran, memotong wajah, dan menyimpan dalam bentuk kompres. Tahapan ini menggunakan library *MultiTask Cascaded Convolutional Neural (MTCNN)* dimana didalam proses diberi kotak batas kemudian dipotong berdasarkan kotak batas, setelah mendapatkan wajah, gambar akan diskalakan (mengubah ukuran)

dengan ukuran sesuai dengan input model yang digunakan yaitu dengan ukuran 160 x 160, lalu menyimpan hasilnya dalam satu file array NumPy terkompresi melalui fungsi *savez\_compressed()*. Setelah tahapan preprocessing selesai maka dilakukan *Feature extraction* dengan FaceNet model dimana tahapan pertama adalah dapatkan penyematan wajah untuk satu wajah terlebih dahulu, kemudian muat dataset wajah yang telah didapatkan dari model MTCNN yang telah dilakukan sebelumnya, berikutnya muat model FaceNet, kemudian konversikan setiap wajah dalam pelatihan dan testing yang diatur menjadi penyematan (*embedding*), simpan array ke satu file dalam format terkompresi, data siap untuk dilakukan proses klasifikasi. Proses untuk pengenalan wajah dilakukan dengan klasifikasi SVM yang digunakan dengan kernel linear dan pengaturan probabilitas true, dimana tahapannya meliputi memuat dataset penyematan wajah yang di peroleh dari proses ekstrasi fitur, kemudian dilakukan normalisasi vektor menggunakan *Normalizer class in scikit-learn*, lalu dilakukan label encode target untuk setiap nama pada dataset menggunakan kelas *LabelEncoder* di *scikit-learn* dilanjutkan fit model menggunakan kelas *SVC* di *scikit-learn*, dengan menggunakan model fit untuk membuat prediksi di setiap kumpulan data train dan test, kemudian menghitung akurasi klasifikasi. Hasil menampilkan prediksi, ekspektasi dan akurasi dari wajah yang di tampilkan dan apabila tidak maka proses akan kembali ke start untuk dilakukan pengecekan.

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Dataset and Experimental Setup

##### 4.1.1 Dataset

###### 1. Dataset1 (*Face dataset*)

Pada dataset ke 1 menggunakan data yang dikumpulkan oleh Prajna Bhandary berupa kumpulan data wajah. Dimana datanya terdiri dari berbagai wajah dengan masker dan tanpa masker berjumlah 1,376 gambar, dimana gambar dengan masker 690 gambar dan tanpa masker 686 gambar. dataset tersebut sebelumnya digunakan untuk detektor masker COVID-19.

Dataset ini akan dibagi menjadi data latih dan uji, yang akan peneliti gunakan sebagai learning model untuk kasus wajah dengan masker maupun tanpa masker sebelum melanjutkan ke penelitian berikutnya. Dataset1 ditunjukkan dalam gambar 4.1.



Gambar 4.1 Sampel dataset wajah yang di develop oleh (Prajna Bhandary, 2020)

## 2. Dataset2 (*Local face without mask*)

Dataset ini merupakan data wajah tanpa masker dengan 62 koleksi gambar dengan empat identitas yang berbeda, yaitu, Aldi Alkulakat, Wawan Riyanto, Elfrida dan Yohan. Masing-masing orang memiliki gambar yang berbeda di mana ukuran gambar juga bervariasi. Data kemudian dibagi untuk data latih 42 gambar dan data uji 20. Ditunjukkan dalam gambar 4.2.



Gambar 4.3 Deteksi wajah tanpa masker dengan MTCNN

## 3. Dataset3 (*Local face with mask*)

Dataset ketiga ini peneliti kumpulkan dengan cara dibuat dari dataset sebelumnya yaitu dataset2 dengan cara memakaikan masker pada wajah dalam

gambar. Dataset ini jumlahnya sama dengan dataset sebelumnya berjumlah 62, yang kemudian dibagi menjadi data latih dan data uji. Yang membedakan hanyalah penggunaan masker pada wajah. Proses pemberian masker dibantu dengan aplikasi berbasis *opensource GNU Image Manipulation Program*. Masker yang digunakan ditunjukkan pada gambar 4.4.

Gambar 4.4 Masker COVID-19 developed by Pyimagesearch

Berikut adalah kumpulan gambar salah satu dari dataset ketiga yaitu wajah dengan masker.



Gambar 4.5 Salah satu dataset3 yang dikumpulkan

Dari yang sudah dikumpulkan dilakukan proses deteksi dengan ukuran 160x160. Ditampilkan dalam gambar 4.6.



Gambar 4.6 deteksi wajah dengan masker dengan MTCNN

Selain dataset masker virtual peneliti juga menguji metode dengan dataset real dari beragam masker wajah. Berikut adalah sampel gambar dari dataset real dengan berbagai macam masker.



Gambar 4.7 Sampel gambar dataset real dengan berbagai macam masker



Gambar 4.8 Deteksi wajah bermasker pada dataset real

#### 4.1.2 Experimental Setup

Penelitian ini menggunakan komputer dengan spesifikasi Intel(R) Core (TM) i3-6100 CPU @ 3.70GHz 3, 70 GHz, RAM 8GB, System type 64-bit, Windows 10, Python adalah bahasa pemrograman yang digunakan dengan python versi (python 3.7) dan Jupyter notebook untuk menjalankan program. Program ini sepenuhnya berbasis pada (*Deep Learning Neural Network*) dan diimplementasikan

menggunakan kerangka kerja Tensorflow. Pustaka python yang digunakan dalam penelitian ini sebagai berikut:

1. Tensorflow (2.3.1)
2. Scipy (1.5.3)
3. Scikit-learn (0.23.2)
4. Opencv (4.4.0)
5. MTCNN (0.1.0)

#### **4.2 Deteksi wajah bermasker menggunakan OpenCV DNN dan TensorFlow**

Deteksi wajah bermasker dilakukan pada dataset1 yang terdiri dari 1736 gambar dengan data training 1100 dan data testing 276. Yang kemudian dilakukan pelabelan sebelum proses deteksi wajah di mulai. Pelabelan ditandai dengan data gambar "*with mask*" dan "*without mask*".

##### **4.2.1. Training data**

Setelah proses pelabelan selesai maka dilakukan training dimana akan difokuskan pada memuat set data deteksi masker wajah dengan melatih model menggunakan TensorFlow dengan Arsitektur *MobileNetV2*. TensorFlow untuk melatih pengklasifikasi untuk secara otomatis mendeteksi apakah gambar terdeteksi sebagai wajah dengan masker wajah atau tidak. Berikut langkah - langkahnya.

##### **1. Pre-process**

Pertama mengambil daftar gambar di direktori dataset, lalu menginisialisasi data dan label. Serta dilakukanya Langkah-langkah pre-pemrosesan termasuk mengubah ukuran menjadi 224×224 piksel, lalu konversi ke format array, dan meningkatkan intensitas piksel dalam gambar input ke rentang [-1, 1] (melalui

preprocess input *convenience function* (). Menambahkan gambar yang telah diproses dan label terkait dengan data dan label yang ada dalam daftar masing-masing, serta memastikan data pelatihan dalam format array NumPy. Hasil dari konversi gambar ke format array ditunjukkan pada gambar 4.9.



Gambar 4.9 Pre-process konversi gambar ke format array

Gambar 4.9 menjelaskan dimana sebelah kanan adalah kumpulan dari data gambar wajah yang kita lihat sebagai manusia kemudian dikonversi ke format array, agar bisa dibaca oleh komputer.

Setelah data pelatihan dalam format array, kemudian dilakukan pengkodean one-hot encoding pada label array ('with mask', 'with mask', 'with mask', ..., 'without mask', 'without mask', 'without mask'). Berikut one-hot encode ([1., 0.], [1., 0.], [1., 0.], ..., [0., 1.], [0., 1.], [0., 1.]), dimana setiap elemen label array terdiri dari array di mana hanya satu indeks yang "hot" (yaitu,1).

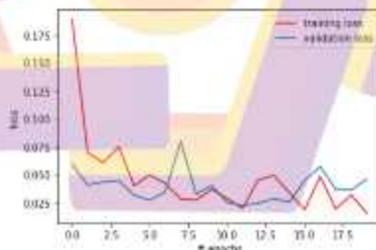
## 2. Training accuracy/loss

Training face mask detector dilatih selama 91 langkah, validasi pada 276 sampel. Hasil ditunjukkan pada tabel 4.1.

Tabel 4.1 Training accuracy/loss

| epoch | waktu          | loss   | accuracy | val_loss | val_accuracy |
|-------|----------------|--------|----------|----------|--------------|
| 1/20  | 39s 430ms/step | 0.1888 | 0.9274   | 0.0593   | 0.9746       |
| 2/20  | 37s 404ms/step | 0.0696 | 0.9733   | 0.0408   | 0.9891       |
| 3/20  | 37s 402ms/step | 0.0606 | 0.9816   | 0.0434   | 0.9783       |
| 4/20  | 37s 406ms/step | 0.0749 | 0.9688   | 0.0447   | 0.9783       |
| 5/20  | 37s 403ms/step | 0.0400 | 0.9871   | 0.0316   | 0.9855       |
| ....  |                |        |          |          |              |
| 20/20 | 38s 412ms/step | 0.0154 | 0.9963   | 0.0459   | 0.9783       |

Tabel 4.1 menjelaskan training accuracy atau loss pada *face mask detection*. Pengujian accuracy atau loss lebih lanjut ditunjukkan dalam bentuk grafik seperti pada (Gambar 4.10) untuk training loss dan validation loss, sedangkan (Gambar 4.11) untuk grafik training accuracy dan validation accuracy.



Gambar 4.10 Grafik Training Loss Dan Validation Loss

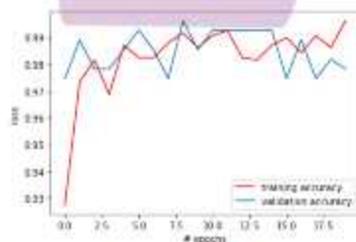
Grafik menjelaskan bahwa garis merah menggambarkan training loss dan garis biru menggambarkan validation loss, dimana dimulai dari epochs 0.0 training lossnya tinggi mencapai 0.180, kemudian mengalami penurunan pada epochs 2.5 dengan loss 0.075, menurun lagi pada epochs ke 5.0 dengan loss di angka 0.50, epochs ke 7.5 training los di angka 0.026, dan naik di epochs 9.5 dengan loss 0.045,

turun pada epochs 10.5 di angka 0.025, naik pada epochs 12.5 dengan loss di angka 0.050, turun pada epochs 15.0 di angka 0.024, naik pada epochs 16.0 dengan loss 0.050 kemudian turun pada epochs 17.5 di angka loss 0.26, turun pada epochs 19.0 di angka 0.010.

Kemudian untuk validation loss dimulai dari epochs 0.0 validation loss berada di angka 0.055, menurun pada epochs 1.5 di angka loss 0.035, naik di epochs 2.5 dengan validation loss 0.048, menurun di epochs 5.0 di angka loss 0.030 dan naik pada epochs 7.5 di angka loss 0.075, turun pada epochs 8.5 di angka loss 0.040, turun lagi di epochs 10.0 di angka 0.030, naik pada epochs 5.0 di angka loss 0.060, kemudian turun pada epochs 17.5 di angka 0.040 dan naik pada epochs 19.0 di angka loss 0.045.

Dari analisis grafik pada gambar 4.19 training loss tertinggi yaitu 0.0180 pada epochs ke 0.0 dan terendah 0.010 pada epochs 19.0, kemudian untuk grafik validation loss tertinggi 0.075 pada epochs 7.5, terendah 0.030 pada epochs 10.0.

Berdasarkan analisis data pada gambar 4.10 terlihat grafik sedikit tanda overfitting, dengan validasi loss lebih rendah dari training loss.



Gambar 4.11 Training Accuracy dan Validation Accuracy

Pada gambar 4.11 garis merah menggambarkan training accuracy dan garis biru menggambarkan validation accuracy, dimulai dari epochs ke 0.0 training loss

berapa pada angka 97, kemudian naik di angka 98, pada epochs ke 2.5, turun pada epochs 0.3 di angka 96.8, naik pada epochs 4.0 di angka 98.5, turun di epochs 5.0 di angka 98, naik pada epochs 7.9 di angka 99, turun pada epochs 10.0 di angka 98.7, kemudian naik pada epochs 11.0 di angka 99, turun di epochs 12.5 di angka 97.9, naik di epochs 15.0 pada angka 98.5, turun di epochs 16.0 di angka 98, naik di epochs 17.5 di angka 99, turun di epochs 18.0 pada angka 98 kemudian naik di angka 100 pada epochs 20.0.

Untuk validation accuracy di mulai dari epochs ke 0.0 berada di angka 97.5 kemudian naik ke angka 99 pada epochs 1.0, turun di epochs 2.5 di angka 97.8, turun pada epochs 3.0 di angka 97, naik ke angka 98.8 pada epochs 5.0, turun di epochs 7.5 di angka 97.5, naik di epochs ke 8.0 di angka 100, turun di angka 98.7 pada epochs 9.0, naik pada epochs ke 10.0 di angka 98.8, turun pada epochs 15.0 di angka 97, naik menjadi 98.7, turun di epochs 16 di angka 97.3, naik pada epochs ke 17.5 di angka 97.9.

Dari analisis grafik pada gambar 4.11 training accuracy tertinggi di angka 100 dengan epochs 20.0, terendah pada epochs 3.0 di angka 96.8 dan untuk validation accuracy tertinggi 100 pada epochs 8.0, terendah pada epochs 16.0 di angka 97.3. Info untuk evaluasi network ditunjukkan pada Tabel 4.2.

### 3. Evaluasi

Untuk mengevaluasi digunakan Confusion matrix, dengan rumus sebagai berikut (Davis & Goadrich, 2006).

|                  |              | Actual Values |              |
|------------------|--------------|---------------|--------------|
|                  |              | Positive (1)  | Negative (0) |
| Predicted Values | Positive (1) | TP            | FP           |
|                  | Negative (0) | FN            | TN           |

Gambar 4.12 Confusion matrix

True Positive (TP)

True Negative (TN)

False Positive (FP)

False Negatif (FN)

$$\text{Akurasi} = (TP + TN) / (TP + FP + FN + TN) \dots\dots\dots(11)$$

$$\text{Precision} = (TP) / (TP + FP) \dots\dots\dots(12)$$

$$\text{Recall} = (TP) / (TP + FN) \dots\dots\dots(13)$$

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \dots\dots\dots(14)$$

Tabel 4.2 Hasil Akurasi, Recall, Precision, F1-Score, Support Dari Training Face Mask Detection

| Nama         | Precision | Recall | f1-score | Support |
|--------------|-----------|--------|----------|---------|
| With mask    | 0.97      | 0.99   | 0.98     | 138     |
| Without mask | 0.99      | 0.97   | 0.98     | 138     |
| Accuracy     |           |        | 0.98     | 276     |
| Macro avg    | 0.98      | 0.98   | 0.98     | 276     |
| Weighted avg | 0.98      | 0.98   | 0.98     | 276     |

Seperti yang terlihat pada Tabel 4.2, peneliti memperoleh 98% akurasi pada set pengujian.

#### 4.2.2. Testing mask

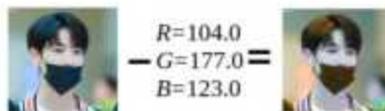
Menerapkan deteksi masker wajah COVID-19 dengan opencv, Setelah deteksi masker wajah dilatih, kemudian beralih ke memuat data dan melakukan deteksi wajah, selanjutnya mengklasifikasikan setiap wajah sebagai *with mask* atau *without mask*. Pada tahapan ini memerlukan tiga import pertama *TensorFlow*, kedua memuat model *MaskNet*, ketiga pre-process gambar input. OpenCV diperlukan untuk tampilan dan manipulasi gambar.

Selanjutnya, memuat model pengklasifikasi wajah dan masker wajah. model detektor wajah yang digunakan yaitu (*deploy.prototxt*) dan (*res10\_300x300\_ssd\_iter\_140000.caffemodel*). langkah berikutnya memuat dan memproses gambar input dengan data image dari foto Renjun NCT Dream dengan masker yang diambil dari google (diakses pada November 29, 2020). OpenCV menyediakan dua fungsi untuk memfasilitasi preprocessing pada gambar yaitu *cv2.dnn.blobFromImage* dan *cv2.dnn.blobFromImages*, dimana Kedua fungsi ini melakukan pengurangan rata-rata, scaling, dan secara opsional bertukar saluran. Berikut langkah dalam pre-process yang ditangani oleh fungsi blob From Image OpenCV.

##### a. Mean subtraction

Yaitu merubah ukuran gambar menjadi 300×300 piksel dengan pengurangan rata - rata ( $\mu$ ),  $R=104.0$ ,  $G=177.0$ , and  $B=123.0$ . Dimana R(red), G(green) dan B(blue), Dengan rumus sebagai berikut (Rosebrock Adrian, 2017).

$$\begin{aligned} R &= R - \mu R \\ G &= G - \mu G \\ B &= B - \mu B \dots\dots\dots (15) \end{aligned}$$



Gambar 4.13 Pengurangan rata-rata

Gambar 4.13 Representasi visual pengurangan rata-rata di mana rata-rata RGB (tengah) telah dihitung dari kumpulan data gambar dan dikurangi dari gambar asli (kiri) yang menghasilkan gambar output (kanan).

- b. Scaling factor,  $\sigma$ , yang menambahkan dalam normalisasi dengan rumus sebagai berikut:

$$R = (R - \mu_R) / \sigma$$

$$G = (G - \mu_G) / \sigma$$

$$B = (B - \mu_B) / \sigma$$

$$\sigma = \text{Standard deviation} \dots \dots \dots (16)$$

Berikut fungsi yang digunakan `blob = cv2.dnn.blobFromImage(image, scalefactor=1.0, size, mean, swapRB=True)`.

Diskusikan tentang setiap parameter sebagai berikut:

1. Image: gambar input yang ingin di preprocess,
2. Scalefactor: melakukan pengurangan berarti, dapat secara opsional menskalakan gambar dengan beberapa faktor. Nilai ini secara default ke '1.0' (yaitu, tanpa penskalaan) Penting juga untuk dicatat bahwa skala faktor harus  $\frac{1}{\sigma}$ .
3. Size.
4. Mean: Ini adalah nilai pengurangan rata-rata. Mereka bisa menjadi 3-tuple dari RGB.

5. `swapRB`: OpenCV mengasumsikan gambar berada dalam urutan saluran BGR; namun, nilai 'rata-rata' mengasumsikan menggunakan pesan RGB. Untuk mengatasi perbedaan ini dapat menukar saluran R dan B dalam gambar dengan menetapkan nilai ini ke 'True'. Secara default OpenCV melakukan pertukaran saluran ini.
- c. OpenCV's `blobFromImage` function

`BlobFromImage` menciptakan 4-dimensional blob dari gambar. Secara opsional mengubah ukuran (*resizes*) and *crops image from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels*. Berikut hasil penggunaan fungsi blob yang ditunjukkan pada gambar 4.14 berikut.



Gambar 4.14 Dimensi blob yang dihasilkan dari blob From Image

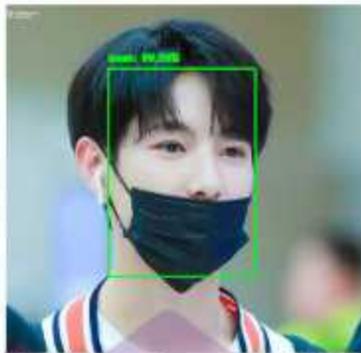
Dari proses tersebut tuple yang dihasilkan memiliki format (`num_images=1, num_channels=3, width=300, height=300`). Setelah melewati blob melalui network kemudian didapatkan area deteksi wajah Dengan *Deep Neural Network*. Hasil ditunjukkan pada gambar 4.15 berikut.

|   |   |            |      |            |            |            |
|---|---|------------|------|------------|------------|------------|
| 0 | 1 | 0.98419729 | .... | 0.27722836 | 0.64010775 | 0.64307065 |
| 0 | 1 | 0.1225033  | .... | 4.2074463  | 4.8407373  | 4.9870415  |
| 0 | 1 | 0.11752322 | .... | 4.00965    | 0.8207079  | 4.9889608  |
| 0 | 0 | 0          | .... | 0          | 0          | 0          |
| 0 | 0 | 0          | .... | 0          | 0          | 0          |
| 0 | 0 | 0          | .... | 0          | 0          | 0          |

Gambar 4.15 Area deteksi wajah

Pada gambar 4.15 area deteksi wajah ditandai dengan warna hitam dan yang berwarna putih artinya bukan merupakan area wajah.

Setelah didapatkan hasil area wajah maka proses selanjutnya menjalankan *Face Region Of Interest(ROI)* melalui model *MaskNet* dengan cara ekstrak *face ROI* via NumPy, kemudian Pre-proses ROI dengan cara yang sama seperti yang dilakukan selama pelatihan, lakukan deteksi masker untuk memprediksi *with mask* atau *without mask*. Dari sini, dibuat anotasi dan untuk menampilkan hasil. Dengan menentukan label kelas berdasarkan probabilitas yang dikembalikan oleh model detektor masker dan menetapkan warna terkait untuk anotasi. Warna akan menjadi "hijau" untuk *with mask* dan "merah" untuk *without mask*. kemudian menggambar label teks (termasuk kelas dan probabilitas), serta persegi panjang kotak batas untuk wajah, menggunakan fungsi gambar OpenCV. Hasil dari deteksi masker wajah COVID-19 ditunjukkan pada gambar 4.15.



Gambar 4.15 Output gambar deteksi masker wajah COVID-19 dengan OpenCV DNN dan Tensorflow

Gambar 4.15 menunjukkan sebuah foto dengan mengenakan masker wajah dan sistem telah berhasil melabeli gambar sebagai "Mask" dengan benar untuk wajah tersebut dan menghasilkan akurasi 99,99%. Dan juga dilakukan pengujian terhadap foto (foto yang dijadikan testing yaitu foto yang di develop by google) dengan wajah tanpa masker yang ditunjukkan pada gambar 4.16 berikut.



Gambar 4.16 Output gambar deteksi No mask dengan OpenCV DNN dan Tensorflow

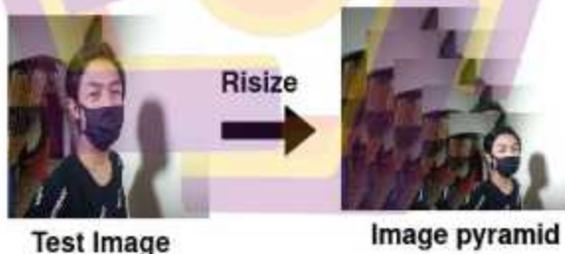
Pada gambar 4.16 memperlihatkan deteksi wajah tanpa masker dengan benar melabeli gambar tersebut sebagai "No mask" dengan akurasi 100,00%.

### 4.3 Proses deteksi wajah menggunakan *Multi Task Cascade Convolutional Neural Network*

Deteksi wajah adalah proses menemukan wajah secara otomatis dalam foto dan melokalkannya dengan menggambar kotak pembatas di sekelilingnya. Dalam penelitian menggunakan *Multi-Task Cascaded Convolutional Neural Network* (MTCNN) untuk deteksi wajah, ini seperti menemukan dan mengekstrak wajah dari foto, tahapan deteksi wajah ini merupakan tahapan dari preprocessing. Berikut tahapan dalam melakukan proses deteksi wajah.

#### 4.3.1 Tahap 1: Image Pyramid

Dalam model ini, membuat piramida gambar, untuk mendeteksi wajah dari semua ukuran yang berbeda, dengan kata lain, membuat salinan yang berbeda dari gambar yang sama dalam berbagai ukuran untuk mencari wajah berukuran berbeda dalam gambar. Image pyramid ditunjukkan dalam gambar 4.17 berikut.



Gambar 4. 17 Image pyramid

Ada dua jenis Piramida Gambar. 1) Piramida Gaussian dan 2) Piramida Laplacian. Tepatnya, tingkat piramida diperoleh secara berurutan sebagai berikut For  $0 < l < N$  (Bergen & Ogden, 1984).

$$G_l(i, j) = \sum_m \sum_n w(m, n) G_{l-1}(2i+m, 2j+n) \dots\dots\dots (17)$$

$N$  = mengacu pada jumlah tingkat dalam piramida

$$G_i = \text{REDUCE} [G_{i-1}]$$

$w(m, n)$  = fungsi pembobotan

Tingkat yang lebih tinggi (Resolusi rendah) dalam Piramida Gaussian dibentuk dengan menghapus baris dan kolom berturut-turut dalam gambar Tingkat bawah (resolusi lebih tinggi). Kemudian setiap piksel di tingkat yang lebih tinggi dibentuk oleh kontribusi dari 5 piksel dalam tingkat yang mendasarinya dengan bobot gaussian. Karena ini tingkat berbeda dalam kepadatan sampelnya diperlukan untuk menginterpolasi nilai sampel baru antara mereka yang berada dalam tingkat tertentu sebelum itu tingkat dikurangi dari berikutnya-bawah Tingkat. Interpolasi dapat dicapai dengan membalikkan proses REDUCE. Berikut rumus yang digunakan untuk Gaussian Pyramid Interpolation.

$$G_{L,k}(i, j) = 4 \sum_m \sum_n G_{L,k-1}\left(\frac{2i+m}{2}, \frac{2j+n}{2}\right) \dots \dots \dots (18)$$

Di sini hanya ketentuan yang  $(2i+m)/2$  dan  $(2j+n)/2$  adalah bilangan bulat yang berkontribusi pada Jumlah. Operasi diperluas menggandakan ukuran gambar dengan setiap iterasi, jadi  $G_{i,1}$ , adalah ukuran  $G_{i,1}$ , dan  $G_{i,1}$  adalah ukuran yang sama dengan gambar asli. Piramida Gaussian level ditampilkan dalam gambar 4.18 berikut.



Gambar 4.18 Piramida Gaussian level

Tingkat piramida bandpass,  $L_0, L_1, \dots, L_N$ , sekarang dapat ditentukan dalam hal tingkat piramida lowpass sebagai berikut:

$$L_1 = G_1 \text{---EXPAND} [G_1+1] \\ = G_1 \text{---}G_1+1,1 \dots \dots \dots (19)$$

Tingkat piramida teratas,  $L_N$ , pertama kali diperluas dan ditambahkan  $L_{N-1}$  untuk membentuk  $G_{N-1}$  maka array ini diperluas dan ditambahkan ke  $L_{N-2}$  untuk memulihkan  $G_{N-2}$ , dan sebagainya. Atau bisa ditulis sebagai berikut.

$$G_0 = \sum L_{l,1} \dots \dots \dots (20)$$

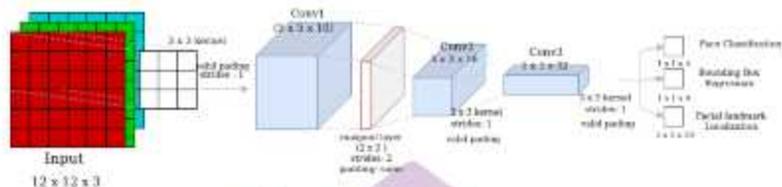
Empat level pertama ditampilkan di gambar 4.19 berikut.



Gambar 4.19 Level Laplacian pyramid

### 4.3.2 Tahap 2: P-Net

Di P-Net, untuk setiap gambar berskala, kernel berjalan melalui gambar dan mencari wajah. Gambar 4.20 menunjukkan struktur dari P-Net.



Gambar 4.20 P-Net (Proposal network)

Pada gambar 4.20 menjelaskan bahwa dalam masing-masing input size  $12 \times 12$  ini, 3 konvolusi dijalankan dengan kernel  $3 \times 3$ . Dengan ketentuan valid padding (tidak ada input padding), dan strides 1 yang artinya filter akan bergeser sebanyak 1 pixels secara horizontal lalu vertical, sehingga menghasilkan output ukuran  $5 \times 5$  pada kasus disini telah mengambil 3 filter tersebut dan oleh karena itu kedalaman conv 1 adalah 10, maka dimensinya adalah,  $5 \times 5 \times 10$ .

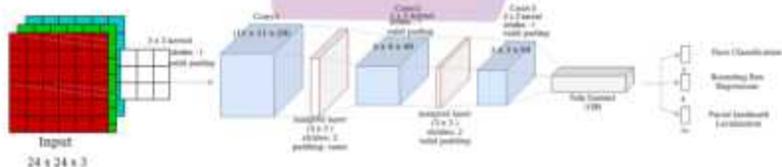
Max Pooling  $2 \times 2$  dengan stride 2, maka pada setiap pergeseran filter, nilai maximum pada area  $2 \times 2$  pixel tersebut yang akan dipilih, sedangkan Average Pooling akan memilih nilai rata-ratanya dan penggunaan same padding yang artinya menambahkan piksel pengisi ekstra di sekitar batas gambar input dengan menetapkan nilai piksel tambahan menjadi nol (0). Lapisan pooling di sini mengambil  $5 \times 5 \times 10$  sebagai input dengan penambahan padding nol dan menerapkan rata-rata kumpulan matriks  $2 \times 2$  yaitu melayangkan matriks  $2 \times 2$  pada input dan mengambil rata-rata dari keempat piksel dan melompat dengan melewati 2 kolom setiap kali sehingga memberikan  $3 \times 3 \times 16$  sebagai output dan menghitung pooling untuk setiap lapisan jadi di sini kedalaman output adalah 16.

Mengambil  $3 \times 3 \times 16$  sebelumnya dengan filter ukuran  $3 \times 3$ , dengan strides 1 yaitu filter akan bergeser sebanyak 1 pixels dan valid padding, sehingga mendapatkan output  $1 \times 1$ , di sini mengambil 32 filter kedalam 16 dan berbelit-belit sehingga mendapatkan output  $1 \times 1 \times 32$ .

Setelah lapisan konvolusi ketiga, jaringan dibagi menjadi dua lapisan. Aktivasi dari lapisan ketiga diteruskan ke dua lapisan konvolusi terpisah, dan lapisan softmax setelah salah satu lapisan konvolusi (softmax menetapkan probabilitas desimal untuk setiap hasil, dan probabilitas menambahkan hingga 1. Dalam hal ini, ia menghasilkan 2 probabilitas: probabilitas bahwa ada wajah di daerah tersebut dan kemungkinan bahwa tidak ada wajah). Konvolusi layer empat (1) menghasilkan probabilitas wajah berada di setiap kotak batas, dan konvolusi layer empat (2) menghasilkan koordinat kotak batas.

### 4.3.3 Tahap 3: R-Net

R-Net memiliki struktur yang mirip, tetapi dengan lebih banyak lapisan. Dibutuhkan kotak batas P-Net sebagai inputnya, dan menyempurnakan koordinatnya. Struktur R-Net sebagai berikut.



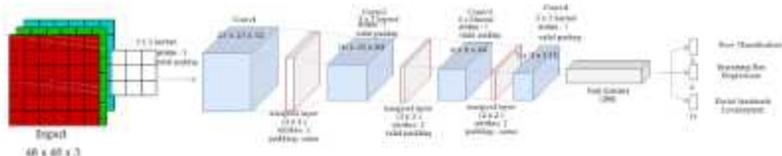
Gambar 4.21 R-Net

Penjelasan dari gambar 4.21 yaitu mengambil  $24 \times 24 \times 3$  sebagai input dengan kernel  $3 \times 3$ . Dan ketentuan valid padding (tidak ada input padding), serta strides 1 yang artinya filter akan bergeser sebanyak 1 pixels secara horizontal lalu

vertical, sehingga menghasilkan output ukuran  $11 \times 11$  pada kasus disini telah mengambil 3 filter tersebut dan oleh karena itu kedalaman conv 1 adalah 28, maka dimensinya adalah,  $11 \times 11 \times 28$ . Dilanjutkan Max Pooling  $3 \times 3$  dengan stride 2 dan penggunaan same padding. Lapisan pooling di sini mengambil  $11 \times 11 \times 28$  sebagai input dengan penambahan padding nol dan menerapkan rata-rata kumpulan matriks  $3 \times 3$  dan melompat dengan melewati 2 kolom setiap kali sehingga memberikan  $4 \times 4 \times 48$  sebagai output dan menghitung pooling untuk setiap lapisan jadi di sini kedalaman output adalah 48. Lalu mengambil output dari lapisan sebelumnya dan melakukan rata-rata pooling dengan langkah 2 yaitu (melewati dua kolom) dengan filter ukuran  $3 \times 3$ , di sini padding adalah valid oleh karena itu mendapatkan dimensi  $3 \times 3 \times 64$ . Selanjutnya mengambil  $3 \times 3 \times 64$  sebagai input dengan kernel  $2 \times 2$ . Dan ketentuan valid padding, serta strides 1 yang artinya filter akan bergeser sebanyak 1 pixels secara horizontal lalu vertical. Kemudian meratakan semua nilai  $3 \times 3 \times 64$  hingga satu lapisan ukuran 128. Demikian pula, R-Net dibagi menjadi dua lapisan pada akhirnya, memberikan dua output: koordinat kotak batas baru dan kepercayaan diri mesin di setiap kotak batas.

#### 4.3.4 Tahap 4: O-Net

O-Net mengambil kotak batas R-Net sebagai input dan menandai koordinat landmark wajah. Struktur O-Net sebagai berikut.



Gambar 4.22 O-Net

Pada Struktur O-Net yang ditunjukkan pada gambar 4.22 mengambil input size  $48 \times 48 \times 3$  dengan ukuran kernel  $3 \times 3$ , strides 1 dan valid padding, dihasilkan kedalaman conv 1 yaitu 32, output  $23 \times 23$ , sehingga dimensinya adalah  $23 \times 23 \times 32$ . Pooling layer dengan ukuran  $3 \times 3$ , strides 2 dan penggunaan padding (same) menghasilkan output conv 2 adalah  $10 \times 10 \times 64$ , mengambil input  $10 \times 10 \times 64$  dengan kernel  $3 \times 3$ , strides 1 dan valid padding dan dilakukan pooling layer dengan ukuran  $3 \times 3$ , strides 1 dengan valid padding menghasilkan output  $4 \times 4$  dengan kedalaman conv 3 yaitu 64 dengan demikian dimensinya adalah  $4 \times 4 \times 64$ , dilanjutkan menggunakan dimensi sebelumnya sebagai input dengan ukuran kernel  $3 \times 3$ , strides 1 dan valid padding dilanjutkan dengan melakukan pooling layer ke tiga dengan ukuran  $2 \times 2$ , strider 2 dan padding (same) sehingga menghasilkan output  $3 \times 3 \times 128$ . Yang dilanjutkan dengan meratakan semua nilai  $3 \times 3 \times 128$  hingga satu lapisan ukuran 256. O-Net pada akhirnya, memberikan 3 output yang berbeda: probabilitas wajah berada di dalam kotak, koordinat kotak batas, dan koordinat landmark wajah (lokasi mata, hidung, dan mulut).

#### **4.3.5 Tiga tugas pelatihan CNN detectors**

Tiga tugas ini meliputi klasifikasi wajah atau non wajah, regresi kotak batas, dan wajah lokalisasi tengara. MTCNN. Menggunakan kehilangan cross-entropi untuk memecahkan wajah klasifikasi, dan kerugian Euclidean untuk setiap sampel (Xiang et al., 2017). Deskripsi tentang fitur wajah dibagi menjadi:

- 1). Klasifikasi wajah manusia dan wajah non-manusia.
- 2). Masalah regresi kotak batas wajah.
- 3). Masalah regresi titik fitur wajah.

Fungsi objektif untuk masalah klasifikasi wajah sebagai berikut.

$$L_i^{cls} = -(y_i^{cls} \log(p_i) + (1 - y_i^{cls})(1 - \log(p_i))) \dots\dots\dots(21)$$

Fungsi tujuan untuk regresi kotak batas (*bounding box regression*) adalah.

$$L_i^{box} = \|\hat{y}_i^{box} - y_i^{box}\|_2^2 \dots\dots\dots(22)$$

Fungsi objektif untuk regresi lima titik fitur wajah adalah.

$$L_i^{landmark} = \|\hat{y}_i^{landmark} - y_i^{landmark}\|_2^2 \dots\dots\dots(23)$$

*Combined loss function* sebagai berikut.

$$\min \sum_{i=1}^N \sum_{v \in \{box, landmark\}} \alpha_v \beta_v^l L_i^v \dots\dots\dots(24)$$



Gambar 4.23 Non-Maximum Suppression

Gambar 4.23 Untuk gambar sebelah kiri adalah gambar bbox sebelum dilakukannya NMS dan yang sebelah kanan yang sudah dilakukan NMS. Metode digunakan untuk dengan cepat menghapus bingkai prediksi dengan kebetulan tinggi dan kalibrasi yang relatif tidak akurat.



Gambar 4.24 Bounding box regression output network

Gambar 4.24 Regresi Bounding-Box dalam output network ini Tujuannya adalah untuk menyempurnakan jendela prediksi ini untuk membawanya lebih dekat ke nilai sebenarnya.

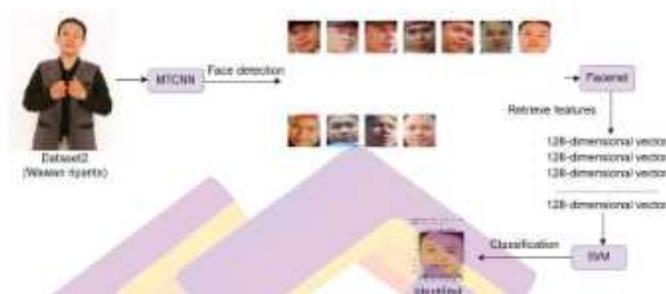


Gambar 4.25 Face detect dengan Combined loss function (dot, box, landmark)

Gambar 4.25 Menghasilkan 5 landmark wajah yaitu ditandai dengan titik(dot) di dalam box dengan keterangan sebagai berikut. ('box': [370, 176, 224, 302], 'confidence': 0.9996788501739502, 'key points': {'left\_eye': (447, 303), 'right\_eye': (550, 293), 'nose': (521, 350), 'mouth\_left': (470, 419), 'mouth\_right': (554, 412)} dimana box menyediakan x, y dari kiri bawah kotak batas, serta lebar dan tinggi kotak. *Confidence* yaitu kemungkinan keyakinan dari prediksi. *key*

points menyediakan dict dengan titik-titik untuk mata kiri, mata kanan, hidung, mulut kiri, dan mulut kanan.

#### 4.4 Identifikasi Wajah Tanpa Masker Pada Dataset2 menggunakan Multi Task Cascade Convolutional Neural Network



Gambar 4. 26 Proses identifikasi wajah salah satu dataset 2 (tanpa masker)

Dalam identifikasi wajah pada dataset2 ini, dimulai dari deteksi wajah dengan MTCNN, dilanjutkan dengan penggunaan FaceNet model untuk membuat *embedding* wajah yang menghasilkan 128-dimensional vectors, berikutnya melakukan pengklasifikasi SVM untuk memprediksi identitas wajah tertentu.

Dalam proses identifikasi ini ada dua mode utama untuk dilakukan yaitu sebagai berikut:

- Verifikasi Wajah: Pemetaan satu-ke-satu dari wajah tertentu terhadap identitas yang diketahui (misalnya apakah ini orangnya?).
- Identifikasi Wajah: Pemetaan satu-ke-banyak untuk wajah tertentu terhadap database wajah yang dikenal (misalnya, siapa orang ini?).

##### 4.4.1 Deteksi Wajah Untuk *Face Recognition*

Tahapan ini merupakan tahapan preprocessing dimana langkah pertama adalah memuat gambar sebagai array NumPy, yang dapat dicapai menggunakan pustaka PIL dan fungsi `open()`, peneliti juga mengkonversi gambar ke RGB, untuk

berjaga-jaga jika gambar memiliki saluran alfa atau hitam dan putih. Selanjutnya, membuat kelas detektor wajah Multi-Task Cascaded Convolutional Neural Network dan menggunakannya untuk mendeteksi semua wajah dalam foto yang dimuat dalam dataset2. Di sini Hasilnya adalah daftar kotak batas, di mana setiap kotak batas menentukan sudut kiri bawah kotak batas, serta lebar dan tinggi. Lalu gunakan koordinat untuk mengekstrak wajah. Kemudian menggunakan PIL *library* untuk mengubah ukuran gambar kecil wajah ini ke ukuran yang diperlukan; secara khusus, model mengharapkan wajah input persegi dengan bentuk 160×160. Berikutnya peneliti menggunakan face dataset2 yang merupakan kumpulan data kecil yang berisi foto-foto. Berikut daftar dataset2 yang dikumpulkan dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Dataset2

| Train          | val            |
|----------------|----------------|
| aldi_alkulakat | aldi_alkulakat |
| elfrida        | elfrida        |
| wawan_riyanto  | wawan_riyanto  |
| yohan          | yohan          |

Dalam deteksi wajah langkah pertama adalah mendeteksi wajah di setiap foto dan mengurangi set data menjadi serangkaian wajah saja. Peneliti menggunakan fungsi *extract\_face()*, dengan dilakukan percobaan dari salah satu direktori data dengan spesifikasi folder berikut (*folder = 'F:/Yanti/Tesis/face detection/data/train/wawan\_riyanto/'*). Hasil dari penggunaan fungsi *extract\_face()* dalam bentuk array NumPy berisi data piksel wajah. Hasil ditampilkan tabel 4.4 berikut.

Tabel 4.4 Array NumPy yang berisi data piksel wajah dari direktori Wawan riyanto

|                  |
|------------------|
| 1 (160, 160, 3)  |
| 2 (160, 160, 3)  |
| 3 (160, 160, 3)  |
| 4 (160, 160, 3)  |
| 5 (160, 160, 3)  |
| 6 (160, 160, 3)  |
| 7 (160, 160, 3)  |
| 8 (160, 160, 3)  |
| 9 (160, 160, 3)  |
| 10 (160, 160, 3) |
| 11 (160, 160, 3) |

Selain data array sebuah gambar diciptakan berisi wajah-wajah yang terdeteksi dalam direktori Wawan riyanto. Berikut hasilnya.



Gambar 4. 27 wajah yang terdeteksi dalam direktori training Wawan riyanto

Pada gambar 4.27 memperlihatkan bahwa setiap wajah berhasil terdeteksi dan menunjukkan bahwa gambar memiliki berbagai pencahayaan, warna kulit, dan orientasi di wajah yang terdeteksi. Setelah proses deteksi wajah berhasil, Selanjutnya, peneliti melakukan perluasan untuk melangkahi setiap subdirektori untuk set data train dan val, mengekstrak wajah, dan menyiapkan set data dengan nama sebagai label output untuk setiap wajah yang terdeteksi.

Peneliti menggunakan fungsi `load_faces()` untuk memuat semua wajah ke dalam daftar untuk direktori dari dataset2. Pemanggilan fungsi `load_faces()` untuk setiap sub direktori folder train dan val. Setiap wajah memiliki satu label, nama, yang dapat diambil dari nama direktori. Kemudian dilakukan pemanggilan `load_dataset()` untuk mengambil nama direktori dan menyimpan hasilnya dalam satu file array NumPy yang terkompresi melalui fungsi `savez_compressed()`.

Dalam proses ini membutuhkan waktu untuk menampilkan hasil dari load dataset train dan val. Pada tabel 4.5 akan menampilkan hasil dari keseluruhan proses yang sudah di deskripsikan sebelumnya.

Tabel 4.5 Hasil proses dari load data training dan validasi

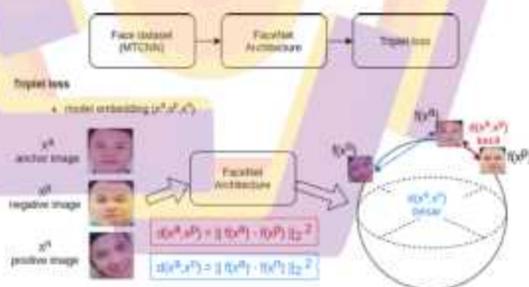
| Hasil   |
|---|
| >loaded 10 examples for class: elfrida        |
| >loaded 11 examples for class: aldi_alkulakat |
| >loaded 11 examples for class: wawan_riyanto  |
| >loaded 10 examples for class: yohan          |
| (42, 160, 160, 3) (42,)                       |
| >loaded 5 examples for class: aldi_alkulakat  |
| >loaded 5 examples for class: elfrida         |
| >loaded 5 examples for class: wawan_riyanto   |
| >loaded 5 examples for class: yohan           |
| (20, 160, 160, 3) (20,)                       |

Pada tabel 4.5 Proses yang berjalan yaitu pertama, semua foto dalam set data “train” dimuat, kemudian wajah diekstraksi, menghasilkan 42 sampel dengan input wajah persegi dan string label kelas sebagai output. Kemudian kumpulan data “val” dimuat, menyediakan 20 sampel yang dapat digunakan sebagai kumpulan

data pengujian. Kedua set data kemudian disimpan ke file array NumPy *Compressed* dengan nama “tanpa-masker.npz”. Kumpulan data siap untuk diberikan ke dalam model *face detection*.

#### 4.4.2. Membuat Penyematan Wajah (*Face Embeddings*)

Tahapan ini merupakan proses *Feature Extraction* dimana dilakukan dengan menggunakan FaceNet Model. Pada tahapan ini dilakukan penyematan wajah atau Face embedding dengan fungsi *get\_embedding()*, dimana Face embedding itu sendiri adalah vektor yang mewakili fitur yang diekstrak dari wajah. Ini kemudian dapat dibandingkan dengan vektor yang dihasilkan untuk wajah lain. Misalnya, vektor lain yang dekat (dengan beberapa ukuran) mungkin orang yang sama, sedangkan vektor lain yang jauh (dengan beberapa ukuran) mungkin orang yang berbeda. Berikut visualisasi prosesnya.



Gambar 4.28 Contoh pada anchor, positif, dan negatif dengan menerapkan kerugian Triplet Loss.

Rumus yang digunakan sebagai berikut.

Triplet loss

$$\sum_i [ \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^c)\|_2^2 + \alpha ]_+ \dots \dots \dots (25)$$

### Triplet Selection

$$\begin{aligned} \text{Argmax } ||f(x^+) - f(x^-)||_2^2 &= \text{Eq(1)} \\ \text{Argmin } ||f(x^+) - f(x^0)||_2^2 &= \text{Eq(2)} \end{aligned} \quad \dots\dots\dots(26)$$

Seperti yang terlihat pada gambar 4.28, pertama memuat set data wajah yang terdeteksi sebelumnya oleh MTCNN menggunakan fungsi *load()* *NumPy*. Kemudian memuat model FaceNet untuk mengubah wajah menjadi penyematan wajah. Di sini menggunakan arsitektur facenet dan menerapkan kerugian *Triplet Loss*.

Ekstraksi fitur ini akan membuat gambar wajah menjadi wajah (penyematan) dalam 128 dimensi. Hasil lengkap mengubah setiap wajah menjadi face embedding di data train dan test(val) ditunjukkan pada gambar 4.29.

```
Loaded: (42, 160, 160, 3) (42,) (20, 160, 160, 3) (20,)
WARNING:tensorflow:No training configuration found in the
el was *not* compiled. Compile it manually.
Loaded Model
(42, 128)
(20, 128)
```

Gambar 4.29 Hasil Face Embedding

Seperti yang terlihat pada gambar 4.29, bahwa dataset wajah dimuat dengan benar dan begitu pula modelnya. Dataset train kemudian diubah menjadi 42 embeddings wajah, masing-masing terdiri dari 128 elemen vektor. 20 contoh dalam dataset pengujian juga sesuai untuk dikonversi menjadi embeddings wajah.

#### 4.4.3 Klasifikasi Wajah (*Face Classification*)

Di Bagian ini, peneliti mengembangkan model untuk mengklasifikasikan embeddings wajah dalam dataset2. Pertama memuat face embeddings dataset yang sudah terbentuk sebelumnya, kemudian normalisasi vektor dengan *L2 norm*

menggunakan *Normalizer class in scikit-learn*. L2 norm yang dihitung sebagai akar kuadrat dari jumlah nilai vektor kuadrat, selanjutnya, variabel target string untuk setiap nama dataset dikonversi menjadi bilangan bulat melalui kelas *LabelEncoder* di *scikit-learn*, lalu fit model dengan *Linear Support Vector Machine (SVM)* menggunakan kelas *SVC* di *scikit-learn* dan mengatur atribut 'kernel' ke 'linear'. Juga probabilitas ketika membuat prediksi, yang dapat dikonfigurasi dengan mengatur 'probabilitas' ke 'True'. Berikutnya, mengevaluasi model. Ini dapat dicapai dengan menggunakan model fit untuk membuat prediksi untuk setiap contoh dalam kumpulan data training dan testing, kemudian menghitung akurasi klasifikasi. Model dievaluasi pada train dan test dataset, dengan akurasi klasifikasi dataset train 42, test 20, accuracy train 100.000, test 100.000.

#### **4.4.3.1. Memplot wajah asli dan Prediksi**

Peneliti memuat set data wajah, khususnya wajah-wajah dalam kumpulan data pengujian. Pertama, memilih contoh acak dari set pengujian, lalu mendapatkan embedding, piksel wajah, prediksi kelas yang diharapkan, dan nama kelas yang sesuai. Selanjutnya menggunakan face embedding sebagai input untuk membuat prediksi tunggal dengan model fit, disini dapat memprediksi bilangan bulat kelas dan probabilitas prediksi. Kemudian mendapatkan nama untuk bilangan bulat kelas yang diprediksi, dan probabilitas untuk prediksi ini, mencetak informasi, memplot piksel wajah bersama dengan nama dan probabilitas yang diprediksi.

Hasil dari pengujian pada dataset2 dengan data foto tanpa mengenakan masker menampilkan prediksi dan ekspektasi dari dataset yang diberikan serta

tingkat confidence yang dihasilkan. Berikut adalah hasil dari pengujian pada dataset dua.

Tabel 4.6 Terdeteksi wajah yang diidentifikasi dengan benar oleh SVM Classifier

| Hasil Identifikasi dari masing masing identitas   |  |
|---|--|
| <p>Predicted: aldi_kulakat (86.99%)<br/>Expected: aldi_kulakat</p> <p>aldi_kulakat (86.99%)</p>  | <p>Predicted: yohan (74.40%)<br/>Expected: yohan</p> <p>yohan (74.40%)</p>                          |
| <p>Predicted: elfrida (85.88%)<br/>Expected: elfrida</p> <p>elfrida (85.88%)</p>                 | <p>Predicted: wawan_riyanto (83.35%)<br/>Expected: wawan_riyanto</p> <p>wawan_riyanto (83.35%)</p>  |

Table 4.6 menunjukkan foto Aldi kulakat yang telah terprediksi dengan benar serta memperoleh keyakinan 86.919%, foto Yohan terprediksi dengan benar dan memperoleh keyakinan 74.404%, foto Elfrida terprediksi dengan benar mendapatkan keyakinan 85.885%, kemudian foto dari Wawan riyanto yang telah terprediksi dengan benar dan mendapat keyakinan 83.350%. Untuk foto yang ditampilkan merupakan foto yang diambil dengan cara pengacakan foto.

Hasil analisis identifikasi wajah pada dataset tanpa masker yang telah dilakukan pengujian disajikan dalam bentuk tabel berikut:

Tabel 4. 7 Analisis foto Aldi alkulakat dari data testing

|                   |   |   |   |   |   |
|-------------------|---|---|---|---|---|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Prediksi</b>   | adi_alkulak<br>at   | adi_alkula<br>kat   | adi_alkulak<br>at   | adi_alkulak<br>at   | adi_alkula<br>kat   |
| <b>Ekspetasi</b>  | adi_alkulak<br>at   | adi_alkula<br>kat   | adi_alkulak<br>at   | adi_alkulak<br>at   | adi_alkula<br>kat   |
| <b>Confidence</b> | 90.148  | 86.479  | 87.747  | 80.180  | 80.514  |

Dari tabel 4.7 semua gambar yang ditampilkan telah terprediksi dengan benar dan mendapat keyakinan tertinggi 90.148% dan terendah 80.180% dari dataset testing Aldi alkulakat, untuk rata - rata yang diperoleh yaitu 85.01%.

Tabel 4.8 Analisis foto Elfrida dari data testing

|                   |   |   |   |   |   |
|-------------------|---|---|---|---|---|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Prediksi</b>   | elfrida   | elfrida   | elfrida   | elfrida   | elfrida   |
| <b>Ekspetasi</b>  | Elfrida   | elfrida   | elfrida   | Elfrida   | elfrida   |
| <b>Confidence</b> | 79.460  | 79.194  | 82.485  | 77.403  | 85.885  |

Dari tabel 4.8 semua gambar yang ditampilkan telah terprediksi dengan benar dan mendapat keyakinan tertinggi 85.885% dan terendah 77.403% dari dataset testing Elfrida, untuk rata - rata keyakinan yang diperoleh yaitu 80,89%.

Tabel 4.9 Analisis foto Yohan dari data testing

|                   |   |   |   |   |   |
|-------------------|---|---|---|---|---|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Prediksi</b>   | yohan   | yohan   | yohan   | yohan   | yohan   |
| <b>Ekspetasi</b>  | yohan   | yohan   | yohan   | yohan   | yohan   |
| <b>Confidence</b> | 57.991  | 74.404  | 57.082  | 71.121  | 74.321  |

Dari tabel 4.9 semua gambar yang ditampilkan telah terprediksi dengan benar dan mendapat keyakinan tertinggi 74.404% dan terendah 57.082% dari dataset testing Yohan, untuk rata - rata yang diperoleh yaitu 66.98%

Tabel 4.10 Analisis foto Wawan Riyanto dari data testing

|                   |   |   |   |   |
|-------------------|---|---|---|---|
| <b>Foto</b>       |  |  |  |  |
| <b>Prediksi</b>   | wawan<br>_riyanto   | wawan<br>_riyanto   | wawan<br>_riyanto   | wawan<br>_riyanto   |
| <b>Ekspetasi</b>  | wawan<br>_riyanto   | wawan<br>_riyanto   | wawan<br>_riyanto   | wawan<br>_riyanto   |
| <b>Confidence</b> | 81.155  | 81.810  | 81.248  | 83.380  |

Dari tabel 4.10 semua gambar yang ditampilkan telah terprediksi dengan benar dan mendapat keyakinan tertinggi 83.380% dan terendah 81.810% dari dataset testing Wawan Riyanto, untuk rata - rata yang diperoleh yaitu 81.90%.

Dari seluruh analisis data yang dihasilkan pada proses identifikasi wajah dengan *Multi Task Cascade Convolutional Neural Network* pada dataset wajah tanpa masker memperoleh rata - rata keyakinan 78.69%.

#### 4.5 Identifikasi wajah bermasker pada dataset3 menggunakan *Multi Task Cascade Convolutional Neural Network*

Setelah melakukan pelatihan pada dataset 1 dimana data telah dilatih untuk mengenali masker wajah dan menghasilkan akurasi tinggi yaitu 98% untuk training, 99,99% untuk testing pada wajah dengan label "Mask". 100.00% untuk wajah dengan label "No mask", dan identifikasi wajah pada dataset2 dengan rata-rata keyakinan sebesar 78,69%. Peneliti melakukan pengujian pada dataset3 yaitu dengan identitas pada dataset kedua dengan wajah yang sudah terpasang masker secara virtual. Berikut gambaran prosesnya pada dataset wajah bermasker.



Gambar 4.30 identifikasi wajah salah satu dataset 3 (dengan masker)

Langkah yang dilakukan peneliti sama dengan langkah yang peneliti gunakan dalam identifikasi wajah pada dataset2 yaitu dengan *Multi-Task Cascaded Convolutional Neural Network* untuk melakukan deteksi wajah dalam proses pengenalan wajah bermasker (*with mask*), di tahap ini bertugas untuk menemukan dan mengekstrak wajah dari foto wajah bermasker. Serta *FaceNet* model digunakan untuk membuat embedding wajah with mask untuk setiap wajah bermasker yang

terdeteksi, selanjutnya akan dikembangkan model pengklasifikasi *Linear Support Vector Machine (SVM)* untuk memprediksi identitas wajah tertentu.

#### 4.5.1 Deteksi Wajah Bermasker (*Faces Mask Detection*)

Deteksi wajah dengan masker dengan Multi-Task Cascaded Convolutional Neural Network dan menggunakannya untuk mendeteksi semua wajah dalam foto yang dimuat dalam dataset3 (with mask) dengan data training 42 dan testing 20. Hasil dari pengujian ini ternyata wajah dengan masker mampu terdeteksi dengan *Facial Landmark Localization* yang dilakukan oleh metode tersebut. Berikut hasil dari *Face Mask Detection* seluruh dataset train masing - masing orang. Yang pertama hasil dari tahapan deteksi wajah dengan masker, menggunakan *Deep Learning*, hasil ditunjukkan pada gambar 4.31.

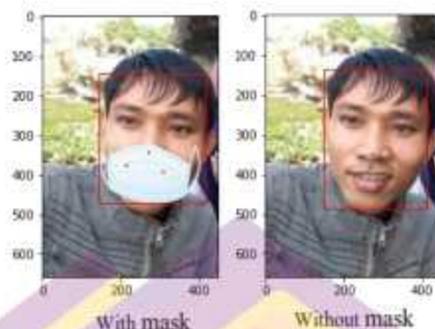


Gambar 4.31 Deteksi wajah bermasker

Pada gambar 4.31 memperlihatkan bahwa wajah dengan masker telah terdeteksi dengan ditandai pada box merah yang tergambar.

Kemudian setelah wajah dengan masker terdeteksi selanjutnya melakukan *facial landmark localization* dengan *Multi-Task Cascaded Convolutional Neural Network*, untuk mendapatkan lokasi mata, hidung dan mulut, dalam kasus wajah dengan masker posisi hidung dan wajah tertutupi oleh masker akan tetapi metode

ini masih mampu menemukan lokasi hidung dan mulut, yang akan ditunjukkan pada gambar 4.32 berupa foto perbandingan foto without mask dan with mask.



Gambar 4. 32 Face detection dengan MTCNN

Seperti yang terlihat pada gambar 4.32 gambar sebelah kiri adalah gambar with mask dan kanan gambar without mask dan memperlihatkan bahwa deteksi lokasi mata, hidung dan mulut dalam posisi yang sama.

Selanjutnya adalah hasil dari *face mask detection* dari masing - masing orang pada dataset ketiga yang menunjukkan semua wajah dalam data training telah terdeteksi, artinya langkah berikutnya bisa dilakukan yaitu proses identifikasi dan verifikasi.



Gambar 4.33 Face mask deteksi dari Aldi alkulakat



Gambar 4.34 Face mask deteksi dari Elfrid



Gambar 4.35 Face mask deteksi dari Yohan



Gambar 4.36 Face mask deteksi dari Wawan riyanto

Setelah semua wajah dengan masker terdeteksi maka selanjutnya peneliti melakukan pengujian dengan 20 data validasi foto wajah dengan masker dan hasilnya metode tersebut mampu memprediksi dengan benar setiap foto yang ditampilkan secara acak dan juga menghasilkan tingkat keyakinan. Berikut hasil yang diberikan.

Tabel 4.11 Terdeteksi wajah dengan masker yang diidentifikasi dengan benar oleh SVM Classifier

| Hasil Identifikasi dari masing masing identitas   |  |
|---|--|
| <p>Predicted: aldi_alkulakat (88.154)<br/>Expected: aldi_alkulakat</p> <p>aldi_alkulakat (88.154)</p>  | <p>Predicted: yohan (77.848)<br/>Expected: yohan</p> <p>yohan (77.848)</p>        |
| <p>Predicted: wawan_riyanto (86.475)<br/>Expected: wawan_riyanto</p> <p>wawan_riyanto (86.475)</p>     | <p>Predicted: elfrida (84.582)<br/>Expected: elfrida</p> <p>elfrida (84.582)</p>  |

Pada Tabel 4.11 menunjukkan foto Aldi alkulakat dengan mengenakan masker telah terprediksi dengan benar dan mendapatkan nilai keyakinan sebesar 88.154%, kemudian untuk Yohan juga telah terprediksi dengan benar dan mendapat keyakinan 77.848%. Wawan riyanto mendapat angka 86.475%, untuk Elfrida dan juga telah terprediksi dengan benar dengan keyakinan 84.582%.

Foto yang ditampilkan merupakan foto yang di diambil dari hasil pengacakan foto secara random. Untuk keseluruhan hasil analisis pada identifikasi wajah bermasker pada dataset3 yang telah dilakukan pengujian disajikan dalam bentuk tabel berikut.

Tabel 4.12 Analisis data wajah Aldi alkulakat menggunakan masker

|                   |   |   |   |   |   |
|-------------------|---|---|---|---|---|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Prediksi</b>   | adi_alkulak<br>kat  | adi_alkulak<br>kat  | adi_alkulak<br>kat  | adi_alkulak<br>at   | adi_alkulak<br>at   |
| <b>Ekspetasi</b>  | adi_alkulak<br>kat  | adi_alkulak<br>kat  | adi_alkulak<br>kat  | adi_alkulak<br>at   | adi_alkulak<br>at   |
| <b>Confidence</b> | 88.154  | 87.427  | 86.942  | 87.829  | 87.899  |

Dari tabel 4.12 semua gambar dengan wajah bermasker yang ditampilkan telah terprediksi dengan benar dan mendapatkan keyakinan tertinggi 88.154% dan terendah 86.942% dari dataset pengujian Aldi alkulakat, untuk rata - rata akurasi yang diperoleh yaitu 87.65%.

Tabel 4.13 Analisis data wajah Elfrida menggunakan masker

|                   |  |  |  |  |  |
|-------------------|--|--|--|--|--|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Prediksi</b>   | elfrida  | elfrida  | elfrida  | elfrida  | elfrida  |
| <b>Ekspetasi</b>  | elfrida  | elfrida  | elfrida  | elfrida  | elfrida  |
| <b>Confidence</b> | 88.871   | 84.582   | 77.929   | 72.941   | 74.266   |

Dari tabel 4.13 semua gambar dengan wajah bermasker yang ditampilkan telah terprediksi dengan benar dan mendapatkan keyakinan tertinggi 88.871% dan terendah 72.941% dari dataset pengujian Elfrida, untuk rata - rata yang diperoleh yaitu 79.72%.

Tabel 4.14 Analisis data wajah Yohan menggunakan masker

|                   |   |   |   |   |   |
|-------------------|---|---|---|---|---|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Prediksi</b>   | yohan   | yohan   | yohan   | yohan   | yohan   |
| <b>Ekspetasi</b>  | yohan   | yohan   | yohan   | yohan   | yohan   |
| <b>Confidence</b> | 67.879  | 78.146  | 73.627  | 74.649  | 72.304  |

Dari tabel 4.14 gambar dengan wajah bermasker yang ditampilkan telah terprediksi dengan benar dan mendapat keyakinan tertinggi 78.146% dan terendah 67.879% dari dataset pengujian Yohan, untuk rata - rata yang diperoleh yaitu 73.32%.

Tabel 4.15 Analisis data wajah Wawan riyanto menggunakan masker

|                   |   |   |   |   |
|-------------------|---|---|---|---|
| <b>Foto</b>       |  |  |  |  |
| <b>Prediksi</b>   | wawan<br>_riyanto   | wawan<br>_riyanto   | wawan<br>_riyanto   | wawan<br>_riyanto   |
| <b>Ekspetasi</b>  | wawan<br>_riyanto   | wawan<br>_riyanto   | wawan<br>_riyanto   | wawan<br>_riyanto   |
| <b>Confidence</b> | 89.012  | 73.276  | 86.475  | 86.507  |

Dari tabel 4.15 dari semua gambar dengan wajah bermasker yang ditampilkan telah terprediksi dengan benar dan mendapatkan keyakinan tertinggi 89.012% dan terendah 81.810% dari dataset pengujian Wawan Riyanto, untuk rata - rata yang diperoleh yaitu. 83.818%.

Dari seluruh analisis data yang dihasilkan pada deteksi wajah bermasker dalam *Face Recognition using Multi Task Cascade Convolutional Neural Network* pada dataset wajah bermasker memperoleh rata - rata 81.18%.

#### 4.6 Pengujian pada dataset real wajah dengan beragam jenis masker

Penelitian melakukan pengujian terhadap dataset real yaitu dataset dengan berbagai gambar wajah yang mengenakan berbagai jenis masker, pengujian dilakukan dengan total data 60 yang kemudian dibagi menjadi data latih 40 dan data testing (validasi) 20 dengan empat identitas yaitu (Wawan Riyanto, Elfrida, Rofik dan Yohan). Berikut hasil pengujian dan analisisnya.

Tabel 4.16 Hasil deteksi wajah dengan MTCNN pada data real (data train)

|               |   |
|---------------|---|
| Elfrida       |    |
| Rofik         |   |
| Wawan Riyanto |  |
| Yohan         |  |

Hasil dan analisis identifikasi wajah berdasarkan pengujian pada dataset dengan berbagai jenis masker. Ditunjukkan pada tabel berikut.

Tabel 4.17 Hasil prediksi, ekspektasi dan confidence dari Elfrida pada dataset real

|                   |   |   |   |   |  |
|-------------------|---|---|---|---|--|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Predicted</b>  | elfrida   | elfrida   | elfrida   | elfrida   | elfrida  |
| <b>Expected</b>   | elfrida   | elfrida   | elfrida   | elfrida   | elfrida  |
| <b>Confidence</b> | 81.222  | 86.754  | 86.149  | 84.014  | 81.884   |

Hasil analisis pada tabel 4.17 yaitu bahwa terdeteksi wajah dengan masker secara real dari Elfrida yang telah terprediksi dengan benar oleh SVM classifier dan memperoleh keyakinan tertinggi yaitu 86.754 % dan terendah 81.222% dengan rata-rata yang diperoleh yaitu 84.005%.

Tabel 4.18 Hasil prediksi, ekspektasi dan confidence dari Rofik pada dataset real

|                   |   |   |   |   |  |
|-------------------|---|---|---|---|--|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Predicted</b>  | rofik   | rofik   | rofik   | rofik   | rofik  |
| <b>Expected</b>   | rofik   | rofik   | rofik   | rofik   | rofik  |
| <b>Confidence</b> | 81.811  | 77.028  | 73.477  | 83.422  | 82.301   |

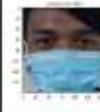
Hasil analisis pada tabel 4.18 bahwa terdeteksi wajah dengan masker secara real dari Rofik yang telah terprediksi dengan benar oleh SVM classifier dan memperoleh keyakinan tertinggi yaitu 83.422 % dan terendah 73.477% dengan rata-rata yang diperoleh yaitu 79.608 %.

Tabel 4.19 Hasil prediksi, ekspektasi dan confidence dari Wawan riyanto pada dataset real

|                   |   |   |   |   |   |
|-------------------|---|---|---|---|---|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Predicted</b>  | wawan riyanto   |
| <b>Expected</b>   | wawan riyanto   |
| <b>Confidence</b> | 91.937  | 85.226  | 91.967  | 84.012  | 86.386  |

Hasil analisis pada tabel 4.19 yaitu bahwa terdeteksi wajah dengan masker secara real dari Wawan riyanto yang telah terprediksi dengan benar oleh SVM classifier dan memperoleh keyakinan tertinggi yaitu 91.967 % dan terendah 84.012% dengan rata - rata yang diperoleh yaitu 87.906 %.

Tabel 4.20 Hasil prediksi, ekspektasi dan confidence dari Yohan pada dataset real

|                   |  |  |  |  |  |
|-------------------|--|--|--|--|--|
| <b>Foto</b>       |  |  |  |  |  |
| <b>Predicted</b>  | yohan  | yohan  | yohan  | yohan  | yohan  |
| <b>Expected</b>   | yohan  | yohan  | yohan  | yohan  | yohan  |
| <b>Confidence</b> | 85.710   | 83.984   | 79.980   | 82.096   | 83.795   |

Hasil analisis pada tabel 4.20 yaitu bahwa terdeteksi wajah dengan masker secara real dari Yohan yang telah terprediksi dengan benar oleh SVM classifier dan memperoleh keyakinan tertinggi yaitu 85.710 % dan terendah 79.980% dengan rata - rata yang diperoleh yaitu 83.113 %.

Dari seluruh analisis data real dengan berbagai wajah dengan beragam masker yang dihasilkan pada identifikasi wajah bermasker menggunakan *Multi Task Cascade Convolutional Neural Network* memperoleh rata - rata keyakinan 83.659%.

#### 4.7 Pengujian diluar dataset latih pada pengujian dataset 3

Penelitian ini melakukan pengujian di luar data latih untuk pengujian pada dataset3 wajah dengan masker secara virtual, dimana data gambar diletakkan pada dataset uji atau validasi untuk kelas wawan riyanto tanpa memasukan data tersebut kedalam data latih. Hasilnya ditunjukkan pada gambar berikut.



Gambar 4. 37 Hasil pengujian diluar data latih

Dari gambar 3.37 menunjukkan bahwa gambar terprediksi dengan salah dimana ditunjukkan pada keterangan expected Wawan riyanto akan tetapi terprediksi sebagai Yohan dengan keyakinan yang rendah yaitu 38.117%. Tujuan dari pengujian ini adalah untuk mengetahui bagaimana hasil apabila dilakukan pengujian diluar dari dataset latih yang di berikan.

Fakta tentang data foto yang digunakan untuk pengujian di luar data latih adalah foto dari adik kandung salah satu dataset 3 yaitu saudara Yohan.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Kesimpulan yang dihasilkan pada penelitian ini diantaranya adalah.

- a. Pada penelitian ini telah dilakukan pengenalan deteksi masker wajah COVID-19 menggunakan OpenCV Deep Neural Network (DNN), TensorFlow dengan model MobileNetV2 yang dilatih dan di uji pada dataset1. Untuk membuat deteksi masker wajah peneliti melatih model dua kelas orang yang mengenakan masker dan orang-orang yang tidak mengenakan masker, hasil dari proses pelatihan mendapatkan akurasi 98% dan pengujian mendapatkan 99.99% untuk wajah dengan label "Mask", 100% untuk wajah dengan label "No mask".
- b. Pada identifikasi wajah yang menunjukkan (ni orangnya?) dan memverifikasi (siapa orang ini?) dari foto wajah berdasarkan dataset2 wajah tanpa masker, menggunakan metode MTCNN, dengan proses deteksi wajah menggunakan MTCNN, serta Facenet model sebagai *Feature Extraction*, FaceNet ini untuk membuat embedding dan tugas klasifikasi menggunakan *Linear Support Vector Machine (SVM)*. Metode yang digunakan di latih dan diuji dengan mengidentifikasi dan memverifikasi empat identitas yaitu Aldi kulakat, Elfrida, Yohan dan Wawan riyanto dengan jumlah dataset train 42 dan val 20. Menghasilkan ketepatan dan rata - rata akurasi dari data pengujian 78.69%.

- c. Metode MTCNN yang dilatih dan di uji untuk identifikasi wajah pada dataset3 (wajah dengan masker virtual) terbukti bahwa metode tersebut mampu mengenali wajah yang diberikan dengan benar dan memperoleh hasil rata - rata keyakinan 81.18%, dengan jumlah dataset train 42 dan val 20 dari empat identitas tersebut.
- d. Metode MTCNN yang di uji untuk mengidentifikasi wajah dengan beragam masker wajah (dataset3 data real) menunjukkan bahwa metode tersebut mampu memprediksi dengan benar setiap wajah yang di ujikan dengan memperoleh rata - rata keyakinan 83,659% dengan data train 40 dan val 20 dari empat identitas yang berbeda.
- e. Metode MTCNN yang diuji dengan dataset di luar data latih ternyata terbukti bahwa hasil yang diperoleh salah memprediksi dengan perolehan keyakinan rendah.

## 5.2. Saran

Terdapat beberapa saran untuk penelitian selanjutnya, diantaranya:

1. Perlu dilakukan pengujian terhadap data besar untuk penelitian selanjutnya untuk mengetahui apakah metode masih baik untuk digunakan.
2. Perlu dilakukan perbandingan antara algoritma untuk mengetahui mana algoritma yang terbaik.

## DAFTAR PUSTAKA

- Abadi, Martin, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." *arXiv preprint arXiv:1603.04467*, 2016.
- Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J. and Ogden, J.M., 1984. Pyramid methods in image processing. *RCA engineer*, 29(6), pp.33-41.
- Agarwal, S., Singh Punn, N., Sonbhadra, S. K., Tanveer, M., Nagabhushan, P., Pandian, S., & Saxena, P. (2020). *Unleashing the power of disruptive and emerging technologies amid COVID-19*. Praveer Saxena.
- Anwar, A. and Raychowdhury, A., 2020. Masked Face Recognition for Secure Authentication. *arXiv preprint arXiv:2008.11104*.
- Arsal, M., Wardijono, B.A. and Angraeni, D., 2020. Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN. *Jurnal Nasional Teknologi dan Sistem Informasi*, 6(1), pp.55-63.
- Bergen, J. R., & Ogden, J. M. (1984). Pyramid Methods in Image Processing Low Carbon Fuels Policies View project Energy System Modelling View project. In *researchgate.net*. <https://www.researchgate.net/publication/246727904>.
- Brown, Matthew, and David G. Lowe. "Invariant Features from Interest Point Groups." In *BMVC*, vol. 4. 2002.
- Brownlee, jason. 2020. How to Develop a Face Recognition System Using FaceNet in Keras. <https://machinelearningmastery.com>, diakses pada 16 November 2020.
- Cahyono, Ferry, Wirawan Wirawan, and Reza Fuad Rachmadi. "Face Recognition System using Facenet Algorithm for Employee Presence." In *2020 4th International Conference on Vocational Education and Training (ICOVET)*, pp. 57-62. IEEE, 2020.
- Cao, Jie, Zhe Su, Liyun Yu, Dongliang Chang, Xiaoxu Li, and Zhanyu Ma. "Softmax cross entropy loss with unbiased decision boundary for image classification." In *2018 Chinese Automation Congress (CAC)*, pp. 2028-2032. IEEE, 2018.
- Chen, Xuefeng, Xin Luo, Xingyao Liu, and Jikang Fang. "Eyes localization algorithm based on prior MTCNN face detection." In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 1763-1767. IEEE, 2019.
- Chowdary, G. Jignesh, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. "Face mask detection using transfer learning of inceptionv3." In *International Conference on Big Data Analytics*, pp. 81-90. Springer, Cham, 2020.
- Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine learning*, 20(3), pp.273-297.
- Damale, Radhika C., and Bazeshree V. Pathak. "Face recognition based attendance system using machine learning algorithms." In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 414-419. IEEE, 2018.
- Dang, Kirti, and Shanu Sharma. "Review and comparison of face detection algorithms." In *2017 7th International Conference on Cloud Computing, Data*

- Science & Engineering-Confluence*, pp. 629-633. IEEE, 2017.
- Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. *ACM International Conference Proceeding Series*, 148, 233–240. <https://doi.org/10.1145/1143844.1143874>
- Feng, S., Shen, C., Xia, N., Song, W., Fan, M. and Cowling, B.J., 2020. Rational use of face masks in the COVID-19 pandemic. *The Lancet Respiratory Medicine*, 8(5), pp.434-436.
- Fukushima, K., & Miyake, S. (1982). *Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition* (pp. 267–285). [https://doi.org/10.1007/978-3-642-46466-9\\_18](https://doi.org/10.1007/978-3-642-46466-9_18)
- Guillermo, M., Pascua, A., Billones, R., & Sybingco, E. (n.d.). COVID-19 Risk Assessment through Multiple Face Mask Detection using MobileNetV2 DNN. In *iscia2020.bit.edu.cn*. Retrieved January 14, 2021, from <http://iscia2020.bit.edu.cn/docs/20201114082420135149.pdf>
- Hjelmås, E., & Low, B. K. (2001). Face detection: A survey. *Computer Vision and Image Understanding*, 83(3), pp.236–274. <https://doi.org/10.1006/cviu.2001.0921>
- Hubel, D.H. and Wiesel, T.N., 1959. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3), pp.574-591.
- Ihsanuddin, "Pengumuman Mendadak Jokowi yang Kejutkan Pasien Corona," Kompas, <https://nasional.kompas.com/>, online accessed May 25, 2020
- Geitgey, A. 2016, *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning*. <https://medium.com/@ageitgey/machine-learning>, diakses pada 11 November 2020.
- G. S. Putri, "WHO Resmi Sebut Virus Corona Covid-19 sebagai Pandemi Global," Kompas, <https://www.kompas.com/>, nline accessed May 25, 2020
- Who coronavirus disease (covid-19) dashboard, <https://covid19.who.int/>, online accessed May 25, 2020
- Jain, Anil K., and Stan Z. Li. *Handbook of face recognition*. Vol. 1. New York: springer, 2011.
- Jose, Edwin, M. Greeshma, Mithun TP Haridas, and M. H. Supriya. "Face recognition based surveillance system using facenet and mtcnn on jetson tx2." In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pp. 608-613. IEEE, 2019.
- Khadatkar, Ashwin, Roshni Khedgaonkar, and K. S. Patnaik. "Occlusion invariant face recognition system." In *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, pp. 1-4. IEEE, 2016.
- Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. *Synthesis Lectures on Computer Vision*, 8(1), 1–207. <https://doi.org/10.2200/s00822ed1v01y201712cov015>
- Kim, G., Suhr, J. K., Jung, H. G., & Kim, J. (n.d.). Face Occlusion Detection by using B-spline Active Contour and Skin Color Information. In *ieeexplore.ieee.org*. Retrieved January 14, 2021, from <https://ieeexplore.ieee.org/abstract/document/5707762/>

- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), 541–551. <https://doi.org/10.1162/nc.1989.1.4.541>
- Li, Y., Zeng, J., Shan, S., & Chen, X. (n.d.). Patch-Gated CNN for Occlusion-aware Facial Expression Recognition. In *ieeexplore.ieee.org*. Retrieved January 14, 2021, from <https://ieeexplore.ieee.org/abstract/document/8545853/>
- Liang, Xiaoxi, Xiaodong Cai, Longze Li, and Yun Chen. "Facial feature extraction method based on shallow and deep fusion CNN." In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 50-53. IEEE, 2017.
- Lihong, Zhao, Song Ying, Zhu Yushi, Zhang Cheng, and Zheng Yi. "Face recognition based on multi-class SVM." In *2009 Chinese Control and Decision Conference*, pp. 5871-5873. IEEE, 2009.
- Loey, M., Manogaran, G., Taha, M.H.N. and Khalifa, N.E.M., 2020. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement*, 167, p.108288.
- Malkauthekar, M. D. (2013). Analysis of Euclidean distance and Manhattan distance measure in face recognition. *IET Conference Publications*, 2013(CP646), 503–507. <https://doi.org/10.1049/cp.2013.2636>
- Militante, S.V. and Dionisio, N.V., 2020, August. Real-Time Facemask Recognition with Alarm System using Deep Learning. In *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)* (pp. 106-110). IEEE.
- Nugroho, Anto Satriyo, Arief Budi Witarto, and Dwi Handoko. "Support vector machine." *Proceeding Indones. Sci. Meeting Cent. Japan*. 2003.
- Nyein, Thida, and Aung Nway Oo. "University Classroom Attendance System Using FaceNet and Support Vector Machine." In *2019 International Conference on Advanced Information Technologies (ICAIT)*, pp. 171-176. IEEE, 2019.
- I. William, D. R. Ignatius Moses Setiadi, E. H. Rachmawanto, H. A. Santoso and C. A. Sari, "Face Recognition using FaceNet (Survey, Performance Test, and Comparison)," 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 2019, pp. 1-6, doi: 10.1109/ICIC47613.2019.8985786.
- Prajna Bhandary. (2020). *Dataset 1*. <https://github.com/prajnasb/observations/tree/master/experiments/data>
- Rowley, H., Baluja, S., pattern, T. K.-I. T. on, & 1998, undefined. (n.d.). Neural network-based face detection. *ieeexplore.Ieee.Org*. Retrieved January 15, 2021, from <https://ieeexplore.ieee.org/abstract/document/655647/>
- Rosebrock, Adrian. 2020. "Pyimagesearch." linea. Available: <https://www.pyimagesearch.com>.
- Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823. 2015.
- Song, F., Zhang, D., Mei, D. and Guo, Z., 2007. A multiple maximum scatter

- difference discriminant criterion for facial feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(6), pp.1599-1606.
- Song, Lingxue, Dihong Gong, Zhifeng Li, Changsong Liu, and Wei Liu. "Occlusion robust face recognition based on mask learning with pairwise differential siamese network." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 773-782. 2019.
- Sun, Y., Wang, X. and Tang, X., 2014. Deep learning face representation by joint identification-verification. *arXiv preprint arXiv:1406.4773*.
- Sung, K.K. and Poggio, T., 1998. Example-based learning for view-based human face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1), pp.39-51.
- Wan, Li, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. "Regularization of neural networks using dropconnect." In *International conference on machine learning*, pp. 1058-1066. PMLR, 2013.
- Xia, Yizhang, Bailing Zhang, and Frans Coenen. "Face occlusion detection based on multi-task convolution neural network." In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 375-379. IEEE, 2015.
- Xiang, Jia, and Gengming Zhu. "Joint face detection and facial expression recognition with MTCNN." In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 424-427. IEEE, 2017.
- Yanagisawa, Hideaki, Takuro Yamashita, and Hiroshi Watanabe. "A study on object detection method from manga images using CNN." In *2018 International Workshop on Advanced Image Technology (IWAIT)*, pp. 1-4. IEEE, 2018.
- Yang, M.H., Kriegman, D.J. and Ahuja, N., 2002. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 24(1), pp.34-58.
- Yin, X. and Liu, X., 2017. Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Transactions on Image Processing*, 27(2), pp.964-975.
- P. Yu, J. Zhu, Z. Zhang, and Y. Han, "A familial cluster of infection associated with the 2019 novel coronavirus indicating possible person-to-person transmission during the incubation period," *The Journal of infectious diseases*, vol. 221(11), pp. 1757-1761, 2020.
- Pratama, Y., Istoningtyas, M. and Rasywir, E., 2019. Pengujian algoritma MTCNN (multi-task cascaded convolutional neural network) untuk sistem pengenalan wajah. *J. Media Inf. Budidarma*, 3(3), pp.240-247.
- Zaytseva, Ekaterina, and Jordi Vitrià. "A search based approach to non maximum suppression in face detection." In *2012 19th IEEE International Conference on Image Processing*, pp. 1469-1472. IEEE, 2012.
- Zhan, S., Tao, Q.Q. and Li, X.H., 2016. Face detection using representation learning. *Neurocomputing*, 187, pp.19-26.
- Zhang, K., Zhang, Z., Li, Z. and Qiao, Y., 2016. Joint face detection and alignment

- using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), pp.1499-1503.
- Shih, Frank. 2010, Face Image Processing and analysis, the Institute of Electrical and Electronics Engineers, Inc. diakses pada 20 Oktober 2020.
- Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539-546. IEEE, 2005.
- Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." In *European conference on computer vision*, pp. 818-833. Springer, Cham, 2014.

