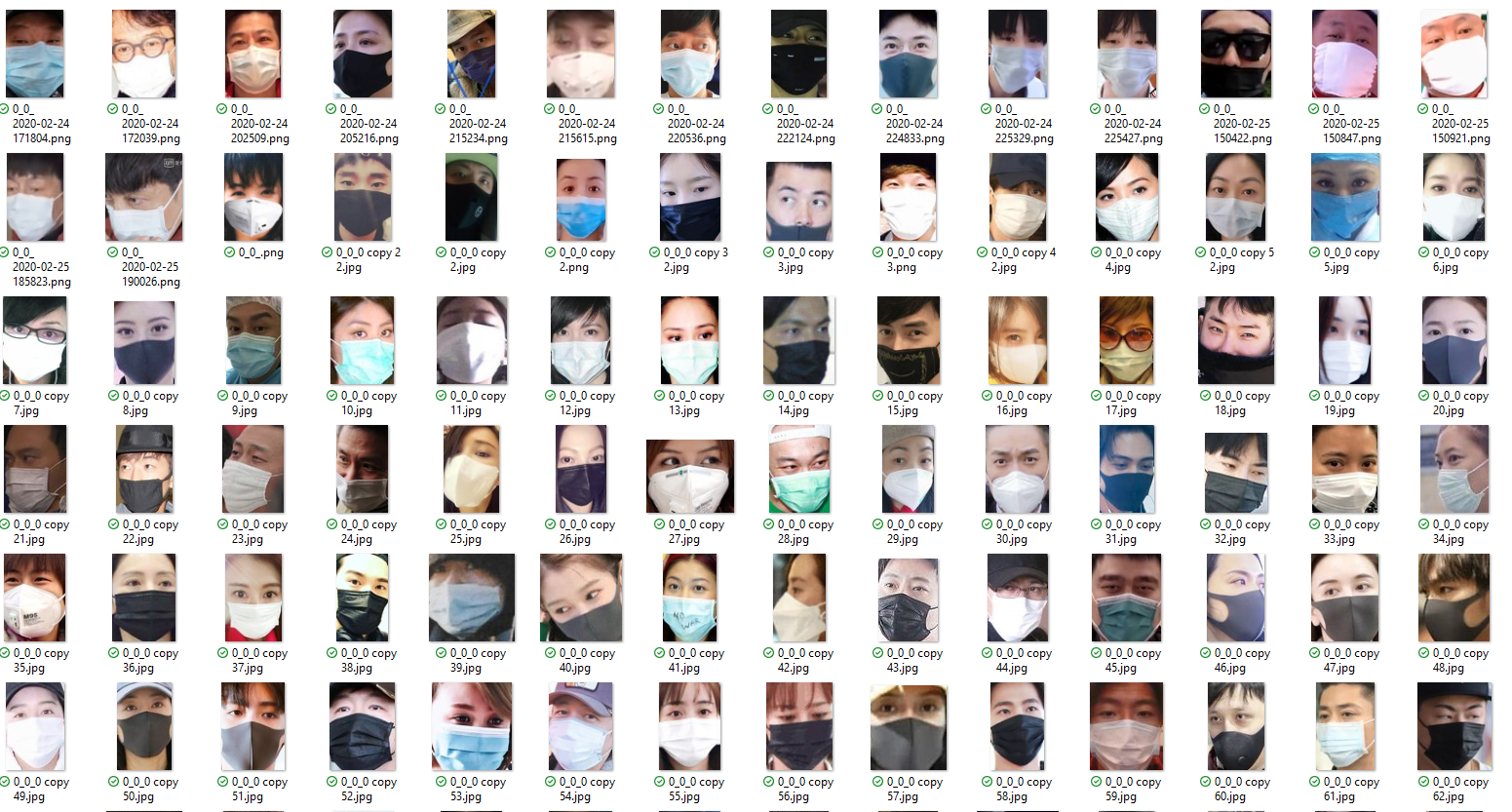
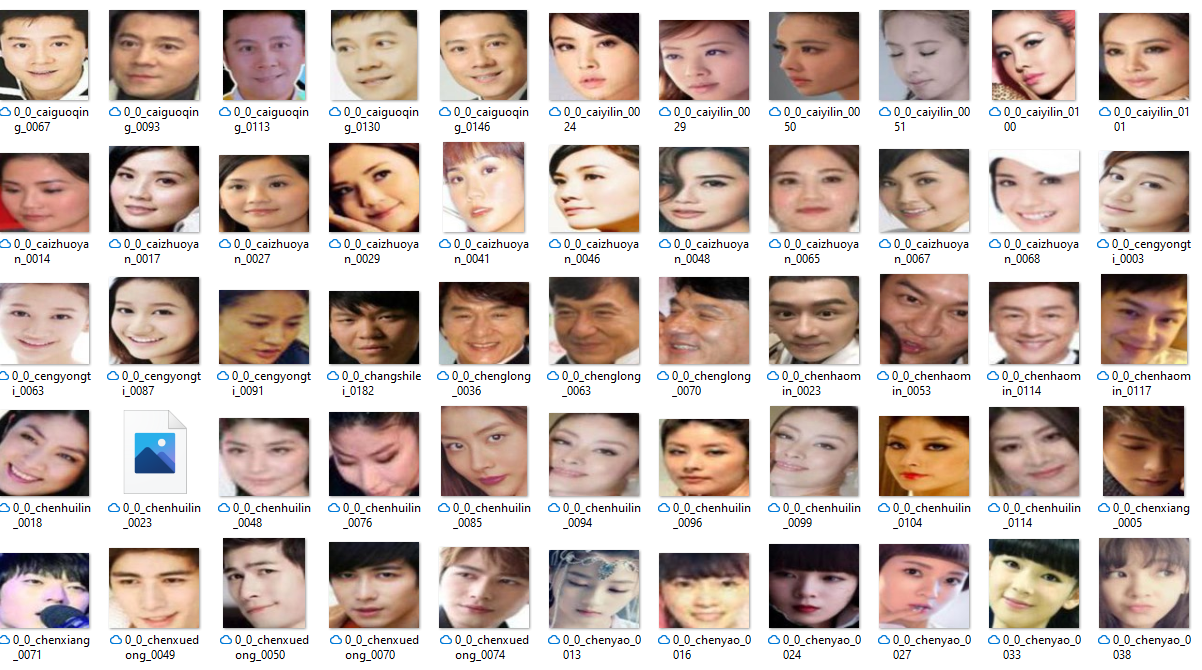
# BAB III METODE PENELITIAN

## 3.1 Deskripsi Data

Data yang digunakan dalam penelitian ini diperolah dari pengumpulan data citra orang yang menggunakan masker dan tidak menggunakan masker, yang merupakan data citra digunakan untuk melatih *machine learning.* Data ini memiliki jumlah 3.833 data citra yang terdiri dari 1.918 tanpa masker dan 1.915 menggunakan masker. Gambar 3.1 merupakan isi dari dataset yang digunakan.



Gambar 3.1 Data penggunaan masker

****

Gambar 3.2 Data penggunaan tanpa masker

## 3.2 Kebutuhan Perangkat *Hardware* dan *Software*

1. Spesifikasi Hardware
2. Processor AMD Ryzen 5 3550H 4 Core 8 Threads
3. RAM 16 GB DDR4 Dual-Channel 2400 Mhz
4. Nvidia GTX 1050 3GB
5. Webcam AUSDOM AF640 1080p
6. Spesifikasi Software
7. Sistem Operasi Windows 10 Home 64—bit
8. Google Collab

## 3.3 Perancangan Sistem

A screenshot of a computer

Description automatically generated with medium confidence

Gambar 3.3 Diagram Perancangan Sistem

Berdasarkan Gambar 3.3 diagram perancangan sistem dibagi menjadi 5 tahap yaitu *Preprocessing, Data Augmentation,* Latih Model CNN, Deteksi Masker dan Hitung Akurasi. Berikut penjelasan masing-masing tahapan tersebut.

### 3.3.1 *Data Augmentation*

proses data augmentasi untuk membuat variasi baru dalam dataset. Terdapat beberapa parameter variasi data augmentasi yang meliputi rotasi, pergeseran, *cropping, zooming,* dan beberapa variasi lainnya yang bertujuan membantu meningkatkan performa model pada *deep learning*.

### 3.3.2 *Preprocessing*

Pada proses *Preprocessing* ini bertujuan untuk mempersiapkan data sebelum dilakukan *Data Augmentation*. Tahap ini bertujuan untuk melakukan normalisasi, transformasi, dan pengubahan ukuran citra sebagai untuk gambar agar sesuai dengan syarat yang diperlukan oleh arsitektur VGG16Net sehingga lebih mudah untuk diproses oleh model *deep learning*. Berikut adalah tahapan preprocessing yang akan dilakukan.

#### **3.3.2.1 Rezise**

*Rezise* merupakan proses untuk mengubah ukuran citra menjadi dimensensi tertentu yang diperlukan oleh arsitektur jaringan VGG16Net. Pada penelitian ini ukuran citra akan di ubah ukurannya menjadi 224 x 224 piksel. Hal ini dilakukan dengan tujuan terdapat beberapa data citra yang ukuran pikselnya berbeda-beda sehingga kemungkinan bisa menimbulkan hasil akurasi berkurang atau rendah.

#### **3.3.2.2 Konversi BGR**

Setelah melakukan proses rezise, langkah berikutnya adalah melakukan proses konversi yang dimana arsitektur VGG16Net menggunakan saluran warna dalam urutan BGR (*Blue, Green, Red*) sebagai input. Jadi, jika awal citra dalam format RGB, maka perlu melakukan konversi saluran warna dari RGB ke BGR.

#### **3.3.2.3 Mean Subtraction**

Setelah dilakukan konversi maka nilai dari BGR akan dihitung rata-rata piksel dari setiap saluran BGR tersebut dari kumpulan data pelatihan VGG16Net. Setelah dihitung, nilai rata-rata tersebut akan dikurangkan dari setiap piksel citra. Hal ini bertujuan untuk membantu dalam memusatkan data dan menghilangkan komponen cahaya tidak relevan. Dalam tahap ini, citra menjadi *zero-centered*

#### **3.3.2.4 Normalisasi**

Setelah memasuki tahap *Mean Subtraction*. Tahap normalisasi ini dilakukan untuk memperolej range piksel yang seragam dan mempermudah proses pelatihan untuk VGG16Net. Normalisasi dilakukan dengan membagi setiap piksel dengan standar deviasi dari setiap saluran warna. Hal ini bertujuan menghasilkan piksel dengan rata-rata nol dan standar deviasi 1. Hal ini untuk mengontrol variasi piksel dan mempermudaj konvergen selama pelatihan

Dalam tahap *Preprocessing*, penelitian ini menggunakan *framework* yang sudah di sediakan oleh *framework* Keras untuk melakukan proses *preprocessing* dalam arsitektur VGG16Net. Salah satu fungsi yang digunakan yaitu “preprocess\_input()” yang tersedia dalam modul “keras.application.vgg16”

### 3.3.4 Latih Model CNN

Dalam tahap ini akan dilakukan pelatihan dengan data training yang telah diaugmentasi. Pada penelitian ini dilakukan dengan menggunakan metode *convolutional neural network* (CNN).

### 3.3.5 Evaluasi Model

Setelah tahap latih model CNN, akan dilakukan evaluasi performa dalam model pada data testing. Dalam hal ini model akan diukur menggunakan *confusion matrix* yang meliputi akurasi, presisi, recall dam F1-score.

### 3.3.5 Model CNN

Setelah model dinyatakan memenuhi kriteria performa *confusion matrix,* model dapat disimpan untuk digunakan pada tahap untuk mendeteksi masker pada citra baru.

## 3.4 Desain Antarmuka

Icon

Description automatically generated

Gambar 3.4 Desain Antar Muka

Desain antar muka pada Gambar 3.4 digunakan untuk melakukan pendeteksi uji data citra menggunakan masker dan tidak menggunakan masker. Kemudian pada panel dalam *frame window* langsung menampilkan output dari *webcam* yang menerima input berupa video dan foto secara *realtime* langsung menampilkan tingkat akurasi pada pendeteksian tersebut.

## 3.5 Skenario Pengujian

Dalam penelitian ini dari jumlah dataset akan dibagi menjadi dua percobaan dari dataset awal berjumlah 3.833 (1.918 tanpa masker dan 1.915 menggunakan masker) dan 800 ( 400 tanpa masker dan 400 menggunakan masker)

### 3.5.1 Skenario Supply Training Test

Skenario sederhana untuk menguji model dari *Convolutional Neural Network* dengan membagi dataset ke dalam dua bagian yaitu data latih dan data uji. Tujuan ini untuk menguji klasifikasi dalam keadaan dataset yang berbeda dan seberapa besar hasil akurasi yang dihasilkan. Dalam penelitian ini scenario yang dijalankan meliputi mempartisi dataset menjadi 90:10, 80:20, dan 70:30.

### 3.5.2 Skenario K-Fold Cross Validation

Skenario pengujian ini dilakukan seleksi k-fold, dalam pengujian ini akan dilakukan percobaan menggunakan tiga jenis k-fold yaitu 3-fold, 5-fold, dan 10-fold, dari 3 percobaan tersebut akan dicari hasil akurasi yang optimal.

### 3.5.3 Skenario Fungsi Pelatihan CNN

Dalam Skenario ini fungsi pelatihan pada *Convolutional Neural Network* akan menggunakan percobaan 3 jenis fungsi aktivasi yaitu *Adaptive Moment Estimation*, *Gradient Descent*, dan *Root Mean* *Square Propagation* untuk membuat sebuah model CNN yang dibuat panjang iterasi atau epochs yang ditentukan seperti 15 iterasi atau lebih untuk mencapai hasil prediksi yang memuaskan. Setelah mendapatkan hasil yang memuaskan maka akan langsung menyimpan model lalu melakukan pengujian deteksi masker dengan data yang baru.

## 3.6 Jadwal Penelitian

Tabel 3.1 Rincia Jadwal Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kegiatan | Waktu | | | | | | | | | | | | |
| Mei | | | | Juni | | | | Juli | | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | Studi Pustaka |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Tahap Persiapan dan Pengajuan Judul |  |  |  |  |  |  |  |  |  |  |  |  |
|  | a. Penyusunan dan pengajuan judul |  |  |  |  |  |  |  |  |  |  |  |  |
|  | b. pengajuan proposal |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Persiapan dan Pre-Processing |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Pengujian Sistem |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Evaluasi Sistem |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Pembuatan Laporan |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

Agarap, A. F. (2018). *Deep Learning using Rectified Linear Units (ReLU)*. http://arxiv.org/abs/1803.08375

Basha, S. H. S., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2019). Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification. *Neurocomputing*, *378*, 112–119. https://doi.org/10.1016/j.neucom.2019.10.008

Gholamalinezhad, H., & Khosravi, H. (2020). *Pooling Methods in Deep Neural Networks, a Review*. https://arxiv.org/abs/2009.07485v1

Hao, X., Zhang, G., & Ma, S. (2016). Deep Learning. *International Journal of Semantic Computing*, *10*(03), 417–439. https://doi.org/10.1142/S1793351X16500045

Kingma, D. P., & Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1412.6980v9

Neapolitan, R. E., & Jiang, X. (2018). Neural Networks and Deep Learning. *Artificial Intelligence*, 389–411. https://doi.org/10.1201/B22400-15

Nurfita, R. D., & Ariyanto, G. (2018). Implementasi Deep Learning berbasis Tensorflow untuk Pengenalan Sidik Jari. *Emitor: Jurnal Teknik Elektro*, *18*(1), 22–27. https://doi.org/10.23917/EMITOR.V18I01.6236

*Pedoman Pencegahan dan Pengendalian CORONAVIRUS DISEASE (COVID-19) Revisi ke-5 - Protokol | Covid19.go.id*. (n.d.). Retrieved May 5, 2023, from https://covid19.go.id/p/protokol/pedoman-pencegahan-dan-pengendalian-coronavirus-disease-covid-19-revisi-ke-5

Putra, W. E., & Putra, W. S. E. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, *5*(1). https://doi.org/10.12962/j23373539.v5i1.15696

Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*.

Yun, H. (2021). Prediction model of algal blooms using logistic regression and confusion matrix. *International Journal of Electrical and Computer Engineering (IJECE)*, *11*(3), 2407. https://doi.org/10.11591/ijece.v11i3.pp2407-2413

Zufar, M., Setiyono, B., & Matematika, J. (2016). Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-time. *Jurnal Sains Dan Seni ITS*, *5*(2), 128862. https://www.neliti.com/id/publications/128862/