

**PENDETEKSIAN *FAKE* MASKER MENGGUNAKAN
METODE CONVOLUTIONAL NEURAL NETWORK (CNN)
DENGAN ARSITEKTUR XCEPTION**

SKRIPSI

Diajukan Oleh:

MUHAMMAD RIDHA

NIM. 180705038

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY**

BANDA ACEH

2022

**PENDETEKSIAN *FAKE* MASKER MENGGUNAKAN
METODE CONVOLUTIONAL NEURAL NETWORK (CNN)
DENGAN ARSITEKTUR XCEPTION**

SKRIPSI

Diajukan kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri Ar-Raniry Banda Aceh
Sebagai Beban Studi Memperoleh Gelar Sarjana Dalam Ilmu Teknologi Informasi

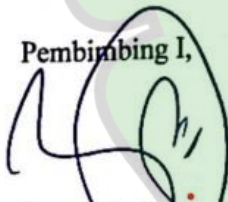
Oleh:

**MUHAMMAD RIDHA
NIM. 180705038**

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**

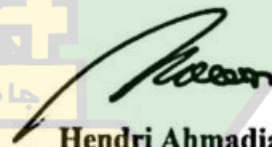
Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I,



**Bustami, M.Sc
NIDN: 2008048601**

Pembimbing II,



**Hendri Ahmadian, S.Si., M.IM
NIDN: 2004018303**

Mengetahui

Ketua Program Studi Teknologi Informasi



**Ima Dwitawati, MBA
NIDN: 0113108204**

**PENDETEKSIAN *FAKE* MASKER MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN
ARSITEKTUR XCEPTION**

SKRIPSI

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir/Skripsi Fakultas Sains
dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus Serta Diterima
Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Ilmu/Prodi Teknologi Informasi

Pada Hari/Tanggal: Sabtu, 24 Desember 2022
30 Jumadil Awal 1444 H
di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Skripsi

Ketua,



Bustami, M.Sc
NIDN: 2008048601

Sekretaris,



Hendri Ahmadian, S.Si., M.IM
NIDN: 2004018303

Penguji I,



Khairan AR, M.KOM
NIDN: 2004078602

Penguji II,



Mufkan Fadhli, S.T., M.T
NIDN: 132811881

Mengetahui

Dean Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh



Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP: 196210021988111001

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan dibawah ini :

Nama : Muhammad Ridha

Nim : 180705038

Program Studi : Teknologi Informasi

Fakultas : Sains dan Teknologi

Judul : Pendeteksian *Fake* Masker Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur Xception

Dengan ini menyatakan bahwa dalam penulisan tugas akhir/skripsi ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggung jawabkan.
2. Tidak melakukan plagiasi terhadap naskah karya orang lain.
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya.
4. Tidak memanipulasi dan memalsukan data.
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

da Aceh
g Menyatakan



METERAI
TEMPEL

794AKX226715804

(Muhammad Ridha)

ABSTRAK

Nama : Muhammad Ridha
NIM : 180705038
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi (FST)
Judul Skripsi : Pendeteksian *fake* masker menggunakan metode Convolutional Neural Network (CNN) dengan arsitektur Xception
Tanggal Sidang : 24 Desember 2022
Tebal Skripsi : 62 halaman
Pembimbing I : Bustami, M.Sc
Pembimbing II : Hendri Ahmadian, S.Si., M.IM
Kata Kunci : Masker, CNN dan arsitektur Xception

Convolutional neural network (CNN) merupakan metode yang terdapat pada *deep learning* yang diklaim sebagai metode terbaik atas permasalahan yang berkaitan dengan *image classification*. Perkembangan metode CNN saat ini telah menghasilkan banyak model atau arsitektur CNN terlatih yang dapat digunakan pada berbagai macam data, salah satunya citra masker wajah untuk pendeteksian secara otomatis. Penelitian ini menggunakan model arsitektur Xception. penelitian ini berfokus pada pendeteksian 3 objek yaitu wajah yang menggunakan masker, tidak bermasker dan *fake* masker. Berdasarkan pembahasan dan hasil analisis yang telah dilakukan dalam penelitian ini, maka diperoleh *accuracy* 99.53% dengan *training* 10 *epoch* dan *accuracy* 99.30% dengan *training* 5 *epoch*. Dari penelitian ini disimpulkan bahwa model arsitektur Xception memiliki tingkat keakuratan yang sangat tinggi dan semakin banyak *epoch* yang dilakukan maka *accuracy* yang didapatkan akan semakin tinggi.

Kata Kunci: Masker, CNN dan Arsitektur Xception.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan penulisan skripsi ini dengan baik. Shalawat dan salam tidak lupa kita sanjung sajikan kepada baginda Nabi besar Muhammad SAW yang telah membawa seluruh umatnya untuk menjadi generasi yang berilmu pengetahuan.

Dengan izin Allah SWT, saya dapat menyelesaikan skripsi yang berjudul **“Pendeteksian Fake Masker Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur Xception”** Dengan harapan penulis bahwa skripsi ini dapat memberikan manfaat bagi pihak yang membutuhkan, menambahkan wawasan serta ilmu pengetahuan.


Penulis menyadari skripsi ini tidak dapat diselesaikan dengan baik tanpa bimbingan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada pihak-pihak yang terlibat secara langsung maupun tidak langsung dalam mendukung kelancaran penulisan skripsi ini baik berupa dukungan, doa maupun bimbingan yang telah diberikan. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

1. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.
2. Ima Dwitawati, M.B.A dan Khairan AR, M.Kom selaku Ketua dan Sekretaris Program Studi Teknologi Informasi Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.
3. Bustami, M.Sc dan Hendri Ahmadian, S.Si., M.IM selaku pembimbing I dan II yang telah mencurahkan waktu, pikiran dan tenaga dalam membimbing penulis demi kesempurnaan skripsi ini. Terima kasih banyak penulis ucapkan, semoga Bapak dan Ibu selalu mendapat rahmat dan perlindungan Allah SWT.
4. Hendri Ahmadian, S.Si., M.IM selaku Penasehat Akademik (PA) penulis selama menempuh pendidikan di Program Studi Teknologi Informasi. Terima kasih banyak telah memberi nasehat dan saran selama ini kepada penulis.

5. Seluruh dosen yang mengajar pada Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan yang sangat berguna bagi penulis selama proses belajar mengajar.
6. Cut Ida Rahmadiana S,Si. Selaku staf prodi Teknologi Informasi yang telah membantu penulis dalam hal administrasi selama menempuh pendidikan di Program Studi Teknologi Informasi.
7. Orang tua terhebat yang penulis cintai, Bapak Rusli, S.Pd dan Ibu Maryani yang telah mendo'akan serta memberikan semangat, kasih sayang yang tiada henti kepada penulis serta seluruh keluarga besar yang telah memberikan nasihat, semangat dan motivasi sehingga penulis menyelesaikan skripsi ini.
8. Taufiqqumullah, Hidayatullah, Zamzami dan Al Asral. Selaku sahabat yang senantiasa mendukung dan memberikan semangat kepada penulis.
9. Sahabat-sahabat Teknologi Informasi yang telah berjuang bersama, berbagi semangat dan suka duka dalam penyelesaian skripsi ini. Terima kasih banyak penulis ucapkan kepada Nisa Afdhila, Muhammad Firdaus, Siska Fadhila, Darul Fata, Aulia Sabri. Penulis sangat berterima kasih atas motivasi dan semangat kalian semua.

Banda Aceh, 20 November 2022

Penulis


Muhammad Ridha

AR - RANIRY

DAFTAR GAMBAR

Gambar 1 Struktur <i>Neuron</i>	8
Gambar 2 <i>Perceptron</i>	9
Gambar 3 <i>Multi Layer Perceptron</i>	10
Gambar 4 <i>Backpropagation</i>	10
Gambar 5 Proses <i>Convolutional Layer</i>	12
Gambar 6 Grafik fungsi aktivasi relu.....	12
Gambar 7 Perbedaan <i>max polling</i> dan <i>average polling</i>	13
Gambar 8 Proses <i>reshape</i> fitur pada <i>flatten</i>	14
Gambar 9 <i>Neural Network</i> Sebelum dan Sesudah <i>Dropout</i>	14
Gambar 10 Arsitektur <i>xception</i>	16
Gambar 11 Proses <i>separable convolution</i>	17
Gambar 12 Tabel perhitungan <i>confusion matrix</i>	18
Gambar 13 Tahapan Penelitian.....	21
Gambar 14 Alur Penelitian.....	23
Gambar 15 Contoh <i>dataset</i> wajah yang memakai masker.....	23
Gambar 16 Contoh <i>dataset</i> wajah yang tidak memakai masker.....	23
Gambar 17 Contoh <i>dataset</i> wajah yang memakai masker palsu.....	23
Gambar 18 Contoh <i>Rescale</i>	24
Gambar 19 Contoh <i>resize</i>	25
Gambar 20 <i>Prepare dataset</i> dan <i>data preprocessing</i>	25
Gambar 21 Model <i>Xception</i> , <i>Training</i> dan <i>Testing</i>	27
Gambar 22 <i>Confusion Matrix</i>	28
Gambar 23 <i>Training</i> dan <i>validation accuracy 10 epoch</i>	31
Gambar 24 <i>Training</i> dan <i>validation loss 10 epoch</i>	32
Gambar 25 <i>Training</i> dan <i>validation accuracy 5 epoch</i>	33
Gambar 26 <i>Training</i> dan <i>validation loss 5 epoch</i>	33
Gambar 27 Hasil <i>confusion matrix 10 epoch</i>	34
Gambar 28 Hasil <i>confusion matrix 5 epoch</i>	35

DAFTAR TABEL

Tabel 1 Jumlah <i>Dataset</i>	24
Tabel 2 Hasil <i>Training 10 epoch</i>	31
Tabel 3 Hasil <i>training 5 epoch</i>	32
Tabel 4 Hasil <i>testing</i> perbandingan antara <i>10 epoch</i> dan <i>5 epoch</i>	34
Tabel 5 <i>Recall, Precision, F1 Score, dan Accuracy</i>	36
Tabel 7 <i>Sample Pengujian</i>	37



DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN SKRIPSI	iii
ABSTRAK	iv
KATA PENGANTAR	iv
DAFTAR GAMBAR	ivii
DAFTAR TABEL.....	iviii
BAB I PENDAHULUAN	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah	3
I.3 Tujuan Penelitian.....	3
I.4 Batasan Masalah.....	3
I.5 Manfaat Penelitian.....	4
BAB II LANDASAN TEORI.....	5
II.1 Penelitian Terdahulu	5
II.2 Masker.....	6
II.3 <i>Computer Vision</i>	6
II.4 <i>Artificial Intelegence</i>	7
II.5 <i>Machine Learning</i>	7
II.6 <i>Deep Learning</i>	7
II.7 <i>Artificial neural network (ANN)</i>	8
II.8 <i>Neuron</i>	8
II.9 <i>Perceptron</i>	9
II.10 <i>Multi Layer Perceptron (MLP)</i>	9
II.11 <i>Backpropagation</i>	10
II.12 <i>Convolutional Neural Network</i>	11
II.12.1 <i>Feature Learning</i>	11
II.12.2 <i>Classification</i>	13
II.13 CNN Arsitektur.....	15
II.14 Xception.....	15
II.15 Bahasa Pemrograman Python	17

II.16 Tensorflow	17
II.17 Keras	18
II.18 Metode Evaluasi	18
II.18.1 <i>Confusion Matrix</i>	18
II.18.2 <i>Accuracy</i>	19
II.18.3 <i>Precision</i>	19
II.18.4 <i>Recall</i>	19
II.18.5 <i>F1 Score</i>	19
BAB III METODE PENELITIAN.....	21
III.1 Tahapan Penelitian.....	21
III.1.1 Perumusan Masalah	21
III.1.2 Studi Literatur dan Pengumpulan data.....	21
III.1.3 Implementasi	22
III.1.4 Evaluasi	22
III.2 Alur Penelitian	22
III.2.1 <i>Dataset</i>	23
III.2.2 <i>Data Preprocessing</i>	24
III.2.3 <i>Training</i>	26
III.2.4 <i>Testing</i>	26
III.2.5 Model	26
III.2.6 Evaluasi Akhir.....	28
III.3 Bahan dan Alat	29
III.4 Tempat dan Waktu.....	29
BAB IV HASIL PENELITIAN	30
IV.1 Hasil Uji Coba	30
IV.1.1 Hasil Proses <i>Training</i>	30
IV.1.2 Hasil Proses <i>Testing</i>	34
IV.1.3 <i>Confusion Matrix</i>	34
IV.1.4 <i>Recall, Precision, f1 Score</i> dan <i>Accuracy</i>	36
IV.1.5 <i>Sample</i> Pengujian.....	37
BAB V KESIMPULAN DAN SARAN.....	41
V.1 Kesimpulan	41

V.2 Saran	41
DAFTAR PUSTAKA	41
LAMPIRAN	41



BAB I

PENDAHULUAN

I.1 Latar Belakang

Masker merupakan alat perlindungan pernafasan yang digunakan untuk melindungi dari menghirup zat-zat berbahaya yang ada di udara, masker digunakan untuk melindungi secara memadai pemakainya karena dengan memakai masker merupakan salah satu cara dalam mengurangi resiko terpaparnya virus (Han & Goleman, Daniel; Boyatzis, Richard; McKee, 2019).

Mengidentifikasi pemakaian masker pada wajah seseorang ialah sesuatu perihal yang gampang dilakukan oleh manusia. Seseorang akan mudah paham, menghafal serta membedakan mana pemakaian masker yang benar, tidak bermasker maupun masker palsu pada seseorang. Namun tidak sama dengan komputer, buat mengidentifikasi kelayakan pemakaian masker pada wajah seorang dibutuhkan perlakuan spesial sehingga bila diberi input suatu gambar, komputer bisa mengetahui apakah dalam gambar tersebut pemakaian masker pada seseorang telah layak ataupun belum. Pengenalan pemakaian masker pada wajah seseorang memiliki tujuan utama yaitu untuk mengenali kelayakan pemakaian masker pada wajah seseorang melalui sebuah citra yang menggunakan *database* masker yang telah disimpan. Maka dari itu sudah banyak metode ataupun algoritma dalam *artificial intelligence* untuk melakukan pendeteksian secara otomatis dimana nanti metode itu akan belajar sendiri seperti metode *Convolutional Neural Network* (CNN).

CNN adalah salah satu jenis *neural network* yang biasa digunakan pada data gambar. CNN bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah image. CNN lebih diandalkan untuk berbagai proses terkait *visual*. CNN ini dipakai untuk mengidentifikasi pola-pola yang terdapat dalam sebuah gambar dengan mengandalkan prinsip-prinsip aljabar linier seperti perkalian matriks. Ada banyak sekali arsitektur yang ada di dalam CNN ini salah satunya adalah Xception.

Xception adalah arsitektur yang memiliki 36 lapisan konvolusi yang membentuk basis ekstraksi ciri dari jaringan. Xception terinspirasi dari arsitektur

Inception yang dimodifikasi dengan mengganti modul pada Inception dengan *depthwise separable convolution*. *Depthwise Separable Convolution* merupakan *framework deep learning* pada TensorFlow dan Keras terdiri dari *depthwise convolution* yaitu konvolusi spasial yang dilakukan secara independent pada masing-masing proses input yang diikuti oleh *pointwise convolution* (Dewi & Ismawan, 2021). sekarang sudah banyak terdapat penelitian mengenai pendeteksian masker secara otomatis.

Terdapat beberapa penelitian terdahulu yang sudah menganalisa pendeteksian masker secara otomatis seperti Ahmad Thorik dkk, dimana dalam penelitian ini menganalisa pendeteksian masker dan tidak bermasker, dataset yang digunakan dalam penelitian ini adalah dataset yang tidak berbentuk *realtime* sehingga jika lokasi pengetesan dilakukan di lokasi yang tidak cukup cahaya maka sistem akan kesulitan untuk mendeteksinya, pada penelitian ini tingkat keakuratan mencapai 92.5%(Thariq & Bakti, 2021). selain itu terdapat penelitian yang dilakukan oleh Muhammad Aminullah, sama dengan penelitian sebelumnya dimana dalam penelitian ini menganalisa pendeteksian masker dan tidak bermasker. namun pada penelitian menggunakan metode *Convolutional Neural Network* (CNN) dan arsitektur MobilnetV2, pada penelitian ini tingkat keakuratan mencapai 90%(Aminullah, 2020). selanjutnya penelitian pendeteksian masker dan tidak bermasker juga dilakukan oleh Darmatasia. Pada penelitian ini menggunakan menggunakan metode CNN dengan arsitektur Xception sebagai model yang digunakan, pada penelitian ini tingkat keakuratan mencapai 94%(Darmasita, 2020).

Pada beberapa penelitian yang telah disebutkan diatas, penelitian tersebut hanya berpaku pada 2 objek yaitu mendeteksi masker dan tidak bermasker. Namun dari penelitian tersebut belum ada penelitian yang mendeteksi masker palsu (*fake mask*) seperti penggunaan tangan, hijab maupun kertas sebagai pengecoh untuk pendeteksian masker. Hal ini menunjukkan penelitian yang telah ada ataupun dibangun belum memenuhi standar untuk protokol kesehatan karena bisa saja mendeteksi tangan, hijab maupun kertas sebagai masker sebagai *output* analisa ataupun sistem.

Berdasarkan paparan diatas, maka dalam penelitian ini akan menganalisa yang bukan hanya bisa mendeteksi masker dan tidak, namun juga bisa mendeteksi masker palsu (*fake mask*). Dalam penelitian ini akan digunakan metode CNN karena CNN merupakan metode yang terdapat pada *deep learning* yang sudah terbukti sebagai metode terbaik atas permasalahan yang berkaitan dengan klasifikasi gambar(Nasichuddin et al., 2018), selain itu pada penelitian ini akan digunakan arsitektur Xception karena merupakan salah satu arsitektur dengan tingkat keakuratan yang tinggi dalam mengklasifikasi citra(Ghani et al., 2021).

I.2 Rumusan Masalah

Berdasarkan indentifikasi masalah yang telah disebutkan sebelumnya, rumusan masalah dalam penelitian ini adalah :

1. Bagaimana mengimplementasi metode CNN dengan arsitektur Xception dalam pendeteksian pada citra wajah yang menggunakan *fake mask*?
2. Bagaimana tingkat akurasi yang didapatkan dari hasil implementasi tersebut?

I.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Untuk mengetahui metode CNN dengan arsitektur Xception dalam pendeteksian pada citra wajah yang menggunakan *fake mask*?
2. Untuk mengetahui tingkat akurasi metode CNN dengan arsitektur Xception dalam pendeteksian *fake mask*.

I.4 Batasan Masalah

Adapun Batasan Masalah dalam penelitian ini adalah :

1. Data yang digunakan adalah citra wajah yang memakai format jpg.
2. Klasifikasi citra ini hanya mencakup 3 model yaitu menggunakan masker, tidak bermasker dan *fake mask*.
3. Pendeteksian *fake mask* hanya mencakup hal-hal yang menempel dibadan seperti tangan, baju, hijab dan kertas.

I.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Mempermudah mengklasifikasi mana yang menggunakan masker, tidak bermasker dan *fake mask* secara otomatis.
2. Mengetahui bagaimana pengimplementasian metode CNN dengan arsitektur Xception dalam mengklasifikasi citra yang memakai masker, tidak memakai masker dan *fake mask*.
3. Mengetahui tingkat akurasi keakuratan metode CNN dengan arsitektur Xception dalam mengklasifikasi citra yang memakai masker, tidak memakai masker dan *fake mask*.
4. Menjadi referensi bagi mahasiswa jika melakukan penelitian tentang metode CNN dan arsitektur Xception.



BAB II

LANDASAN TEORI

II.1 Penelitian Terdahulu

Terdapat beberapa penelitian terdahulu yang sudah menganalisa ataupun membuat sistem deteksi masker secara otomatis. Salah satunya adalah penelitian yang dilakukan oleh Nyoman Purnama dan Putu Kusuma Negara, pada penelitian ini membangun sistem yang bisa mendeteksi wajah yang memakai masker dan tidak. Metode yang digunakan dalam penelitian ini adalah *sequential exploratory*, penelitian ini dilakukan melalui dua tahapan. Tahapan pertama menggunakan metode kualitatif untuk memperoleh rancangan sistem dan tahapan yang kedua menggunakan metode kuantitatif untuk menguji efektifitas sistem. Proses *Training* dataset dilakukan dengan aplikasi berbasis Phyton. Sedangkan pada proses pengujian dilakukan dengan aplikasi *mobile* berbasis android dengan menggunakan Android Studio dan Framework Tensorflow. Arsitektur yang digunakan pada penelitian ini adalah MobileNet versi 2 dan menghasilkan tingkat akurasi sebesar 90% (Nyoman & Putu Kusuma Negara, 2021).

Selain itu penelitian mengenai pendeteksian masker secara otomatis juga dilakukan oleh Restin Alfinda Zai dan Fajar Astuti, metode yang digunakan dalam penelitian ini adalah CNN dan Viola Jones. Penelitian ini menggunakan 1300 dataset yang di pisah menjadi 4 katagori yaitu masker medis, bukan medis, masker salah dan tidak bermasker. Pada penelitian ini menghasilkan tingkat akurasi sebesar 92.3% (Hermawati & Zai, 2021).

Selanjutnya penelitian tentang pendeteksian masker secara otomatis juga dilakukan oleh Mohammad Farid Naufal dan Selvia Ferdina Kusuma, dalam penelitian ini menggunakan arsitektur Xception. Dataset yang digunakan dalam penelitian ini terdiri dari dua katogori yaitu memakai masker dan tidak memakai masker. Pada penelitian ini menghasilkan tingkat akurasi sebesar 96% (Naufal & Kusuma, 2021).

Berdasarkan penelitian terdahulu yang telah disebutkan, diketahui bahwa belum adanya penelitian yang menganalisa maupun mebangun sistem untuk

mendeteksi masker palsu (*fake mask*). Oleh karena itu, pada penelitian ini akan menganalisa yang tidak hanya mendeteksi masker dan tidak bermasker namun juga bisa menganalisa untuk mendeteksi masker palsu (*fake mask*). Metode yang digunakan dalam penelitian ini adalah CNN dengan arsitektur Xception.

II.2 Masker

Masker merupakan alat perlindungan pernafasan yang digunakan sebagai metode untuk melindungi individu dari menghirup zat-zat berbahaya atau kontaminan yang berada di udara (Raden Muhammad Ali et al., 2020). Masker sendiri mempunyai banyak tipe, salah satunya adalah *Air Purifying Respirators* (APR), masker ini menggunakan filter atau *catridge* yang dapat mencegah zat-zat berbahaya yang berada di udara. Manfaat memakai masker diantaranya adalah mengurangi resiko penularan virus, mencegah penyebaran tanpa gejala dan melindungi diri dan orang lain.

Pemakaian masker wajah adalah salah satu cara yang direkomendasikan untuk mengurangi resiko penularan virus. Oleh karena itu pemakaian masker menjadi salah satu cara pencegahan yang dianjurkan berbagai lembaga kesehatan seperti *World Health Organization* (WHO) (Kusuma et al., 2021). Pada Penelitian ini pengertian masker dibagi menjadi 3 yaitu :

1. Menggunakan masker yaitu area wajah yang menggunakan masker.
2. Tidak menggunakan masker yaitu area wajah yang tidak menggunakan masker.
3. Masker palsu (*fake mask*) yaitu area wajah yang ditutupi menggunakan tangan, baju, kertas dan hijab yang bertujuan untuk menyerupai masker.

II.3 Computer Vision

Computer vision merupakan ilmu yang membuat komputer dapat mengenali suatu keadaan dan objek fisik nyata dari sebuah gambar atau citra untuk dapat membuat suatu keputus. *Computer vision* membuat komputer dapat menerima informasi secara visual seperti manusia. Sehingga, menjadikan komputer “*acts like human sight*”. Beberapa Kemampuan itu adalah: *object detection*, *image classification* dan *image segmentation* (Miceli et al., 2021).

II.4 Artificial Intelligence

Artificial Intelligence adalah Suatu ilmu dan teknik yang membahas tentang pembuatan mesin cerdas dimana agar sebuah komputer mampu melakukan pekerjaan yang masih dilakukan dengan baik oleh manusia pada saat ini. Layaknya manusia, sebuah komputer menjadi cerdas dengan belajar dari pengalaman dengan mempelajari data (Sutanto, 2021).

II.5 Machine Learning

Machine Learning merupakan cabang *artificial intelligence (AI)* yang berfokus di pengolahan data, dan algoritma ini bekerja dengan cara meniru manusia belajar dan mempertinggi akurasi perlahan-lahan. *Machine learning* termasuk komponen penting di ilmu yang berfokus pada pengembangan data. Melalui penggunaan statistic. Algoritma *machine learning* dilatih buat membentuk penjabaran maupun prediksi data dalam pengembangannya. Pengembangan data yang diolah oleh *machine learning* meliputi wawasan utama dari *artificial intelligence* serta pengambilan keputusan dalam sebuah sistem.

Secara menyeluruh bisa dikatakan *machine learning* adalah metode analisis data yang dilakukan menggunakan otomatisasi pembuatan model analitis. Sebagai salah satu cabang dari *artificial intelligence*, *machine learning* didasari pernyataan bahwa sistem belajar melalui data, mengidentifikasi pola serta mengambil keputusan dengan sedikit campur tangan manusia dalam proses tersebut (Chang et al., 2018).

II.6 Deep Learning

Deep learning atau lebih dikenal dengan sebutan *deep structured learning* dan *hierarchical learning* adalah cabang dari *machine learning* yang menggunakan *artificial neural networks (ANN)*. Dasarnya, *deep learning* merupakan teknik pembelajaran yang dilakukan mesin dengan meniru cara kerja saraf otak manusia.

Seperti halnya saraf otak manusia kompleks, *deep learning* bisa mendeteksi objek, mengenali suara, serta lainnya. *deep learning* bisa belajar dengan mandiri dan beradaptasi dengan sejumlah besar data, sehingga dapat menyelesaikan permasalahan yang kompleks sekalipun. *Deep learning* di desain buat menganalisa data menggunakan pola logika yang sama seperti manusia membuat keputusan.

Deep learning mempunyai kemampuan menangkap fitur atau karakteristik relevan dari suatu data buat memecahkan sebuah masalah.

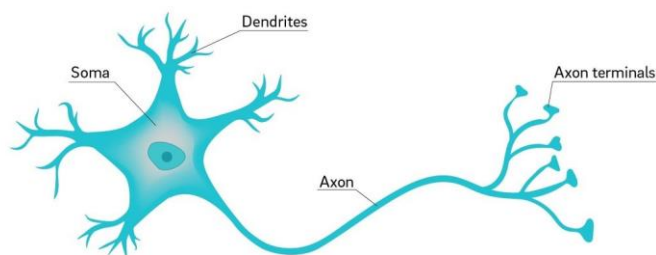
Seperti *neural network* pada dasarnya, *layer* di *deep learning* dibagi menjadi tiga *layer* yaitu *input layer*, *hidden layer*, dan *output layer*. Semua *layer* ini mempunyai peran masing-masing. *Input layer* yang berfungsi buat menampung data mentah semisal bunyi, teks, maupun gambar. *Hidden layer* bertugas buat memeriksa serta mengklasifikasi data sesuai referensi yang telah dimiliki. *Output layer* berfungsi buat menunjukkan hasil penelitian serta klasifikasi terstruktur dari suatu data. *Deep learning* memiliki beberapa algoritma dan yang paling sering digunakan salah satunya adalah *convolutional Neural Network*.

II.7 Artificial neural network (ANN)

Artificial neural network (ANN) atau jaringan saraf tiruan merupakan sebuah algoritma dalam *deep learning* yang terinspirasi dari sistem jaringan *neuron* atau saraf pada otak manusia. *Neuron* adalah dasar dari sebuah jaringan saraf tiruan yang berfungsi sebagai unit pengolah informasi. Saat ini ANN telah digunakan dalam menangani permasalahan seperti klasifikasi gambar dan pengenalan bahasa (TRIATMAJA, 2018).

II.8 Neuron

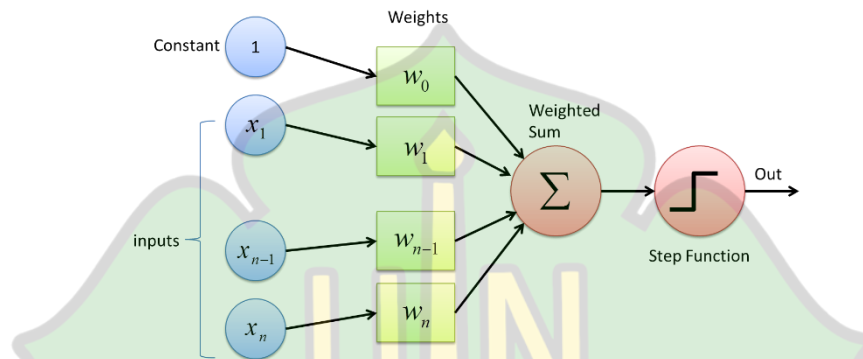
Neuron atau saraf adalah pembawa informasi. *Neuron* mengirimkan informasi keseluruhan sistem saraf menggunakan *impuls* listrik dan sinyal kimiawi. *Neuron* terdiri atas 3 bagian, yaitu badan sel yang di dalamnya terdapat *nukleus*, *dendrit*, dan *akson*. *Nukleus* berfungsi mengontrol seluruh aktivitas sel dan berisi materi genetik. *Dendrit* adalah cabang pendek yang bertugas menerima pesan untuk sel. Sedangkan *akson* adalah cabang panjang yang berfungsi mengirimkan pesan dari sel (Kholik, 2021).



Gambar 1 Struktur *Neuron* (Kholik, 2021).

II.9 Perceptron

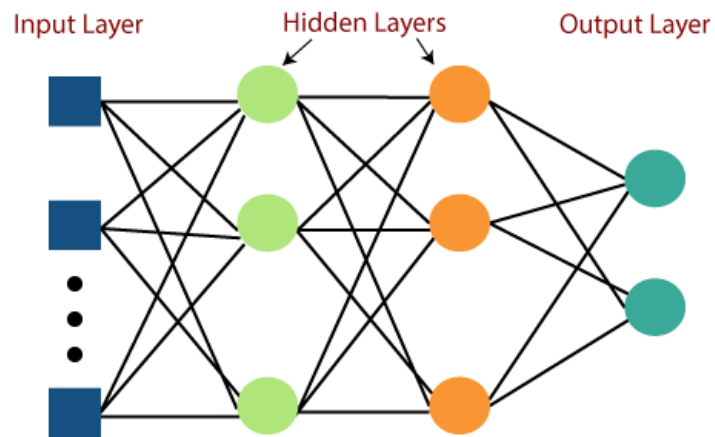
Komponen dasar pembangun jaringan saraf tiruan adalah *perceptron*. Ilmuwan yang pertama kali menemukan *perceptron* adalah Frank Rosenblatt dari *Cornell Aeronautical Library* pada tahun 1957 (Rosenblatt, 1958). *Perceptron* terinspirasi dari *neuron* pada jaringan saraf manusia. *Perceptron* dan *neuron* merupakan hal yang sama pada jaringan saraf tiruan. Suatu perceptron terdiri dari 5 bagian yaitu Input (x_i) dan Output (y), Penjumlahan (Σ), Bobot atau weights (W_i) dan bias (W_0), dan Fungsi aktivasi (f) (Desai & Shah, 2021).



Gambar 2 *Perceptron* (Desai & Shah, 2021).

II.10 Multi Layer Perceptron (MLP)

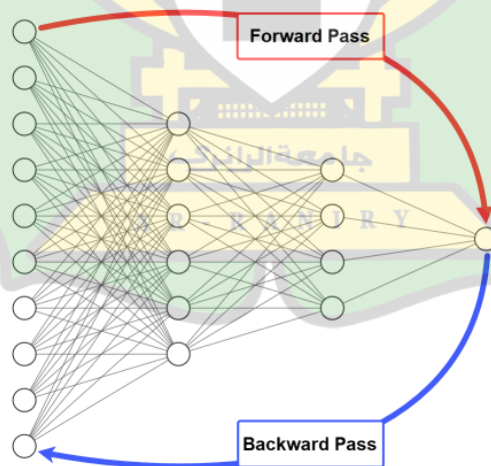
Multi layer perceptron (MLP) merupakan jaringan saraf yang terdiri dari *layer input*, *hidden layer*, *layer output*. *Layer input* merupakan layer yang hanya menampung angka-angka. Sedangkan *hidden layer* dan *layer output* adalah layer yang mempunyai beberapa *perceptron*. Pada jaringan saraf tiruan, hidden layer bisa ada atau tidak sama sekali. Namun untuk *layer input* dan *layer output* harus selalu ada. Pada sebuah jaringan saraf tiruan tidak ada aturan baku dalam menetapkan jumlah perceptron dan *hidden layer*. Semakin banyak jumlah *hidden layer* yang dipakai, maka semakin kompleks masalah yang dapat diselesaikan, namun juga semakin lama jaringan saraf tersebut untuk belajar (Yu et al., 2021).



Gambar 3 *Multi Layer Perceptron* (Yu et al., 2021).

II.11 *Backpropagation*

Propagasi balik atau *backpropagation* dipublikasikan melalui sebuah penelitian yang dilakukan oleh Geoffrey Hinton, David Rumelhart, dan Ronald Williams pada tahun 1985 (Rumelhart & McClelland, 1985). Propagasi balik merupakan suatu metode yang digunakan dalam pelatihan MLP. Metode *backpropagation* bekerja dengan melakukan *forward propagation* pada awal proses dan kemudian menyimpan *error* yang dihasilkan. Lalu *error* tersebut digunakan untuk mengubah nilai bobot setiap *perceptron* dalam arah mundur (*backward*) (Firzatullah, 2021).



Gambar 4 *Backpropagation* (Firzatullah, 2021).

II.12 Convolutional Neural Network

Convolutional neural network (CNN) merupakan sebuah algoritma *deep learning* yang di desain buat pengolahan data 2 dimensi. *Convolutional neural network* umumnya di pergunakan buat menyelidiki serta mendeteksi *feature* pada suatu gambar. Bila *neural network* pada dasarnya memakai data *array* satu dimensi (1D) menjadi input, di algoritma *Convolutional neural network* data yang dipakai menjadi input adalah data 2 dimensi (2D). Sama seperti *neural network* pada dasarnya, *convolutional neural network* terdiri dari banyak *neuron* yang mempunyai *weight*, *bias*, dan fungsi aktivasi.

Tahapan proses di *Convolutional neural network* mempunyai 2 bagian utama, yaitu *feature learning* dan *classification*. Di bawah ini penjelasan dari kedua bagian tersebut.

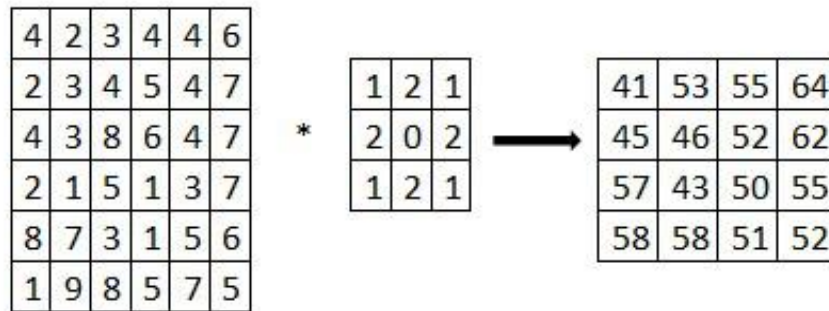
II.12.1 Feature Learning

Feature learning merupakan proses *encoding* dari suatu gambar menjadi *feature* yang berupa nilai-nilai yang merepresentasikan gambar tersebut. *Future learning* terdiri dari beberapa *layer* yang saling bekerjasama buat mengambil ciri suatu gambar (Developers Breach, 2020). berikut penjelasan setiap *layer* pada *feature learning*.

1. Convolution Layer

Semua data yang melewati *layer* ini akan menemui proses konvolusi. Tahap ini mengkonversi setiap filter ke semua bagian data awal serta menghasikan sebuah *feature map* 2D. *Layer* ini mempunyai filter mempunyai panjang, tinggi serta tebal seperti dengan data awal. Setiap filter akan mengalami pergeseran dan operasi “dot” antara data awal dan nilai filter (Ajit et al., 2020). Pada tahapan *convolution layer* setiap citra *input* dengan matrix 6x6 yang dimasukkan akan dikali dengan setiap kernel 3x3 untuk proses konvolusi, setiap kernel akan dikali dengan citra *input* yang dibelah menjadi matrix 3x3 juga, hasil perkalian antara citra *input* yang sudah dibelah menjadi matrix 3x3 akan dengan setiap karnel lalu akan ditambahkan semuanya dan hasil dari penambahan tersebut akan menjadi *output* sehingga menempati di sebuah *cell*

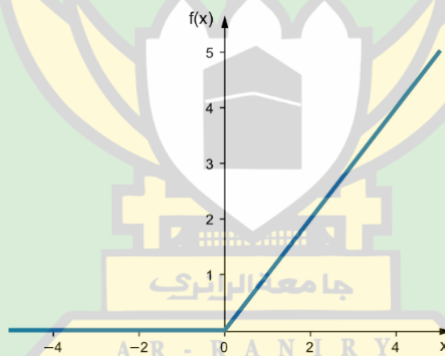
pada citra yang baru yang disebut citra hasil. Gambar 5 menunjukkan proses *convolutional layer*.



Gambar 5 Proses *Convolutional Layer*.

2. *Rectified Linear Unit (ReLU) activation*

Fungsi dari ReLU adalah mengaplikasikan fungsi aktivasi nilai *output* dari *convolution layer*. *Output* dari fungsi aktivasi ditulis sebagai 0 apabila inputnya negatif. Apabila inputnya positif, maka nilai *output* akan sama dengan nilai *input* fungsi aktivasi itu sendiri (Agarap, 2018). Gambar 6 merupakan grafik fungsi aktivasi di *ReLU layer*.

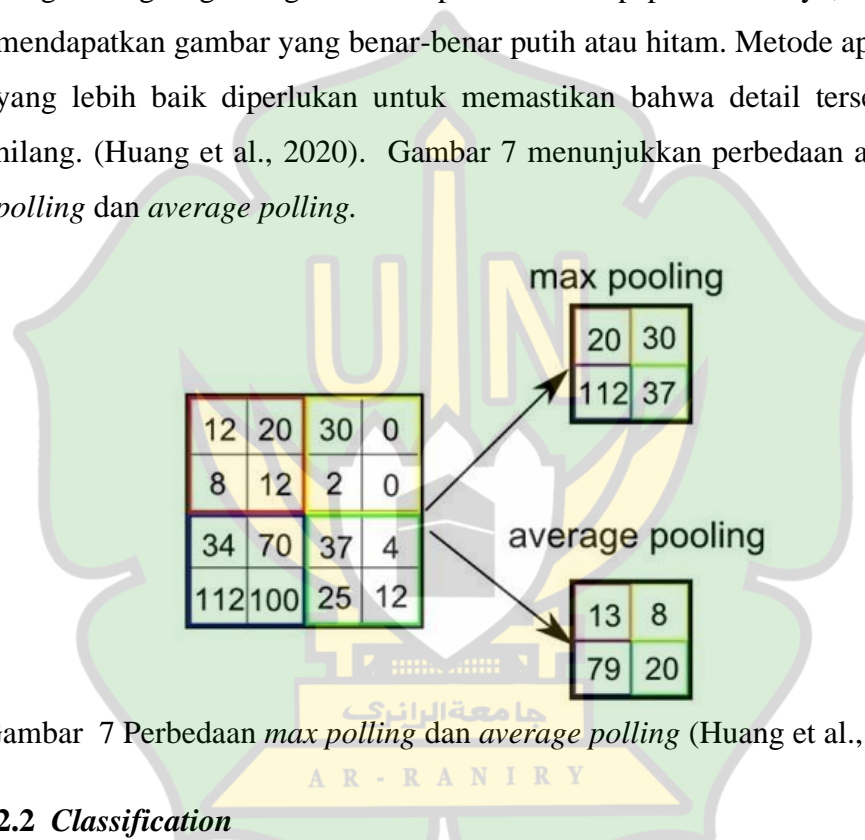


Gambar 6 Grafik fungsi aktivasi *relu* (Agarap, 2018).

3. *Pooling Layer*

Pooling layer adalah salah satu tahapan proses dari *convolutional neural network* yang bertujuan untuk mengurangi dimensi dalam *matrixs* yang sudah melalui tahapan konvolusi. Tujuan dari *layer* ini buat meningkatkan kecepatan proses komputasi. Karena sesudah melewati *layer* ini, parameter yang akan di proses semakin sedikit, sehingga kesempatan akan *overfitting* makin sedikit. *Pooling layer* mempunyai filter ukuran tertentu yang akan melakukan proses *sliding window* terhadap *matrixs* input.

Ada 2 jenis yang sering digunakan, yaitu *max pooling* dan *average pooling*. *Max pooling* mengambil nilai paling besar dari *matrixs* input. *Average pooling* mengambil nilai rata-rata dari *matrixs* input. Fungsi dari pooling ini adalah untuk mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi *down-sampling*. *Down-sampling* digunakan ketika gambar dibuat lebih kecil. Meskipun mengecilkan gambar tidak memerlukan pengisian ruang baru. Misalnya, pertimbangkan gambar yang terdiri dari piksel hitam dan putih bergantian. Jika mengecilkan gambar ini menjadi setengah ukurannya dengan langsung mengambil sampel nilai setiap piksel lainnya, Anda akan mendapatkan gambar yang benar-benar putih atau hitam. Metode aproksimasi yang lebih baik diperlukan untuk memastikan bahwa detail tersebut tidak hilang. (Huang et al., 2020). Gambar 7 menunjukkan perbedaan antara *max pooling* dan *average pooling*.



Gambar 7 Perbedaan *max pooling* dan *average pooling* (Huang et al., 2020).

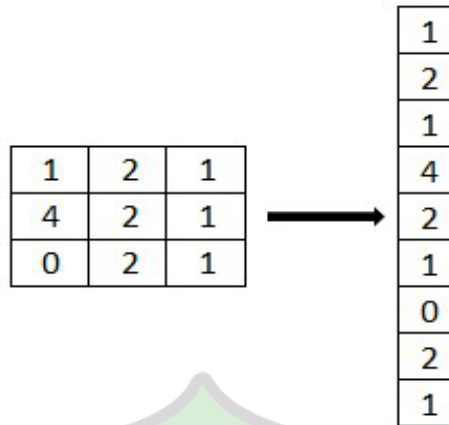
II.12.2 Classification

Tahap ini berfungsi buat klasifikasi terhadap setiap *neuron* yang sudah melalui ekstraksi di tahap sebelumnya. Tahap *classification* terdiri dari 4 layer yang berhubungan antara satu layer dengan lainnya (Budi et al., 2021). Berikut penjelasan setiap *layer* pada *classification*.

1. *Flatten*

Flatten berfungsi buat melahirkan kembali (*reshape*) *feature map* yang berasal *multidimensional array* menjadi sebuah vektor. Hal ini dilakukan supaya nilai tersebut bisa dipergunakan sebagai input di *fully connected*

layer(Sultana et al., 2022). Gambar 8 menunjukkan proses *reshape* pada *flatten*.



Gambar 8 Proses *reshape* fitur pada *flatten*.

2. Dropout Regularization

Dropout adalah teknik regularisasi pada *neural network*, di tahapan ini *neuron* biasanya di pilih secara acak dan tidak digunakan pada proses *training*. Caranya adalah menghilangkan beberapa *neuron*, artinya menghilangkan sementara dari jaringan yang ada. Tahap ini bertujuan meminimalisir terjadinya *overfitting*. Selain itu, teknik ini dapat meningkatkan kecepatan proses learning pada *convolutional neural network* secara menyeluruh (Kholik, 2021). Gambar 9 menunjukkan perbedaan *neural network* yang menggunakan *droupout* dan tidak menggunakan *droupout*.



Gambar 9 *Neural Network* Sebelum dan Sesudah *Dropout*

3. *Fully Connected Layer*

Fully connected layer terdiri dari *hidden layer*, *activation function*, *output layer*, dan *loss function*. *Layer* ini biasanya digunakan di *multilayer perceptron* yang bertujuan untuk melakukan transformasi di dimensi data supaya data-data tadi bisa diklasifikasikan secara linear. Input pada *layer* ini yaitu data yang berasal dari *feature learning*. Data yang diproses pada *layer* ini adalah data yang berbentuk *vektor*, yang sebelumnya sudah diproses oleh *flatten*.

4. *Softmax*

Softmax atau biasa disebut dengan *Softmax Classifier* biasanya digunakan untuk mengklasifikasi kelas yang lebih dari dua. Standar klasifikasi yang biasanya dilakukan adalah tugas untuk klasifikasi kelas *biner*. *Softmax* bisa saja menghitung peluang untuk semua label. Dari label yang ada akan diambil sebuah *vektor* nilai riil dan merubahnya menjadi *vektor* dengan nilai antara nol dan satu yang bila semua dijumlahkan akan bernilai satu (Ali & Mallaiah, 2021).

II.13 CNN Arsitektur

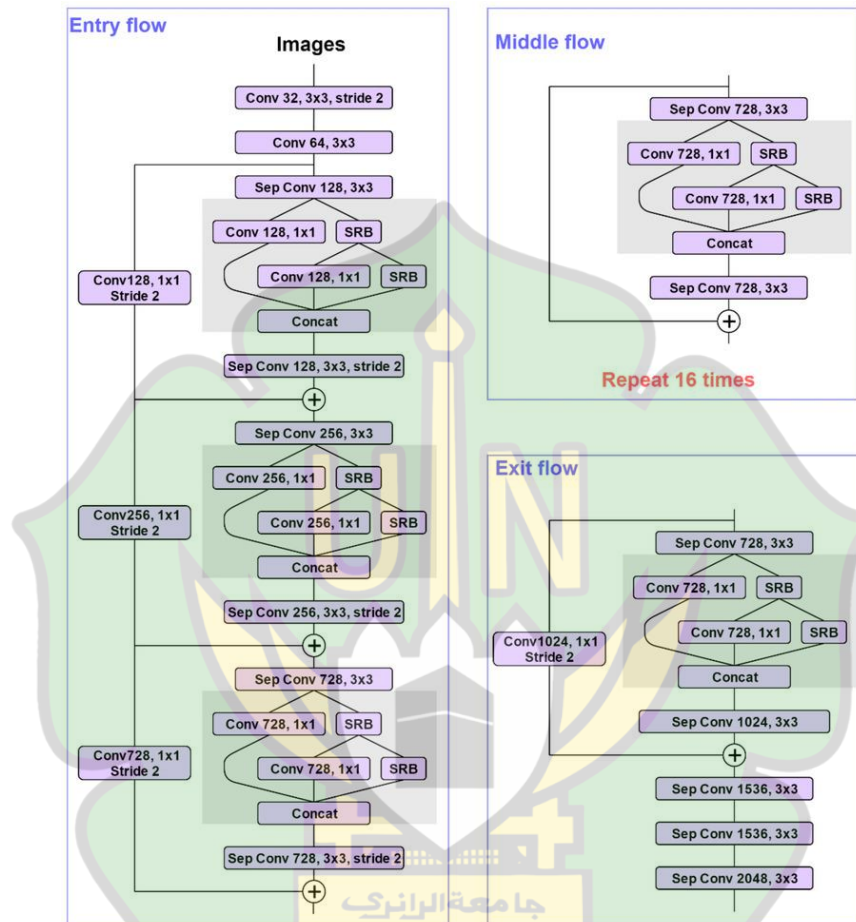
CNN adalah salah satu jenis *neural networks* yang ada pada *deep learning*, CNN biasanya mengolah data dalam bentuk citra. CNN bekerja menggunakan *kernel*. *Kernel* tersebut akan mengekstrak fitur dari *input* menggunakan operasi *konvolusi*. Kelebihan CNN adalah parameter *sharing* yang dapat membantu mengurangi jumlah parameter pada keseluruhan sistem dan membuat beban komputasinya berkurang. Selain itu juga CNN memiliki keunggulan *spatial features*. *Spatial features* mengacu pada susunan piksel dan hubungan antar piksel dalam sebuah citra. Hal ini memudahkan untuk melakukan identifikasi suatu objek, lokasi suatu objek, dan hubungannya dengan citra yang lain (Iii & Teori, n.d.).

Terdapat banyak arsitektur CNN yang bisa digunakan untuk melakukan klasifikasi. Seperti contoh VGG16, InceptionV3, ResNet50, dan Xception. Namun pada penelitian ini arsitektur yang dipilih adalah xception.

II.14 Xception

Arsitektur Xception memiliki 36 lapisan *konvolusi* yang membentuk basis ekstraksi ciri dari jaringan. Xception terinspirasi dari arsitektur Inception yang dimodifikasi dengan mengganti modul pada Inception dengan *depthwise separable*

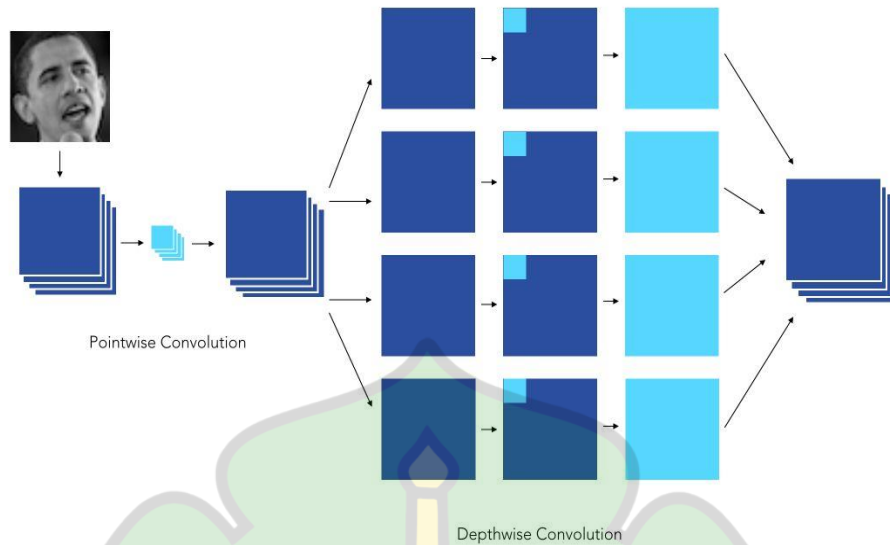
convolution. Depthwise Separable Convolution merupakan *framework deep learning* pada TensorFlow dan Keras terdiri dari *depthwise convolution* yaitu konvolusi spasial yang dilakukan secara independent pada masing-masing proses input yang diikuti oleh *pointwise convolution* (Dewi & Ismawan, 2021). Gambar 10 menunjukkan arsitektur Xception.



Gambar 10 Arsitektur xception (Dewi & Ismawan, 2021).

Data pertama-tama melewati *entry flow*, kemudian melalui *middle flow* dimana pada tahapan ini diulang sebanyak delapan kali, dan terakhir melalui *exit flow*. Semua lapisan *Convolution* dan *Separable Convolution* diikuti dengan *normalization batch* supaya mempercepat proses *training*. Di dalam *Separable Convolution* dibagi menjadi 2 langkah utama yaitu *Depthwise Convolution* dan *Pointwise Convolution*, Langkah pertama yaitu *Pointwise Convolution* memproses data dengan matrix 1x1 dan selanjutnya akan masuk ke Langkah *Depthwise*

Convolution yang memproses data dengan matrix 3x3(Shaheed et al., 2022). Gambar 11 menunjukkan langkah *Separable Convolution*.



Gambar 11 Proses *separable convolution* (Shaheed et al., 2022).

II.15 Bahasa Pemrograman Python

Python merupakan salah satu Bahasa yang *interpretatif* multiguna. tidak sama dengan bahasa lainnya yang susah buat dibaca serta dipahami, python lebih fokus di pembacaan *sintaxs* supaya mudah buat memahami. Hal ini menghasilkan Python menjadi mudah dipelajari. Python juga memiliki beragam *library* yang mendukung dalam pembuatan sebuah aplikasi termasuk kecerdasan buatan.

II.16 Tensorflow

Tensorflow adalah antarmuka buat mengekspresikan prosedur algoritma *Machine Learning* dan buat mengeksekusi sintaks dengan memakai data yang dimiliki oleh objek atau sasaran yang ada. Tensorflow dapat membedakan objek antara objek pertama dan kedua. Oleh karena itu dibutuhkan data buat mengenali sebuah gambar yang banyak serta kompleks. Sistem ini sangat sempurna, fleksibel serta dapat dipergunakan untuk mengutarakan beberapa algoritma, termasuk algoritma pelatihan dan inferensi model jaringan syaraf tiruan. Tensorflow mempunyai fitur buat menjalankan pelatihan model menggunakan CPU dan GPU yang memiliki pelatihan yang lebih cepat. TensorFlow ini dipergunakan menjadi *backend* dari Keras, yang merupakan belakang layer dari *software* atau website

yang menggunakan bahasa pemrograman Python serta dikombinasikan menggunakan *library* Keras dari *deep learning*(Sembiring et al., 2020).

II.17 Keras

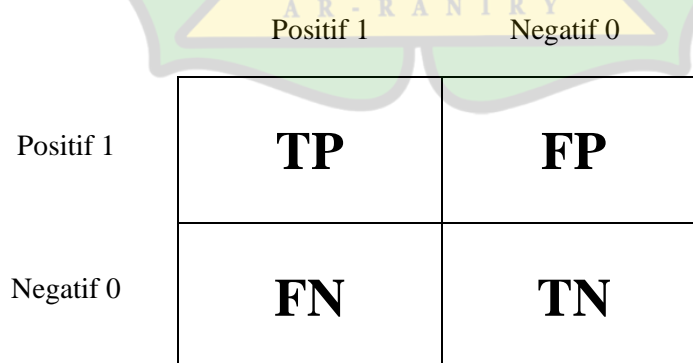
Keras merupakan API yang menggunakan jaringan saraf tingkat tinggi, membantu jalannya *deep learning* dan *artificial intelligence*. Keras menggunakan bahasa pemrograman Python dan bisa berjalan pada TensorFlow, CNTK, atau Theano. Keras merupakan *neural network library* yang mudah dipergunakan. Seiring perkembangan *deep learning*, *keras* menyediakan *pretrained model* atau model yang sudah terlatih yang dapat digunakan untuk menyelesaikan berbagai macam masalah seperti *image classification* dan permasalahan lainnya. Pada keras, model yang sudah terlatih ini disebut dengan *keras application*. (Putra, 2018).

II.18 Metode Evaluasi

Untuk mengetahui hasil atau kesimpulan maka dilakukan evaluasi pada tahap akhir penelitian ini. Evaluasi pada penelitian ini dilakukan dengan menggunakan *confusion matrix*, yaitu *recall*, *precision*, *f1 score* dan *accuracy*.

II.18.1 Confusion Matrix

Ketika mengevaluasi performa prosedur algoritma di *machine learning* biasanya menggunakan metode *Confusion Matrix*. *Confusion Matrix* merepresentasikan prediksi serta kondisi yang sebenarnya dari data yang didapatkan oleh *machine learning*. Gambar 8 menunjukkan perhitungan pada *confusion matrix*.



	Positif 1	Negatif 0
Positif 1	TP	FP
Negatif 0	FN	TN

Gambar 12 Tabel perhitungan *confusion matrix*.

Keterangan :

True Positive (TP) : jumlah data positif yang diprediksi benar.

False Positive (FP): jumlah data negatif yang diprediksi sebagai data positif.

False Negative (FN): jumlah data positif yang diprediksi sebagai data negative.

True Negative (TN) : jumlah data negatif yang diprediksi benar.

II.18.2 Accuracy

Accuracy dapat menunjukkan seberapa akurat model dapat mengklasifikasikan dengan benar. Secara matematis *accuracy* ditulis sebagai berikut:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

II.18.3 Precision

Precision dapat didefinisikan sebagai rasio dari jumlah total data positif yang diprediksi bernilai benar dengan jumlah total data positif yang diprediksi. *High Precision* menunjukkan data berlabel positif memang positif (Pratiwi et al., 2021). Secara matematis *precision* ditulis sebagai berikut:

$$precision = \frac{TP}{TP + FP} \quad (2)$$

II.18.4 Recall

recall menunjukkan beberapa persen data kategori positif yang terklasifikasi dengan benar oleh sistem. *High Recall* menunjukkan kelas dikenali dengan baik. Secara matematis *recall* ditulis sebagai berikut:

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

II.18.5 F1 Score

F1 score adalah *Harmonic Mean* antara *presisi* dan *recall*. Nilai *F1 score* berada antara 0 dan 1 dimana 1 adalah nilai terbaik. *F1-Score* yang baik mengindikasikan model klasifikasi memiliki *precision* dan *recall* yang baik (Ainur Rohman & Arifianto, 2021). Secara matematis *F1 Score* ditulis sebagai berikut:

$$f1 \text{ score} = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} \quad (4)$$

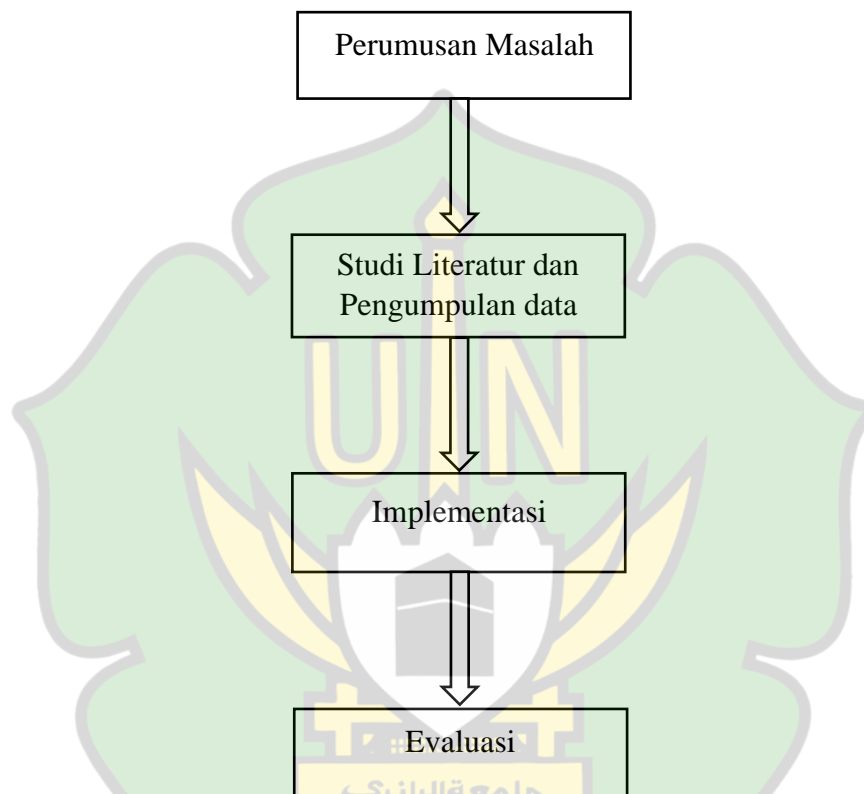


BAB III

METODE PENELITIAN

III.1 Tahapan Penelitian

Adapun tahapan penelitian yang akan dilakukan dalam penulisan skripsi ini dapat dilihat pada gambar 13.



Gambar 13 Tahapan Penelitian.

III.1.1 Perumusan Masalah

Perumusan masalah merupakan pernyataan atau gagasan spesifik mengenai ruang lingkup masalah yang akan diteliti.

III.1.2 Studi Literatur dan Pengumpulan data

Studi literatur adalah cara yang dipakai untuk menghimpun data-data atau sumber-sumber yang berhubungan dengan topik yang diangkat dalam suatu penelitian (Habsy, 2017).

III.1.3 Implementasi

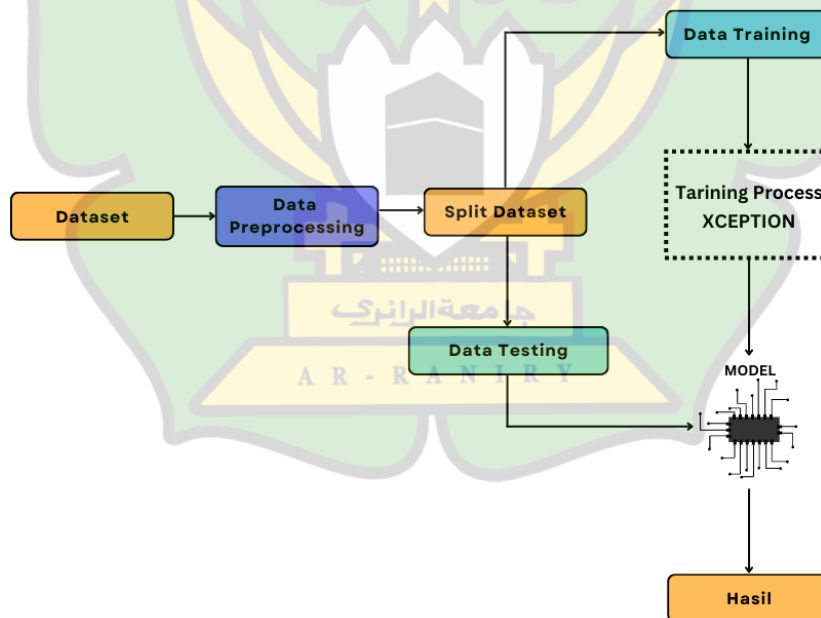
Implementasi adalah tahap penerapan sekaligus pengujian pada penelitian yang dibuat (Amput, 2019).

III.1.4 Evaluasi

Evaluasi bertujuan untuk mengidentifikasi kekuatan dan kelemahan dari suatu penelitian yang sedang digunakan, mengetahui tersedia atau tidaknya suatu informasi saat diperlukan, dan mengetahui bahwa informasi yang diberikan dalam aplikasi disajikan secara akurat, handal, dan tepat (Darmoyo, 2020).

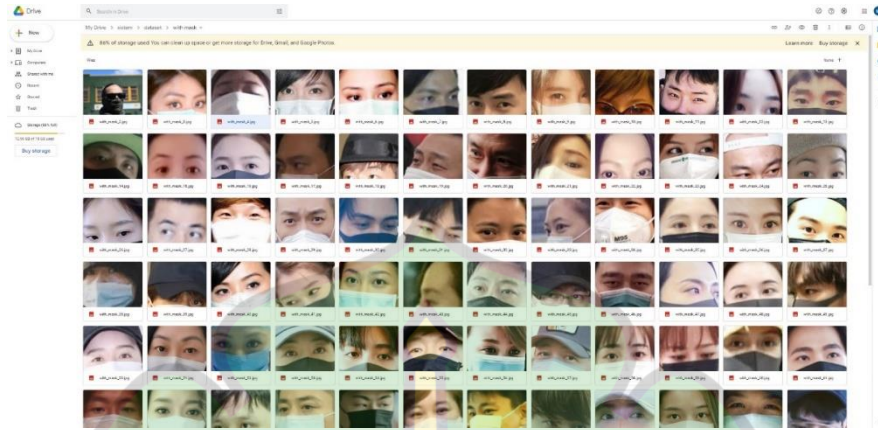
III.2 Alur Penelitian

Alur penelitian pada penelitian melalui beberapa tahapan yang dimulai dari dataset, data *preprocessing* setelah itu dataset akan di *splitting* atau dibagi menjadi dua yaitu data *training* yang bertujuan untuk melatih semua dataset dan data *testing* untuk membandingkan dataset semula dengan data yang sudah di *training*, gambaran alur penelitian ini dapat dilihat pada gambar 14.



III.2.1 Dataset

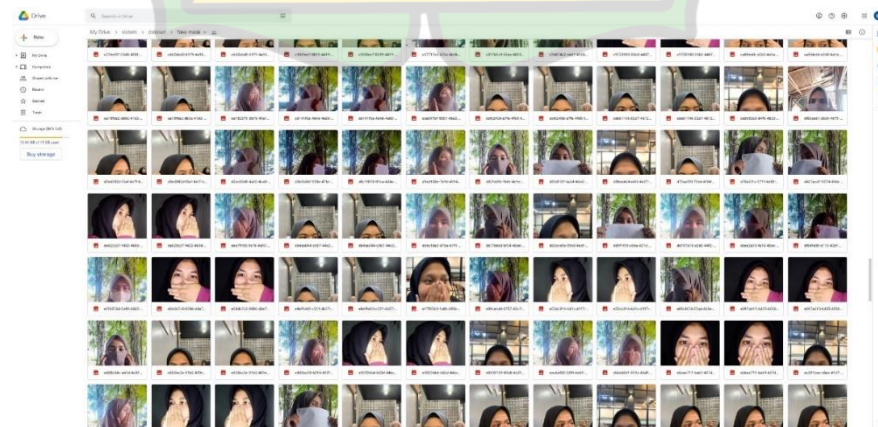
Dataset yang digunakan pada penelitian ini adalah data-data berupa gambar wajah yang menggunakan masker, tidak bermasker dan masker palsu, dataset tersebut dibagi sesuai dengan kelasnya masing masing seperti pada gambar 15,16 dan 17.



Gambar 15 Contoh dataset wajah yang memakai masker.



Gambar 16 Contoh dataset wajah yang tidak memakai masker.



Gambar 17 Contoh dataset wajah yang memakai masker palsu

Dalam penelitian ini dataset yang digunakan diperoleh dengan dua metode yaitu :

1. Dataset yang memakai masker dan tidak bermasker diperoleh dari website Kaggle.
2. Dataset yang memakai masker palsu diperoleh dari pemotretan secara langsung.

Pada penelitian ini banyaknya dataset yang digunakan adalah 7715 citra dan dibagi menjadi 3 kelas. Pembagian kelas dataset bisa dilihat pada tabel 1.

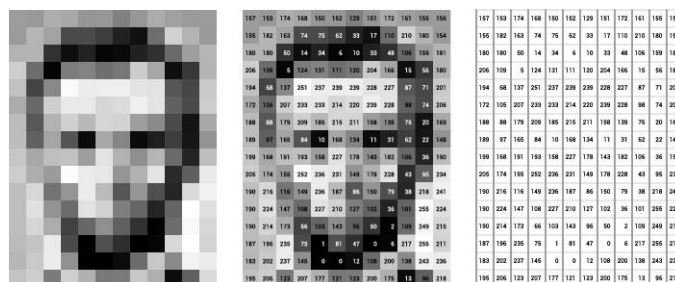
Tabel 1 Jumlah Dataset.

Kelas	Jumlah	Sumber
With mask	3000	Kaggle
No Mask	3000	Kaggle
Fake mask	1715	Pemotretan Langsung

III.2.2 Data Preprocessing

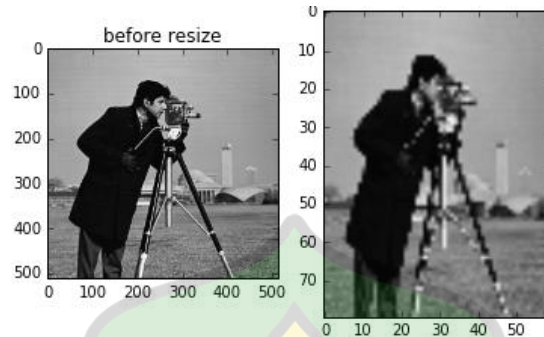
Tahap data *preprocessing* bertujuan agar dataset siap digunakan oleh model untuk dipelajari. Beberapa proses yang terjadi pada tahap ini adalah sebagai berikut:

1. *Splitting* dataset, Proses *splitting* dilakukan untuk membagi dataset menjadi 2 bagian yaitu data *training* sebesar 90% dan data *testing* sebesar 10%(Leiner et al., n.d.).
2. *Rescale*, Pada proses ini nilai dari setiap *pixel* pada citra diskalakan menjadi nilai antara 0 dan 1. *Rescale* dilakukan agar proses komputasi saat *training* berjalan lebih cepat karena hanya memproses angka dengan nilai antara 0 dan 1 dimana angka tersebut mempunyai alokasi memori yang kecil(Damen et al., 2022). Contoh citra yang sudah melalui proses *rescale* dapat dilihat pada gambar 18.



Gambar 18 Contoh *Rescale*.

3. *Resize*, Pada proses ini ukuran citra di ubah dari 256x256 piksel menjadi 299x299 piksel. *Resize* dilakukan karena model seperti Xception, menerima gambar dengan ukuran 299x299 secara *default*(Immanuel & Setiabudi, n.d.). Contoh citra yang sudah melalui proses *resize* dapat dilihat pada gambar 19.



Gambar 19 Contoh *resize*.

```
# connect to google drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

#add the path general where the classes subpath are allocated
path = 'gdrive/MyDrive/sistem/dataset'

classes=["fake mask", "no mask", "with mask"]
num_classes = len(classes)
batch_size = 16

***No Augmentation on the Test set Images**
datagen = ImageDataGenerator(rescale=1./255,
                             validation_split=0.1) #9:1

#loading the images to training set
train_gen = datagen.flow_from_directory(directory=path,
                                       target_size=(299, 299),
                                       class_mode='categorical',
                                       subset='training',
                                       shuffle=True, classes=classes,
                                       batch_size=batch_size,
                                       color_mode="rgb")

#loading the images to test set
test_gen = datagen.flow_from_directory(directory=path,
                                       target_size=(299, 299),
                                       class_mode='categorical',
                                       subset='validation',
                                       shuffle=False, classes=classes,
                                       batch_size=batch_size,
                                       color_mode="rgb")
```

Gambar 20 *Prepare dataset dan data preprocessing*.

Pada *pseudocode* diatas dijelaskan tahap pertama adalah melakukan *mount* terhadap *google drive* supaya dataset yang tersimpan pada *drive* bisa dipanggil. Selanjut dilakukan *Data Augmentation*. *Data augmentation* adalah sebuah teknik

memanipulasi sebuah data tanpa kehilangan inti atau esensi dari data tersebut. Untuk data berupa *Image*, kita bisa lakukan *rotate*, *flip*, *crop*, dll.

III.2.3 Training

Training adalah proses bagi model untuk mempelajari data. Data yang digunakan adalah data *training* yang sudah dipersiapkan sebelumnya pada tahap data *preprocessing*. Data *training* yang digunakan pada penelitian ini adalah 6944 dataset.

Hasil dari proses *training* dapat dilihat dari nilai *accuracy* dan *loss score* yang ditunjukkan oleh model. Nilai *accuracy* menunjukkan kemampuan model dalam mengenali dan memprediksi data dengan benar, semakin tinggi nilai *accuracy* maka model semakin baik. Sebaliknya, *loss score* adalah nilai kesalahan model dalam mengenali dan memprediksi data, semakin rendah nilai *loss score* maka model semakin baik.

III.2.4 Testing

Setelah proses *training* selesai maka selanjutnya dilakukan *testing* model. Data yang digunakan adalah data testing yang sudah dipersiapkan sebelumnya yaitu 772 dataset.

III.2.5 Model

Pada tahap ini akan dibangun beberapa model menggunakan Metode CNN dengan Keras. model yang dibangun adalah xception. Arsitektur ini mirip dengan arsitektur InceptionResnetV2 – hanya saja pada proses *feature learning* menggunakan Xception *layer*. *Layer – layer* yang terdapat di dalam model Xception dapat dilihat pada gambar 6. Xception bekerja mirip seperti InceptionResnetV2 dengan melakukan *feature learning* pada *layer* konvolusi yang terpisah. kemudian hasil *feature learning* digunakan untuk klasifikasi atau pendeteksian. Gambar 20 menunjukkan model arsitektur Xception.


```

# Xception
learning_rate = 1e-5
epochs = 10

Xception = tf.keras.applications.Xception(input_shape = (299, 299, 3),
                                          weights = 'imagenet',
                                          include_top = False)

Xception.trainable = True

model = tf.keras.Sequential([
    Xception,
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(3, activation = "softmax")],
    name = "Xception_Categorical_Classification")

model.compile(optimizer = Adam(learning_rate = learning_rate, decay = learning_rate / epochs),
              loss = "categorical_crossentropy",
              metrics = ["accuracy"])

# check point callback
checkpoint_path = "training/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                save_weights_only=True,
                                                verbose=1)

# fit model
history=model.fit(train_gen,
                  validation_data= test_gen,
                  steps_per_epoch=100,
                  batch_size=batch_size,
                  epochs= epochs,
                  callbacks=[cp_callback])

```

Gambar 21 Model Xception, *Training* dan *Testing*.

Pada *pseudocode* diatas dijelaskan tahapan pembangunan model. *Learning rate*, merupakan salah satu parameter *training* untuk menghitung nilai koreksi bobot pada waktu proses *training*. Nilai *learning rate* ini berada pada *range* nol (0) sampai (1). Semakin besar nilai *learning rate*, maka proses *training* akan berjalan semakin cepat. Pada penelitian ini *learning rate* yang digunakan sebanyak 5 dan menggunakan 10 *epoch* yang artinya proses training akan diulang sebanyak 10 kali.

Input shape pada model ini menggunakan 3 shape dan image dengan *size* 299x299, bobot dari masing-masing layer yang sudah di training berdasarkan bobot imagenet dan include topnya yaitu false yang bererti perintah untuk menyertakan apakah model yang digunakan akan disertakan dengan top layer dari arsitektur Xception. Layer yang akan dilalui pada model ini adalah flatten yang

berfungsi membuat input yang memiliki banyak dimensi menjadi satu dimensi. Dense layer pada model ini adalah 3 dan fungsi aktivasinya adalah softmax.

Sesudah model selesai dibuat maka selanjutnya akan dilakukan proses *training* dan *testing* sesuai dengan jumlah *epoch* yang sudah diatur pada pembuatan model sebelumnya.

III.2.6 Evaluasi Akhir

Pada tahapan terakhir dalam penelitian ini dilakukan evaluasi untuk mengetahui hasil kinerja dari setiap model. Hasil kinerja dapat diketahui dari perhitungan menggunakan *confusion matrix*, *recall*, *precision*, *f1 score* dan *accuracy*.

```
# confusion matrix
# transform the predictions into array such as [0,0,1,2...]
predictions = np.array(list(map(lambda x: np.argmax(x), y_pred)))
#Retrieve the True classes of the test set
y_true=test_gen.classes
# Build Confusion Matrix
CMatrix = pd.DataFrame(confusion_matrix(y_true, predictions), columns=classes, index =classes)

plt.figure(figsize=(12, 6))
ax = sns.heatmap(CMatrix, annot = True, fmt = 'g' ,vmin = 0, vmax = 250,cmap = 'Blues')
ax.set_xlabel('Predicted',fontsize = 14,weight = 'bold')
ax.set_xticklabels(ax.get_xticklabels(),rotation =0);
ax.set_ylabel('Actual',fontsize = 14,weight = 'bold')
ax.set_yticklabels(ax.get_yticklabels(),rotation =0);
ax.set_title('Confusion Matrix - Test Set',fontsize = 16,weight = 'bold',pad=20);

#Accuracy Result
acc = accuracy_score(y_true, predictions)
#Precision, Recall and F-Score (For the whole dataset)
results_all = precision_recall_fscore_support(y_true, predictions, average='macro',zero_division = 1)
#Precision, Recall and F-Score (For each Class)
results_class = precision_recall_fscore_support(y_true, predictions, average=None, zero_division = 1)

#Organise the Results into a DataFrame
metric_columns = ['Precision', 'Recall', 'F-Score','S']
all_df = pd.concat([pd.DataFrame(list(results_class)).T,pd.DataFrame(list(results_all)).T])
all_df.columns = metric_columns
all_df.index = ['COVID', 'Normal', 'Viral Pneumonia','Total']

print('**Overall Results**')
print('Accuracy Result: %.2f%%'%(acc*100)) #Accuracy of the whole Dataset
print('Precision Result: %.2f%%'%(all_df.iloc[3,0]*100))#Precision of the whole Dataset
print('Recall Result: %.2f%%'%(all_df.iloc[3,1]*100))#Recall of the whole Dataset
print('F-Score Result: %.2f%%'%(all_df.iloc[3,2]*100))#F-Score of the whole Dataset
```

Gambar 22 Confusion Matrix.

Pada tahapan ini akan dilakukan proses representasi hasil dari proses training dan testing, selanjutnya dilakukan perhitungan keakuratan menggunakan *precision*, *recall*, *f1 Score* dan *training*.

III.3 Bahan dan Alat

Pada penelitian ini bahasa pemrograman yang digunakan adalah Python 3.7.12. Selain itu pada penelitian ini juga memanfaatkan beberapa fitur atau library yaitu :

1. Tensorflow, merupakan framework untuk metode *Deep Neural Network*.
2. Os untuk membuat dan menghapus *directory* (folder), mengambil isinya, mengubah dan mengidentifikasi *directory*.
3. Numpy, merupakan *library* yang disediakan python dalam memudahkan operasi komputasi tipe data numerik.
4. Pandas, merupakan modul untuk memuat *file* ke dalam tabel *virtual* menyerupai spreadsheet.
5. Seaborn, merupakan *library* untuk membuat grafik dan statistik.
6. Matplotlib adalah kumpulan fungsi yang membuat beberapa perubahan pada gambar. Misalnya, membuat gambar, membuat areaplot dalam gambar, menambah label di plot dan lainnya.

Notebook yang digunakan dalam penelitian ini adalah *Google Collab* dengan spesifikasi RAM 12 GB, Disk 78 GB, dan GPU 15 GB.

III.4 Tempat dan Waktu

Penelitian ini dilaksanakan pada laboratorium program studi Teknologi Informasi Universitas Islam Negeri Ar-Raniry Banda Aceh pada tanggal 21 Juli 2022 sampai 20 November 2022.

BAB IV

HASIL PENELITIAN

IV.1 Hasil Uji Coba

Berikut hasil uji coba pada penelitian pendeteksian *fake mask* menggunakan metode CNN dengan arsitektur Xception.

IV.1.1 Hasil Proses Training

Salah satu bagian terpenting dari berhasilnya proses identifikasi ekspresi berbasis citra ini adalah bagusnya hasil dari proses *training*. Baiknya hasil dari proses *training* akan memberikan dampak yang sangat tinggi terhadap hasil yang didapatkan pada proses uji coba nanti.

Setelah arsitektur dibentuk dan dilakukan proses model. Maka algoritma akan langsung melakukan pelatihan pada data yang sudah disiapkan sebelumnya. Seperti yang sudah dibahas pada sub bab sebelumnya, data yang akan digunakan untuk proses *training* adalah sebesar 90% data. Data tersebut akan dieksekusi oleh Xception untuk diekstraksi yang dipelajari fiturnya.

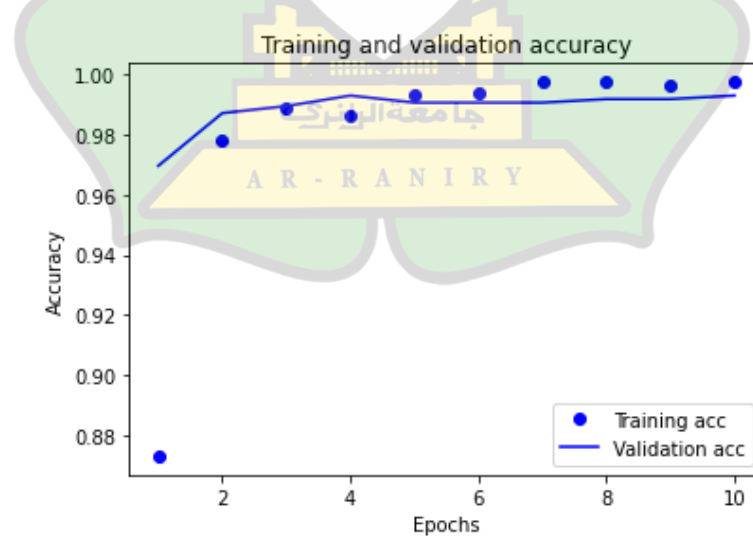
Parameter literasi yang digunakan pada proses *training* ini sebanyak 10 dan 5 *epoch*. Epoch merupakan *hyperparameter* yang menentukan berapa kali algoritma *deep learning* bekerja melewati seluruh dataset baik secara *forward* maupun *backward*. Dengan nilai *batch size* sebesar 100, *batch size* adalah jumlah *training* sampel yang digunakan dalam satu *batch* untuk sekali iterasi. dalam istilah yang lebih sederhana, satu *epoch* tercapai ketika semua *batch* telah berhasil dilewatkan melalui jaringan saraf satu kali. Jadi proses *training* akan berlangsung dan diulang-ulang sebanyak 10 untuk *training* pertama dan 5 kali untuk *training* ke dua supaya memperoleh ekstraksi ciri dari fitur yang dibutuhkan. Kemudian untuk nilai *learning rate* yang digunakan untuk melakukan *update* bobot adalah 0.001. setelah proses *training* selesai, kemudian dilakukan perhitungan akurasi untuk mendapatkan nilai akurasi hasil *training*. Berikut tabel hasil *training* 10 dan 5 *epoch* dimana pada tabel training terdiri dari jumlah *epoch*, *training accuracy*, *validation*

accuracy, training loss, validation loss dan waktu yang dibutuhkan untuk setiap *epoch(time)*.

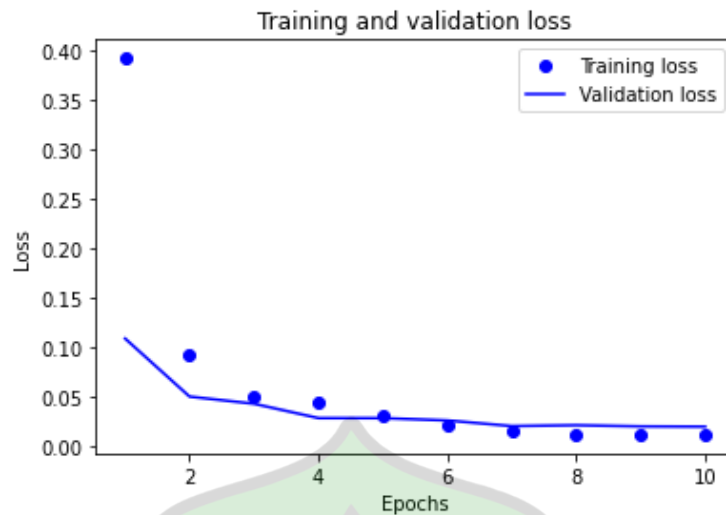
Tabel 2 Hasil *Training* 10 *epoch*.

Epoch	Training Acc	Validation acc	Taining Loss	Validation Loss	Time (S)
1	0.8731	0.9696	0.3924	0.1091	814
2	0.9781	0.9871	0.0934	0.0505	402
3	0.9887	0.9895	0.0503	0.0432	334
4	0.9861	0.9930	0.0442	0.0287	289
5	0.9931	0.9907	0.0314	0.0287	232
6	0.9937	0.9907	0.0221	0.0264	178
7	0.9975	0.9907	0.0150	0.0207	159
8	0.9975	0.9918	0.0124	0.0216	126
9	0.9962	0.9918	0.0122	0.0204	110
10	0.9975	0.9930	0.0123	0.0201	112

Berdasarkan hasil *training* model Xception yang ditunjukkan pada tabel 2. Model di *training* menggunakan 10 *epoch*. Pada *epoch* terakhir model mempunyai nilai *training accuracy* 0.9975, *validation accuracy* 0.9930, *training loss* 0.0123, dan *validation loss* 0.0201. Grafik yang dihasilkan dari proses *training* 10 *epoch* dapat dilihat pada gambar 23 dan 24.



Gambar 23 *Training* dan *validation accuracy* 10 *epoch*.



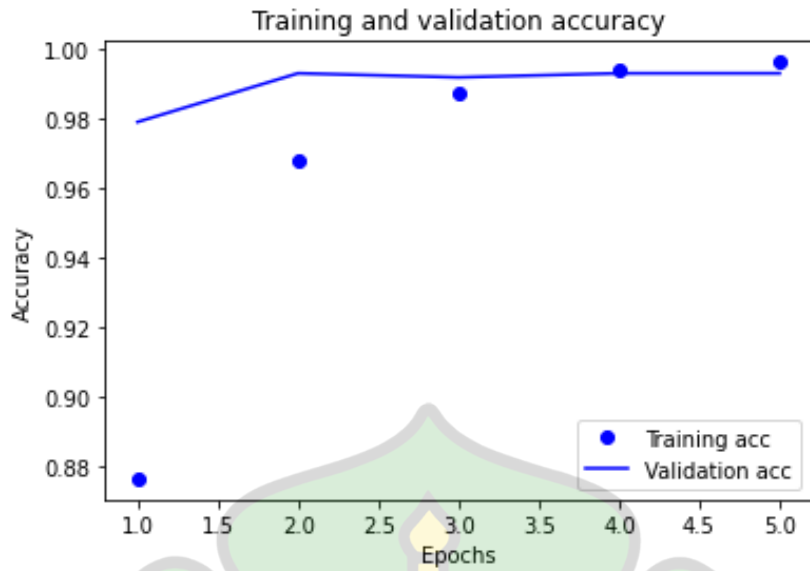
Gambar 24 Training dan validation loss 10 epoch.

Hasil yang diperoleh dari proses *training* dapat dilihat juga dalam bentuk grafik dimana pada grafik ini tanda titik diketahui sebagai *training* dan tanda garis sebagai validasi. Grafik *training* diatas menunjukkan bahwa *accuracy* mencapai 0.98+ dan begitu juga dengan validasi, sedangkan untuk *loss training* dan *loss validasi* tidak mencapai 0.1.

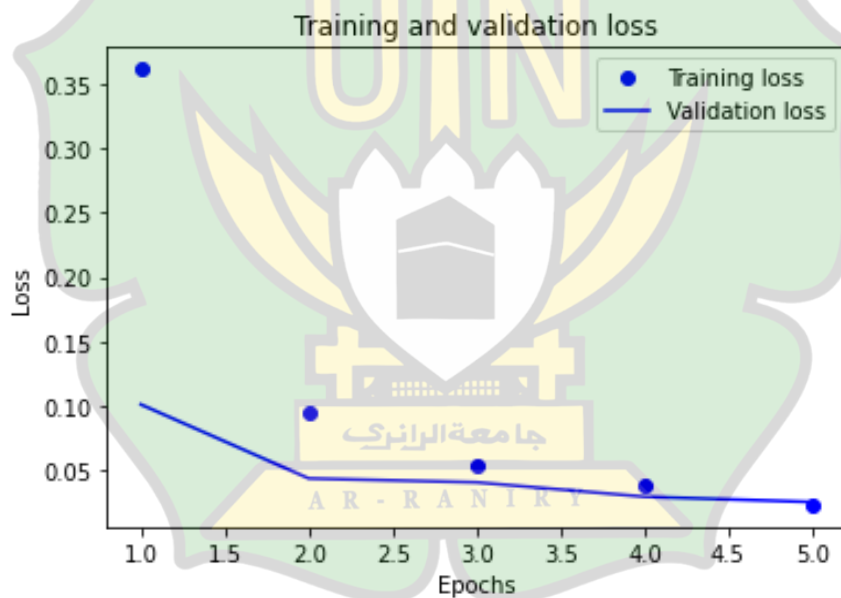
Tabel 3 Hasil *training* 5 epoch.

Epoch	Training Acc	Validation acc	Taining Loss	Validation Loss	Time (S)
1	0.8813	0.9638	0.3520	0.1176	628
2	0.9775	0.9848	0.0904	0.0613	323
3	0.9912	0.9895	0.0423	0.0410	266
4	0.9880	0.9895	0.0376	0.0323	206
5	0.9931	0.9907	0.0294	0.0287	185

Berdasarkan hasil *training* model Xception yang ditunjukkan pada tabel 2. Model di *training* menggunakan 5 *epoch*. Pada *epoch* terakhir model mempunyai nilai *training accuracy* 0.9931, *validation accuracy* 0.9907, *training loss* 0.0294, dan *validation loss* 0.0287. Diagram yang dihasilkan dari proses *training* 5 *epoch* dapat dilihat pada gambar 25 dan 26.



Gambar 25 Training dan validation accuracy 5 epoch.



Gambar 26 Training dan validation loss 5 epoch.

Berdasarkan hasil *training* 10 dan 5 *epoch*, diketahui bahwa hasil dari *training* menggunakan 10 epoch lebih tinggi tingkat *accuracy* dan lebih rendah tingkat *loss* yang dihasilkan.

IV.1.2 Hasil Proses Testing

Setelah proses *training* selesai maka selanjutnya dilakukan *testing* model. Dimana pada proses ini data yang sudah di *training* akan di bandingkan dengan data yang sudah dipersiapkan pada proses *preprocessing*. Dataset yang digunakan pada proses testing adalah 10%.

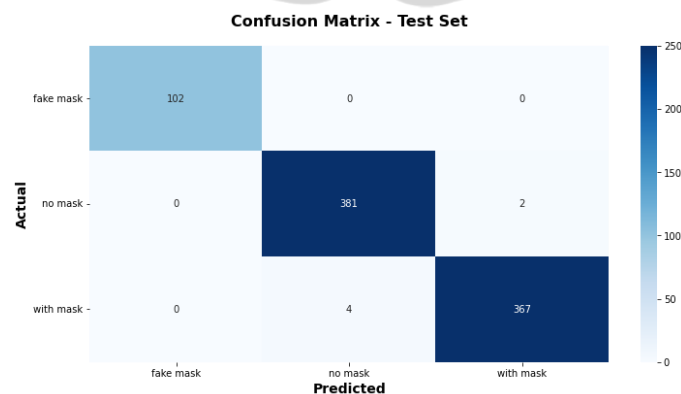
Tabel 4 Hasil *testing* perbandingan antara 10 *epoch* dan 5 *epoch*.

Epoch	Testing Acc	Testing Loss	Time (S)
10	0.9930	0.0201	154
5	0.9930	0.0249	155

Pada Tabel 4 menunjukkan nilai *accuracy* dan *loss score* yang dihasilkan oleh model xception pada proses *testing*. Dari tabel 4 dapat dilihat Xception adalah model yang memiliki keakuratan yang tinggi yaitu *accuracy* 0,9953 dan *loss score* 0,0249 dengan akumulasi waktu 155s yang di running dengan 10 *epoch* dan *accuracy* 0,9930 serta *loss score* 0,0249 dengan akumulasi waktu 155s yang di running dengan 5 *epoch*.

IV.1.3 Confusion Matrix

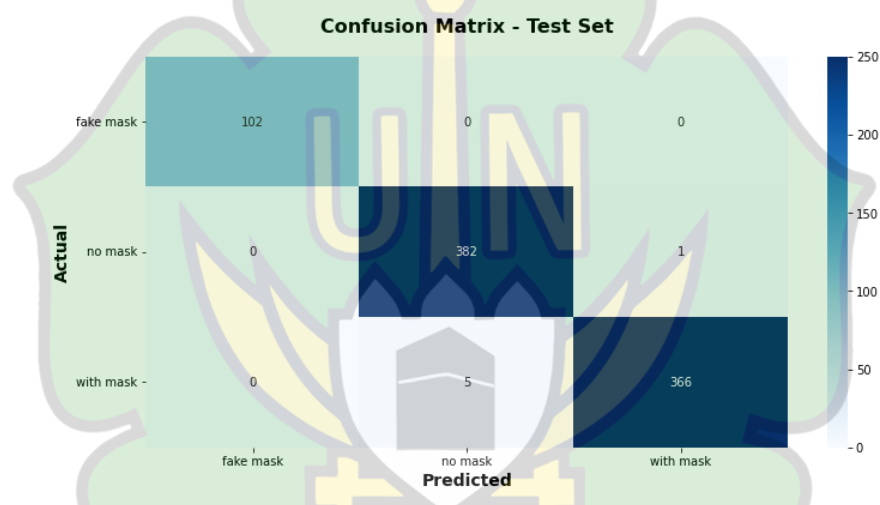
Setelah proses *training* dan *testing* selesai maka selanjutnya dilakukan proses pengukuran. Model melakukan prediksi terhadap data *testing* citra wajah. Kemudian hasil prediksi tersebut diukur menggunakan *confusion matrix*, *recall*, *precision*, *f1 score*, dan *accuracy*.



Gambar 27 Hasil *confusion matrix* 10 *epoch*.

Dari *confusion matrix* model Xception dengan 10 *epoch* yang dihasilkan dari prediksi terhadap data *testing* maka dapat disimpulkan:

1. Dari total 102 citra pada kelas *fake mask*, sebanyak 102 citra yang diprediksi dengan benar, 0 citra yang diprediksi sebagai *no mask* dan 0 citra yang diprediksi sebagai *with mask*.
2. Dari total 383 citra pada kelas *no mask*, sebanyak 381 citra yang diprediksi dengan benar, 0 citra yang diprediksi sebagai *fake mask* dan 2 citra yang diprediksi sebagai *with mask*.
3. Dari total 371 citra pada kelas *with mask*, sebanyak 367 citra diprediksi dengan benar, 0 citra yang diprediksi sebagai *fake mask* dan 4 citra yang diprediksi sebagai *no mask*.



Gambar 28 Hasil *confusion matrix* 5 *epoch*.

Dari *confusion matrix* model Xception dengan 5 yang dihasilkan dari prediksi terhadap data *testing* maka dapat disimpulkan:

1. Dari total 102 citra pada kelas *fake mask*, sebanyak 102 citra yang diprediksi dengan benar, 0 citra yang diprediksi sebagai *no mask* dan 0 citra yang diprediksi sebagai *with mask*.
2. Dari total 383 citra pada kelas *no mask*, sebanyak 382 citra yang diprediksi dengan benar, 0 citra yang diprediksi sebagai *fake mask* dan 1 citra yang diprediksi sebagai *with mask*.

3. Dari total 371 citra pada kelas *with mask*, sebanyak 366 citra diprediksi dengan benar, 0 citra yang diprediksi sebagai *fake mask* dan 5 citra yang diprediksi sebagai *no mask*.

IV.1.4 Recall, Precision, f1 Score dan Accuracy

Berdasarkan *confusion matrix* maka langkah selanjutnya yang dilakukan adalah mencari nilai *Recall*, *Precision*, *F1 Score*, dan *Accuracy*. Nilai *Recall* yang tinggi menunjukkan model dapat mengenali *class* dengan baik. Pada tabel 5 terlihat bahwa nilai *Recall* yaitu 99,65%. *Precision* adalah nilai yang menunjukkan ketepatan model dalam memprediksi sebuah *class* dengan benar. Pada tabel 5 terlihat bahwa *Precision* mempunyai nilai yaitu 99,65%. *F1 score* merupakan *Harmonic Mean* antara *presicion* dan *Recall*. Nilai *f1 score* yang tinggi mengindikasikan model memiliki *Precision* dan *Recall* yang baik. Pada tabel 5 terlihat *F1 score* mempunyai nilai yaitu 99.65%. sedangkan *Accuracy* dapat menunjukkan seberapa akurat model dapat melakukan pendeteksian dengan benar terhadap keseluruhan data. Pada tabel 5 terlihat *accuracy* mempunyai nilai yaitu 99.53% %. Dan *Recall*, *Precision*, *F1 Score*, dan *Accuracy* dengan *training 5 epoch* adalah *Recall* 99.46%, *Precision* 99.48%, *F1 Score* 99.46% dan *Accuracy* 99.30%. Jadi diketahui bahwa semakin banyak proses di *training* maka akan semakin besar tingkat akurasi yang didapat.

Tabel 5 *Recall*, *Precision*, *F1 Score*, dan *Accuracy*.

10 epoch	
Accuracy	99.53%
precision	99.65%
recall	99.65%
f1 Score	99.65%

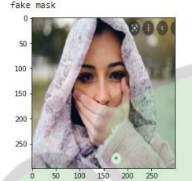

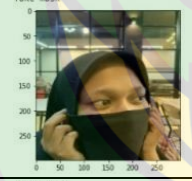

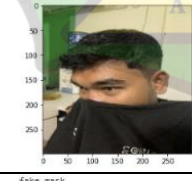
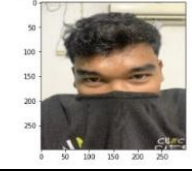
Tabel 6 *Recall*, *Precision*, *F1 Score*, dan *Accuracy*.


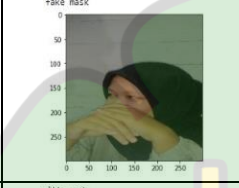
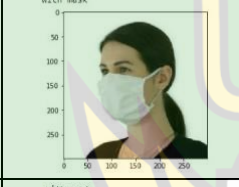
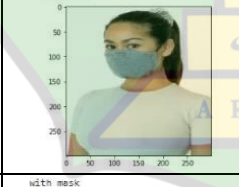

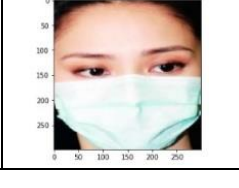
5 epoch	
Accuracy	99.30%
precision	99.48%
recall	99.46%
f1 Score	99.47%

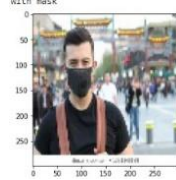
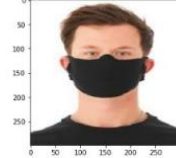




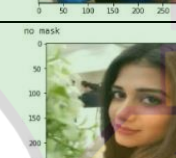
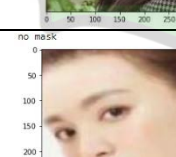
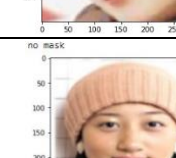
IV.1.5 Sample Pengujian

Pada bagian ini akan dilakukan pengujian terhadap analisa metode CNN dengan arsitektur Xception pada objek menggunakan masker, tidak bermasker dan *fake mask*. Pengujian ini dilakukan dengan menggunakan 30 citra, dimana masing-masing citra memiliki kelas yang berbeda-beda dan dibagi menjadi 10.

Tabel 7 Sample Pengujian.

Kelas	Citra	Waktu(s)	Ouput	Status Identifikasi
Fake mask		22s	fake mask	Benar
Fake mask		19s	fake mask	Benar
Fake mask		21s	fake mask	Benar
Fake mask		22s	fake mask	Benar
Fake mask		23s	fake mask	Benar
Fake mask		19s	fake mask	Benar

Fake mask		20s	fake mask	Benar
Fake mask		20s	fake mask	Benar
Fake mask		22s	fake mask	Benar
Fake mask		21s	fake mask	Benar
With mask		19s	With mask	Benar
With mask		23s	With mask	Benar
With mask		22s	With mask	Benar
With mask		23s	With mask	Benar
With mask		19s	With mask	Benar

With mask		19s	With mask	Benar
With mask		22s	With mask	Benar
With mask		18s	With mask	Benar
With mask		19s	With mask	Benar
With mask		18s	With mask	Benar
No mask		23s	No mask	Benar
No mask		22s	No mask	Benar
No mask		20s	No mask	Benar
No mask		20s	No mask	Benar

No mask		18s	No mask	Benar
No mask		19s	No mask	Benar
No mask		23s	No mask	Benar
No mask		22s	No mask	Benar
No mask		22s	No mask	Benar
No mask		21s	No mask	Benar

Dari hasil pengujian yang dilakukan menggunakan 30 citra yang dibagi masing-masing 10 menjadi *fake mask*, 10 menjadi *with mask* dan 10 menjadi *no mask*, semua pengujian terhadap citra dideteksi dengan benar.

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Pada penelitian ini pendeteksian citra wajah yang memakai masker, tidak bermasker dan masker palsu yang dilakukan dengan model Xception, Dataset yang digunakan adalah citra wajah berjumlah 7715 yang terdiri dari 3 class yaitu memakai masker dengan jumlah 3000 dataset, tidak bermasker dengan jumlah 3000 dataset dan masker palsu 1715. Data *training* yang digunakan sebanyak 90% dan untuk data *testing* sebanyak 10%. Masukkan wajah yang digunakan oleh model berukuran 299x299 *pixel* dan *epoch* yang digunakan dalam penelitian ini sebesar 10 dan 5 *epoch*.

Berdasarkan pembahasan dan hasil analisis yang telah dilakukan dalam penelitian ini, maka kesimpulan yang dapat diambil adalah model Xception merupakan salah satu arsitektur terbaik dengan nilai keakuratan yang tinggi yaitu mencapai 99.53% dengan proses *training* 10 *epoch* dan 99.30% dengan proses *training* 5 *epoch*. jadi diketahui bahwa semakin banyak proses di *training* maka akan semakin besar tingkat akurasi yang didapat.

V.2 Saran

Dari hasil yang telah diuji dalam penelitian ini, peneliti memberikan beberapa saran untuk penelitian yang lebih lanjut dalam konteks yang sama, yaitu sebagai berikut:

1. Menggunakan *pretrained* model lainnya yang tersedia di dalam keras untuk melakukan pendeteksian seperti MobileNet, Inception, VG19 dan Resnet serta mengetahui keakuratan model terbaik.

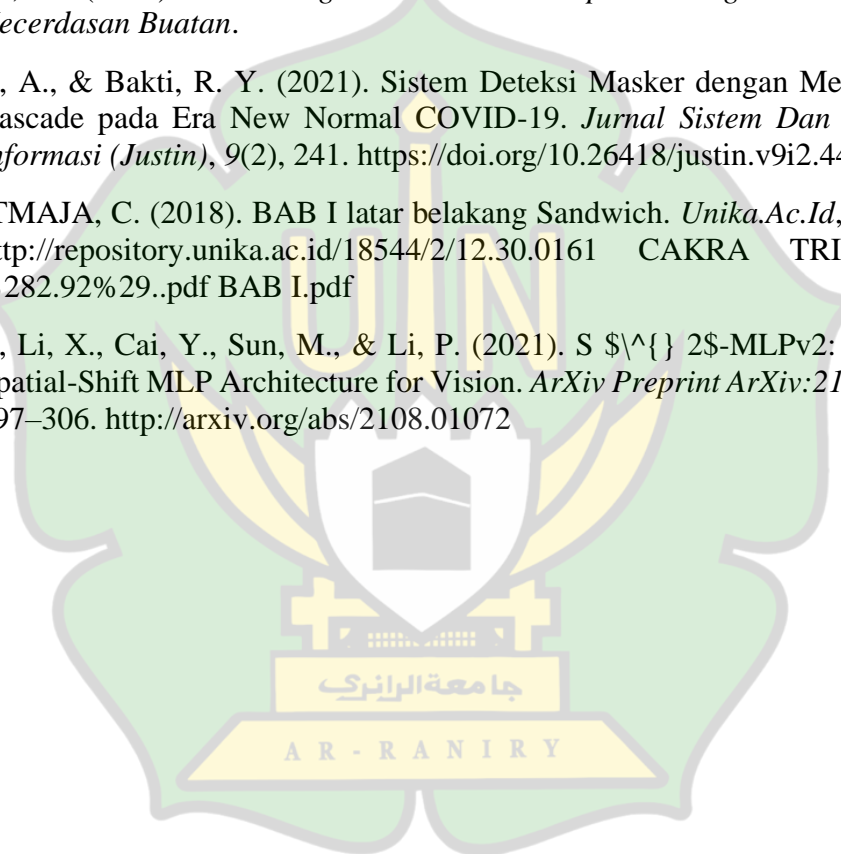
DAFTAR PUSTAKA

- Agarap, A. F. (2018). *Deep Learning using Rectified Linear Units (ReLU)*. 1, 2–8. <http://arxiv.org/abs/1803.08375>
- Ainur Rohman, M., & Arifianto, D. (2021). Penerapan Metode Euclidean Probability dan Confusion Matrix dalam Diagnosa Penyakit Koi. *Jurnal Smart Teknologi*, 2(2), 122–130. <file:///C:/Users/dokta/Downloads/4992-14056-2-PB.pdf>
- Ajit, A., Acharya, K., & Samanta, A. (2020). A Review of Convolutional Neural Networks. *International Conference on Emerging Trends in Information Technology and Engineering, Ic-ETITE 2020, October*. <https://doi.org/10.1109/ic-ETITE47903.2020.049>
- Ali, A. A. A., & Mallaiah, S. (2021). Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout. *Journal of King Saud University - Computer and Information Sciences*, 34(6), 3294–3300. <https://doi.org/10.1016/j.jksuci.2021.01.012>
- Aminullah, M. (2020). *Alat Deteksi Masker Dengan Metode Convolutional Neural Network Untuk Tunanetra Pada Era New Normal*. 321–324.
- Amput, F. (2019). Perancangan Sistem Informasi Reservasi Kamar Berbasis Web Pada Hotel Karmila Bandung. *Jurnal Sistem Informatika Dan Informasi*, 1, 1–476.
- Apandi, T. H., & Sugianto, C. A. (2018). *Analisis Komparasi Machine Learning Pada Data Spam Sms*. 12(1). <https://doi.org/10.31227/osf.io/adhx8>
- Budi, R. S., Patmasari, R., & Saidah, S. (2021). Klasifikasi Cuaca Menggunakan Metode Convolutional Neural Network (Cnn) Weather Classification Using Convolutional Neural Network (Cnn) Method. *E-Proceeding of Engineering : Vol.8, No.5 Oktober 2021*, 8(5), 5047–5052.
- Chang, S., Cohen, T., & Ostdiek, B. (2018). What is the machine learning? *Physical Review D*, 97(5), 56009. <https://doi.org/10.1103/PhysRevD.97.056009>
- Damen, D., Doughty, H., Farinella, G. M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., & Wray, M. (2022). Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130(1), 33–55. <https://doi.org/10.1007/s11263-021-01531-2>
- Darmasita. (2020). Deteksi Penggunaan Masker Menggunakan Xception Transfer Learning. *JURNAL INSTEK (Informatika Sains Dan Teknologi)*, 5, 279–288.
- Darmoyo, J. (2020). *Desain Sistem Secara Umum*. 1–21.
- Desai, M., & Shah, M. (2021). An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). *Clinical EHealth*, 4(2021), 1–11. <https://doi.org/10.1016/j.ceh.2020.11.002>

- Developers Breach. (2020). Convolutional Neural Network | Deep Learning. *Developersbreach.Com*. <https://developersbreach.com/convolution-neural-network-deep-learning/>
- Dewi, N., & Ismawan, F. (2021). Implementasi Deep Learning Menggunakan Cnn Untuk Sistem Pengenalan Wajah. *Faktor Exacta*, 14(1), 34. <https://doi.org/10.30998/faktorexacta.v14i1.8989>
- Firzatullah, R. M. (2021). Menggunakan Sistem Pendukung Keputusan Penentuan Uang Kuliah Tunggal Universitas XYZ Menggunakan Algoritma Backpropagation. *Petir*, 14(2), 170–180. <https://doi.org/10.33322/petir.v14i2.996>
- Ghani, M. A., Fahrizal, F., & Lawi, A. (2021). Implementasi Arsitektur Xception Untuk Klasifikasi Citra Covid-19 Radiography. *Konferensi Nasional Ilmu Komputer (KONIK)*, 413–419.
- Habsy, B. A. (2017). Seni Memahami Penelitian Kuliitatif Dalam Bimbingan Dan Konseling : Studi Literatur. *JURKAM: Jurnal Konseling Andi Matappa*, 1(2), 90. <https://doi.org/10.31100/jurkam.v1i2.56>
- Han, E. S., & goleman, daniel; boyatzis, Richard; Mckee, A. (2019). Bahaya Gas Sulfur Dan Akibat Terhadap Manusia. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
- Hermawati, F. A., & Zai, R. A. (2021). Sistem Deteksi Pemakaian Masker Menggunakan Metode Viola-Jones dan Convolutional Neural Networks (CNN). *Proceeding KONIK (Konferensi Nasional Ilmu Komputer)*, 5, 182–187.
- Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., Zhang, L., Fan, G., Xu, J., Gu, X., Cheng, Z., Yu, T., Xia, J., Wei, Y., Wu, W., Xie, X., Yin, W., Li, H., Liu, M., ... Cao, B. (2020). Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *The Lancet*, 395(10223), 497–506. [https://doi.org/10.1016/S0140-6736\(20\)30183-5](https://doi.org/10.1016/S0140-6736(20)30183-5)
- Iii, B. A. B., & Teori, L. (n.d.). *Bab iii. landasan teori 3.1*. 8–10.
- Imanuel, A., & Setiabudi, D. H. (n.d.). *Penerapan Convolutional Neural Network dengan Pre-Trained Model Xception untuk Meningkatkan Akurasi dalam Mengidentifikasi Jenis Ras Kucing*. www.kaggle.com
- Karo, M. B. (2012). *Perilaku Hidup Bersih dan Sehat (PHBS) Strategi Pencegahan Penyebaran Virus Covid-19*. 1–4.
- Kholik, A. (2021). Klasifikasi Menggunakan Convolutional Neural Network (Cnn) Pada Tangkapan Layar Halaman Instagram. *Jdmsi*, 2(2), 10–20.
- Kusuma, U. F., Arsyad, N., Lavinia, M. S., Rahayu, S., Pasaribu, M. K. K., & Syakurah, R. A. (2021). Analisis Penelusuran Masker Sebagai Protokol Kesehatan Saat Pandemi Covid-19 Di Indonesia: Studi Google Trends. *VISIKES: Jurnal Kesehatan Masyarakat*, 20(1). <https://doi.org/10.33633/visikes.v20i1.4191>

- Leiner, J., Duan, B., Wasserman, L., & Ramdas, A. (n.d.). *Data fission : splitting a single data point. 2.*
- Miceli, M., Yang, T., Naudts, L., Schuessler, M., Serbanescu, D., & Hanna, A. (2021). Documenting computer vision datasets: An invitation to reflexive data practices. *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 161–172. <https://doi.org/10.1145/3442188.3445880>
- Munawar, E. (2020). Studi Perilaku Masyarakat Aceh Dalam Menghadapi Pandemi Covid-19. *The 2nd Seminar on Population, Family and Human Resources*, 1–9. <https://eprints.latbangdjogja.web.id/147/3/03>. KTI POP - Prosiding.pdf
- Nasichuddin, M. A., Adji, T. B., & Widyawan, W. (2018). Performance Improvement Using CNN for Sentiment Analysis. *IJITEE (International Journal of Information Technology and Electrical Engineering)*, 2(1). <https://doi.org/10.22146/ijitee.36642>
- Naufal, M. F., & Kusuma, S. F. (2021). Pendeteksi Citra Masker Wajah Menggunakan CNN dan Transfer Learning. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(6), 1293. <https://doi.org/10.25126/jtiik.2021865201>
- Nugroho, V. A., Adi, D. P., Wibowo, A. T., Sulistyono, M. T., & Gumelar, A. B. (2021). Klasifikasi Jenis Pemeliharaan dan Perawatan Container Crane menggunakan Algoritma Machine Learning. *Matics*, 13(1), 21–27. <https://doi.org/10.18860/mat.v13i1.11525>
- Nyoman, P., & Putu Kusuma Negara. (2021). Deteksi Masker Pencegahan Covid19 Menggunakan Convolutional Neural Network Berbasis Android. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(3), 576–583. <https://doi.org/10.29207/resti.v5i3.3103>
- Pratiwi, B. P., Handayani, A. S., & Sarjana, S. (2021). Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi Wsn Menggunakan Confusion Matrix. *Jurnal Informatika Upgris*, 6(2), 66–75. <https://doi.org/10.26877/jiu.v6i2.6552>
- Putra, A. W. P. (2018). Aplikasi Deteksi Jatuh Pada Ponsel Pintar Android (Voice Alert) Menggunakan Sensor Akselerometer. *Fakultas Keperawatan: Universitas Islam Indonesia*, 1–47.
- Raden Muhammad Ali, S., Resty Varia, T., & Djazuly, C. (2020). Analisis Faktor Risiko Kematian Dengan Penyakit Komorbid Covid-19. *Jurnal Keperawatan Silampari*, 4(1), 1689–1699.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. <https://doi.org/10.1037/h0042519>
- Rumelhart, D. E., & McClelland, J. L. (1985). Learning Internal Representations by Error Propagation. *Defense Technical Information Center Technical Report*.

- Sembiring, Z., Safitri, H. R., Studi, P., Komputer, T., Medan, N., Studi, P., Informasi, S., Medan, N., & Wajah, D. (2020). Deteksi Wajah Bermasker Berbasis Tensorflow-Keras. *Jikstra*, 02(02), 45–55.
- Shaheed, K., Mao, A., Qureshi, I., Abbas, Q., Kumar, M., & Zhang, X. (2022). Finger-vein presentation attack detection using depthwise separable convolution neural network. *Expert Systems with Applications*, 198(March), 116786. <https://doi.org/10.1016/j.eswa.2022.116786>
- Sultana, S., Iqbal, M. Z., Selim, M. R., Rashid, M. M., & Rahman, M. S. (2022). Bangla Speech Emotion Recognition and Cross-Lingual Study Using Deep CNN and BLSTM Networks. *IEEE Access*, 10, 564–578. <https://doi.org/10.1109/ACCESS.2021.3136251>
- Sutanto, M. (2021). *Perlindungan Hukum Atas Ciptaan Yang Dihasilkan Oleh Kecerdasan Buatan*.
- Thariq, A., & Bakti, R. Y. (2021). Sistem Deteksi Masker dengan Metode Haar Cascade pada Era New Normal COVID-19. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, 9(2), 241. <https://doi.org/10.26418/justin.v9i2.44309>
- TRIATMAJA, C. (2018). BAB I latar belakang Sandwich. *Unika.Ac.Id*, 2010, 12. [http://repository.unika.ac.id/18544/2/12.30.0161_CAKRA TRIATMAJA %282.92%29..pdf](http://repository.unika.ac.id/18544/2/12.30.0161_CAKRA_TRIATMAJA%282.92%29..pdf) BAB I.pdf
- Yu, T., Li, X., Cai, Y., Sun, M., & Li, P. (2021). S²-MLPv2: Improved Spatial-Shift MLP Architecture for Vision. *ArXiv Preprint ArXiv:2108.01072*, 297–306. <http://arxiv.org/abs/2108.01072>



LAMPIRAN

IMPORT LIBRARY

```
# Dataset
from google.colab import drive
# Preprocessing
from keras.preprocessing.image import ImageDataGenerator
# Model
import tensorflow as tf
from tensorflow.keras.optimizers import Adam
from keras import layers
from keras.callbacks import ReduceLRonPlateau, EarlyStopping
import os
# Evaluate
from google.colab import files
from tensorflow.keras.preprocessing import image
from sklearn.metrics import precision_recall_fscore_support, accuracy_score
import numpy as np
import seaborn as sns
from sklearn.metrics import confusion_matrix, classification_report
import pandas as pd
import matplotlib.pyplot as plt
from keras import models
import cv2
from keras import models
```

Prepare Dataset

```
# connect to google drive
drive.mount('/content/gdrive') / Connect ke Drive

#add the path general where the classes subpath are allocated
path = 'gdrive/MyDrive/sistem/dataset' / Menunjukkan Direktori Dataset

classes=["fake mask", "no mask", "with mask"] / Class Yang ada Di Drive
num_classes = len(classes)
batch_size = 16 (Batch Size Dataset Di jadikan 16)

***No Augmentation on the Test set Images**
datagen = ImageDataGenerator(rescale=1./255,
                             validation_split=0.1) #9:1

#loading the images to training set
train_gen = datagen.flow_from_directory(directory=path,
```

```

        target_size=(299, 299),
        class_mode='categorical',
subset='training',
        shuffle=True, classes=classs
, batch_size=batch_size,
        color_mode="rgb")
#loading the images to test set
test_gen = datagen.flow_from_directory(directory=path,
        target_size=(299, 299),
        class_mode='categorical',
        subset='validation',
        shuffle=False, classes=classes
        batch_size=batch_size,
        color_mode="rgb")

```

Model

```

learning_rate = 1e-5
epochs = 10

Xception = tf.keras.applications.Xception(input_shape = (299, 299, 3),
        weights = 'imagenet',
        include_top = False)
Xception.trainable = True

model = tf.keras.Sequential([
        Xception,
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(4, activation = "softmax")],
        name = "Xception_Categorical_Classification")

model.compile(optimizer = Adam(learning_rate = learning_rate, decay = lea
rning_rate / epochs),
        loss = "categorical_crossentropy",
        metrics = ["accuracy"])

```

Training & Testing

```

# fit model
history=model.fit(train_gen,

```

```

        validation_data= test_gen,

        steps_per_epoch=100,

        batch_size=batch_size,

        epochs= epochs)

#evaluate model

model.evaluate(test_gen)

```

Output

```

import matplotlib.pyplot as plt
%matplotlib inline

acc=history.history['accuracy']
val_acc=history.history['val_accuracy']
loss=history.history['loss']
val_loss=history.history['val_loss']

epochs=range(1,len(acc)+1)
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

```

Confusion Matrics

```

# confusion matrix
# transform the predictions into array such as [0,0,1,2...]
predictions = np.array(list(map(lambda x: np.argmax(x), y_pred)))
#Retrieve the True classes of the test set

```

```

y_true=test_gen.classes
# Build Confusion Matrix
CMatrix = pd.DataFrame(confusion_matrix(y_true, predictions), columns=classes, index =classes)

plt.figure(figsize=(12, 6))
ax = sns.heatmap(CMatrix, annot = True, fmt = 'g' ,vmin = 0, vmax = 250, cmap = 'Blues')
ax.set_xlabel('Predicted',fontsize = 14,weight = 'bold')
ax.set_xticklabels(ax.get_xticklabels(),rotation =0);

ax.set_ylabel('Actual',fontsize = 14,weight = 'bold')
ax.set_yticklabels(ax.get_yticklabels(),rotation =0);
ax.set_title('Confusion Matrix - Test Set',fontsize = 16,weight = 'bold', pad=20);

```

Recall, Precision, F1 Score, Accuracy

```

#Accuracy Result
acc = accuracy_score(y_true, predictions)
#Precision, Recall and F-Score (For the whole dataset)
results_all = precision_recall_fscore_support(y_true, predictions, average='macro',zero_division = 1)
#Precision, Recall and F-Score (For each Class)
results_class = precision_recall_fscore_support(y_true, predictions, average=None, zero_division = 1)

#Organise the Results into a Dataframe
metric_columns = ['Precision','Recall', 'F-Score','S']
all_df = pd.concat([pd.DataFrame(list(results_class)).T,pd.DataFrame(list(results_all)).T])
all_df.columns = metric_columns
all_df.index = ['COVID', 'Normal', 'Viral Pneumonia','Total']

print('**Overall Results**')
print('Accuracy Result: %.2f%%'%(acc*100)) #Accuracy of the whole Dataset
print('Precision Result: %.2f%%'%(all_df.iloc[3,0]*100))#Precision of the whole Dataset
print('Recall Result: %.2f%%'%(all_df.iloc[3,1]*100))#Recall of the whole Dataset
print('F-Score Result: %.2f%%'%(all_df.iloc[3,2]*100))#F-Score of the whole Dataset

```

Running

```
# test with image
```

```
%matplotlib inline

uploaded = files.upload()

for fn in uploaded.keys():

    # predicting images
    path = fn
    new_img = image.load_img(path, target_size=(299,299))
    img = image.img_to_array(new_img)
    img = img/255
    img = np.expand_dims(img, axis=0)
    prediction = model.predict(img)
    prediction = np.argmax(prediction,axis=1)
    print(prediction)
    print(classes[prediction[0]])
    plt.imshow(new_img)
```



RIWAYAT HIDUP PENULIS

DATA PRIBADI

Nama : Muhammad Ridha
NIM : 180705038
Jurusan : Teknologi Informasi
Fakultas : Sains dan Teknologi
Tempat, Tanggal Lahir : Desa Bueng, 30 Januari 2000
Jenis Kelamin : Laki-Laki
Alamat : Desa Meunasah Bueng, Kec. Ulim,
Kab. Pidie Jaya
Telp/Hp : 082298567524
Email : mrdha01@gmail.com



RIWAYAT PENDIDIKAN

2006-2012 : Min 1 Ulim
2012-2015 : SMP Swasta Ummul Ayman Samalanga
2012-2018 : SMA Negeri Unggul Pidie Jaya
2018-2022 : S1 Teknologi Informasi, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Ar-Raniry Banda Aceh