

**PEMANFAATAN DEEP LEARNING MENGGUNAKAN ARSITEKTUR
GOOGLNET DAN INCEPTION-V3 UNTUK PENGEMBANGAN
SISTEM PRESENSI BERBASIS FACE RECOGNITION**

(SKRIPSI)

Oleh

**ARIB YUSRON HAMDANI
NPM 2017051078**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

**PEMANFAATAN DEEP LEARNING MENGGUNAKAN ARSITEKTUR
GOOGLNET DAN INCEPTION-V3 UNTUK PENGEMBANGAN
SISTEM PRESENSI BERBASIS FACE RECOGNITION**

OLEH

ARIB YUSRON HAMDANI

SKRIPSI

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

ABSTRAK

PEMANFAATAN DEEP LEARNING MENGGUNAKAN ARSITEKTUR GOOGLNET DAN INCEPTION-V3 UNTUK PENGEMBANGAN SISTEM PRESENSI BERBASIS FACE RECOGNITION

Oleh

ARIB YUSRON HAMDANI

Dalam dekade terakhir, terjadi kemajuan yang signifikan dalam teknologi pengenalan biometrik yaitu teknologi *face recognition* yang menjadi salah satu aspek yang paling menarik perhatian dimana teknologi ini mampu untuk mengidentifikasi individu berdasarkan ciri wajah. Meskipun teknologi *face recognition* telah diadopsi oleh sejumlah organisasi sebagai solusi untuk manajemen kehadiran, beberapa tantangan penting masih ada seperti variasi kondisi pencahayaan, variasi ekspresi wajah, serta perubahan sudut pandang yang dapat mempengaruhi akurasi sistem secara signifikan. Oleh karena itu, peningkatan akurasi dan ketahanan terhadap perubahan lingkungan merupakan kunci dalam meningkatkan efektivitas teknologi *face recognition* dalam manajemen kehadiran. Penelitian ini bertujuan untuk mengatasi tantangan-tantangan yang masih ada dalam manajemen kehadiran dengan menggabungkan teknologi *face recognition* berbasis *deep learning* dengan menggunakan 2 perbandingan arsitektur GoogleNet dan Inception-V3 dalam rangka pengembangan sistem presensi berbasis pengenalan wajah. Studi ini menggunakan dataset yang dilakukan augmentasi, dataset diaugmentasi menjadi 4 bagian, dan akan dibedakan menjadi 3 variasi hyperparameter, dengan total iterasi 12 pelatihan. Hasil yang didapatkan menunjukkan bahwa arsitektur Inception-V3 mengungguli performa arsitektur GoogleNet dalam 11 dari 12 percobaan. Dengan hasil performa Inception-V3 mencapai akurasi maksimum sebesar 88%, dibandingkan dengan performa akurasi maksimum pada model GoogleNet sebesar 60%. Temuan ini menunjukkan bahwa arsitektur Inception-V3 lebih cocok untuk pengenalan wajah dalam sistem kehadiran, karena performa akurasi dan konsistensi hasil pelatihan yang unggul. Studi ini memberikan wawasan yang signifikan mengenai pemilihan arsitektur pembelajaran mendalam yang optimal untuk aplikasi berbasis pengenalan wajah.

Kata kunci: *Deep Learning*, GoogleNet, Inception-V3, Pengenalan Wajah, Sistem Presensi, Perbandingan Model, Augmentasi Datasets, Machine Learning.

ABSTRACT

UTILIZATION OF DEEP LEARNING USING GOOGLNET ARCHITECTURE AND INCEPTION-V3 FOR DEVELOPMENT OF A FACE RECOGNITION BASED PRESENCE SYSTEM

By

ARIB YUSRON HAMDANI

In the last decade, there has been significant progress in biometric recognition technology, namely facial recognition technology, which has become one of the aspects that has attracted the most attention, where this technology is able to identify individuals based on facial characteristics. Although facial recognition technology has been adopted by a number of organizations as a solution for attendance management, several important challenges still exist such as variations in lighting conditions, variations in facial expressions, and changes in viewing angles that can significantly affect system accuracy. Therefore, increasing accuracy and resilience to environmental changes is key to increasing the effectiveness of facial recognition technology in attendance management. This research aims to overcome the challenges that still exist in attendance management by combining deep learning-based facial recognition technology using 2 comparisons of the GoogleNet and Inception-V3 architectures in the context of developing a facial recognition-based attendance system. This study uses a dataset that has been augmented, the dataset is augmented into 4 parts, and will be divided into 3 hyperparameter variations, with a total of 12 training iterations. The results obtained show that the Inception-V3 architecture outperforms the GoogleNet architecture in 11 out of 12 experiments. With the performance results, Inception-V3 reaches a maximum accuracy of 88%, compared to the maximum accuracy performance on the GoogleNet model of 60%. These findings indicate that the Inception-V3 architecture is more suitable for face recognition in presence systems, due to its superior accuracy performance and consistency of training results. This study provides significant insights into the selection of optimal deep learning architectures for facial recognition-based applications.

Keywords: Deep Learning, GoogleNet, Inception-V3, Face Recognition, Presence Systems, Model Comparison, Datasets Augmentation, Machine Learning.

Judul Skripsi : **PEMANFAATAN DEEP LEARNING
MENGUNAKAN ARSITEKTUR
GOOGLENET DAN INCEPTION-V3
UNTUK PENGEMBANGAN SISTEM
PRESENSI BERBASIS FACE
RECOGNITION**

Nama Mahasiswa : **Arib Ysron Hamdani**

Nomor Pokok Mahasiswa : **2017051078**

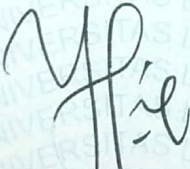
Program Studi : **S1 Ilmu Komputer**

Jurusan : **Ilmu Komputer**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing


Anie Rose Irawati, S.T., M.Cs.
NIP. 19791031 200604 2 002


Yohana Tri Utami, S.Kom., M.Kom.
NIP. 19900110 201903 2 010

2. Ketua Jurusan Ilmu Komputer


Dwi Sakethi, S.Si., M.Kom.
NIP. 19680611 1998021 001

MENGESAHKAN

1. Tim Penguji

Ketua : **Anie Rose Irawati, S.T., M.Cs.**

Sekretaris : **Yohana Tri Utami, S.Kom., M.Kom.**

Penguji Utama : **Didik Kurniawan S.Si., M.T.**

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi : **10 Juni 2024**

PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul **“Pemanfaatan Deep Learning Menggunakan Arsitektur Googlenet dan Inception-V3 Untuk Pengembangan Sistem Presensi Berbasis Face Recognition”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil jiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 10 Juni 2024



Arib Yusron Hamdani

NPM. 2017051078

RIWAYAT HIDUP



Penulis lahir di Bandar Lampung pada tanggal 16 April 2002 sebagai anak kedua dari tiga bersaudara dari pasangan Bapak Damanuri S.Pd. dan Ibu Indun Iswati. Penulis telah selesai menepuh Pendidikan tingkat Sekolah Dasar (SD) di SDN 2 Beringin Raya pada tahun 2014. Kemudian Sekolah Menengah Pertama (SMP) di SMPN 14 Bandar Lampung pada tahun 2017 dan Sekolah Menengah Atas (SMA) di SMAN 9 Bandar Lampung pada tahun 2020. Dan pada tahun 2020 penulis terdaftar sebagai mahasiswa pada Jurusan Ilmu Komputer FMIPA Universitas Lampung melalui jalur masuk SBMPTN.

Selama menjadi mahasiswa di jurusan Ilmu Komputer, penulis aktif dalam berbagai kegiatan akademik maupun kegiatan organisasi. Kegiatan yang dilakukan adalah sebagai berikut,

1. Menjadi Asisten Dosen untuk mata kuliah Matematika Dasar di Jurusan Ilmu Komputer pada tahun ajaran Ganjil 2021/2022.
2. Menjadi Asisten Dosen untuk mata kuliah Struktur Data dan Algoritma di Jurusan Ilmu Komputer pada tahun ajaran Genap 2021/2022.
3. Menjadi Asisten Dosen untuk mata kuliah Komunikasi Data dan Jaringan Komputer di Jurusan Ilmu Komputer pada tahun ajaran Ganjil 2022/2023.
4. Menjadi Asisten Dosen untuk mata kuliah Pemrograman Mobile Lanjut dan mata kuliah Pemrograman Web Lanjut di Jurusan Ilmu Komputer pada tahun ajaran Ganjil 2023/2024.
5. Mengikuti program Magang Jurusan Ilmu Komputer di PT. Sentra Vidya Utama selama 6 bulan pada tahun 2023.
6. Menjadi panitia dalam kegiatan Seminar Nasional Bersama FMIPA Unila pada tahun 2021 dan 2023.
7. Menjadi Anggota Bidang Internal Himpunan Mahasiswa Ilmu Komputer (HIMAKOM) pada tahun 2021-2022.

PERSEMBAHAN

Segala puji saya haturkan kepada Allah SWT atas segala Rahmat dan Karunia-Nya. Sholawat serta salam senantiasa juga tercurahkan kepada Nabi Muhammad SAW sehingga saya dapat menyelesaikan skripsi ini.

Ku persembahkan skripsi ini kepada:

Kedua Orang Tua dan Saudara

Yang senantiasa memberikan yang terbaik, dan selalu memberikan doa-nya disetiap waktu. Terimakasih karena selalu memberi dukungan dan mendidik dengan penuh kasih sayang.

MOTTO

“You can plan, but Allah is Best Planner”

“Do your best, pray until the end, and leave the rest to Allah”

SANCAWACANA

Puji syukur kehadiran Allah SWT. yang telah memberikan rahmat dan hidayat-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini tepat waktu. Sholawat serta salam penulis sanjungkan kepada Nabi dan Rasul Muhammad SAW yang penulis harapkan syafaatnya di hari akhir kelak.

Skripsi yang berjudul “Pemanfaatan Deep Learning Menggunakan Arsitektur Googlenet dan Inception-V3 Untuk Pengembangan Sistem Presensi Berbasis Face Recognition” ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana jurusan Ilmu Komputer Universitas Lampung.

Dalam proses penelitian ini, sangat banyak orang-orang yang terlibat dalam pelaksanaannya. Oleh sebab itu, penulis mengucapkan terima kasih yang sebesar besarnya kepada:

1. Allah SWT, yang telah memberikan hidayah kesehatan dan kemampuan untuk menyelesaikan tugas akhir ini.
2. Kedua Orang Tua penulis yang tidak henti-hentinya selalu memberikan segala doa dan semangat serta mempercayai penulis untuk menyelesaikan skripsi ini.
3. Kakak dan Adik penulis yang telah kebersamai serta mendoakan penulis untuk tiap kesempatannya.
4. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Ibu Anie Rose Irawati, ST, M.Cs. selaku pembimbing utama, pembimbing akademik, dan sekretaris jurusan Ilmu Komputer yang telah membimbing dan memberikan banyak arahan yang membantu kepada penulis sehingga penelitian yang penulis lakukan dapat menjadi lebih baik.

6. Ibu Yohana Tri Utami, S.Kom., M.Kom. sebagai pembimbing kedua yang telah memberikan bimbingan dalam penulisan skripsi serta arahan yang sangat membantu sehingga penulisan skripsi ini dapat terselesaikan.
7. Bapak Didik Kurniawan, S.Si., MT. sebagai pembahas yang telah banyak memberikan masukan, dan saran yang bermanfaat bagi penulis dalam menyusun serta menyelesaikan skripsi.
8. Bapak, Ibu Dosen dan semua Staf Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan banyak ilmu dan membantu segala urusan administrasi yang penulis butuhkan.
9. Nasywa Nathania Wirawan yang telah membantu dan memberi dukungan selama penyusunan berkas hingga penulisan skripsi terselesaikan.
10. Teman-teman “Nyelow” Adi, Alif, Eryan, Daffa, Azka, dan Irvandra yang telah kebersamai dan membantu dari awal masa perkuliahan.
11. Teman-teman “Alianshe” Kenzo, Thoriq, Rifan, Yadzka, Taufik, dan Renal yang telah menemani dan menghibur selama masa perkuliahan.
12. Teman-teman “OOP” Rifki, Azka, Gatra, Aldo, Feny, Darin, Syha, dan Syifa yang telah berjuang bersama melewati suka dan duka sejak sekolah menengah atas (SMA).
13. Teman-teman “9 Sekawan” Wira, Reza, Mute, Eski, Eko, Nuril, Yuda, dan Nanda yang selalu menghibur penulis dan bermain bersama dikala jenuh.
14. Seluruh pihak yang terlibat dalam proses penelitian skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa masih terdapat kekurangan dalam penulisan skripsi ini. Namun, semoga skripsi ini dapat memberikan keberkah untuk ilmu pengetahuan dan dapat bermanfaat sehingga dapat membantu semua pihak.

Bandar Lampung, 10 Juni 2024

Arib Yusron Hamdani

NPM. 2017051078

DAFTAR ISI

Halaman

DAFTAR ISI.....	i
DAFTAR TABEL	iii
DAFTAR GAMBAR.....	iv
I. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	4
II. TINJAUAN PUSTAKA.....	5
2.1. Penelitian Terdahulu.....	5
2.2. Pengenalan Wajah	6
2.3. <i>Deep Learning</i>	7
2.4. <i>Convolutional Neural Network (CNN)</i>	8
2.5. <i>Inception</i>	13
2.6. <i>Dataset</i>	16
2.7. <i>Confusion Matrix</i>	17
2.8. <i>Web Service</i>	19

III. METODE PENELITIAN	20
3.1. Tempat dan Waktu	20
3.2. Perangkat Penelitian	20
3.3. Tahapan Penelitian	21
IV. HASIL DAN PEMBAHASAN	26
4.1. Pengumpulan Dataset	26
4.2. <i>Preprocessing Dataset</i>	26
4.3. Pemisahan Dataset.....	30
4.4. Pembuatan Model.....	30
4.5. <i>Training Model</i>	31
4.6. Pembuatan <i>Web Service</i>	35
4.7. Pengujian	36
4.8. Pembahasan	42
V. SIMPULAN DAN SARAN	45
5.1. Simpulan.....	45
5.2. Saran	45
DAFTAR PUSTAKA	46

DAFTAR TABEL

Tabel	Halaman
1. Penelitian terdahulu	5
2. <i>Hyperparameter</i> pada <i>convolutional layer</i>	9
3. Perbedaan GoogleNet dan Inception-V3	14
4. Struktur arsitektur GoogleNet	15
5. Struktur arsitektur Inception-V3	15
6. Hasil 4 kemungkinan <i>confusion matrix</i>	17
7. Skema training model yang dilakukan.....	32
8. Hasil perhitungan metrik tiap kelas pada GoogleNet	38
9. Hasil perhitungan metrik tiap kelas pada Inception-V3	40
10. Hasil prediksi pada <i>web service</i>	41
11. Perbandingan model pada skema terbaik model GoogleNet	42
12. Perbandingan model pada skema terbaik model Inception-V3	43
13. Hasil dari semua skema	43

DAFTAR GAMBAR

Gambar	Halaman
1. Gambaran bidang ilmu <i>deep learning</i>	7
2. Ilustrasi <i>convolutional neural network</i>	8
3. Konvolusi pertama (A), Konvolusi kedua (B), Konvolusi terakhir (C).	10
4. Ilustrasi <i>max pooling</i> 2x2 dengan <i>stride</i> 2.	11
5. Ilustrasi <i>fully connected layer</i>	12
6. Ilustrasi <i>inception module</i>	13
7. Ilustrasi <i>layer Inception-V3</i>	14
8. Alur Tahapan Penelitian.	21
9. Hasil <i>Preprocessing</i>	22
10. Contoh hasil <i>cropping</i> dan <i>resize</i>	27
11. Dataset original pada salah satu mahasiswa.	27
12. Hasil <i>convert color</i> pada <i>grayscale</i>	28
13. Hasil 4 <i>augment</i> pada 4 <i>Augment dataset</i>	29
14. Hasil augmentasi random erasing	29
15. Visualisasi arsitektur GoogleNet.	31
16. Visualisasi layer arsitektur Inception-V3.....	31
17. Grafik akurasi saat <i>training</i> model GoogleNet.	33
18. Grafik <i>loss</i> saat <i>training</i> model GoogleNet.	33
19. Grafik akurasi saat <i>training</i> model Inception-V3.	34
20. Grafik <i>loss</i> saat <i>training</i> model Inception-V3.....	34
21. Tampilan <i>web service</i> yang dibuat.....	35
22. Hasil <i>confusion matrix</i> pada prediksi GoogleNet.	37
23. Hasil <i>confusion matrix</i> pada prediksi Inception-V3.	39
24. Tampilan <i>Postman</i> untuk melakukan prediksi.....	41

I. PENDAHULUAN

1.1. Latar Belakang

Manajemen kehadiran atau presensi merupakan komponen utama dalam pengelolaan sumber daya manusia di berbagai organisasi, termasuk perusahaan, institusi pendidikan, dan entitas pemerintah. Efisiensi dalam pengumpulan, pemrosesan, dan pengelolaan data kehadiran memiliki dampak signifikan pada produktivitas, pengukuran kinerja, serta kepatuhan terhadap regulasi yang berkaitan dengan tenaga kerja. Pada umumnya, sistem presensi mengandalkan proses manual yang melibatkan catatan kehadiran yang dapat rentan terhadap kesalahan manusiawi dan manipulasi. Menurut (Retriana Bela, 2013) jika kondisi kerja memenuhi syarat dan mendukung, kinerja pegawai dapat meningkat.

Dalam dekade terakhir, terjadi kemajuan yang signifikan dalam teknologi pengenalan biometrik, dengan teknologi *face recognition* menjadi salah satu aspek yang paling menarik perhatian. Kemampuan teknologi *face recognition* ini untuk mengidentifikasi individu berdasarkan ciri-ciri wajah mereka. Keuntungan utama *face recognition* adalah kemampuannya untuk mengenali individu tanpa memerlukan interaksi fisik atau perangkat khusus, menjadikannya solusi yang sangat relevan dalam lingkungan kerja yang semakin mobilitas dan terdistribusi.

Meskipun teknologi *face recognition* telah diadopsi oleh sejumlah organisasi sebagai solusi untuk manajemen kehadiran, beberapa tantangan penting masih ada. Variasi kondisi pencahayaan, variasi ekspresi wajah, serta perubahan sudut pandang adalah beberapa faktor yang dapat mempengaruhi akurasi sistem secara signifikan. Oleh karena itu,

peningkatan akurasi dan ketahanan terhadap perubahan lingkungan merupakan tantangan kunci dalam meningkatkan efektivitas teknologi *face recognition* dalam manajemen kehadiran.

Berbagai penelitian sebelumnya yang dilakukan oleh (Madhu dkk., 2019) telah mencoba menerapkan teknologi *face recognition* dalam berbagai konteks, termasuk dalam sistem presensi. Salah satu metode yang biasa digunakan yaitu dengan memanfaatkan *deep learning*. *Deep learning* merupakan sub ilmu dari *machine learning* yang terinspirasi dari struktur korteks otak manusia, dengan mengadopsi jaringan syaraf tiruan dengan banyak lapisan tersembunyi (Santoso & Ariyanto, 2018).

Beberapa penelitian seperti (Manashti dkk., 2023) telah memanfaatkan *deep learning* menggunakan salah satu model, *convolutional neural network* (CNN). Penelitian ini menggunakan arsitektur GoogleNet dan Inception-V3 untuk meningkatkan akurasi pengenalan wajah, studi-studi kasus semacam ini memberikan wawasan berharga tentang keberhasilan dan kendala dalam pengembangan solusi manajemen kehadiran berbasis wajah yang lebih efisien. Kedua arsitektur tersebut merupakan arsitektur yang menggunakan *inception*. Googlenet sendiri juga dikenal sebagai Inception-V1 yang merupakan akar perkembangan dari Inception-V3.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengatasi tantangan-tantangan yang masih ada dalam manajemen kehadiran dengan menggabungkan teknologi *face recognition* berbasis *deep learning*. Fokus utama penelitian ini adalah penggunaan arsitektur GoogleNet dan Inception-V3, hal ini karena menurut penelitian terdahulu oleh (Jahandad dkk., 2019) pada arsitektur *Inception* memiliki kelebihan dalam melakukan komputasi dengan efisien. Hal tersebut sesuai untuk meningkatkan akurasi dan efisiensi dalam pengenalan wajah. Dengan demikian, penelitian ini diharapkan akan memberikan kontribusi penting dalam pengembangan sistem manajemen kehadiran yang lebih canggih, akurat, dan efisien dalam menghadapi tantangan lingkungan kerja yang beragam.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan di atas, rumusan masalah penelitian ini adalah bagaimana mengembangkan suatu sistem yang memiliki kapabilitas untuk melakukan pencatatan kehadiran menggunakan *face recognition* dengan akurasi yang optimal.

1.3. Batasan Masalah

Berdasarkan pemaparan latar belakang diatas, batasan masalah penelitian ini yaitu:

1. Penelitian ini berfokus pada pengembangan model *convolutional neural network* implementasi teknologi *face recognition* berbasis *deep learning*
2. Dataset yang digunakan adalah *file* berupa gambar dengan ekstensi *.jpg* atau *.png*
3. Pengembangan model menggunakan arsitektur GoogleNet dan Inception-V3.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menghitung akurasi model dengan arsitektur GoogleNet dan Inception-V3
2. Mengembangkan *Web Services* menggunakan GoogleNet dan Inception-V3 untuk mengenali wajah.

1.5. Manfaat Penelitian

Berdasarkan latar belakang, penelitian ini bermanfaat untuk:

1. Mempermudah dan meningkatkan efektivitas dalam manajemen kehadiran
2. Meningkatkan efisiensi waktu dalam proses presensi.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Beberapa penelitian yang pernah dilakukan sebelumnya yang membahas mengenai sistem presensi berbasis *face recognition* menggunakan CNN dengan arsitektur GoogleNet dan Inception-V3. Pada Tabel 1 merupakan beberapa penelitiannya.

Tabel 1. Penelitian terdahulu

Penulis	Judul	Metode	Hasil
(Santoso & Ariyanto, 2018)	Implementasi <i>Deep Learning</i> Berbasis Keras Untuk Pengenalan Wajah	Penelitian menggunakan <i>convolutional neural network</i> (CNN) menggunakan <i>library keras</i>	Berhasil mendapatkan tingkat kecocokan data menggunakan sebesar 98,57% dari 30 data <i>testing</i>
(Zufar, 2016)	<i>Convolutional Neural Networks For Real-Time Face Recognition</i>	Sistem pengenalan wajah dengan konstruksi model 7 <i>layer</i>	Mendapatkan keberhasilan mengklasifikasi lebih dari 87%

Penulis	Judul	Metode	Hasil
(Sa'idah dkk., 2022)	Modifikasi <i>Convolutional Neural Network</i> Arsitektur GoogLeNet dengan <i>Dull Razor Filtering</i> untuk Klasifikasi Kanker Kulit	Peneliti menggunakan arsitektur GoogleNet untuk melakukan klasifikasi dua jenis kanker kulit	Mendapatkan hasil 97,73% dan <i>loss</i> 1,7063.
(Jahandad dkk., 2019)	<i>Offline Signature Verification Using Deep Learning Convolutional Neural Network (CNN) Architectures Googlenet Inception-V1 And Inception-V3</i>	Melakukan perbandingan antara GoogleNet dan Inception-V3.	Mendapatkan hasil bahwa meskipun Inception-v3 memiliki kompleksitas lebih tinggi, Inception-v1 memberikan kinerja yang lebih baik untuk citra tanda tangan dua dimensi.

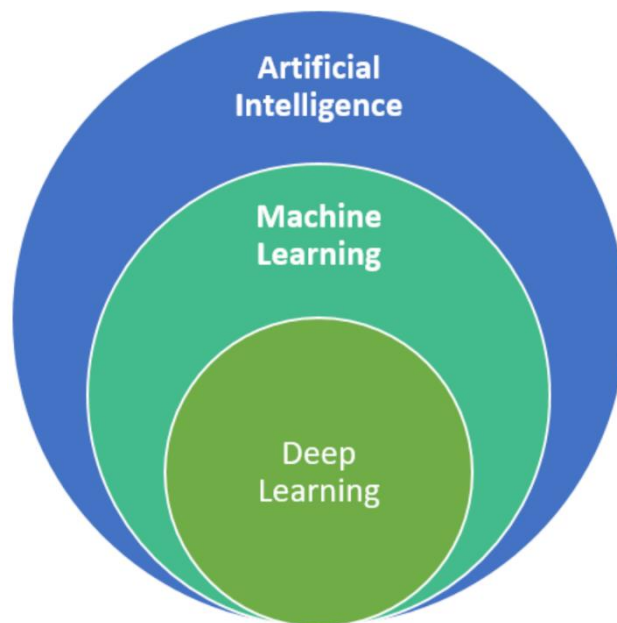
2.2. Pengenalan Wajah

Pengenalan wajah atau yang biasa dikenal dengan *face recognition*. Pengenalan wajah merupakan salah satu jenis biometrik berdasarkan fisiologis. Biometrik dibagi menjadi dua kategori yaitu berdasarkan perilaku seseorang seperti ritme pengetikan, gaya berjalan atau suara dan berdasarkan fisiologis seperti sidik jari, wajah, iris mata (Zufar, 2016). Di antara jenis biometrik fisiologis yang disebutkan sebelumnya, pengenalan wajah masih menjadi bidang penelitian sejak tahun 1960 dengan cakupan luas untuk perbaikan berkelanjutan.

Pengenalan wajah memiliki beberapa tantangan diantaranya kecerahan yang berbeda beda, sudut pengambilan gambar, dan adanya aksesoris pada wajah seperti janggut, kumis, dan kacamata. Dengan demikian pengenalan wajah dapat dikembangkan dengan tujuan mencari efisiensi pengenalan wajah.

2.3. *Deep Learning*

Deep Learning merupakan cabang dari *machine learning* yang merupakan bagian dari *artificial intelligence*. Pada Gambar 1 merupakan gambaran bidang ilmu *deep learning* dalam bentuk diagram.



Gambar 1. Gambaran bidang ilmu *deep learning*.

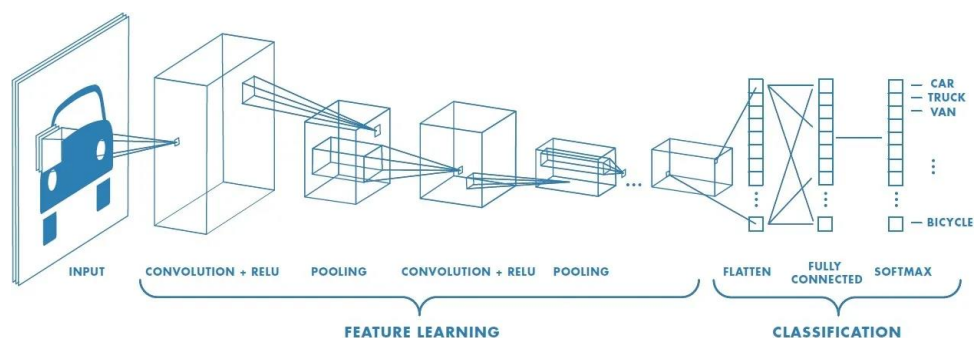
Menurut (Santoso & Ariyanto, 2018) *deep learning* merupakan sub ilmu dari *machine learning* yang terinspirasi dari struktur korteks otak manusia, dengan mengadopsi jaringan syaraf tiruan dengan banyak lapisan tersembunyi.

Seiring berkembangnya ilmu pengetahuan mengenai *deep learning* sekarang banyak *library* pada bahasa pemrograman yang berfokus mempelajari jaringan syaraf tiruan, salah satu *library* yang terkenal seperti

keras. Keras merupakan *library* pada bahasa pemrograman python. Semakin maraknya pengembangan bidang *deep learning* yang bertujuan untuk mengajari mesin untuk mengenali pola yang terdapat pada data yang diberikan, dengan demikian mesin tersebut dapat melakukan perhitungan dan pengenalan pola yang ada pada data.

2.4. *Convolutional Neural Network (CNN)*

Convolutional neural network merupakan salah satu jenis dari *deep learning*. Menurut (Arrofiqoh & Harintaka, 2018) *Convolutional neural network* termasuk dalam jenis *deep learning* karena kedalamannya. *Convolutional neural network* merupakan salah satu *neural network* yang biasa digunakan dalam mengolah data berupa citra, nantinya *convolutional neural network* digunakan untuk mendeteksi dan mengenali objek pada sebuah citra. Menurut (Sa'idah dkk., 2022) salah satu keunggulan *convolutional neural network* yaitu tidak membutuhkan metode ekstraksi ciri tertentu, hal ini membuat tahapan proses klasifikasi citra dapat berjalan lebih cepat. Namun, membutuhkan lebih banyak data. Pada *Convolutional neural network* terdapat beberapa layer yang bertujuan untuk melakukan filter pada setiap prosesnya. Prosesnya disebut dengan proses training. Pada Gambar 2 terdapat ilustrasi dari *convolutional neural network*.



Gambar 2. Ilustrasi *convolutional neural network*.
Sumber: (Santoso & Ariyanto, 2018)

Pada saat *training*, menurut (Santoso & Ariyanto, 2018) pada proses training terdapat 3 tahapan yaitu *Convolutional layer*, *Pooling layer*, dan *Fully connected layer*.

a. Convolutional Layer

Dalam matematika *convolutional* atau konvolusi adalah sebuah operasi matematika dari dua fungsi yang menghasilkan fungsi lainnya. Pada Persamaan 1 merupakan pengolahan citra konvolusi yang digunakan adalah konvolusi diskrit, dan didefinisikan sebagai berikut:

$$h(x, y) = (f * g)(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b)g(x - a, y - b) \quad (1)$$

Fungsi $f(x, y)$ merepresentasikan citra dan $g(x, y)$ merepresentasikan kernel konvolusi atau filter. Pada *convolutional layer* menjadi *layer* pertama yang menerima inputan berupa citra.

Menurut (Zufar, 2016) pada *convolutional layer* memiliki beberapa *hyperparameter* dan parameter, pada Tabel 2 merupakan beberapa *hyperparameter* yang ada pada lapisan *convolutional layer*.

Tabel 2. *Hyperparameter* pada *convolutional layer*

No	Hyperparameter	Keterangan
1	<i>Input Size</i>	Dimensi citra masukan yang akan diproses
2	<i>Kernel</i>	Jumlah layer yang akan dilakukan konvolusi
3	<i>Kernel Size</i>	Ukuran dari kernel berupa matriks
4	<i>Stride</i>	Jumlah pergeseran filter pada proses konvolusi
5	<i>Padding</i>	Jumlah penambahan nilai di sekitar input citra

Ilustrasi pada *convolutional layer* dengan menggunakan *input size* (5x5), *kernel size* (3x3), dan *stride* 1 dengan *output size* (3x3) dapat dilihat pada Gambar 3.

(A)

152	165	142	145	143
137	36	40	44	40
99	30	42	42	44
42	25	37	38	48
33	25	36	36	50

 $\times$

1	0	-1
1	0	-1
1	0	-1

 $=$

108		

(B)

152	165	142	145	143
137	36	40	44	40
99	30	42	42	44
42	25	37	38	48
33	25	36	36	50

 $\times$

1	0	-1
1	0	-1
1	0	-1

 $=$

108	-4	

(C)

152	165	142	145	143
137	36	40	44	40
99	30	42	42	44
42	25	37	38	48
33	25	36	36	50

 $\times$

1	0	-1
1	0	-1
1	0	-1

 $=$

108	-4	-6
30	-22	-7
6	-4	-5

Gambar 3. Konvolusi pertama (A), Konvolusi kedua (B), Konvolusi terakhir (C).

Operasi konvolusi pada *convolutional layer* akan dilakukan sebanyak yang diinginkan sesuai dengan arsitektur yang digunakan. Hasil dari operasi konvolusi ini akan menjadi input untuk tahapan selanjutnya yaitu *pooling layer*.

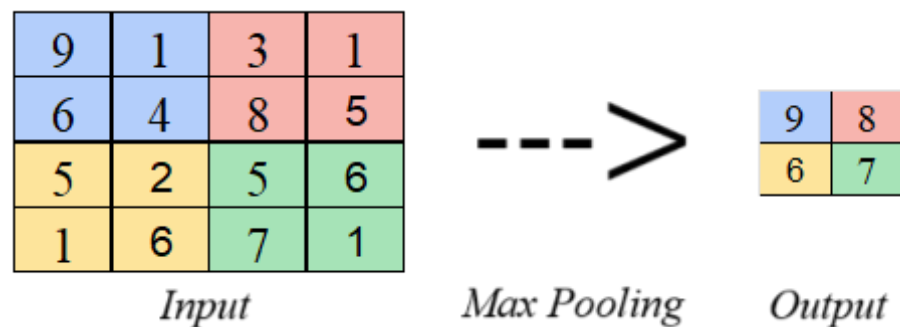
b. *Pooling Layer*

Pooling layer merupakan tahapan yang sangat berpengaruh pada *convolutional neural network* dan dilakukan setelah proses konvolusi. Menurut (Nurcahyati dkk., 2022) proses *pooling layer* merupakan proses perhitungan dengan melakukan pengurangan ukuran matriks, proses ini dilakukan setelah *convolutional layer*.

Pada *pooling layer* ada beberapa metode yang digunakan yaitu *max pooling*, *average pooling*, *standard deviation pooling*, dan *lp pooling*. Dari beberapa metode yang ada, menurut (Pratama dkk., 2023) metode yang paling sering digunakan adalah *max pooling* dan *average pooling*.

Max pooling menurut (Hardi & Sundari, 2022) yaitu membagi *output* dari *convolutional layer* menjadi matriks yang lebih kecil dengan memilih nilai maksimum dari setiap masukan. Sedangkan *average pooling* menurut (Nurkhasanah & Murinto, 2021) digunakan untuk menghitung rata rata dari matriks masukan untuk mendapatkan *output* matriks yang lebih kecil.

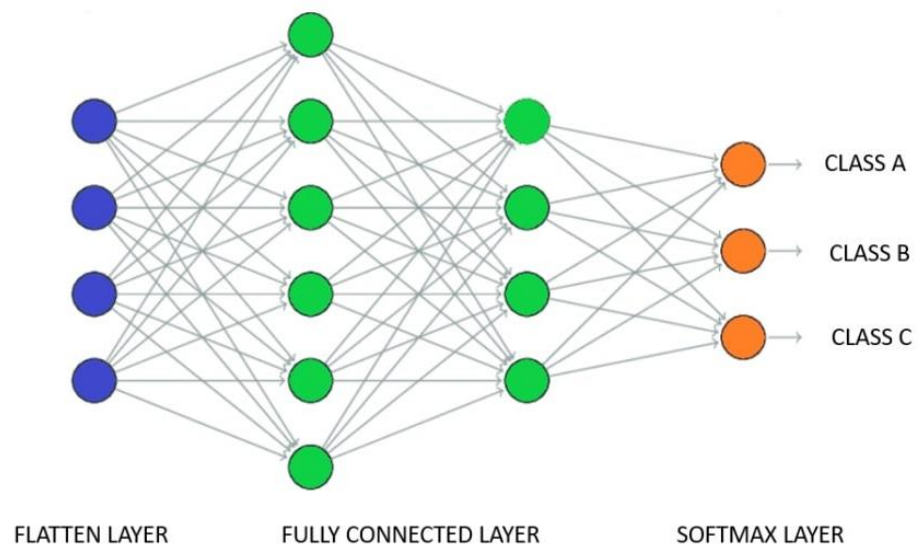
Proses *pooling layer* ini menghasilkan sebuah matriks yang nantinya akan diproses pada tahap selanjutnya yaitu *fully connected layer*. Pada Gambar 4 dapat dilihat ilustrasi *max pooling* yaitu dengan mengambil nilai maksimum dari matriks input dengan filter 2x2 dan *stride* 2.



Gambar 4. Ilustrasi *max pooling* 2x2 dengan *stride* 2.

c. *Fully Connected Layer*

Fully Connected Layer proses yang dilakukan untuk mengolah *input* dari *pooling layer* dengan menggunakan *flatten*. Menurut (Albelwi & Mahmood, 2017) *Fully Connected Layer* berfungsi untuk menyatukan semua node menjadi satu dimensi. Pada Gambar 5 terdapat ilustrasi dari *fully connected layer*.



Gambar 5. Ilustrasi *fully connected layer*.

Pada *fully connected layer* dilakukan beberapa metode yang membuat *layer* terakhir ini dapat dikenali dan paling berkorelasi dengan kelas yang ada. Pada *fully connected layer* biasanya diikuti oleh fungsi aktivasi seperti *relu*, *sigmoid*, dan *softmax* untuk menghasilkan output klasifikasi.

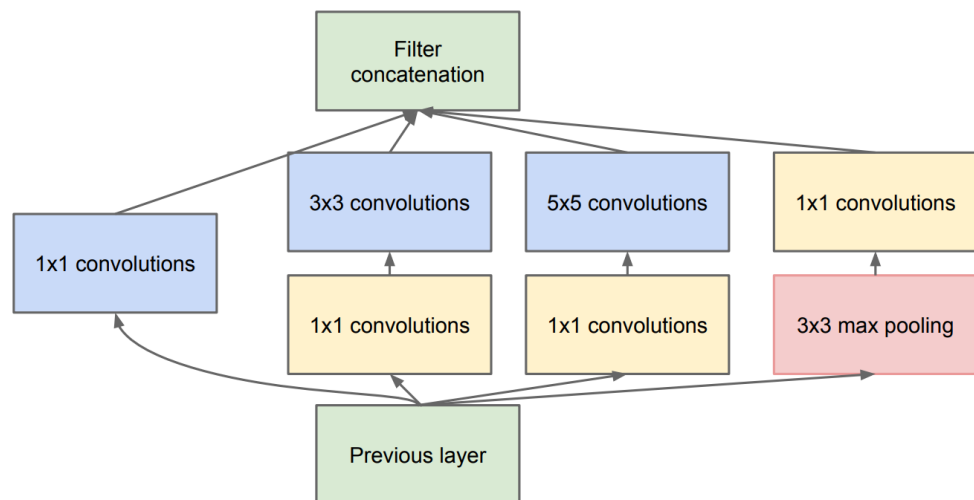
Menurut (Amiral dkk., 2021) Fungsi aktivasi sendiri berfungsi sebagai penghasil mekanisme yang membantu *output neuron* untuk dikenali dan berkorelasi dengan kelas yang ada. Beberapa fungsi aktivasi memiliki ciri masing masing. Pada *relu*, input dengan nilai negatif akan diubah menjadi nilai nol, pada *sigmoid* mengubah nilai input menjadi 0 hingga 1, dan pada *softmax* menghasilkan probabilitas kelas.

Dengan seiringnya pengembangan *convolutional neural network* banyak arsitektur yang dikembangkan dengan tujuan yang berbeda beda. Pada pengembangannya menggunakan variasi yang berbeda beda untuk mendapatkan efisiensi dan memaksimalkan tujuan pengembangan.

2.5. Inception

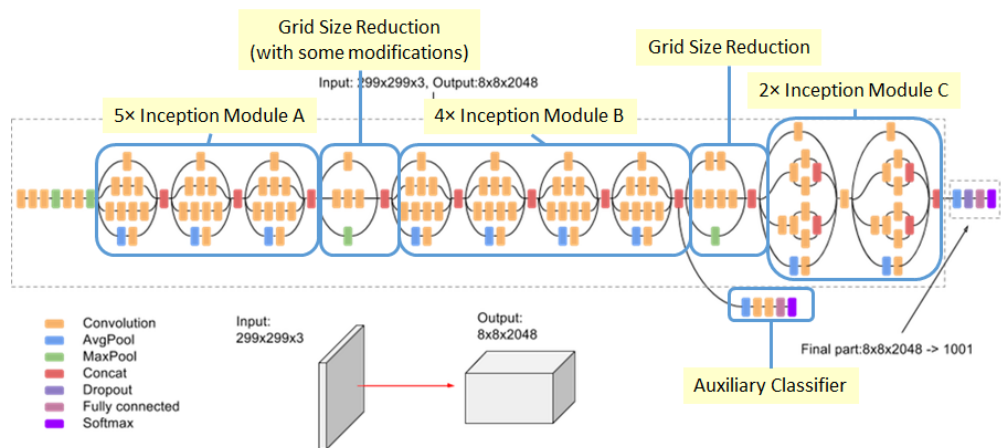
Inception merupakan salah satu jenis arsitektur *convolutional neural network*. Arsitektur *inception* sendiri memiliki ciri yaitu menggunakan *inception module*. *Inception module* merupakan komponen yang disusun secara paralel dengan masing masing ukuran kernel konvolusi yang berbeda. Secara umum ukuran kernel yang digunakan yaitu 1x1, 3x3, 5x5, dan *pooling layer*.

Pengembangan *Inception* sudah dilakukan sejak tahun 2014 oleh (Szegedy dkk., 2014) yang mengembangkan arsitektur dengan menggunakan *inception module naïve version* dengan tambahan pengurangan dimensi dan diberi nama GoogleNet atau dikenal juga sebagai arsitektur Inception-V1. Pada Gambar 6 merupakan ilustrasi dari *inception module*-nya.



Gambar 6. Ilustrasi *inception module*.

Pada tahun 2016 arsitektur ini kembali dikembangkan oleh (Szegedy dkk., 2016). Pengembangan ini diberi nama dengan Inception-V3 dengan mengembangkan dari versi sebelumnya. Sama dengan beberapa versi sebelumnya yaitu dengan menggunakan *inception module*. Dapat dilihat Pada gambar 7 merupakan ilustrasi pada Inception-V3.



Gambar 7. Ilustrasi *layer* Inception-V3.
Sumber: (Pratiwi & Pardede, 2021)

Dengan seiring perkembangannya, arsitektur inception mengembangkan dengan mengubah struktur arsitektur. Pada penelitian oleh (Jahandad dkk., 2019) menjelaskan beberapa perbedaan yang dimiliki oleh GoogleNet dan Inception-V3, pada Tabel 3 merupakan beberapa perbedaan yang terdapat pada GoogleNet dan Inception-V3 dalam bentuk strukturnya.

Tabel 3. Perbedaan GoogleNet dan Inception-V3

<i>Parameter</i>	<i>GoogleNet</i>	<i>Inception-V3</i>
Banyak Operasi	4 Juta Operasi	12 Juta operasi
Convolution Filter Size	1x1,3x3,5x5	5x5 diganti menjadi 2 filter 3x3
Banyak Layer	22 Layer depth	Lebih dari 48 Layer depth
Factorizing Convolution	Tidak digunakan	Filter Factorizing

Struktur model sesuai dengan publikasi ilmiah yang dibuat untuk mendefinisikan struktur dari arsitektur GoogleNet dan Inception-V3. Pada Tabel 4 dan Tabel 5 Merupakan bentuk struktur arsitektur GoogleNet dan Inception-V3

Tabel 4. Struktur arsitektur GoogleNet

Type	Patch Size/Stride	Output Size
Convolution	7×7/2	112×112×64
Max Pool	3×3/2	56×56×64
Convolution	3×3/1	56×56×192
Max Pool	3×3/2	28×28×192
Inception (3a)	-	28×28×256
Inception (3b)	-	28×28×480
Max Pool	3×3/2	14×14×480
Inception (4a)	-	14×14×512
Inception (4b)	-	14×14×512
Inception (4c)	-	14×14×512
Inception (4d)	-	14×14×528
Inception (4e)	-	14×14×832
Max Pool	3×3/2	7×7×832
Inception (5a)	-	7×7×832
Inception (5b)	-	7×7×1024
Avg Pool	7×7/1	1×1×1024
Dropout	-	1×1×1024
Linear	-	1×1×1000
Softmax	-	1×1×1000

Tabel 5. Struktur arsitektur Inception-V3

Type	Patch Size/Stride	Output Size
Convolution	3×3/2	149×149×32
Convolution	3×3/1	147×147×32
Conv Padded	3×3/1	147×147×64
Pool	3×3/2	73×73×64
Convolution	3×3/1	71×71×80
Convolution	3×3/2	35×35×192
Convolution	3×3/1	35×35×288

Type	Patch Size/Stride	Output Size
3 x Inception	-	$17 \times 17 \times 768$
5 x Inception	-	$8 \times 8 \times 1280$
2 x Inception	-	$8 \times 8 \times 2048$
Pool	8×8	$1 \times 1 \times 2048$
Linear	-	$1 \times 1 \times 1000$
Softmax	-	$1 \times 1 \times 1000$

2.6. Dataset

Dataset merupakan kumpulan data yang digunakan untuk melakukan operasi dalam riset. Pada *machine learning dataset* digunakan untuk melakukan *training*, *validation*, dan *testing* model. *Dataset* pada *machine learning* digunakan pada beberapa tahapan yaitu *preprocessing dataset*, *training dataset*, dan *testing dataset*.

a. Preprocessing Dataset

Preprocessing dataset merupakan proses pengkondisian dataset agar dapat sesuai dengan kebutuhan sistem. Menurut (Santoso & Ariyanto, 2018) *dataset* menjadi masukan dan akan menjadi masukan pada masukan lain.

Beberapa metode yang digunakan pada *preprocessing dataset* pada klasifikasi yaitu *resize*, *cropping*, dan *convert color*. Semua metode yang digunakan sesuai dengan kebutuhan.

b. Training Dataset

Training dataset merupakan proses pelatihan dataset yang akan melakukan pengenalan terhadap *dataset* yang dimiliki untuk dapat dikenali dengan menggunakan *convolutional neural network* yang membentuk suatu model.

Metode *training model* ini biasanya tergantung dari penggunaan. Beberapa metode *training* yang dapat digunakan yaitu *convolutional neural network* tadi dengan arsitektur yang bermacam macam seperti GoogleNet, Inception-V3, dan Sebagainya.

c. *Testing Dataset*

Pada proses ini dataset akan digunakan sebagai *testing* model yang telah dibuat. Menurut (Santoso & Ariyanto, 2018) Proses ini menghasilkan hasil akhir dari tingkat keakurasian dataset dengan klasifikasi yang dimiliki model.

2.7. *Confusion Matrix*

Confusion matrix merupakan salah satu metode evaluasi berbentuk tabel yang dapat digunakan untuk melakukan evaluasi hasil *training* model klasifikasi. Metode ini bekerja dengan membandingkan hasil prediksi model dengan dataset yang dimiliki. *Confusion matrix* menggambarkan hasil klasifikasi dengan 4 kemungkinan yaitu, *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* (Uly dkk., 2023). Pada Tabel 6 merupakan penjelasan dari 4 kemungkinan tersebut.

Tabel 6. Hasil 4 kemungkinan *confusion matrix*

Aktual \ Prediksi	Positive	Negative
Positive	True Positive (TP)	<i>False Negative (FN)</i>
Negative	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Dengan menggunakan *confusion matrix*, dapat dilakukan perhitungan beberapa metrik lainnya seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Dengan demikian metrik tersebut dapat dijadikan perbandingan kinerja model.

a. Accuracy

Accuracy dapat dihitung dengan membagi jumlah prediksi benar dengan total jumlah prediksi. Pada persamaan 2 merupakan rumus dari *accuracy*.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

b. Precision

Precision adalah metrik evaluasi yang mengukur seberapa akurat model yang dibuat dalam memprediksi jumlah positif yang benar dari semua prediksi positif. Dengan demikian dapat mengukur seberapa baik model dalam membatasi jumlah data yang salah namun diprediksi benar. Pada persamaan 3 merupakan rumus perhitungan matematis dari *precision*.

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

c. Recall

Recall merupakan metrik evaluasi yang mengukur kemampuan model dalam melakukan prediksi data benar dengan hasil prediksi yang benar. Pada persamaan 4 merupakan rumus perhitungan matematis pada *Recall*.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

d. F1-Score

F1-Score merupakan metrik yang mengukur kinerja model klasifikasi dengan menggunakan *precision* dan *recall*. Perhitungan dilakukan dengan mengambil rata rata antara *precision* dan *recall*, sehingga dapat melihat kekurangan dalam kinerja klasifikasi yang disebabkan ketidakseimbangan *True Positive* (TP) dan *False Negative* (FN). Pada persamaan 5 merupakan rumus perhitungan pada F1-Score.

$$F1_Score = 2 \frac{(recall * precision)}{(recall + precision)} \quad (5)$$

2.8. *Web Service*

Web Service merupakan sebuah layanan yang disediakan untuk berinteraksi dengan sistem lain melalui internet. Menurut (Painem & Soetanto, 2020) *Web Service* memiliki dua metode *simple object access protocol (SOAP)* dan *representational state transfer (REST)*. Menurut (Fakhrun & Gumilang, 2018) perbedaan pada *SOAP* dan *REST* yaitu pada format *response*, pada *SOAP* menghasilkan *response* berupa XML, dan *REST* dapat menghasilkan *response* XML, JSON, dan format *text* lainnya.

III. METODE PENELITIAN

3.1. Tempat dan Waktu

Penelitian dilakukan di Laboratorium Komputasi Dasar Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Penelitian ini dilaksanakan pada semester Ganjil Tahun Akademik 2023/2024 dimulai pada bulan November 2023 hingga bulan Februari 2023.

3.2. Perangkat Penelitian

Pada penelitian ini terdapat beberapa perangkat keras dan perangkat lunak yang digunakan yaitu:

a. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah laptop dengan spesifikasi sebagai berikut:

Merk	: HP
Tipe	: HP Pavilion x360 Convertible
Model	: 14-dh1001tx
CPU	: Intel i3-10110U
GPU	: Nvidia MX130
RAM	: 8GB
Penyimpanan	: SSD 512GB

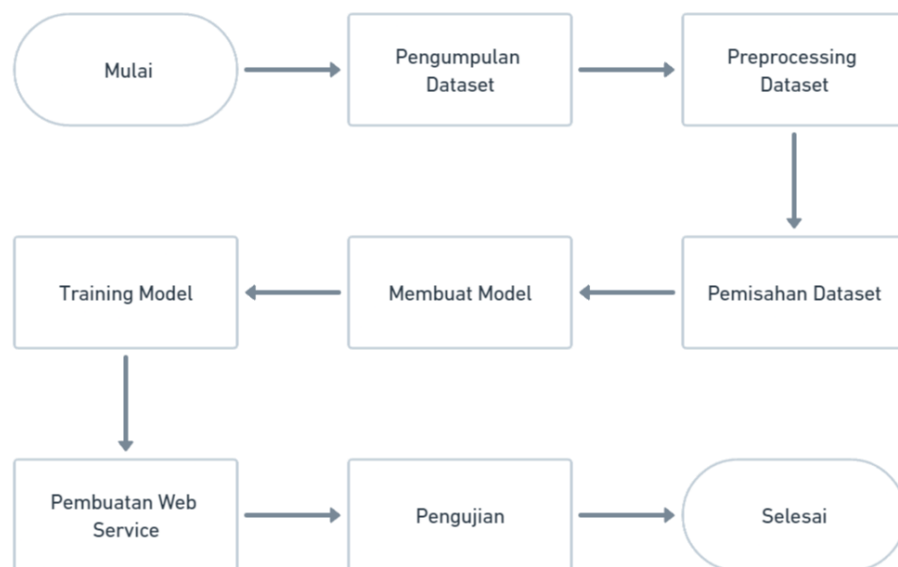
b. Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini yaitu:

1. Windows 11 Home 64-bit digunakan sebagai sistem operasi
2. Visual Studio Code versi 1.84.2 digunakan sebagai kode editor
3. Python 3.10.11 sebagai bahasa pemrograman yang digunakan
4. Postman v10.15 sebagai *software* untuk melakukan pengujian API.

3.3. Tahapan Penelitian

Alur tahapan penelitian ini dapat dilihat pada Gambar 8 yang menyajikan bagaimana alur dari penelitian ini.



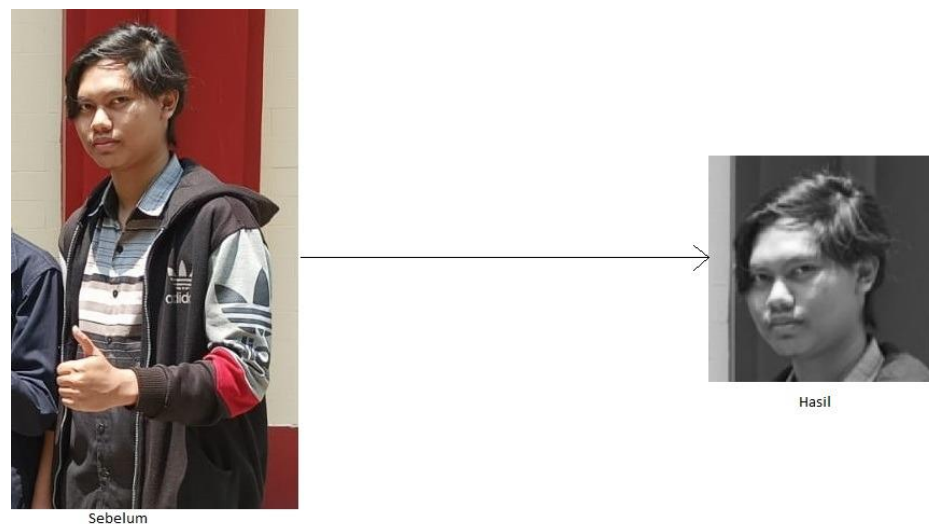
Gambar 8. Alur Tahapan Penelitian.

a. Pengumpulan *Dataset*

Pada tahapan penelitian ini, metode pengumpulan dataset dilakukan dengan mengumpulkan 20 foto dari masing-masing 20 mahasiswa. Proses ini bertujuan untuk memperoleh data yang akan digunakan untuk melakukan *training* model untuk mengenali 20 mahasiswa tersebut.

b. *Preprocessing Dataset*

Dataset yang digunakan masih perlu dilakukan *preprocessing* agar mendapatkan hasil yang efisien. Pada penelitian ini menggunakan *preprocessing* yang sama pada kedua arsitektur yang digunakan, yaitu dengan melakukan *resize*, *cropping*, *convert color*, dan *random erasing*. Pada *convert color*, menggunakan beberapa fungsi untuk melakukan *preprocessing* yang sesuai dengan arsitektur GoogleNet dan Inception-V3 yaitu penggunaan fungsi perubahan warna, saturasi warna, dan ketajaman gambar. Diharapkan penggunaan beberapa *preprocessing* ini dapat meningkatkan hasil performa model yang baik. Pada Gambar 9 merupakan salah satu contoh tampilan sebelum dan sesudah data dilakukan *preprocessing*.



Gambar 9. Hasil *Preprocessing*.

c. *Pemisahan Dataset*

Pada penelitian ini *dataset* akan dibagi menjadi 3 bagian yaitu data latih (*training dataset*), Data validasi (*validation dataset*), dan data uji (*testing dataset*). Data latih merupakan data yang akan digunakan untuk melakukan *training* model, data validasi merupakan data yang nantinya

akan digunakan untuk melakukan validasi data pada proses *training* model, dan data uji merupakan data yang nanti digunakan untuk melakukan pengujian pada saat melakukan proses pengujian.

d. Membuat Model

Model akan dibuat dengan menggunakan metode *convolutional neural network* dengan menggunakan arsitektur GoogleNet dan Inception-V3. Pada GoogleNet model dibangun dengan menggunakan *inception block* sesuai dengan publikasi ilmiah oleh (Szegedy dkk., 2014). Pada Inception-V3 akan dibuat dengan menggunakan *library* yang ada pada *tensorflow* yang mengacu pada publikasi ilmiah oleh (Szegedy dkk., 2016).

e. Training Model

Pada penelitian ini *training* dilakukan dengan menggunakan beberapa *hyperparameter training* modelnya. Dengan menggunakan 2 arsitektur yang berbeda, agar dapat mendapatkan perbedaan yang jelas maka menggunakan *hyperparameter training* model yang sama. Untuk *hyperparameter* yang digunakan nanti akan disesuaikan dengan kebutuhan dan dengan menggunakan metode yang paling efisien.

f. Pembuatan Web Service

Web Service pada penelitian ini akan menggunakan *framework* FastAPI yang merupakan *framework* yang dikembangkan menggunakan Python. Pemilihan FastAPI sebagai *web service* yang akan digunakan karena *framework* ini paling mudah untuk berintegrasi dengan bahasa pemrograman Python. Setelah mendapatkan hasil dari beberapa tahapan sebelumnya, hasil tersebut berupa model yang nantinya akan diimplementasikan menggunakan *framework* FastAPI.

Untuk melakukan prediksi menggunakan model yang sudah dilatih, *web service* menerima masukan sebuah *file* gambar yang dikirim melalui *endpoint* yang dibuat. Gambar akan diproses dengan melakukan *resize* dan akan dilakukan prediksi menggunakan model dengan ekstensi *file* “.h5”. Setelah gambar selesai diprediksi *web service* akan melakukan *response* berisi hasil prediksi dalam format JSON.

g. Pengujian

Pengujian pada penelitian ini bertujuan untuk memastikan model yang dibuat dapat mendapatkan hasil yang efisien dan mendapatkan akurasi yang tinggi. Pada pengujian *web service* bertujuan untuk memastikan dapat berjalan sesuai dengan apa yang sudah direncanakan. Skenario pengujian yang akan dibuat yaitu

1. Skenario Pengujian Model

Pada pengujian model ini skenario yang dibuat adalah dengan melakukan pengujian menggunakan *testing dataset* yang telah dipisah pada tahap sebelumnya. Diharapkan *testing dataset* yang dimiliki dapat diprediksi dengan benar dan memiliki akurasi yang tinggi

Pada proses ini juga model akan dilakukan evaluasi dengan menggunakan *confusion matrix* dan menghitung metrik berupa akurasi, *precision*, *recall*, dan *f1-score*. Metrik tersebut akan dihitung menggunakan hasil prediksi dari *testing dataset* yang berjumlah 40 foto, yaitu 2 foto dari masing masing mahasiswa.

2. Skenario Pengujian Web Service

Pada pengujian *web service* akan dilakukan pengujian dengan melakukan *input* dengan menggunakan *file* yang sudah disiapkan. File *input* yang digunakan adalah foto yang diambil dari *testing*

dataset. *File* tersebut akan diproses dalam *web service*. Harapannya *response* dapat mengembalikan hasil prediksi yang sesuai dengan NPM dari mahasiswa tersebut dalam format JSON.

Jika *input* merupakan *file* selain gambar, *response* akan mengembalikan peringatan bahwa *file* harus berupa gambar atau *return error (415)*. Jika mengakses *endpoint* yang salah akan mengembalikan peringatan *endpoint* tidak ditemukan atau mengembalikan *error (404)*.

Proses pengujian dilakukan dengan menggunakan *web service* dibuat yang memiliki kedua model dengan format model '.h5'. Model yang akan digunakan pada pengujian *web service* ini adalah model yang memiliki kinerja terbaik dari model GoogleNet dan Inception-V3.

V. SIMPULAN DAN SARAN

5.1. Simpulan

Simpulan dari penelitian yang telah dilaksanakan yaitu:

1. Penelitian ini berhasil mengembangkan model CNN dengan arsitektur GoogleNet dan Inception-V3 untuk pengenalan wajah dan berhasil mendapatkan akurasi 60% pada model GoogleNet dan 88% pada model Inception-V3.
2. Dari 12 percobaan yang dilakukan, Inception-V3 memiliki kinerja yang lebih baik dibandingkan dengan GoogleNet yaitu sebanyak 11 percobaan lebih unggul arsitektur Inception-V3 dan 1 percobaan keduanya memiliki kinerja yang seimbang.
3. Penelitian ini berhasil mengembangkan *web service* dengan menggunakan *framework FastAPI* yang dapat melakukan prediksi berupa dengan melakukan *input* berupa gambar dan mengembalikan hasil prediksi dengan format JSON.

5.2. Saran

Berdasarkan hasil dari penelitian ini, maka didapatkan saran untuk melanjutkan penelitian yaitu:

1. Mencoba mengembangkan model GoogleNet dan Inception-V3 dengan mengubah *hyperparameter* lain agar lebih efektif dalam pengenalan wajah.
2. Mengembangkan sistem presensi dengan fitur jadwal dan dengan memanfaatkan *web service* yang telah dibuat.
3. Mengembangkan IoT dalam bentuk kamera yang memanfaatkan model sehingga dapat melakukan pengenalan wajah untuk presensi.

DAFTAR PUSTAKA

- Albelwi & Mahmood. 2017. A framework for designing the architectures of deep Convolutional Neural Networks. *Entropy*. 19
- Amiral, Adrian Surya Saputra, Zufar Majid Suprayogi, Rosa Amalia, Ahnaf Dhiyaul Khoir, Hastuti, & Maulana Syarif. 2021. Pembangkitan Musik Klasik Menggunakan Metode Long-Short Term Memory.
- Arrofiqoh & Harintaka. 2018. Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi. *GEOMATIKA*. 24 : 61.
- Fakhrun & Gumilang. 2018. Rancangan Web Service Dengan Metode REST API Untuk Integrasi Aplikasi Mobile dan Website Pada Bank Sampah.
- Hardi & Sundari. 2022. Pengenalan Telapak Tangan Menggunakan Convolutionall Neural Network (CNN). *Jurnal Rekayasa Perangkat Lunak*. 4
- Jahandad, Sam, Kamardin, Amir Sjarif, & Mohamed. 2019. Offline signature verification using deep learning convolutional Neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3. pp. 475–483.
- Madhu, Adapa, Vatsavaya, & Pothula. 2019. Face Recognition Based Attendance System Using Machine Learning.
- Manashti, Pirnia, Manashty, Ujan, Toews, & Duhaime. 2023. PSDNet: Determination of Particle Size Distributions Using Synthetic Soil Images and Convolutional Neural Networks.
- Nurcahyati, Makhfuddin Akbar, & Zahara. 2022. Klasifikasi Citra Penyakit Pada Daun Jagung Menggunakan Deep Learning Dengan Metode Convolution Neural Network (Cnn). 2 : 43–51.

- Nurkhasanah & Murinto. 2021. Klasifikasi Penyakit Kulit Wajah Menggunakan Metode Convolutional Neural Network Classification of Facial Skin Diseases Using the Method of the Convolutional Neural Network. *SAINTEKS*. 18
- Painem & Soetanto. 2020. Sistem Presensi Pegawai Berbasis Web Service Menggunakan Metode Restfull Dengan Keamanan JWT Dan Algoritma Haversine. *Fountain of Informatics Journal*. 5 : 6.
- Pratama, Rasywir, Fachruddin, Kisbianty, & Irawan. 2023. Eksperimen Layer Pooling menggunakan Standar Deviasi untuk Klasifikasi Dataset Citra Wajah dengan Metode CNN. *Building of Informatics, Technology and Science (BITS)*. 5
- Pratiwi & Pardede. 2021. Image Captioning Menggunakan Metode Inception-V3 dan Transformer.
- Retriana Bela. 2013. Pengaruh Pemanfaatan Teknologi Informasi Terhadapkinerja Individual Pada Kantor Dinas Kependudukan dan Pencatatan Sipil Kabupaten Wonogiri.
- Sa'idah, Putu, Nugraha Suparta, & Suhartono. 2022. Modifikasi Convolutional Neural Network Arsitektur GoogLeNet dengan Dull Razor Filtering untuk Klasifikasi Kanker Kulit.
- Santoso & Ariyanto. 2018. Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Jurnal Teknik Elektro*. 18
- Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke, & Rabinovich. 2014. Going Deeper with Convolutions.
- Szegedy, Vanhoucke, Ioffe, & Shlens. 2016. Rethinking the Inception Architecture for Computer Vision.
- Uly, Hendry, & Iriani. 2023. CNN-RNN Hybrid Model for Diagnosis of COVID-19 on X-Ray Imagery. *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*. 14 : 57–67.
- Zufar. 2016. Convolutional Neural Networks For Real-Time Face Recognition.