

**KLASIFIKASI JENIS PATAH TULANG PADA
MANUSIA MENGGUNAKAN METODE
*CONVOLUTIONAL NEURAL
NETWORK (CNN)***

Oleh

GUSTI NUZLAN DJAFRI

T3121124

SKRIPSI

**Untuk Memenuhi Salah Satu Syarat Ujian
Guna Memperoleh Gelar Sarjana**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2025**

PERSETUJUAN SKRIPSI

KLASIFIKASI JENIS PATAH TULANG PADA MANUSIA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Oleh

GUSTI NUZLAN DJAFRI

T3121124

SKRIPSI

Untuk memenuhi salah satu syarat ujian

Guna memperoleh gelar sarjana.

Program Studi Teknik Informatika

Ini telah disetujui oleh Tim Pembimbing

Gorontalo, Februari 2025

Pembimbing I


YASIN ARIL MUSTOFA, M.Kom
NIDN.0926088503

Pembimbing II


SUDIRMAN S. PANNA, M.Kom
NIDN.0924038205

PENGESAHAN SKRIPSI

KLASIFIKASI JENIS PATAH TULANG PADA MANUSIA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK(CNN)

Oleh

GUSTI NUZLAN DJAFRI

T3121124

Diperiksa oleh Panitia Ujian Strata Satu (S1)

Universitas Ichsan Gorontalo

1. Ketua Penguji
Zohrahayaty, M.Kom
2. Anggota
Rofiq Harun, M.Kom
3. Anggota
Yusrianto Malago, M.Kom
4. Anggota
Yasin Aril Mustofa, M.Kom
5. Anggota
Sudirman S. Panna, M.Kom



Mengetahui,



Dekan Fakultas Ilmu Komputer
Iryan A. Salihi, M.Kom
NIDN.0928028101



Ketua Program Studi
Sudirman S. Panna, M.Kom
NIDN.0924038205

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa:

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (skripsi saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari tim pembimbingan.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar Pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, Februari 2025

Yang membuat Pernyataan,


Gusti Nuzlan Djafri

ABSTRACT

GUSTI NUZLAN DJAFRI. T3121124. CLASSIFICATION OF HUMAN FRACTURE TYPES USING CONVOLUTIONAL NEURAL NETWORK METHOD

This study aims to develop an automatic classification model for human fracture types using the Convolutional Neural Network (CNN) method, known for its ability to automatically extract images/features without additional extraction techniques. The dataset used in this study consists of 1,521 X-ray images categorized into 10 different fracture types. The model training process involves using 80% of the images for training and 20% for testing. The CNN architecture includes several convolutional and pooling layers, and fully connected layers designed to create an accurate classification model. The results indicate that the CNN model achieves an accuracy rate of 95% in classifying fracture types. Other evaluation metrics, such as precision, recall, and F1-score, show excellent performance for most fracture types. Specifically, classes such as Comminuted Fracture, Spiral Fracture, and Hairline Fracture indicate 100% accuracy. Although some classes, such as Avulsion Fracture, exhibit lower accuracy, the model still demonstrates significant potential for accurately identifying and classifying fracture types. The confusion matrix analysis further confirms that the model is reliable for automatic fracture classification. This study significantly contributes to the AI-based diagnostic system development in the medical field, particularly in assisting medical personnel in quickly and accurately identifying various types of fractures. The findings also pave the way for future studies to improve classification accuracy, especially for classes with lower accuracy, and for applying CNN models to larger datasets or other medical images, such as CT scans or MRIs.

Keywords: classification, human fracture types, Convolutional Neural Network

ABSTRAK

GUSTI NUZLAN DJAFRI. T3121124. KLASIFIKASI JENIS PATAH TULANG PADA MANUSIA MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK*

Penelitian ini bertujuan untuk mengembangkan model klasifikasi otomatis jenis patah tulang pada manusia menggunakan metode Convolutional Neural Network (CNN). CNN dipilih karena kemampuannya untuk mengekstraksi fitur citra secara otomatis tanpa memerlukan teknik ekstraksi tambahan. Dalam penelitian ini, dataset yang digunakan berjumlah 1521 citra X-ray yang dikategorikan ke dalam 10 jenis fraktur berbeda. Proses pelatihan model dilakukan menggunakan data training yang terdiri dari 80% citra dan 20% citra digunakan untuk data testing. Arsitektur CNN yang diterapkan mencakup beberapa lapisan konvolusi, pooling, dan fully connected untuk menghasilkan model klasifikasi yang akurat. Hasil penelitian menunjukkan bahwa model CNN berhasil mencapai tingkat akurasi sebesar 95% dalam mengklasifikasikan jenis patah tulang. Metrik evaluasi lainnya, seperti presisi, recall, dan f1-score, menunjukkan performa yang sangat baik pada sebagian besar jenis fraktur, dengan kelas-kelas seperti "Comminuted fracture", "Spiral fracture", dan "Hairline fracture" mencapai akurasi 100%. Meskipun ada beberapa kelas dengan akurasi lebih rendah, seperti pada "Avulsion fracture", model tetap menunjukkan potensi yang sangat baik dalam mengidentifikasi dan mengklasifikasikan jenis fraktur. Analisis menggunakan confusion matrix lebih lanjut mengonfirmasi bahwa model ini dapat diandalkan untuk klasifikasi patah tulang secara otomatis. Penelitian ini memberikan kontribusi signifikan dalam pengembangan sistem diagnosis otomatis berbasis AI untuk bidang medis, terutama dalam membantu tenaga medis dalam mengidentifikasi berbagai jenis patah tulang dengan cepat dan akurat. Temuan ini juga membuka peluang untuk penelitian lebih lanjut dalam meningkatkan akurasi klasifikasi, terutama pada kelas dengan akurasi rendah, serta penerapan model CNN pada dataset yang lebih besar atau citra medis lainnya seperti CT scan atau MRI.

Kata kunci: klasifikasi, jenis patah tulang manusia, *Convolutional Neural Network*



KATA PENGANTAR

Alhamdulillah penulis panjatkan kehadiran Allah SWT atas berkat Rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan usulan penelitian ini untuk memenuhi salah satu syarat penyusunan skripsi program studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo. Salam dan Taslim kepada junjungan Nabi Muhammad Shallahu ‘Alaihi wa Sallam atas perjuangan beliau yang telah mengantarkan kita dari alam kebodohan ke alam yang penuh ilmu pengetahuan, usulan penelitian ini penulis beri judul **“Klasifikasi Jenis Patah Tulang Pada Manusia Menggunakan Metode *Convolutional Neural Network* (CNN).”**

Penulis menyadari bahwa usulan penelitian ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Oleh karena itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan terima kasih kepada;

1. Bapak Dr. Abdul Gaffar La Tjoke, M.Si., selaku ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Ibu Dr. Juriko Abdussamad, M.Si., selaku rektor Universitas Ichsan Gorontalo;
3. Bapak Irvan A. Salihi, M.Kom., selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, M.Kom., selaku Wakil Dekan I Bidang Akademik;
5. Ibu Irma Surya Kumala Idris, M.Kom., selaku Wakil Dekan II Bidang Kepegawaian, Administrasi Umum, dan Keuangan;
6. Bapak Yasin Aril Mustofa, M.Kom., selaku Pembimbing Utama yang telah membimbing dan membantu penulis dalam penyusunan Usulan Penelitian ini;
7. Bapak Sudirman S. Panna, M.Kom., selaku ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer sekaligus Pembimbing Pendamping yang telah membimbing penulis dalam penyusunan Usulan Penelitian ini;

8. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai ilmu yang berguna dan bermanfaat bagi penulis di masa depan;

9. Dengan penuh rasa syukur, penulis mengucapkan terima kasih kepada kedua orang tua yang selalu menjadi sumber kekuatan, doa, dan dukungan dalam setiap langkah perjalanan akademik penulis. Terima kasih atas segala doa yang tidak pernah putus, kasih sayang yang tulus, serta motivasi yang selalu menguatkan penulis dalam menghadapi setiap tantangan. Setiap pencapaian dalam hidup saya, termasuk terselesaikannya skripsi ini, adalah berkat dukungan serta pengorbanan mereka. Semoga papa dan mama senantiasa diberikan keberkahan usia dan rezeki.;

10. Disetiap perjalanan, selalu ada sosok yang hadir sebagai pelipur lara, penguat semangat, dan teman berbagi cerita. Kepada seluruh teman-teman terbaik penulis mengucapkan banyak terima kasih atas kebersamaan, dukungan, dan tawa yang menemani setiap Langkah dalam proses penyelesaian skripsi ini. Semoga pertemanan ini tetap abadi, dan kesuksesan ini bisa kita rayakan bersama.;

Semoga Allah, SWT melimpahkan balasan atas jasa dari mereka kepada penulis. Dengan demikian penulis mengharapkan Usulan Penelitian ini bisa mendapatkan kritik dan saran yang konstruktif. Selanjutny penulis berharap Usulan Penelitian ini bisa bermanfaat untuk perkembangan dan peningkatan illmu pengetahuan bagi para pembaca.

Gorontalo, 2025

Penulis

DAFTAR ISI

PERSETUJUAN SKRIPSI	ii
PENGESAHAN SKRIPSI	iii
PERNYATAAN SKRIPSI	iv
<i>ABSTRACT</i>	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiii
BAB	I
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II LANDASAN TEORI	5
2.1 Tinjauan Studi.....	5
2.2 Tinjauan Pustaka.....	7
2.2.1 Patah Tulang atau Fraktur Pada Manusia.....	7
2.2.2 Klasifikasi.....	9
2.2.3 X-Ray (Sinar X)	10
2.2.4 <i>Convolution Neural Network (CNN)</i>	11
2.3 Kerangka pikir	18
BAB III METODE PENELITIAN	19
3.1	
Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian..	19
3.2	
Pengumpulan Data.....	19

3.3	
Pemodelan	20
3.3.1 Pre-processing.....	20
3.3.2 Data Training	20
3.3.3 Training CNN	20
3.3.4 Data Testing.....	22
3.3.5 Model.....	22
3.3.6 Hasil Klasifikasi CNN	22
3.3.7 Evaluasi.....	22
BAB IV HASIL PENELITIAN	23
4.1 Hasil Pengumpulan Data	23
4.2 Hasil Pemodelan	23
4.2.1 Pra Pengolahan Data	23
4.2.2 CNN	24
4.3 Evaluasi Model	30
4.3.1 <i>Confusion Matrix</i>	30
BAB V PEMBAHASAN	36
5.1 Objek Penelitian	36
5.2 Pembahasan Model	36
5.2.1 Proses Pelatihan Model	36
5.2.2 <i>Training</i> dan <i>Validation Loss</i>	38
5.2.3 <i>Training</i> dan <i>Validation Accuracy</i>	40
5.2.4 Evaluasi Model	42
BAB VI PENUTUP	46
6.1 Kesimpulan	46
6.2 Saran	46
DAFTAR PUSTAKA	48

DAFTAR GAMBAR

Gambar 2.1:	Arsitektur <i>Convolutional Neural Network</i> (CNN).....	12
Gambar 2.2:	Proses <i>Convolution</i>	12
Gambar 2.3:	Proses <i>Pooling Layer</i>	13
Gambar 2.4:	<i>Max Pooling</i>	14
Gambar 2.5:	<i>Average Pooling</i>	14
Gambar 2.6:	<i>Flatten</i>	15
Gambar 2.7:	<i>Fully Connected Layer</i>	16
Gambar 2.8:	<i>Dropout</i>	16
Gambar 3.1:	Pemodelan.....	20
Gambar 4.1:	Model Arsitektur	25
Gambar 4.2:	Proses Konvolusi Dengan Kernel 3x3	26
Gambar 4.3:	Feature Map	27
Gambar 4.4:	Max Pooling	27
Gambar 4.5:	Matrix dan Vektor	28
Gambar 4.6:	Fully Connected Layer	28
Gambar 5.1:	Jumlah Data di Setiap Kelas	36
Gambar 5.2:	Training Dataset Dengan 10 Epoch	36
Gambar 5.3:	Training dan Validation Loss	38
Gambar 5.4:	Training dan Validation Accuracy	40
Gambar 5.5:	Evaluasi Hasil Kinerja Metode	42
Gambar 5.6:	Confusion Matrix	44

DAFTAR TABEL

Tabel 2.1:	Penelitian Terkait.....	5
Tabel 2.2:	<i>Hyperparameter Pada Convolutional Layer</i>	13
Tabel 2.3:	<i>Confusion Matrix</i>	18
Tabel 4.1:	Dataset Jenis Patah Tulang	24
Tabel 4.2:	Hasil Data Testing	30
Tabel 4.3:	<i>Confusion Matrix</i>	32
Tabel 4.4:	Hasil <i>Confusion Matrix</i>	35

DAFTAR LAMPIRAN

Lampiran 1: Data.....	50
Lampiran 2: Kode Program.....	53
Lampiran 3: Riwayat Hidup Peneliti.....	69

BAB I

PENDAHULUAN

1.1 Latar Belakang

Fraktur atau patah tulang merupakan kondisi dimana susunan tulang pada tubuh terputus, sehingga tidak lagi utuh. Kondisi ini dapat menyebabkan timbulnya rasa nyeri, terjadinya luka, hingga mengakibatkan perubahan bentuk pada tulang. Fraktur bisa terjadi di bagian tulang mana saja, namun lebih sering terjadi pada tulang tangan, pinggul, selangka, lengan atas dan kaki. Adapun risiko terjadinya patah tulang atau fraktur antara lain: kekurangan asupan kalsium dan vitamin D, berjenis kelamin wanita, terutama yang sudah berusia 50 tahun ke atas, kebiasaan merokok, mengonsumsi minuman beralkohol secara berlebihan, pernah mengalami fraktur sebelumnya, *sedentary lifestyle* (gaya hidup yang kurang aktif bergerak), mengonsumsi obat kortikosteroid dalam jangka panjang, penderita penyakit diabetes melitus, rheumatoid arthritis, serta gangguan sistem pencernaan dan kelenjar endokrin [1].

Klasifikasi adalah proses pengelompokan dan pengkategorian sesuatu berdasarkan ciri-ciri tertentu. Fraktur atau patah tulang dapat diklasifikasikan berdasarkan beberapa jenis, yaitu : fraktur dislokasi (*Dislocation Fracture*), Fraktur Oblik (*Oblique Fracture*), Fraktur spiral (*Spiral Fracture*), Fraktur Kominutif (*Comminuted Fracture*), Fraktur Linear (*Longitudinal Fracture*), Fraktur Greenstick (*Greenstick Fracture*), Fraktur Impacted (*Impacted Fracture*), Fraktur Stres (*Hairline Fracture*), Fraktur Avulsi (*Avulsion Fracture*), dan Fraktur Patologis (*Pathological Fracture*).

Metode *Convolutional Neural Network* (CNN) adalah merupakan pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. *Convolutional Neural Network* (CNN) termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi serta dapat memproses gambar menggunakan matriks bobot yang disebut filter atau fitur. Metode *Convolutional Neural Network* (CNN) dapat digunakan untuk mengatasi masalah klasifikasi gambar jenis patah tulang pada manusia sehingga dapat secara otomatis mengekstraksi ciri penting dari setiap citra. Selain itu, metode

Convolutional Neural Network (CNN) juga lebih efisien dibandingkan metode *Neural Network* lainnya terutama untuk memori dan kompleksitas.

Terdapat berbagai macam Metode yang dapat digunakan untuk klasifikasi seperti penelitian yang telah dilakukan oleh Kosrat Dlashad Ahmed (2023) yang menggunakan algoritma *Naive Bayes* (NB), *Decision tree*, *Nearest Neighbors* dan *Random Forest* (RF) dalam *Detection of bone fracture based on machine learning techniques* dengan Tingkat akurasi yang diperoleh 64%, 80%, 83% dan 85% [2]. Adapun penelitian lain dilakukan oleh Xiaofeng Hu, Jianmin Chen, Xiaofei Zheng, Jianmei Li dan Mingwei Zhou (2023) yang menggunakan metode *Naive Bayes* (NB), *Support Vector Machine* (SVM) dan *Random Forest* (RF) dalam *Establishment and application of TSDPSO-SVM model combined with multi-dimensional feature fusion method in the identification of fracture-related infection* dengan tingkat akurasi yang di dapatkan 92%, 92,28% dan 94,06% [3]. Penelitian lainnya juga dilakukan oleh B. Sumanth, G. Ramkumar, Vinay Sivasamy (2022) yang menggunakan metode *Artificial Neural Network* (ANN) dan *Support Vector Mechine* dalam *Bone Fracture Identification using ANN Algorithm and Comparing the Accuracy with SVM Algorithm* dengan menghasilkan tingkat akurasi 85% untuk ANN dan 86% untuk SVM [4]. Adapun penelitian lain dilakukan oleh Torangto. Yos P. Situngkir (2022) dalam Klasifikasi penyakit pada daun kentang menggunakan pengolahan citra dengan metode *Convulotional Neural Network* (CNN). Pengujian dilakukan dengan skenario *confusion matrix* untuk menentukan tingkat akurasi yang dihasilkan oleh model CNN dan berdasarkan dari hasil klasifikasi yang dilakukan, didapatkan tingkat akurasi sebesar 98,2% [5].

Penggunaan metode *Convolutional Neural Network* (CNN) dalam mendeteksi fraktur tulang dapat memberikan solusi yang efektif karena CNN secara otomatis dapat mengekstraksi fitur dari citra medis seperti X-ray, CT scan, atau MRI tanpa memerlukan ekstraksi fitur manual. CNN bekerja dengan memproses citra menggunakan layer-layer konvolusi yang bertugas mengenali pola-pola visual penting yang berkaitan dengan fraktur, seperti garis patah, pergeseran tulang, atau keretakan. Karena kemampuan CNN untuk menangkap detail lokal dari citra dan mengenali pola-pola yang kompleks, model ini sangat cocok untuk

mengidentifikasi perbedaan kecil yang mungkin sulit dideteksi oleh algoritma lain atau bahkan oleh mata manusia [6].

Kontribusi utama dari penelitian ini adalah pengembangan model CNN yang dapat diandalkan untuk klasifikasi jenis tulang manusia, yang diharapkan dapat digunakan untuk mengembangkan system mendiagnosa kondisi tulang dan membantu tenaga medis. Dengan adanya penelitian ini, diharapkan pengklasifikasian tulang dapat dilakukan dengan lebih cepat, akurat dan konsisten, mengurangi beban kerja tenaga medis dan meningkatkan kualitas layanan kesehatan. Selain itu, penelitian ini juga di harapkan dapat mendorong penggunaan teknologi kecerdasan buatan dalam bidang kesehatan, meningkatkan inovasi dan pengembangan teknologi medis.

Berdasarkan uraian latar belakang diatas, penulis tertarik melakukan sebuah penelitian dengan judul **“Klasifikasi Jenis Patah Tulang Pada Manusia dengan Menggunakan Metode *Convolutional Neural Network* (CNN)”**.

1.2 Identifikasi Masalah

Sesuai dengan hasil uraian latar belakang, berdasarkan bentuk atau pola patahannya, patah tulang atau fraktur pada manusia terbagi atas beberapa macam jenis, sehingga perlu untuk diklasifikasikan.

1.3 Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan masalah “Bagaimana hasil kinerja penerapan metode *Convolutional Neural Network* (CNN) untuk menguji dan mengukur tingkat akurasi dalam klasifikasi jenis patah tulang pada manusia”.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah “Untuk mengetahui hasil uji dan tingkat akurasi penerapan metode *Convolutional Neural Network* (CNN) dalam klasifikasi jenis patah tulang pada manusia”.

1.5 Manfaat Penelitian

Dengan adanya penelitian yang dilakukan diharapkan memberikan manfaat secara teoritis dan praktis yakni:

1. Secara teoritis, memberikan masukan bagi perkembangan ilmu pengetahuan dan teknologi, khususnya pada bidang ilmu komputer, yaitu berupa Penerapan metode CNN dalam klasifikasi jenis patah tulang.
2. Secara praktis, penelitian ini diharapkan dapat memberikan wawasan dan pengetahuan tentang klasifikasi jenis patah tulang pada manusia.
3. Secara akademis, diharapkan hasil penelitian dapat dijadikan acuan mahasiswa Universitas Ichsan Gorontalo atau kampus lain, sebagai bahan referensi melakukan penelitian selanjutnya yang berhubungan dengan klasifikasi jenis patah tulang.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Penelitian mengenai penggunaan *Convolutional Neural Network* (CNN) dalam bidang kesehatan telah menunjukkan hasil yang menjanjikan dalam beberapa tahun terakhir. CNN adalah arsitektur deep learning yang dirancang untuk memproses data dalam bentuk grid, seperti gambar dan telah digunakan dengan sukses dalam berbagai aplikasi medis. Adapun penelitian terkait patah tulang dengan penerapan algoritma *Convolutional Neural Network* (CNN) sebagai berikut:

Tabel 2.1 Penelitian Terkait

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
.	Torangto. Yos P. Situngkir	klasifikasi penyakit pada daun kentang mengunaka n pengolahan citra dengan metode <i>Convolution al Neural Network</i> (CNN)	2022	<i>Convolutional Neural Network</i> (CNN)	Pengujian dilakukan dengan skenario <i>confusion matrix</i> untuk menentukan tingkat akurasi yang dihasilkan oleh model CNN. Berdasarkan dari hasil klasifikasi

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
					yang dilakukan, didapatkan Tingkat akurasi sebesar 98,2%.
1	Rinisha Bagaria, Sulochana Wadhwani, A. K. Wadhwani	Bone Fracture Detection in X-ray Images using Convolutional Neural Network (CNN)	2022	<i>Convolutional Neural Network (CNN)</i>	Pengujian dengan model CNN pada penelitian ini, mendapatkan nilai akurasi sebesar 90%. Dimana metode ini, dapat digunakan dengan akurat untuk penelitian.
2	Rika Rokhana, Joko,	Convolutional Neural Network	2019	<i>Convolutional Neural Network (CNN)</i>	Untuk kasus ini, arsitektur

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
	Priambodo, Tita Karlita, I Made Gede Sunarya, Eko Mulyanto Yuniarno, I Ketut Eddy Purnama Mauridhi Hery Purnomo.	(CNN) untuk pendeteksi patah tulang Femur pada Citra Ultrasonik B-Mode			CNN memberikan hasil terbaik dengan Tingkat akurasi 95,3%.

2.2 Tinjauan Pustaka

2.2.1 Patah Tulang atau Fraktur pada Manusia

Patah tulang atau fraktur adalah kondisi tulang patah atau retak sehingga bentuk atau bahkan posisinya berubah. Penyebab utama patah tulang adalah karena adanya tekanan atau benturan yang kuat ke tulang yang melebihi kekuatan dari tulang itu sendiri. Meski umumnya disebabkan oleh benturan yang kuat, patah tulang juga bisa terjadi akibat benturan ringan bila tulang sudah mengalami pengeroposan atau osteoporosis.

Tingkat keparahan patah tulang atau fraktur, tergantung pada seberapa kuat tekanan yang diterima tulang. Akibat perbedaan kekuatan tersebut, jenis atau macam-macam patah tulang bisa beragam, yang kemudian diklasifikasikan berdasarkan bentuk atau pola dari patahan tulang. [7]

Klasifikasi patah tulang atau fraktur berdasarkan bentuk atau pola patahan yang umum terjadi, yakni :

1. Fraktur Dislokasi (*Dislocation Fracture*)

Dislokasi adalah kondisi ketika tulang di sendi bergeser atau keluar dari posisi normalnya. Semua persendian di tubuh dapat mengalami dislokasi, terutama bila terjadi benturan akibat kecelakaan atau terjatuh ketika berolahraga.[8]

2. Fraktur Oblik (*Oblique Fracture*)

Oblique fracture atau fraktur oblik adalah jenis patah tulang yang memiliki pola patahan miring atau diagonal. Kondisi ini biasanya terjadi karena ada tekanan atau pukulan dari sudut tertentu, yaitu atas atau bawah. Tipe fraktur ini tergolong ke dalam patah tulang lengkap atau total. Namun, fraktur oblik ada yang bersifat *displaced* atau bergeser maupun *nondisplaced*.

3. Fraktur Spiral atau *Spiral Fracture*

Fraktur spiral juga merupakan bagian dari jenis patah tulang lengkap atau total. Tipe fraktur ini terjadi Ketika tulang yang patah telah terpelintir atau berputar dari titiknya.

4. Fraktur Kominutif atau *Comminuted Fracture*

Jenis fraktur kominutif juga merupakan bagian dari patah tulang lengkap atau total. Pada fraktur kominutif, tulang pecah menjadi tiga bagian atau lebih dan tidak lagi sejajar. Umumnya, fraktur ini terjadi diarea tulang kecil yang rentan patah, seperti ditangan atau kaki, akibat kecelakaan mobil atau kejadian serius lainnya.

5. Fraktur Linear (*Longitudinal Fracture*)

Bentuk fraktur ini sejajar dengan panjang tulang, baik disepanjang atau hampir sepanjang tulang tersebut. Tipe ini juga merupakan macam patah tulang total atau lengkap.

6. Fraktur Greenstick (*Greenstick Fracture*)

Fraktur greenstick tergolong ke dalam tipe patah tulang parsial atau tidak lengkap. Kondisi ini terjadi ketika tulang yang patah atau retak hanya disatu sisi, sedangkan sisi lainnya tidak, sehingga tulang dapat menekuk atau bengkok.

7. Fraktur Impacted (*Impacted Fracture*)

Fraktur *impacted* juga tergolong kedalam jenis patah tulang tidak lengkap atau parsial. Kondisi ini terjadi ketika tulang yang patah hanya terjadi disatu sisi, tetapi patahan tersebut tidak sampai terlepas. Ujung-ujung patahan tersebut saling mendorong atau menekan satu sama lain sehingga retakan atau patahan tulang tampak menonjol.

8. Fraktur Stres (*Hairline Fracture*)

Fraktur stres atau disebut juga dengan *hairline* (garis rambut) umumnya dialami oleh atlet atau seseorang yang melakukan gerakan terus menerus menekan tulang. Biasanya, jenis patah tulang ini terjadi dikaki atau tungkai kaki. Sesuai namanya tipe fraktur stres berbentuk seperti garis rambut atau hanya terdapat retakan kecil ditulang.

9. Fraktur Avulsi atau *Avulsion Fracture*

Jenis fraktur avulsi terjadi ketika fragmen tulang, yaitu tendon atau ligamen, terlepas dari tulang. Fragmen tulang yang terlepas itu biasanya menarik atau mengambil bagian dari tulang. Fraktur avulsi ini umumnya disebabkan oleh adanya gaya tarikan kuat pada tulang dan biasanya terjadi pada sendi lutut dan bahu.

10. Fraktur Patologis atau *Pathological Fracture*

Fraktur patologis umumnya terjadi karena kondisi medis atau penyakit tertentu yang melemahkan tulang, seperti osteoporosis. Penderita osteoporosis memiliki tulang yang rapuh dan lemah, sehingga lebih mudah patah daripada tulang yang sehat.

2.2.2 Klasifikasi

Kata klasifikasi berasal dari “*classification*” akar kata “*to classify*” (bahasa inggris), yang memiliki arti menggolongkan dan menempatkan benda-benda yang sama disuatu tempat. Pengertian lain dari klasifikasi menurut istilah perpustakaan yaitu mengumpulkan bahan pustaka seperti buku atau lainnya dengan cara sistematis dalam sebuah golongan atau kelas dengan menggunakan ciri khusus untuk mempermudah pengguna dalam menemukan buku tersebut.

Klasifikasi adalah pengelompokan yang sistematis dari sejumlah obyek, gagasan, buku atau benda-benda lain kedalam kelas atau golongan tertentu berdasarkan ciri-ciri yang sama. [10]

2.2.3 X-Ray (Sinar-X)

Sinar-X ditemukan oleh Wilhelm Conrad Rontgen, berkebangsaan Jerman pada tahun 1895. Penemuannya diilhami dari hasil percobaan sebelumnya antara lain dari J.J Thomson mengenai tabung katoda dan Heinrich Hertz tentang fotolistrik. Kedua percobaan tersebut mengamati gerak electron yang keluar dari katoda menuju ke anoda yang berada dalam tabung kaca yang hampa udara. Pemnbangkit sinar-X berupa tabung hampa udara yang didalamnya terdapat filament yang juga sebagai katoda dan terdapat komponen anoda. Jika filamen dipanaskan maka akan keluar elektron dan apabila antara katoda dan anoda diberi beda potensial yang tinggi, elektron akan dipercepat menuju ke anoda. Dengan percepatan elektron tersebut maka akan terjadi tumbukan tak kenyal sempurna antara elektron dengan anoda, akibatnya terjadi pancaran radiasi sinar-X.[11]

Sinar-X adalah radiasi elektromagnetik yang digunakan untuk menghasilkan gambar bagian dalam tubuh manusia. Pencitraan X-ray akan menciptakan gambar bagian dalam tubuh. Gambar-gambar ini akan menunjukkan bagian-bagian tubuh dalam berbagai nuansa hitam dan putih. Pemeriksaan X-ray bertujuan untuk membantu dokter mendiagnosis dan memantau beberapa kondisi tubuh. Misalnya, keluhan seperti infeksi, pembusukan gigi, patah tulang, radang sendi, osteoporosis, ataupun kanker tulang.

X-ray adalah jenis radiasi yang dapat menembus tubuh. Sinar radiasi ini tidak dapat dilihat dengan mata telanjang dan tidak dapat dirasakan. Cara kerja X-ray yang menampakkan perbedaan warna dari putih, abu-abu, hingga hitam.

- Warna hitam menandakan bahwa X-ray mengenai udara.
- Bila mengenai logam atau bagian tubuh yang padat, seperti tulang, sebagian besar partikel X-ray terblokir. Hasil X-ray pun akan tampak berwarna putih.
- Jika sinar X-ray mengenai otot, lemak, dan cairan, hasilnya akan memunculkan warna abu-abu.

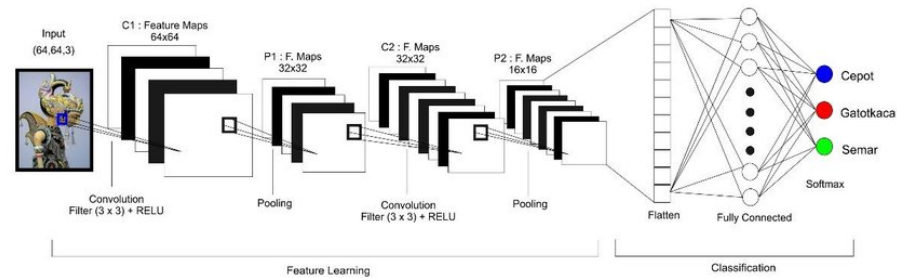
2.2.4 *Convolutional Neural Network (CCN)*

Convolutional Neural Network (CCN) adalah metode klasifikasi yang termasuk dalam kelompok *deep learning* yang menggunakan lapisan konvolusi untuk mengonvolusikan sebuah input dengan sebuah filter.

Proses deteksi dan pengenalan sering menggunakan metode-metode classifier seperti *Hidden Markov Model*, *Support Vector Machine* (SVM), *Artificial Neural Network*, atau metode kombinasi *Adaptive Boost*. Sejalan dengan perkembangan perangkat keras yang mendukung pengolahan *big data* dan komputasi tingkat tinggi, berkembanglah metode Deep Learning (DL). DL adalah sebuah bidang keilmuan baru dalam bidang *Machine Learning* yang menyajikan komposisi transformasi data nonlinear. Salah satu jenis DL adalah *Deep Neural Network* (DNN), dengan kelebihan yaitu dapat memiliki lapisan lebih dari dua. CNN termasuk dalam jenis DNN karena memiliki jaringan yang dalam dan berlapis-lapis.

CNN adalah *supervised-feed forward* DL, yang merupakan pengembangan dari *Multilayer Perceptron* (MLP) dan didesain untuk mengolah data berdimensi besar sehingga banyak diaplikasikan pada data citra/visual. Penggunaan CNN dalam pengenalan dan klasifikasi data mulai berkembang pesat sejak berlangsungnya kompetisi *ImageNet Large-Scale Visual Recognition Challenge* pada tahun 2010. Setelah itu, mulai dikenal beberapa arsitektur CNN seperti *Alexnet* yang memiliki arsitektur delapan lapisan, *GoogLeNet* memiliki 22 lapisan, dan *Resnet* yang memiliki 100 lapisan lebih. [12]

Secara garis besar, CNN tidak terlalu jauh berbeda dengan Neural Network biasanya. CNN terdiri dari neuron yang memiliki weight, bias dan activation function. Arsitektur Convolutional Neural Network dapat dilihat pada gambar 2.1.

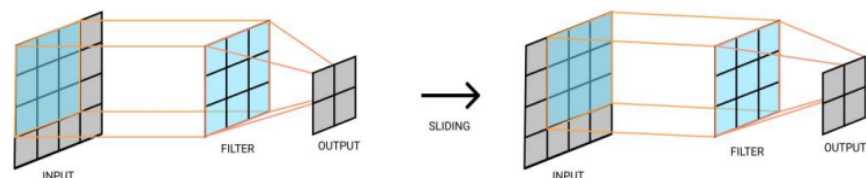


Gambar 2.1 Arsitektur *Convolutional Neural Network* [13]

Dari gambar 2.1 dapat dilihat ada 3 layer utama dalam *Convolutional Neural Network*, yaitu:

1. *Convolutional layer*

Convolutional layer merupakan hasil dari proses convolutional dimana proses tersebut melakukan pengenalan pola, pola yang dimaksud meliputi tepi, bentuk, warna dan tekstur dalam suatu citra. Pola yang dihasilkan tergantung pada filternya. Filternya berupa matriks dimana nilai awal adalah nilai random, kemudian ukuran filter menentukan nilai baris dan kolom.



Gambar 2.2 Proses *Convolution*

Filter akan memproses dot product pada neuron yang berada dalam ruang lingkup filter, kemudian melakukan proses sliding untuk menghitung nilai selanjutnya. Hasil dari proses ini akan menjadi keluaran dari proses konvolusi. Proses ini akan berlanjut sampai sebagai masukan yang diterima. Convolutional layer memiliki hyperparameter dan parameter. Hyperparameter pada layer ini menjadi acuan untuk menentukan jumlah dan ukuran hasil ekstraksi layer, dapat dilihat pada tabel 2.2.

Tabel 2.2 *Hyperparameter* pada *Convolutional layer*

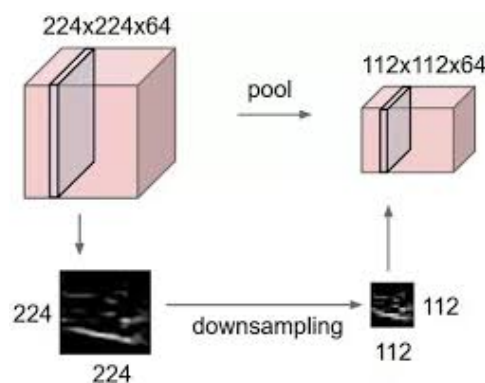
NO	<i>Hyperparameter</i>	Keterangan
1.	Depth	Kedalaman layer atau jumlah layer konvolusi.
2.	Stride	Jumlah pergeseran filter pada proses konvolusi.
3.	Zero-padding	Jumlah penambahan nilai intensitas nol di daerah sekitar input gambar.

2. *Rectified Linear Units (ReLU)*

Aktivasi ReLu (*Rectified Linear Unit*) adalah lapisan aktivasi pada model CNN yang mengaplikasikan fungsi $f(x) = \max(0, x)$ yang artinya fungsi ini menconvert semua nilai negatif menjadi nol terhadap nilai piksel pada input citra. Maka aktivasi ini menormalisasikan nilai sehingga tidak ada nilai yang di bawah nol.

3. *Pooling Layer*

Pooling layer bekerja dengan blok spasial yang bergerak sepanjang ukuran *feature pattern*. *Pooling Layer* juga biasanya berada setelah *convolution layer*. Prinsip *Pooling Layer* terdiri dari filter dengan ukuran dan stride yang akan dipindahkan ke seluruh area *feature map*. *Pooling layer* akan mereduksi ukuran spasial dan jumlah parameter dalam jaringan serta mempercepat komputasi dan mengontrol terjadinya overfitting.

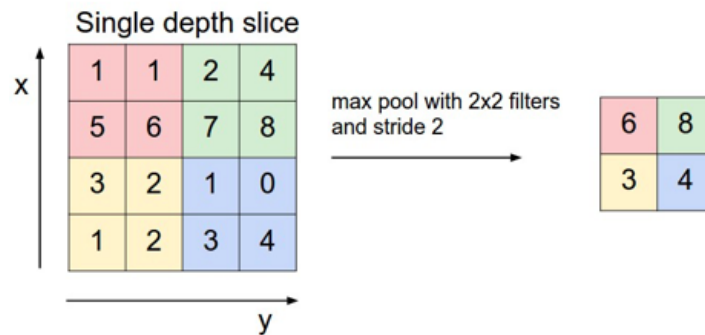
**Gambar 2.3** Proses pooling Layer

Ukuran pergeseran blok pada umumnya adalah ukuran pada dimensi blok (HxH) itu sendiri sehingga tidak ada overlapping seperti pada Convolutional Layer. Pergerakan blok diikuti dengan perhitungan pooling pada masukan pola fitur (u). Pada layer ini dimana *pooling layer* tidak memiliki parameter karena sudah ditentukan sebelumnya (fixed). Ada beberapa macam tipe *pooling layer* antara lain:

a. Max Pooling

Max pooling merupakan metode standar yang digunakan untuk mendapatkan output pola fitur (z) dengan cara mencari nilai paling besar pada blok spasial pada setiap pergerakan.

$$F_{\max(x)} = \max\{x_i\}_{i=0}^N \quad (2-1)$$

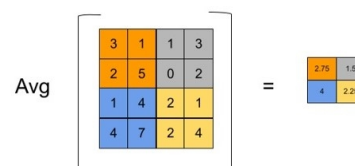


Gambar 2.4 Max Pooling

b. Average Pooling

Average Pooling mengetahui pola fitur keluaran dengan menghitung nilai rata-rata masukan pola fitur pada blok spasial HxH piksel di setiap wilayah pergerakan. Average pooling juga, menyimpan informasi spasial pada pola fitur sebelumnya.

$$F_{Average}(X) = \frac{1}{N} \sum_{i=1}^N x_i \quad (2-2)$$



Gambar 2.5 Average Pooling

c. Lp Pooling

Lp Pooling adalah tipe Pooling Layer generalisasi dari *Max Pooling* dan *average pooling*.

$$F_{T \text{ Max Avg } (X)} = \begin{cases} \text{Max} \{Y_i\}_{i=0}^K, Y_i \geq T \\ \frac{1}{K} \sum_{i=1}^K Y_i, Y_i < T \end{cases} \quad (2-3)$$

Ketika nilai $K = 1$, *Pooling Layer* bekerja sebagai *average pooling* dan jika nilai $K = \infty$ *Pooling Layer* bekerja sebagai *Max Pooling*. Sebagai contoh, input ke *Pooling Layer* dengan ukuran $W1 \times H1 \times D1$. Untuk mendapatkan ukuran output dari pooling layer, dapat menggunakan persamaan 2-4.

$$W2 = \frac{w1-f}{s} + 1 = H2 \quad (2-4)$$

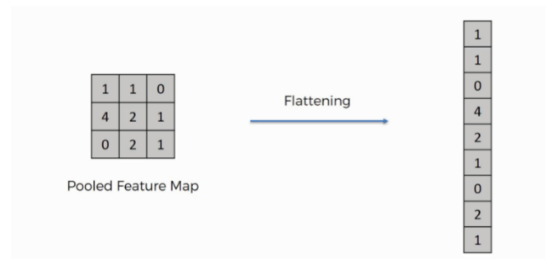
$$D2 = D1$$

*padding biasanya dilakukan pada *pooling layer*, namun mungkin ditemukan.

Dimensi keluaran dari *pooling layer* juga menggunakan rumus yang sama seperti pada *convolutional layer*. *Pooling layer* juga memiliki tujuan untuk mengurangi ukuran *feature map* (downsampling), agar bisa mempercepat proses komputasi, karena lebih sedikit parameter yang diperbarui dan mengatasi overfitting.

4. Flatten

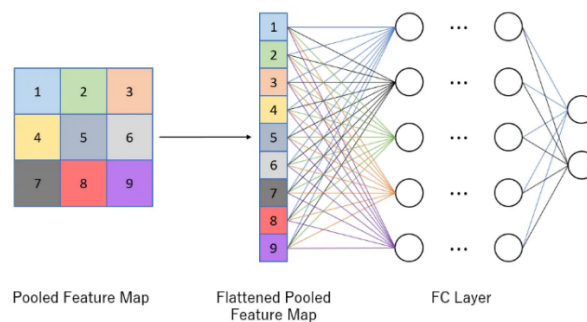
Flattening adalah Dimana operasi matriks bisa berubah menjadi *vector* satu dimensi. Proses ini mengubah feature map yang didapatkan layer sebelumnya menjadi *vector* satu dimensi agar *feature map* tersebut dapat diklasifikasikan dengan *fully-connected layer* dan *sigmoid* [14].



Gambar 2.6 *flatten* [14]

5. *Fully connected layer*

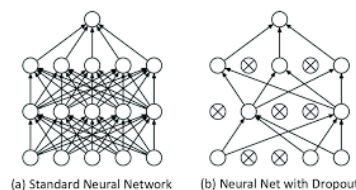
Fully connected layer pengambilan input dari hasil output pooling layer yang berupa feature map dalam bentuk *multidimensional array*. Lapisan-lapisan ini akan dibentuk kembali feature map dan menghasilkan vector sebanyak n - dimensi dimana n adalah jumlah kelas output yang harus dipilih program. Misalnya, jika lapisan terdiri dari 500 neuron, fungsi aktivasi *sigmoid* akan diterapkan untuk mengembalikan daftar probabilitas terbesar untuk setiap 10 label kelas sebagai klasifikasi akhir jaringan. [15]



Gambar 2.7 *Fully Conneted Layer* [16]

6. *Dropout*

Dropout digunakan untuk mengurangi *overfitting* pada model *fully connected layer*.



Gambar 2.8 *Dropout* [18]

Dalam perhitungan layer *neural network*, *dropout* akan memilih secara acak *neuron* yang layak dinonaktifkan sehingga tidak terlalu banyak weight dan neuron yang terlibat. Probabilitas neuron yang dilibatkan dalam perhitungan akan ditetapkan misalnya antara 50% sampai 70%. Neuron yang tidak aktif, ditandai dengan tanda silang [19]

7. *Confusion matrix*

Confusion matrix adalah metode yang digunakan untuk melakukan perhitungan akurasi pada konsep *data mining*. *Confusion matrix* memberikan Keputusan yang diperoleh dari nilai *performance* klasifikasi berdasarkan objek dengan benar atau salah. Metode ini berisikan informasi actual dan prediksi pada sistem klasifikasi.

Tabel 2.3 *Confusion Matrix*

Nilai prediksi	Nilai sebenarnya	
	<i>True</i>	<i>False</i>
<i>True</i>	<i>True Positive</i> (A)	<i>False positive</i> (B)
<i>False</i>	<i>True negative</i> (C)	<i>False negative</i> (D)

Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data dan dapat diperoleh dari nilai akurasi, presisi dan recall [5]

$$\text{Akurasi} = \frac{A+D}{A+B+C+D} * 100\%$$

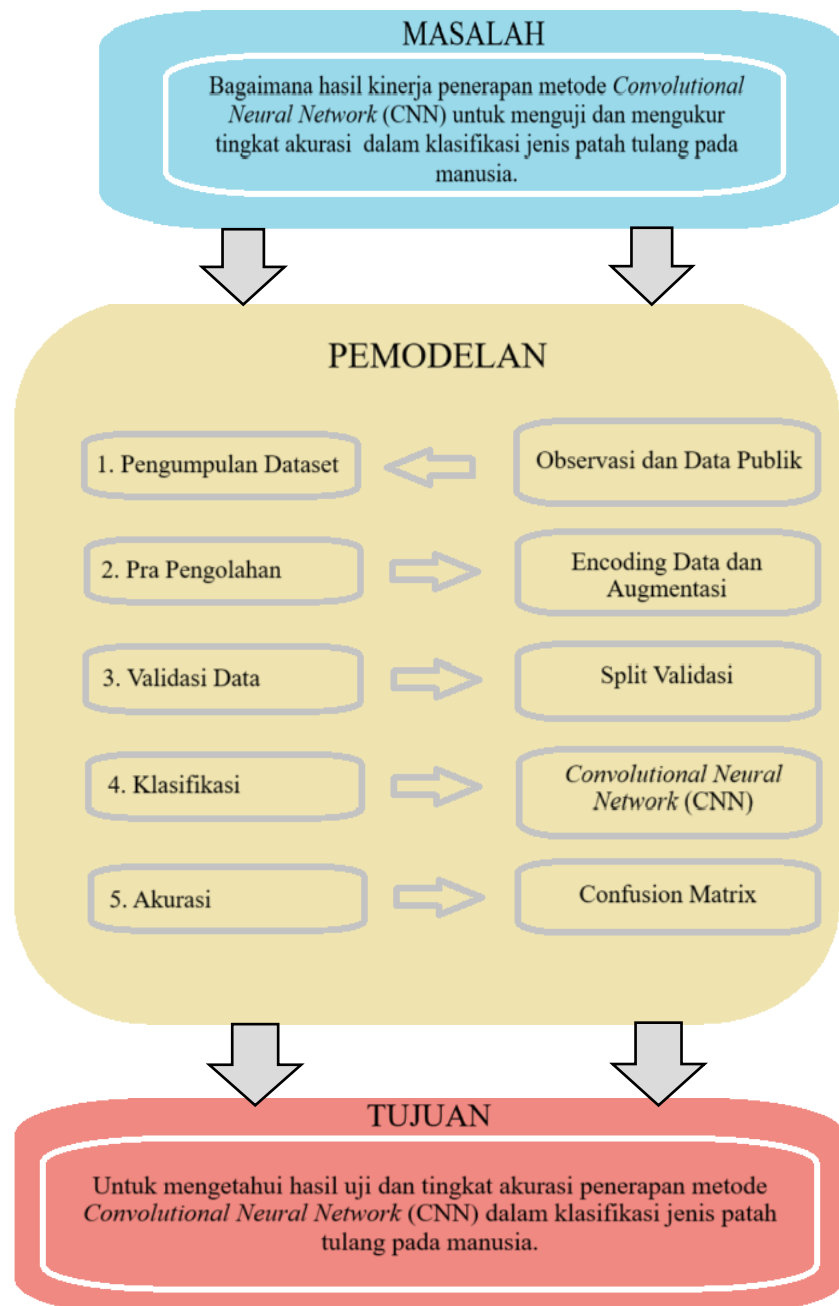
$$\text{Presisi} = \frac{A}{A+C} * 100\%$$

$$\text{Recall} = \frac{A}{D+A} * 100\%$$

Keterangan:

- A (True Positive) → jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif.
- B (False Positive) → jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif.
- C (False Negative) → jumlah data dengan nilai sebenarnya positif dan nilai prediksi negative.
- D (True Negative) → jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif.

2.3 Kerangka Pikir



BAB III

METODE PENELITIAN

3.1 Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian

Dilihat dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dilihat berdasarkan jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif. Dilihat dari perlakuan terhadap data, maka penelitian ini merupakan penelitian konfirmatori.

Penelitian ini menggunakan metode penelitian *eksperiment*. Dengan demikian, jenis penelitian ini adalah penelitian *eksperimental*. Subjek dalam penelitian ini adalah klasifikasi untuk mendapat hasil akhir.

Penelitian ini dimulai dari bulan Mei 2024.

3.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah dataset public dengan nama FRACTURE DATA SET, dataset yang diunduh dari <https://universe.roboflow.com/corso-rphcb/bone-break-classification/dataset/2>, Dataset ini merupakan data yang pada umumnya digunakan untuk *eksperiment* klasifikasi jenis patah tulang pada manusia.

Berikut penjelasan IMAGE DATA SET:

Nama Dataset	: FRACTURE DATASET
Jumlah seluruh data	: 1521 Images
Jumlah data training	: <i>Comminuted</i> fraktur : 134 Images <i>Dislocation</i> fraktur : 137 Images <i>Spiral</i> fraktur : 74 images <i>Avulsion</i> fraktur : 109 images <i>Greenstick</i> fraktur : 106 images <i>Hairline</i> fraktur : 101 images <i>Impacted</i> fraktur : 75 images <i>Longitudinal</i> fraktur : 68 images <i>Oblique</i> fraktur : 69 images <i>Pathological</i> fraktur : 116 images
Jumlah data testing	: <i>Comminuted</i> fraktur : 14 Images

Dislocation fraktur : 19 Images

Spiral fraktur : 12 images

Avulsion fraktur : 14 images

Greenstick fraktur : 16 images

Hairline fraktur : 10 images

Impacted fraktur : 9 images

Longitudinal fraktur : 12 images

Oblique fraktur : 16 images

Pathological fraktur : 18 images

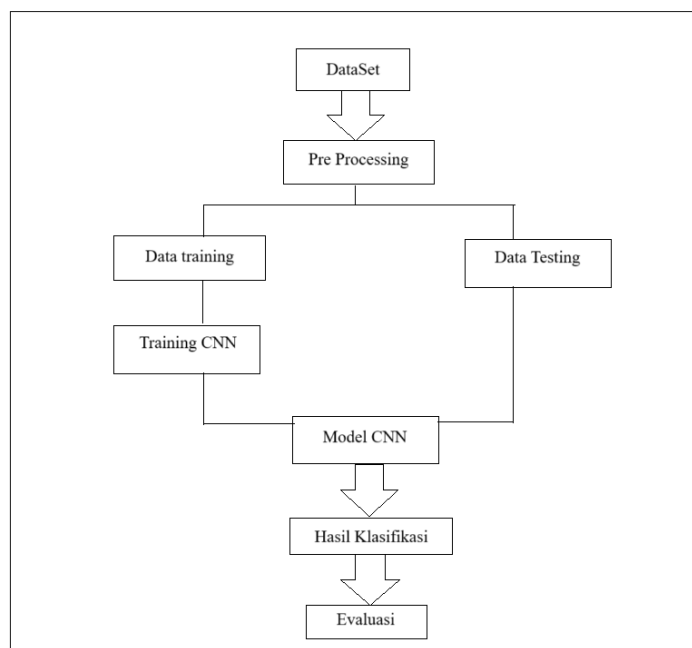
Ukuran gambar : 50 x 50 Pixel

Jenis Fracture : *Comminuted* fraktur, *Dislocation* fraktur, *Spiral* fraktur, *Avulsion* fraktur, *Greenstick* fraktur, *Hairline* fraktur, *Impacted* fraktur, *Oblique* fraktur, dan *Pathological* fraktur.

Format : JPG

Quality : Low quality

3.3 Pemodelan



Gambar 3.1 Pemodelan

3.3.1 Pre-processing

Dari dataset atau data public akan dilakukan transformasi atau encoding data, selanjutnya augmentasi gambar atau memanipulasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda. Agar memperbanyak jumlah data yang akan diolah dan mendapatkan model yang terbaik.

3.3.2 Data Training

Data training berupa data yang telah diseimbangkan jumlahnya untuk diolah menggunakan proses resampling data yang nantinya akan dilatih menggunakan algoritma *Convolutional Neural Network* (CNN), jumlah data training yang digunakan berupa 80% dari sampel patah tulang pada manusia dengan jenis yang berbeda yaitu *Comminuted* fraktur, *Dislocation* fraktur, *Spiral* fraktur, *Avulsion* fraktur, *Greenstick* fraktur, *Hairline* fraktur, *Impacted* fraktur, *Oblique* fraktur dan *Pathological* fraktur. Data training ini berupa hasil undersampling data terhadap patah tulang pada manusia dan setelah itu akan diproses training CNN.

3.3.3 Training CNN

Training menggunakan *Convolutional Neural Network* (CNN) yaitu menjadikan data training menjadi data inputan. Algoritma *Convolutional Neural Network* (CNN) bekerja dengan cara memanfaatkan proses konvolusi dengan menggerakkan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah gambar, selanjutnya komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan, setelah itu nilai matrix akan dinormalisasikan dengan fungsi aktivasi RELU, kemudian akan dilakukan proses *pooling layer* untuk mengecilkan ukuran gambar, 3 (tiga) proses tersebut adalah proses fitur ekstraksi dan selanjutnya akan dilakukan proses klasifikasi, pada proses klasifikasi terdapat proses flatten Dimana proses ini mengubah bentuk matriks ke vektor atau bentuk satu dimensi tetapi tetap mempertahankan informasi spasial dari tiap gambar, setelah itu akan dilakukan proses *fully connected layer* untuk menentukan classnya. Dari proses training ini menghasilkan model yang sudah dilatih dan akan menjadi ketentuan untuk dijadikan pengenalan data testing.

3.3.4 Data Testing

Data testing merupakan data yang telah diseimbangkan jumlahnya yang digunakan untuk menguji data training. Jumlah yang digunakan berupa 20% dari sampel patah tulang pada manusia dengan jenis berbeda. Data testing digunakan untuk mengetahui berapa tingkat keberhasilan untuk melakukan klasifikasi terhadap jenis patah tulang pada manusia.

3.3.5 Model

Merupakan hasil dari proses training dari algoritma Convolutional Neural Network (CNN) menggunakan data training maka didapatkan model yang sudah dilatih dari proses training tersebut, maka model inilah yang akan diimplementasikan untuk mengklasifikasi data testing.

3.3.6 Hasil Klasifikasi CNN

Proses klasifikasi dilakukan untuk mengklasifikasi setiap hasil akhir dari berbagai jenis patah tulang atau fraktur pada manusia, akan menampilkan hasil klasifikasi akhir berupa *Comminuted* fraktur, *Dislocation* fraktur, *Spiral* fraktur, *Avulsion* fraktur, *Greenstick* fraktur, *Hairline* fraktur, *Impacted* fraktur, *Oblique* fraktur, *Pathological* fraktur dan *Longitudinal* fraktur. Kemudian akan dilihat tingkat akurasi setiap jenis fraktur menggunakan Confusion Matrix.

3.3.7 Evaluasi

Pada proses evaluasi bertujuan untuk mengetahui hasil kinerja dari metode yang digunakan, evaluasi dilakukan pada seluruh data testing kemudian target *output* yang dihasilkan akan diubah kedalam *Confussion Matrix* untuk dinilai tingkat akurasinya.

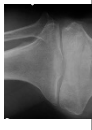
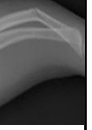
BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Data set yang digunakan dalam penelitian ini adalah dataset *public* dengan nama Patah Tulang dataset yang di unduh dari <https://universe.roboflow.com/corso-rphcb/bone-break-classification/dataset/2>, Dataset ini merupakan data yang pada umumnya digunakan dalam ekperiment untuk mendeteksi patah tulang manusia. Dataset ini sudah berukuran 50 x 50 dengan format gambar jpg dengan warna hitam putih. Dengan menggunakan jumlah data testing 140 image dan data training 989 image.

Tabel 4.1 Data set Jenis Patah Tulang

CF	DF	SF	AF	GF	HF	IF	OF	PF	LF
									
									
									

4.2 Hasil Pemodelan

4.2.1 Pra Pengolahan Data

Pra pengolahan yaitu proses awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahapan ini citra tulang yang sudah ada di konversikan agar di dapatkan data yang sesuai dengan kebutuhan penelitian. Fungsi dari tahapan ini untuk melakukan encoding data atau perubahan data label dari kata menjadi data bilangan atau numerik. Selanjutnya dilakukan augmentasi agar dapat menambah jumlah data menjadi lebih banyak. Pra pengolahan dalam penelitian ini dilakukan dengan beberapa tahap yaitu:

1. Encoding data (*Label Encoder*)

Pada proses ini akan dilakukan transformasi label kata menjadi bentuk numerik agar dapat diolah oleh program.

Comminuted Fracture → 0

Disalocation Fracture → 1

Spiral Fracture → 2

Avulsion Fracture → 3

Greenstick Fracture → 4

Hairline Fracture → 5

Impacted Fracture → 6

Oblique Fracture → 7

Phatologic fracture → 8

Longitudinal Fracture → 9

2. Augmentasi

Tahap augmentasi adalah Teknik manipulasi pada data yang ada tanpa kehilangan inti dari data tersebut. Pada penelitian ini dilakukan Teknik augmentasi terhadap data training dengan mengatur rescale 1./225, rotation range 30, zoom range 0.2, width dan height shift range 0.1, dan horizontal flip beberapa cara tersebut membantu model yang akan disusun serta mempermudah untuk melatih model tersebut.

4.2.2 CNN

Penerapan metode CNN terdapat beberapa tahapan yaitu input data, menentukan pengaturan untuk model yang akan di terapkan sampai pada output dari kelas klasifikasi serta mengatur nilai *epoch* untuk melatih data agar mendapatkan hasil akhir yang sesuai. Model arsitektur CNN yang digunakan pada penelitian ini adalah model sequential dan berikut tahapan penerapan model CNN pada penelitian ini;

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 256, 256, 3)	0
conv2d (Conv2D)	(None, 128, 128, 32)	896
batch_normalization (BatchNormalization)	(None, 128, 128, 32)	128
leaky_re_lu (LeakyReLU)	(None, 128, 128, 32)	0
dropout (Dropout)	(None, 128, 128, 32)	0
conv2d_1 (Conv2D)	(None, 64, 64, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 64, 64, 64)	256
leaky_re_lu_1 (LeakyReLU)	(None, 64, 64, 64)	0
dropout_1 (Dropout)	(None, 64, 64, 64)	0
conv2d_2 (Conv2D)	(None, 32, 32, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 32, 32, 128)	512
leaky_re_lu_2 (LeakyReLU)	(None, 32, 32, 128)	0
dropout_2 (Dropout)	(None, 32, 32, 128)	0
conv2d_3 (Conv2D)	(None, 16, 16, 256)	295,168
batch_normalization_3 (BatchNormalization)	(None, 16, 16, 256)	1,024
leaky_re_lu_3 (LeakyReLU)	(None, 16, 16, 256)	0
dropout_3 (Dropout)	(None, 16, 16, 256)	0
flatten (Flatten)	(None, 65536)	0
dense (Dense)	(None, 100)	6,553,700
batch_normalization_4 (BatchNormalization)	(None, 100)	400
leaky_re_lu_4 (LeakyReLU)	(None, 100)	0
dropout_4 (Dropout)	(None, 100)	0
dense_1 (Dense)	(None, 10)	1,010

Gambar 4.1 Model Arsitektur

Dimulai dengan lapisan input yang menerima data citra dengan ukuran 256x256x3. Lapisan pertama adalah conv2d yang menghasilkan fitur dengan ukuran 128x128x32, diikuti dengan normalisasi batch (batch normalization) yang bertujuan untuk menstabilkan proses pelatihan dan mempercepat konvergensi model. Setelah itu, lapisan leaky ReLU digunakan untuk menambahkan non-linearitas ke dalam jaringan, yang dilanjutkan dengan lapisan dropout untuk mengurangi overfitting.

Arsitektur selanjutnya menggunakan dua lapisan konvolusi tambahan (conv2d), masing-masing diikuti oleh normalisasi batch dan leaky ReLU, di mana lapisan-lapisan ini bertugas mengekstraksi fitur yang lebih mendalam dari citra input. Lapisan dropout kembali diterapkan untuk menghindari pembelajaran yang berlebihan pada data pelatihan. Setelah dua blok konvolusi lebih lanjut, citra yang telah diekstraksi fitur-fiturnya tersebut diratakan melalui lapisan flatten, yang mengubah output menjadi satu dimensi.

Lapisan dense bertindak sebagai lapisan penuh yang menghubungkan fitur yang dipelajari sebelumnya untuk klasifikasi lebih lanjut, diikuti oleh lapisan normalisasi batch yang membantu mempercepat proses pelatihan. Sebagai bagian dari proses klasifikasi, lapisan dense terakhir menghasilkan output dengan 10 neuron, yang merepresentasikan klasifikasi jenis patah tulang yang ditentukan. Model yang digunakan mengandalkan kombinasi konvolusi mendalam, normalisasi, dan dropout untuk meningkatkan kinerja dalam mengidentifikasi dan mengklasifikasikan citra jenis patah tulang dengan akurasi tinggi.

Proses pertama metode CNN yaitu:

1. *Convolutional Layer*

Pada layer ini akan dilakukan perhitungan antara dua matriks. Dimana satu matriks adalah parameter yang dipelajari atau kernel, dan matriks lainnya adalah pixel citra.

Perhitungan manual metode CNN untuk proses konvolusi.

Dari pixel matriks 50 x 50 x 1 diambil contoh dengan ukuran matriks citra = 8 x 8 x 1, kernel = 3 x 3 x 1, S=1, P=1.

Awalnya akan dihitung output volumenya: $v = \frac{W-F+2P}{S} + 1$

W = input size, F = kernel size, P = padding, S = stride, dan V = Volume

Masukkan nilai $V = \frac{8-3+2(1)}{1} + 1 = 8$ jadi output volumenya 8 x 8 x 1

0	0	0	0	0	0	0	0	0	0
0	0	0	96	136	23	25	2	0	0
0	2	113	166	160	21	25	10	0	0
0	26	23	107	175	58	25	24	11	0
0	10	26	22	193	98	21	24	12	0
0	10	26	24	103	160	70	23	23	0
0	11	10	24	40	226	104	43	22	0
0	0	10	25	23	77	134	104	2	0
0	0	1	10	25	21	10	0	0	0
0	0	0	0	0	0	0	0	0	0

0	1	1
0	1	0
0	0	1

kernel

Gambar 4.2 Proses Konvolusi dengan kernel 3 x 3

$$= 0 \times 0 + 0 \times 1 + 0 \times 1 + 0 \times 0 + 0 \times 1 + 0 \times 0 + 0 \times 0 + 2 \times 0 + 113 \times 1$$

$$= 113$$

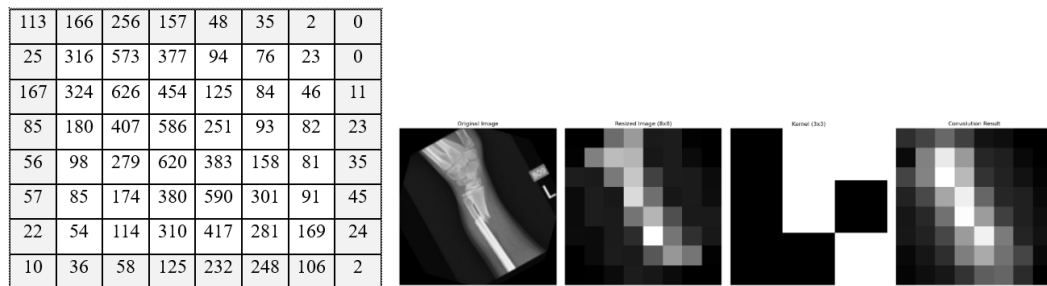
$$= 0 \times 0 + 0 \times 1 + 0 \times 1 + 0 \times 0 + 0 \times 1 + 96 \times 0 + 2 \times 0 + 113 \times 0 + 166 \times 1$$

$$= 166$$

$$= 0 \times 0 + 0 \times 1 + 0 \times 1 + 0 \times 0 + 96 \times 1 + 136 \times 0 + 113 \times 0 + 166 \times 0 + 160 \times 1$$

$$= 256$$

Dan sampai perhitungan terakhir kemudian dari perhitungan tersebut di dapatkan output konvolusi atau feature map dengan ukuran 8 x 8 x 1.

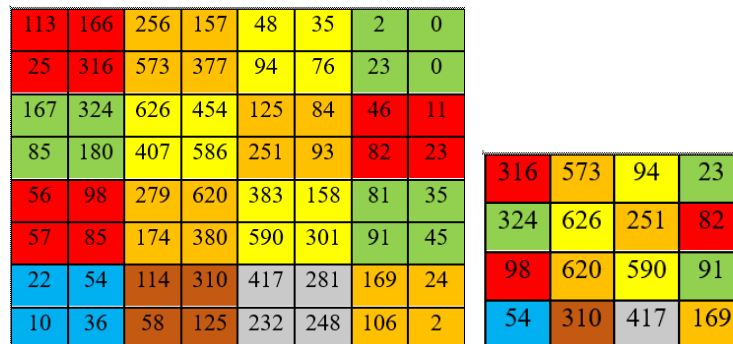


Gambar 4.3 *Feature Map*

Setelah mendapatkan output konvolusi atau feature map maka akan dilakukan Pooling Layer atau mengurangi ukuran dimensi feature map agar bisa mempercepat komputasi. Feature map adalah pemetaan citra atau intisari citra yang dipetakan dari hasil konvolusi.

2. Pooling Layer

Pada tahapan konvolusi didapatkan feature map selanjutnya akan dilakukan pooling layer menggunakan ukuran filter 2 x 2. Maka pada saat menggunakan max pooling tiap filter matriks akan diambil nilai paling besar dan akan dimasukkan ke dalam feature map. Pada matriks dibawah tiap blok warna menandakan tiap pergeseran window dengan matriks 2 x 2 masing – masing window akan diambil nilai terbesarnya, kemudian nilai yang paling besar akan menjadi feature map baru.

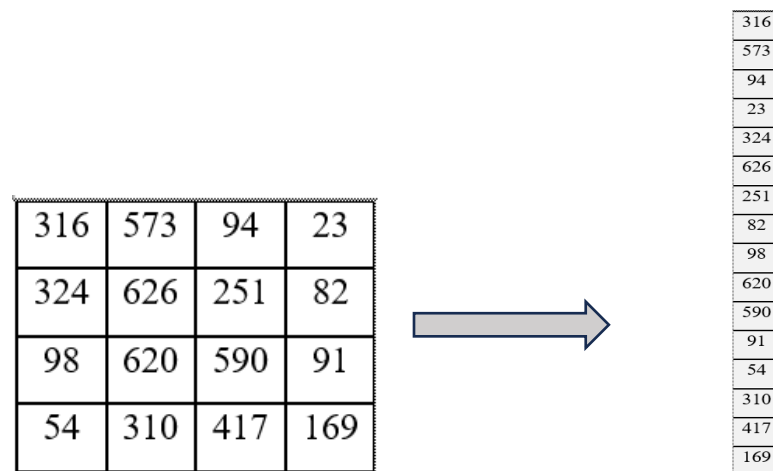


Gambar 4.4 *Max Pooling*

Dari hasil max pooling diatas di dapatkan feature map dengan ukuran 4 x 4. Proses konvolusi dan pooling layer ini akan terus terjadi sesuai dengan arsitektur model CNN yang digunakan. Setelah mendapatkan nilai max pooling maka akan masuk pada proses *flattening*.

3. Flatten

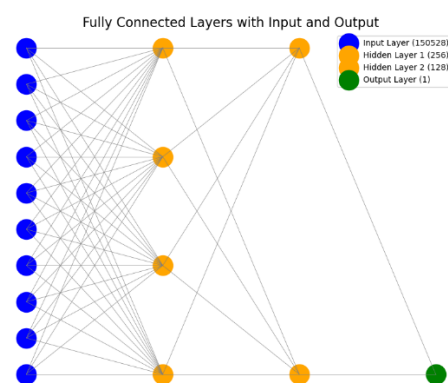
Pada tahap ini feature map akan melalui proses flattening atau mengubah matriks menjadi bentuk vector (satu dimensi).



Gambar 4.5 Matriks dan Vektor

4. Fully Connected Layer

Tahapan ini berguna untuk mengklasifikasikan data masukan. Pada proses training layer ini berguna untuk mencari nilai bobot terbaik.



Gambar 4.6 Fully Connected Layer

1. Inisialisasi semua bobot,

Bobot awal atau $w = 0$, untuk semua inputan.

2. Fungsi aktivasi yang digunakan pada penelitian ini adalah $\sigma = \text{sigmoid}$

3. Laju pembelajaran atau *learning rate*

Pada penelitian ini ditentukan nilai learning ratenya $\alpha = 0,0001$

4. Menghitung persamaan nilai output:

$$f(x) = \sum_{x=i}^n (x_1 w_1 + x_2 w_2 + \dots + x_n w_n)$$

$$\sigma y_j(p) = \frac{e^{x_j}}{\sum_{j=i} e^{x_j}}$$

$$f(x) = \sum_{x=i}^{16} (x_1 w_1 + x_2 w_2 + \dots + x_{16} w_{16})$$

$$f(x) =$$

$$\sum_{x=i}^{16} (316.0 + 573.0 + 94.0 + 23.0 + 324.0 + 626.0 + 251.0 + 82.0 + 98.0 + 620.0 + 590.0 + 91.0 + 54.0 + 310.0 + 417.0 + 169.0)$$

$$f(x) = \sum_{x=i}^n ($$

$$316.0 + 573.0 + 94.0 + 23.0 + 324.0 + 626.0 + 251.0 + 82.0 + 98.0 + 620.0 + 590.0 + 91.0 + 54.0 + 310.0 + 417.0 + 169.0)$$

$$f(x) = \sum_{x=i}^n ($$

$$316.0 + 573.0 + 94.0 + 23.0 + 324.0 + 626.0 + 251.0 + 82.0 + 98.0 + 620.0 + 590.0 + 91.0 + 54.0 + 310.0 + 417.0 + 169.0)$$

$$f(x) = \sum_{x=i}^n ($$

$$316.0 + 573.0 + 94.0 + 23.0 + 324.0 + 626.0 + 251.0 + 82.0 + 98.0 + 620.0 + 590.0 + 91.0 + 54.0 + 310.0 + 417.0 + 169.0)$$

$$f(x) = \sum_{x=i}^n ($$

Sigmoid:

$$\sigma(a) = \frac{1}{1+e^{-a}} = \frac{1}{1+2.71828^{-0}} = 2$$

Nilai sama dengan nilai output lainnya dikarenakan menggunakan nilai $w = 0$

5. Hitung gradien error untuk layer output

$$e_k(p) = y_{dk}(p) - y_k(p)$$

$$\delta_k(p) = y_{dk}(p) * [1 - y_k(p)] * e_k(p)$$

Untuk data pertama, diharapkan nilai $y_{dk} = 1$, sedangkan nilai hasil keluaran yang didapatkan $y_{dk} = 2$ nilai neuron output semuanya sama karena nilai $w = 0$.

$$e_1(1) = y_d - y_1(1) = 1 - 2$$

$$\delta_1(1) = y_1(1) * [1 - y_1(1)] * e_1(1) = 2 * [1 - 2] * -1 = 2$$

6. Menghitung koreksi bobot

Untuk $\Delta w_{11}, \Delta w_{12}, \Delta w_{13}, \Delta w_{14}$:

$$\Delta w_{11} = a * y_1(1) * \delta_1(1) = 0,0001 * 316 * 2 = 0,0632$$

$$\Delta w_{12} = a * y_2(1) * \delta_1(1) = 0,0001 * 573 * 2 = 0,1146$$

$$\Delta w_{13} = a * y_3(1) * \delta_1(1) = 0,0001 * 94 * 2 = 0,0188$$

$$\Delta w_{14} = a * y_4(1) * \delta_1(1) = 0,0001 * 23 * 2 = 0,0046$$

7. Perbaharui bobot

Dengan di perolehnya nilai pengkoreksi bobot maka proses memperbaharui nilai bobot dengan menggunakan persamaan:

$$w_{jk}(p + 1) = w_{jk}(p) + \Delta w_{jk}(p)$$

$$w_{jk}(p + 1) = w_{11}(p) + \Delta w_{11}(p) = 0 + 0,0632 = 0,0632$$

$$w_{jk}(p + 1) = w_{12}(p) + \Delta w_{12}(p) = 0 + 0,1146 = 0,1146$$

Setelah itu untuk iterasi berikutnya, kembali ke langkah 5 dan diulangi sampai mendapatkan hasil yang sesuai. Setelah itu jika didapatkan hasil yang sesuai, lalu di klasifikasikan citra sesuai dengan kelasnya.

4.3 Evaluasi Model

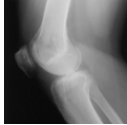
4.3.1 Confusion Matrix

Penelitian ini menggunakan *confusion matrix* sebagai metode. Dalam perhitungan akurasi pada penerapan pengklasifikasian tulang manusia. Evaluasi kinerja klasifikasi tulang manusia didasarkan dari jumlah pengujian objek salah dan benar yang dideteksi dapat dilihat dari tabel berikut:

Tabel 4.2 Hasil Data Testing

Image Testing	Klasifikasi	Aktual	image	klasifikasi	Aktual
---------------	-------------	--------	-------	-------------	--------

	<i>Comminute d Fracture</i>	<i>Comminute d Fracture</i>		<i>Comminute d Fracture</i>	<i>Comminute d Fracture</i>
---	-------------------------------------	-------------------------------------	---	-------------------------------------	-------------------------------------

Image Testing	Klasifikasi	Aktual	Image testing	Klasifikasi	Aktual
	<i>Oblique Fracture</i>	<i>Oblique Fracture</i>		<i>Oblique fracture</i>	<i>Oblique fracture</i>
	<i>Dislocation fracture</i>	<i>Dislocation fracture</i>		<i>Dislocation fracture</i>	<i>Dislocation fracture</i>
	<i>Longitudina l Fracture</i>	<i>Longitudina l Fracture</i>		<i>Longitudina l fracture</i>	<i>Longitudina l fracture</i>
	<i>Greenstick Fracture</i>	<i>Greenstick Fracture</i>		<i>Greenstick fracture</i>	<i>Greenstick fracture</i>
	<i>Pathologic Fracture</i>	<i>Pathologic Fracture</i>		<i>Pathologic fracture</i>	<i>Pathologic fracture</i>
	<i>Impacted Fracture</i>	<i>Impacted Fracture</i>		<i>Impacted fracture</i>	<i>Impacted fracture</i>
	<i>Spiral Fracture</i>	<i>Spiral Fracture</i>		<i>Spiral fracture</i>	<i>Spiral fracture</i>
	<i>Avulsion Fracture</i>	<i>Longitudina l Fracture</i>		<i>Avulsion fracture</i>	<i>Avulsion fracture</i>
	<i>Hairline Fracture</i>	<i>Hairline Fracture</i>		<i>Hairline fracture</i>	<i>Hairline fracture</i>

Tabel 4.3 *Confusion Matrix*

	CF	DF	SF	AF	GF	HF	IF	OF	PF	LF
CF	2	0	0	0	0	0	0	0	0	0
DF	0	2	0	0	0	0	0	0	0	0
SF	0	0	2	0	0	0	0	0	0	0
AF	0	0	0	1	0	0	0	0	0	1
GF	0	0	0	0	2	0	0	0	0	0
HF	0	0	0	0	0	2	0	0	0	0
IF	0	0	0	0	0	0	2	0	0	0
OF	0	0	0	0	0	0	0	2	0	0
PF	0	0	0	0	0	0	0	0	2	0
LF	0	0	0	0	0	0	0	0	0	2

Jumlah data yang diklasifikasikan dengan benar = 19

Jumlah data yang diklasifikasikan dengan salah = 1

Menghitung nilai *Accuracy* :

$$Accuracy = \frac{nl\ benar}{jml\ data\ (n)} \times 100\%$$

$$Comminuted\ Fracture = \frac{2}{2} \times 100\% = 100$$

$$Dislocation\ Fracture = \frac{2}{2} \times 100\% = 100$$

$$Spiral\ Fracture = \frac{2}{2} \times 100\% = 100$$

$$Avulsion\ Fracture = \frac{1}{2} \times 100\% = 50$$

$$Greenstick\ Fracture = \frac{2}{2} \times 100\% = 100$$

$$Hairline\ Fracture = \frac{2}{2} \times 100\% = 100$$

$$Impacted\ Fracture = \frac{2}{2} \times 100\% = 100$$

$$\text{Oblique Fracture} = \frac{2}{2} \times 100\% = 100$$

$$\text{Pathologic Fracture} = \frac{2}{2} \times 100\% = 100$$

$$\text{Longitudinal Fracture} = \frac{2}{2} \times 100\% = 100$$

$$\text{Accuracy} = \frac{2+2+2+1+2+2+2+2+2+2}{20} \times 100\% = \frac{19}{20} \times 100\% = 0.95 = 95\%$$

Menghitung nilai TP, TN, FP, FN sesuai dengan kelas masing-masing:

Comminuted Fracture

$$\text{TP} = 2 \quad \text{TN} = 0$$

$$\text{FN} = 0 \quad \text{FP} = 0$$

Dislocation Fracture

$$\text{TP} = 2 \quad \text{TN} = 0$$

$$\text{FP} = 0 \quad \text{FN} = 0$$

Spiral Fracture

$$\text{TP} = 2 \quad \text{TN} = 0$$

$$\text{FN} = 0 \quad \text{FP} = 0$$

Avulsion Fracture

$$\text{TP} = 1 \quad \text{TN} = 0$$

$$\text{FN} = 0 \quad \text{FP} = 1$$

Greenstick Fracture

$$\text{TP} = 2 \quad \text{TN} = 0$$

$$\text{FN} = 0 \quad \text{FP} = 0$$

Hairline Fracture

$$\text{TP} = 2 \quad \text{TN} = 0$$

$$\text{FN} = 0 \quad \text{FP} = 0$$

Impacted Fracture

$$\text{TP} = 2 \quad \text{TN} = 0$$

$$\text{FN} = 0 \quad \text{FP} = 0$$

Oblique Fracture

$$\text{TP} = 2 \quad \text{TN} = 0$$

$$\text{FN} = 0 \quad \text{FP} = 0$$

Pathologic Fracture

$$\begin{array}{ll} TP = 2 & TN = 0 \\ FN = 0 & FP = 0 \end{array}$$

Longitudinal Fracture

$$\begin{array}{ll} TP = 2 & TN = 0 \\ FN = 0 & FP = 0 \end{array}$$

Persamaan precision dan recall:

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

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

Hitung nilai precision dan recall *Comminuted Fracture, Dislocation Fracture, Spiral Fracture, Avulsion Fracture, Greenstick Fracture, Hairline Fracture, Impacted Fracture, Oblique Fracture, Pathological Fracture, dan Longitudinal Fracture* :

$$\text{Comminuted Fracture: precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Dislocation Fracture: precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Spiral Fracture : precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Avulsion Fracture : precision} = \frac{1}{1+1} = 0.5 = 50 \quad \text{recall} = \frac{1}{1+0} = 1 = 100$$

$$\text{Greenstick Fracture : precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Hairline Fracture: precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Impacted Fracture: precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Oblique Fracture: precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Pathological Fracture: precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Longitudinal Fracture: precision} = \frac{2}{2+0} = 1 = 100 \quad \text{recall} = \frac{2}{2+0} = 1 = 100$$

$$\text{Precision} = \frac{100+100+100+50+100+100+100+100+100+100}{10} = 95$$

$$\text{Recall} = \frac{100+100+100+100+100+100+100+100+100+100}{10} = 100$$

Tabel 4.4 Hasil *Confusion Matrix*

Klasifikasi	TP	TN	FP	FN	Hasil %		
					<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
<i>Comminuted Fracture</i>	2	0	0	0	100	100	100
<i>Dislocation Fracture</i>	2	0	0	0	100	100	100
<i>Spiral Fracture</i>	2	0	0	0	100	100	100
<i>Avulsion Fracture</i>	1	0	1	0	50	50	100
<i>Greenstick Fracture</i>	0	0	0	0	100	100	100
<i>Hairline Fracture</i>	2	0	0	0	100	100	100
<i>Impacted Fracture</i>	2	0	0	0	100	100	100
<i>Oblique Fracture</i>	2	0	0	0	100	100	100
<i>Pathological Fracture</i>	2	0	0	0	100	100	100
<i>Longitudinal Fracture</i>	2	0	0	0	100	100	100
Rata-rata					95%	95%	100%

F1-score

$$\text{F1 - score} = 2 \times \frac{(\text{recall} \times \text{precision})}{(\text{recall} + \text{precision})}$$

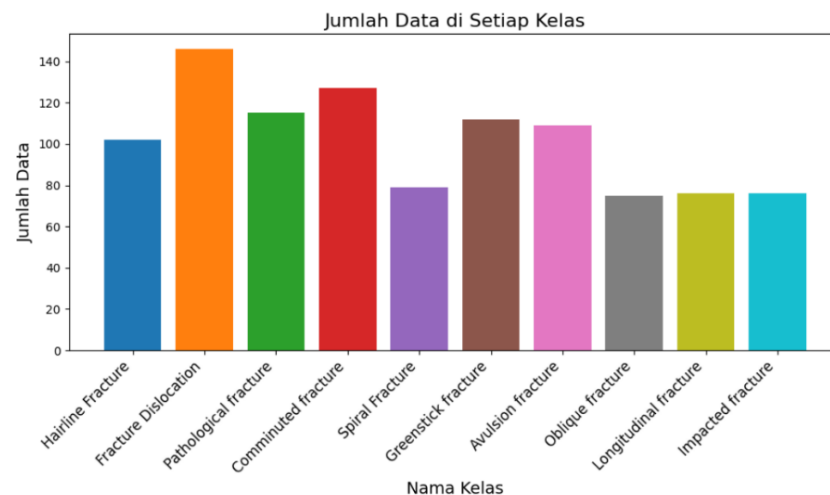
$$\text{F1 - score} = 2 \times \frac{(100 \times 95)}{(100+95)} = 2 \times \frac{9500}{195} = 2 \times 48.71 = 97.42\%$$

BAB V

PEMBAHASAN

5.1 Objek Penelitian

Data yang digunakan pada penelitian ini terdiri dari data public dengan nama PATAH TULANG DATA SET, data yang di ambil terdiri dari 10 *class* yaitu berupa *Comminuted* fraktur, *Dislocation* fraktur, *Spiral* fraktur, *Avulsion* fraktur, *Greenstick* fraktur, *Hairline* fraktur, *Impacted* fraktur, *Oblique* fraktur, *Pathological* fraktur dan *Longitudinal* fraktur. Dataset yang di unduh dari <https://universe.roboflow.com/cursor-rphcb/bone-break-classification/dataset/2>, masing-masing data training ada 989 image dan data testing ada 140.



Gambar 5.1 Jumlah Data di Setiap Kelas

5.2 Pembahasan Model

5.2.5 Proses Pelatihan Model

```
Epoch 1/10
32/32 ————— 46s 1s/step - accuracy: 0.1187 - loss: 2.7486 - val_accuracy: 0.1696 - val_loss: 2.1847
Epoch 2/10
32/32 ————— 40s 1s/step - accuracy: 0.3570 - loss: 1.9235 - val_accuracy: 0.3125 - val_loss: 1.8883
Epoch 3/10
32/32 ————— 40s 1s/step - accuracy: 0.4802 - loss: 1.5551 - val_accuracy: 0.6161 - val_loss: 1.2447
Epoch 4/10
32/32 ————— 40s 1s/step - accuracy: 0.6768 - loss: 1.0496 - val_accuracy: 0.8393 - val_loss: 0.6362
Epoch 5/10
32/32 ————— 41s 1s/step - accuracy: 0.8661 - loss: 0.6045 - val_accuracy: 0.9018 - val_loss: 0.4089
Epoch 6/10
32/32 ————— 42s 1s/step - accuracy: 0.9609 - loss: 0.3132 - val_accuracy: 0.9464 - val_loss: 0.2668
Epoch 7/10
32/32 ————— 41s 1s/step - accuracy: 0.9729 - loss: 0.2212 - val_accuracy: 0.9375 - val_loss: 0.2449
Epoch 8/10
32/32 ————— 43s 1s/step - accuracy: 0.9818 - loss: 0.1439 - val_accuracy: 0.9375 - val_loss: 0.2148
Epoch 9/10
32/32 ————— 39s 1s/step - accuracy: 0.9961 - loss: 0.1075 - val_accuracy: 0.9464 - val_loss: 0.1910
Epoch 10/10
32/32 ————— 42s 1s/step - accuracy: 0.9895 - loss: 0.0976 - val_accuracy: 0.9464 - val_loss: 0.1817
```

Gambar 5.2 Training Dataset dengan 10 *Epoch*

Gambar 5.2 menunjukkan hasil keluaran dari proses pelatihan model deep learning yang dilakukan selama 10 *epoch*. Setiap *epoch* terdiri dari 32 langkah, di mana kinerja model dievaluasi berdasarkan beberapa metrik, termasuk akurasi, loss, dan akurasi validasi. Proses pelatihan menunjukkan kemajuan yang signifikan dari *epoch* pertama hingga *epoch* terakhir, dengan peningkatan kinerja model seiring waktu saat model beradaptasi dengan data.

Pada *epoch* awal (1-3), akurasi pelatihan relatif rendah, dimulai pada 11,87% dan perlahan meningkat menjadi 48,02%. Akurasi validasi, yang menggambarkan kemampuan model untuk menggeneralisasi pada data yang belum pernah dilihat sebelumnya, juga rendah pada tahap awal ini, dimulai pada 16,96% dan naik menjadi 61,61%. Tahapan awal ini adalah hal yang biasa bagi model yang baru mulai belajar, di mana bobot-bobot model masih dioptimalkan dan jaringan mulai mengenali pola-pola dasar. Loss, yang mengukur perbedaan antara nilai yang diprediksi dan nilai yang benar, pada awalnya tinggi, mencerminkan prediksi model yang belum optimal. Pada *epoch* 1, loss tercatat 2,7486, dan pada *epoch* 3, loss ini turun menjadi 1,5551, menunjukkan adanya perbaikan.

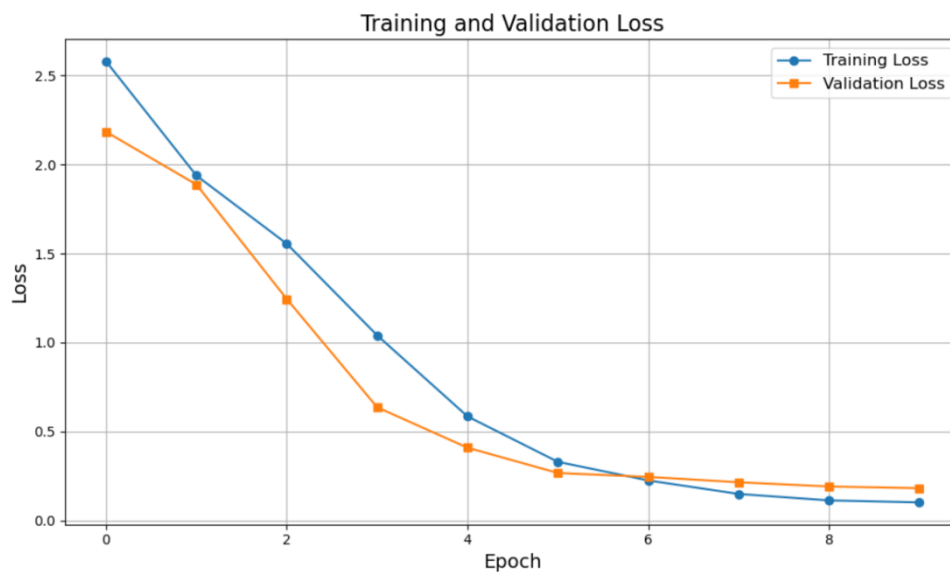
Peningkatan yang signifikan pada akurasi pelatihan dan akurasi validasi mulai terlihat. Pada *epoch* 5, akurasi pelatihan mencapai 86,61%, dan akurasi validasi meningkat menjadi 90,18%. Hal ini menunjukkan bahwa model mulai mempelajari pola yang lebih kompleks dan mulai menunjukkan kinerja yang baik tidak hanya pada data pelatihan tetapi juga pada data validasi yang tidak terlihat sebelumnya. Penurunan loss semakin mendukung hal ini, karena prediksi model semakin mendekati nilai yang sebenarnya. Loss validasi turun dari 0,6362 pada *epoch* 4 menjadi 0,2668 pada *epoch* 6, yang menunjukkan peningkatan dalam kemampuan generalisasi model.

Pada *epoch* 7 hingga 10, model mencapai tingkat kinerja yang sangat baik, dengan akurasi pelatihan melebihi 98% pada *epoch* 8. Akurasi validasi stabil pada sekitar 94%, mencerminkan keseimbangan yang baik antara pembelajaran pola pada data pelatihan dan pemeliharaan kemampuan generalisasi pada data baru. Nilai loss terus menurun secara konsisten, dengan loss validasi terakhir pada *epoch*

10 tercatat sebesar 0,1817. Penurunan ini menunjukkan bahwa model mendekati kinerja optimal, dengan overfitting yang minimal.

Hasil yang diperoleh menunjukkan efektivitas model dalam belajar dari data dan meningkatkan kinerjanya seiring waktu. Akurasi meningkat secara konsisten dengan setiap *epoch*, sementara loss menurun, menunjukkan bahwa model semakin cakap dalam membuat prediksi yang akurat. Akurasi dan loss validasi memberikan bukti bahwa model tidak hanya menguasai data pelatihan tetapi juga dapat menggeneralisasi dengan baik pada contoh yang belum terlihat. Pada *epoch* 10, model telah mencapai keseimbangan yang baik antara akurasi tinggi dan loss rendah, yang merupakan hasil yang diinginkan untuk model deep learning, karena menunjukkan ketahanan dan keandalan dalam aplikasi dunia nyata.

5.2.6 Training dan Validation Loss



Gambar 5.3 Grafik *Training* dan *Validation Loss*

Gambar 5.3 menunjukkan hubungan antara training loss dan validation loss selama 10 *epoch* pada sebuah model machine learning. Grafik ini menampilkan dua garis yang berbeda: satu untuk training loss (berwarna biru) dan satu lagi untuk validation loss (berwarna oranye). Kedua metrik loss ini menurun seiring bertambahnya jumlah *epoch*, yang menunjukkan bahwa model sedang belajar dan meningkatkan kinerjanya.

Pada *epoch* awal (1-2), baik training loss maupun validation loss relatif tinggi. Hal ini menunjukkan bahwa model masih berada pada tahap awal pembelajaran, di mana model kesulitan untuk membuat prediksi yang akurat pada data pelatihan maupun data validasi. Seiring dengan berjalannya waktu, nilai loss secara bertahap menurun, dengan training loss menunjukkan penurunan yang lebih cepat dibandingkan dengan validation loss. Ini adalah pengamatan yang khas, karena model menjadi lebih baik dalam menyesuaikan diri dengan data pelatihan pada fase awal pembelajaran. Kurva training loss turun tajam dari sekitar 2,5 menjadi sekitar 1,5 pada *epoch* 2, yang menunjukkan peningkatan signifikan dalam kemampuan model untuk memprediksi data pelatihan dengan benar.

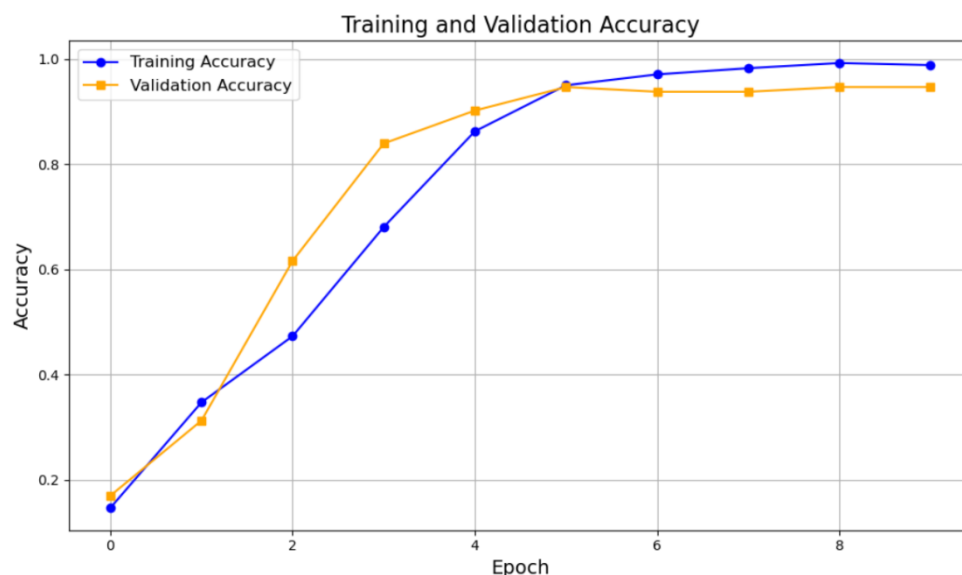
Mulai *epoch* 3 dan seterusnya, kedua kurva terus menurun, meskipun dengan laju yang lebih lambat. Pada *epoch* 4, training loss mendekati 1, dan validation loss turun menjadi sekitar 1,5. Meskipun kedua metrik ini terus membaik, laju perubahan menurun, yang merupakan tanda bahwa model mendekati keadaan yang stabil. Perbedaan antara training loss dan validation loss secara bertahap menjadi lebih konsisten, dengan validation loss menunjukkan sedikit penurunan dari *epoch* 4 hingga *epoch* 6. Pola ini menunjukkan bahwa model tidak mengalami overfitting, karena jarak antara kinerja pada data pelatihan dan data validasi tetap stabil.

Pada *epoch* yang lebih akhir (7-10), penurunan pada kedua training loss dan validation loss menjadi jauh lebih kecil. Pada *epoch* 10, training loss mencapai titik stabil di sekitar 0,4, sementara validation loss stabil di sekitar 0,5. Ini menunjukkan bahwa model telah mencapai titik di mana penurunan nilai loss tidak lagi signifikan, yang berarti model telah berhasil mempelajari pola-pola utama dalam data. Keterdekatan antara kurva training loss dan validation loss semakin memperkuat ide bahwa model ini dapat menggeneralisasi dengan baik, karena model menunjukkan kinerja yang serupa pada data pelatihan dan data validasi yang belum terlihat.

Grafik pada gambar 5.3 menunjukkan bahwa model telah dilatih dengan efektif, dengan peningkatan yang stabil selama *epoch* berlangsung. Penurunan nilai loss, khususnya konvergensi antara training loss dan validation loss, menunjukkan

bahwa model telah mempelajari pola-pola yang relevan dalam data dan dapat memprediksi dengan akurat, sambil mempertahankan kemampuan generalisasi. Jarak yang relatif kecil antara *training loss* dan *validation loss* pada akhir proses pelatihan adalah tanda positif, karena ini mencerminkan model yang seimbang dengan baik dan tidak mengalami *overfitting* pada data pelatihan. Hasil ini menunjukkan bahwa proses pelatihan telah menghasilkan model yang dapat berfungsi dengan baik baik pada data pelatihan maupun pada data validasi yang tidak terlihat sebelumnya.

5.2.7 Training dan Validation Accuracy



Gambar 5.4 Grafik *Training dan Validation Accuracy*

Gambar 5.4 menunjukkan grafik yang menggambarkan perubahan akurasi pelatihan (*training accuracy*) dan akurasi validasi (*validation accuracy*) seiring berjalannya waktu selama 10 *epoch*. Grafik ini mengilustrasikan bagaimana kinerja model berkembang selama proses pelatihan, baik pada data pelatihan maupun data validasi.

Pada awal *epoch* (*epoch* 0 hingga 2), baik akurasi pelatihan maupun validasi menunjukkan peningkatan yang cukup signifikan. Akurasi pelatihan (berwarna biru) mulai dari sekitar 0,2 pada *epoch* pertama dan naik tajam hingga mencapai sekitar 0,6 pada *epoch* kedua. Akurasi validasi (berwarna oranye) mengikuti tren serupa, meskipun dengan sedikit perbedaan, dari sekitar 0,2 menjadi lebih dari 0,5

pada *epoch* kedua. Hal ini menunjukkan bahwa model mulai mengidentifikasi pola-pola dalam data pelatihan, meskipun pada tahap ini akurasi validasi belum sebaik akurasi pelatihan, yang merupakan hal yang wajar pada fase awal pelatihan.

Pada *epoch* 3 hingga 6, kedua kurva menunjukkan peningkatan yang lebih moderat. Akurasi pelatihan terus meningkat dengan laju yang stabil, mencapai sekitar 0,8 pada *epoch* 4 dan mendekati 1 pada *epoch* 6. Akurasi validasi juga terus meningkat, meskipun dengan kecepatan yang sedikit lebih lambat dibandingkan dengan training accuracy, mencapai sekitar 0,75 pada *epoch* 6. Penurunan laju peningkatan pada *epoch* 4 hingga 6 ini menunjukkan bahwa model mulai mencapai titik di mana pengoptimalan menjadi lebih sulit dan ada sedikit kesulitan dalam meningkatkan akurasi lebih jauh pada data validasi.

Memasuki *epoch* 7 hingga 10, kedua kurva mencapai titik jenuh, dengan akurasi pelatihan mendekati 1,0, sedangkan akurasi validasi stabil di sekitar 0,94. Selama periode ini, meskipun akurasi pelatihan mencapai nilai puncak, akurasi validasi tampaknya melambat, menunjukkan bahwa meskipun model berhasil menguasai data pelatihan, ada sedikit penurunan dalam kemampuannya untuk menggeneralisasi pada data yang belum pernah dilihat sebelumnya. Hal ini bisa mengindikasikan adanya sedikit overfitting, di mana model terlalu fokus pada data pelatihan, namun tetap menunjukkan kemampuan yang sangat baik dalam pengujian validasi.

Grafik pada gambar 5.4 menunjukkan bahwa model mengalami peningkatan yang sangat baik dalam hal akurasi, baik pada data pelatihan maupun data validasi, seiring berjalannya waktu. Keduanya menunjukkan tren peningkatan yang sangat jelas, meskipun dengan sedikit perbedaan dalam laju kenaikan. Akurasi pelatihan mencapai angka yang sangat tinggi, mencerminkan kemampuan model dalam mengidentifikasi pola dalam data pelatihan, sementara akurasi validasi yang stabil menunjukkan bahwa model berhasil menggeneralisasi dengan baik pada data yang tidak terlihat. Meskipun ada sedikit pelambatan dalam peningkatan validasi, hasil ini tetap menunjukkan model yang sangat efektif dan dapat diandalkan untuk tugas yang diberikan.

5.2.8 Evaluasi Metode

Classification Report:

	precision	recall	f1-score	support
Avulsion fracture	0.79	1.00	0.88	11
Comminuted fracture	1.00	0.93	0.97	15
Fracture Dislocation	0.94	1.00	0.97	17
Greenstick fracture	1.00	0.90	0.95	10
Hairline Fracture	0.91	1.00	0.95	10
Impacted fracture	1.00	0.88	0.93	8
Longitudinal fracture	1.00	0.92	0.96	12
Oblique fracture	0.90	0.90	0.90	10
Pathological fracture	1.00	0.92	0.96	12
Spiral Fracture	1.00	1.00	1.00	7
accuracy			0.95	112
macro avg	0.95	0.94	0.95	112
weighted avg	0.95	0.95	0.95	112

Gambar 5.5 Hasil Evaluasi Kinerja Metode

Gambar 5.5 menunjukkan laporan klasifikasi untuk sebuah model yang mengevaluasi akurasi prediksinya pada berbagai jenis patah tulang. Pada Gambar 5.5 menunjukkan matriks kunci seperti presisi, recall, dan f1-score untuk setiap kelas patah tulang, serta akurasi keseluruhan model. Metrik-metrik ini sangat penting untuk menilai sejauh mana model dapat membedakan antara berbagai jenis patah tulang dan seberapa efektif kinerjanya di berbagai kategori.

Jenis patah tulang yang dievaluasi meliputi *Avulsion fracture*, *Comminuted fracture*, *Fracture Dislocation*, *Greenstick fracture*, *Hairline fracture*, *Impacted fracture*, *Longitudinal fracture*, *Oblique fracture*, *Pathological fracture*, dan *Spiral fracture*. Untuk setiap jenis patah tulang, nilai presisi, recall, dan f1-score dilaporkan. Presisi menunjukkan proporsi prediksi positif yang benar dari semua prediksi positif yang dibuat oleh model, sementara recall menggambarkan proporsi prediksi positif yang benar dari semua instance positif yang sebenarnya dalam dataset. F1-score adalah rata-rata harmonis dari presisi dan recall, memberikan keseimbangan antara keduanya.

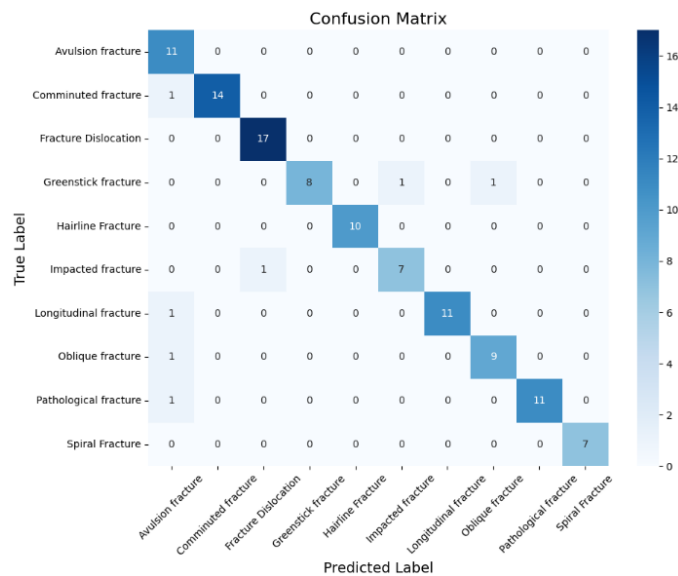
Model ini menunjukkan kinerja yang sangat baik di sebagian besar kategori patah tulang. Misalnya, presisi dan recall untuk kelas "*Comminuted fracture*" sangat tinggi, dengan presisi 1,00 dan *recall* 0,93, menghasilkan f1-score yang kuat

yaitu 0,97. Demikian pula, kelas "*Greenstick fracture*" dan "*Spiral fracture*" menunjukkan presisi, recall, dan f1-score yang sempurna, masing-masing 1,00, yang menunjukkan akurasi model dalam memprediksi jenis patah tulang ini. Kelas "*Fracture Dislocation*" dan "*Pathological fracture*" juga menunjukkan nilai presisi dan recall yang tinggi, dengan f1-score masing-masing 0,97 dan 0,96, yang semakin menegaskan ketangguhan model.

Tetapi, tidak semua kelas patah tulang mencapai kinerja sempurna. Kelas "*Avulsion fracture*" menunjukkan presisi yang sedikit lebih rendah yaitu 0,79 tetapi dapat menebusnya dengan recall sempurna 1,00. Ini menunjukkan bahwa meskipun model kadang-kadang salah mengklasifikasikan patah tulang lain sebagai "*Avulsion*", ia dapat dengan andal mengidentifikasi semua instance asli dari jenis patah tulang ini, menghasilkan recall yang lebih tinggi. F1-score untuk kelas ini adalah 0,88, mencerminkan kompromi ini. Kelas "*Impacted fracture*" juga menunjukkan recall tinggi 1,00, dengan presisi 1,00 dan f1-score 0,93, yang menunjukkan kinerja yang baik meskipun ada sedikit penurunan pada recall.

Akurasi keseluruhan model adalah 0,95, yang berarti 95% dari prediksi yang dibuat oleh model adalah benar. Ini adalah hasil yang sangat baik, yang semakin didukung oleh skor rata-rata makro dan rata-rata berbobot, yang juga masing-masing sebesar 0,95. Rata-rata makro menghitung kinerja rata-rata di seluruh kelas, sementara rata-rata berbobot mempertimbangkan ketidakseimbangan kelas dengan memberi bobot lebih pada kelas yang lebih sering muncul. Kedua rata-rata ini mengonfirmasi bahwa model bekerja secara konsisten di seluruh jenis patah tulang, tanpa ada kelas yang menunjukkan kinerja yang buruk.

Model ini menunjukkan akurasi tinggi dan kinerja yang sangat baik di semua kategori patah tulang, dengan hasil yang sangat kuat dalam hal presisi dan recall untuk sebagian besar kelas. Variasi kecil dalam presisi dan recall di berbagai kelas menunjukkan area di mana model dapat dioptimalkan lebih lanjut, tetapi secara keseluruhan, model ini sangat efektif untuk klasifikasi patah tulang.



Gambar 5.6 *Confusion Matrix*

Gambar 5.6 menunjukkan *confusion matrix* yang digunakan untuk mengevaluasi kinerja model dalam mengklasifikasikan berbagai jenis patah tulang. Matriks ini memberikan gambaran yang jelas tentang sejauh mana model dapat memprediksi dengan benar atau keliru untuk setiap jenis patah tulang berdasarkan data yang digunakan. Setiap baris dalam matriks mewakili kelas nyata (*true label*), sementara setiap kolom mewakili prediksi yang dibuat oleh model (*predicted label*).

Pada diagonal utama matriks, yang menunjukkan jumlah prediksi yang benar untuk setiap kelas, terdapat angka yang cukup tinggi untuk sebagian besar jenis patah tulang. Misalnya, pada kelas "*Comminuted fracture*", model berhasil mengklasifikasikan 14 dari 15 contoh dengan benar, yang menunjukkan bahwa model ini sangat baik dalam mengenali patah tulang jenis ini. Hal yang sama berlaku untuk "*Fracture Dislocation*" yang memiliki 17 prediksi yang benar. Keakuratan model dalam mengklasifikasikan jenis-jenis patah tulang ini mengindikasikan bahwa model memiliki performa yang kuat dalam mengenali ciri khas dari berbagai jenis patah tulang.

Namun, di luar diagonal utama, terdapat beberapa kesalahan klasifikasi yang menunjukkan bahwa model tidak selalu berhasil memprediksi kelas dengan akurasi sempurna. Sebagai contoh, pada kelas "*Avulsion fracture*", meskipun ada

11 prediksi yang benar, terdapat 1 kesalahan di mana model salah mengklasifikasikan contoh sebagai "*Longitudinal fracture*". Ini menunjukkan bahwa model memiliki beberapa kesulitan dalam membedakan antara jenis patah tulang yang mungkin memiliki ciri fisik yang mirip atau gejala yang tumpang tindih. Selain itu, pada kelas "*Greenstick fracture*", model mengklasifikasikan 8 contoh dengan benar, namun ada 1 contoh yang salah diklasifikasikan sebagai "*Pathological fracture*" dan 1 lainnya sebagai "*Hairline fracture*".

Matriks pada Gambar 5.7 juga memperlihatkan bahwa model cenderung memiliki performa yang sangat baik pada kelas-kelas dengan jumlah contoh yang lebih besar, seperti "*Fracture Dislocation*" dan "*Longitudinal fracture*", di mana sebagian besar prediksi yang dibuat adalah benar. Namun, pada kelas-kelas dengan jumlah contoh yang lebih sedikit, seperti "*Spiral fracture*", model menunjukkan kinerja yang sangat baik dengan 7 prediksi yang benar dan tidak ada kesalahan. Hal ini mengindikasikan bahwa meskipun jumlah data terbatas untuk beberapa kelas, model mampu mengklasifikasikan dengan akurasi yang tinggi.

BAB VI

PENUTUP

6.1 Kesimpulan

Hasil penelitian menunjukkan bahwa penerapan metode CNN dalam klasifikasi jenis patah tulang pada manusia dapat memberikan hasil yang memuaskan dengan akurasi mencapai 95%. Model yang dibangun mampu mengidentifikasi dan mengklasifikasikan jenis patah tulang dengan tingkat akurasi yang baik, meskipun ada beberapa kesalahan pada kelas-kelas tertentu seperti pada "Avulsion fracture". Secara keseluruhan, model CNN yang diterapkan menunjukkan kinerja yang sangat baik dalam mengidentifikasi fraktur berdasarkan citra X-ray. Tingkat akurasi model pada data testing menunjukkan hasil yang sangat baik, dengan rata-rata akurasi sebesar 95%. Meskipun ada perbedaan antara akurasi pelatihan dan validasi, model menunjukkan kemajuan yang signifikan dalam proses pelatihan dengan peningkatan akurasi yang stabil hingga mencapai nilai maksimal pada *epoch* terakhir. Analisis menggunakan *Confusion Matrix* menunjukkan bahwa model memiliki performa yang sangat baik pada sebagian besar kelas fraktur, seperti pada "*Comminuted fracture*" dan "*Spiral fracture*" dengan akurasi 100%. Hasil ini menunjukkan bahwa CNN sangat efektif dalam mengklasifikasikan jenis patah tulang tertentu, meskipun beberapa kelas, seperti "Avulsion fracture", menunjukkan akurasi yang lebih rendah.

6.2 Saran

1. Penelitian selanjutnya perlu fokus pada pengembangan teknik augmentasi data atau penerapan arsitektur model yang lebih kompleks untuk meningkatkan presisi pada kelas-kelas dengan akurasi rendah, seperti "Avulsion fracture". Penggunaan dataset yang lebih besar dan beragam juga dapat membantu memperbaiki kinerja model, terutama dalam mengatasi masalah pada kelas minoritas yang kurang terrepresentasi.
2. Salah satu arah penelitian yang dapat dipertimbangkan adalah penerapan transfer learning dengan menggunakan model pretrained seperti VGG16, ResNet, atau InceptionV3. Pendekatan ini memungkinkan pemanfaatan

model yang telah terlatih pada dataset besar untuk meningkatkan akurasi klasifikasi pada dataset yang lebih terbatas, seperti citra X-ray patah tulang.

3. Untuk meningkatkan akurasi diagnosis, penelitian dapat mengeksplorasi penggunaan data multi-modal, termasuk citra CT scan atau MRI, sebagai pelengkap citra X-ray. Selain itu, pengembangan model dengan interpretabilitas yang lebih tinggi akan penting untuk meningkatkan kepercayaan profesional medis terhadap sistem diagnosis otomatis, dengan memberikan penjelasan yang jelas mengenai alasan di balik prediksi yang dihasilkan.

DAFTAR PUSTAKA

- [1] Siloam hospitals medical team, “Fraktur (Patah Tulang) - Penyebab, Gejala, dan Penanganannya.” Accessed: Sep. 24, 2024. [Online]. Available: <https://www.siloamhospitals.com/en/informasi-siloam/artikel/apa-itu-fraktur>
- [2] K. Dlshad Ahmed and R. Hawezi, “Detection of bone fracture based on machine learning techniques,” *Measurement: Sensors*, vol. 27, Jun. 2023, doi: 10.1016/j.measen.2023.100723.
- [3] X. Hu, J. Chen, X. Zheng, J. Li, and M. Zhou, “Establishment and application of TSDPSO-SVM model combined with multi-dimensional feature fusion method in the identification of fracture-related infection,” *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-46526-w.
- [4] G. R. V. S. B Sumanth, “<https://ieeexplore.ieee.org/abstract/document/9996074>,” *Bone Fracture Identification using ANN Algorithm and Comparing the Accuracy with SVM Algorithm*, 2022.
- [5] “KLASIFIKASI PENYAKIT PADA DAUN KENTANG MENGGUNAKAN PENGOLAHAN CITRA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN).”
- [6] M. Zein Ersyad, K. N. Ramadhani, and A. Arifianto, “Pengenaln Bentuk Tangan dengan Convolutional Neural Network (CNN).”
- [7] Ihda fadila, “<https://hellosehat.com/muskuloskeletal/patah-tulang/jenis-fraktur/>,” *Mengenal Jenis-jenis Fraktur atau Patah Tulang yang Umum Terjadi*, Sep. 2021.
- [8] Dr.pittara, “<https://www.alodokter.com/dislokasi>,” *Dislokasi*, Aug. 2022.
- [9] Fracture and orthopaedic clinic LTD, “<https://www.faoonline.com/home/conditions/shoulder/intraarticular-fractures>”.
- [10] Tatiana. Tomljanovic, *Cougars*. Smartbook Media Inc., 2016.
- [11] “UJI KESESUAIAN BERKAS SINAR-X DI RUMAH SAKIT BHAYANGKARA JAMBI.”

- [12] R. Rokhana *et al.*, “Convolutional Neural Network untuk Pendeteksian Patah Tulang Femur pada Citra Ultrasonik B-Mode,” 2019.
- [13] Triano nurhikmat, “https://www.researchgate.net/figure/Gambar-51-Arsitektur-Jaringan-Gambar-51-merupakan-arsitektur-jaringan-pada-proses_fig8_325578703,” 2018.
- [14] SuperDataScience Team, “Convolutional Neural Networks (CNN): Step 3 - Flattening,” <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening>, 2018.
- [15] “()”, doi: 10.15680/IJIRCCE.2016.
- [16] Alejandro ito aramendia, “<https://medium.com/@alejandro.itoaramendia/convolutional-neural-networks-cnns-a-complete-guide-a803534a1930>,” *Convolutional Neural Networks (CNNs) : A Complete Guide*, 2024.
- [17] Pavan belagatti, “<https://www.singlestore.com/blog/a-guide-to-sigmoid-activation-function/>,” *Understanding the Sigmoid Activation Function: A Comprehensive Guide*, 2024.
- [18] Giorgio roffo, “https://www.researchgate.net/figure/9-An-illustration-of-the-dropout-mechanism-within-the-proposed-CNN-a-Shows-a_fig23_317277576,” 2017.
- [19] Gabriele deluca, “<https://www.baeldung.com/cs/ml-relu-dropout-layers>,” *How ReLU and Dropout Layers Work in CNNs*, 2024.

Lampiran 1 : DATA

Comminuted Fracture



Dislocation fracture



Spiral Fracture



Avulsion Fracture



Greenstick Fracture



Hairline Fracture



Impacted Fracture



Oblique Fracture



Pathologic Fracture



Longitudinal Fracture



Lampiran 2 : KODE PROGRAM

1. Image Processing

```
import os
import numpy as np
import matplotlib.pyplot as plt

import tensorflow as tf

from tensorflow import keras
from keras import (
    backend as K,
    losses,
    optimizers,
    layers,
    models,
    utils,
    callbacks
)

from pathlib import Path
data_dir = "dataset_2"

# load the train data

train_data = utils.image_dataset_from_directory(
    data_dir,
    labels="inferred",
    label_mode="int",
    validation_split=0.1,
    subset="training",
    shuffle=True,
```

```

color_mode="rgb",
image_size=(256,256),
batch_size=64,
seed=40
)

#load the test data
validation_data = utils.image_dataset_from_directory(
data_dir,
labels="inferred",
label_mode="int",
validation_split=0.1,
subset="validation",
color_mode="rgb",
image_size=(256,256),
batch_size=64,
seed=42
)

# Inspect class names
class_names = train_data.class_names
print("Class names:", class_names)

import matplotlib.cm as cm # Untuk palet warna

# Hitung jumlah data di setiap kelas
class_counts = {}
for images, labels in train_data.unbatch():
label = labels.numpy()
class_counts[label] = class_counts.get(label, 0) + 1

```

```

# Ubah indeks label menjadi nama kelas
class_counts_named = {class_names[label]: count for label, count in
class_counts.items()}

# Buat daftar warna berbeda menggunakan colormap
num_classes = len(class_counts_named)
colors = cm.tab10(range(num_classes)) # Menggunakan colormap "tab10"

# Plot grafik batang dengan warna berbeda
plt.figure(figsize=(10, 6))
plt.bar(
class_counts_named.keys(),
class_counts_named.values(),
color=colors
)
plt.title("Jumlah Data di Setiap Kelas", fontsize=16)
plt.xlabel("Nama Kelas", fontsize=14)
plt.ylabel("Jumlah Data", fontsize=14)
plt.xticks(rotation=45, ha="right", fontsize=12)
plt.tight_layout()
plt.show()

def display_random_samples(data, class_names, num_samples=16,
grid_size=(4, 4)):
"""
Menampilkan sampel gambar acak beserta nama kelas dalam bentuk grid.
Args:
data: Dataset (contoh: train_data atau validation_data)
class_names: List nama kelas
num_samples: Jumlah gambar yang ingin ditampilkan
grid_size: Ukuran grid (jumlah baris dan kolom)

```

```

"""

# Ambil sampel acak dari dataset
images = []
labels = []
for img_batch, label_batch in data.unbatch().take(num_samples):
    images.append(img_batch.numpy())
    labels.append(label_batch.numpy())

# Atur ukuran grid
fig, axes = plt.subplots(grid_size[0], grid_size[1], figsize=(12, 12))
axes = axes.flatten()

for img, label, ax in zip(images, labels, axes):
    # Tampilkan gambar
    ax.imshow(img.astype("uint8"))
    ax.set_title(f'Class: {class_names[label]}', fontsize=10)
    ax.axis("off") # Hapus sumbu

# Hapus subplot kosong jika ada
for ax in axes[len(images):]:
    ax.axis("off")

plt.tight_layout()
plt.show()

# Tampilkan 16 sampel acak dari train_data
display_random_samples(train_data, class_names, num_samples=16,
grid_size=(4, 4))

# data preprocessing
def preprocess(img):

```



```
img = tf.cast(img, "float32") / 255.0
```

```
return img
```

```
train_dataset = train_data.map(lambda x, y: (preprocess(x), y))
```

```
val_dataset = validation_data.map(lambda x, y: (preprocess(x), y))
```

```
# get train data and labels
```

```
x_train = []
```

```
y_train = []
```

```
for images, labels in train_dataset:
```

```
#append train data to x_train
```

```
x_train.append(images.numpy())
```

```
#append labels to y_train
```

```
y_train.append(labels.numpy())
```

```
# concatenate all the batches to get the full list
```

```
x_train = np.concatenate(x_train, axis=0)
```

```
y_train = np.concatenate(y_train, axis=0)
```

```
# get val data and labels
```

```
x_val = []
```

```
y_val = []
```

```
for images, labels in val_dataset:
```

```
#append train data to x_train
```

```
x_val.append(images.numpy())
```

```
#append labels to y_train
```

```
y_val.append(labels.numpy())
```

```

# concatenate all the batches to get the full list
x_val = np.concatenate(x_val, axis=0)
y_val = np.concatenate(y_val, axis=0)
# get single batch from train data

def sample_batch(data):
    batch = data.take(1).get_single_element()
    if isinstance(batch, tuple):
        batch = batch[0]
    return batch.numpy()
class_names_dict = {
    0: 'Avulsion fracture',
    1: 'Comminuted fracture',
    2: 'Fracture Dislocation',
    3: 'Greenstick fracture',
    4: 'Hairline Fracture',
    5: 'Impacted fracture',
    6: 'Longitudinal fracture',
    7: 'Oblique fracture',
    8: 'Pathological fracture',
    9: 'Spiral Fracture'
}
# function to display images

def display(images, n=10, size=(15,5), cmap="gray_r", as_type="float32"):
    indices = np.random.choice(len(images), n)

    num_images_per_row = 4

    num_rows = (n + num_images_per_row - 1) // num_images_per_row

```

```

# Set up the subplots
fig, axes = plt.subplots(num_rows, num_images_per_row, figsize=size)

#plt.figure(figsize=size)
for i, index in enumerate(indices):
    #plt.subplot(1, n, i+1)

    label_index = y_train[index]

    # Convert integer label to class name
    class_name = class_names_dict.get(label_index, f'Unknown Class {label_index}')

    # Calculate the position of the subplot in the grid
    row = i // num_images_per_row
    col = i % num_images_per_row

    axes[row, col].imshow(images[index].astype(as_type), cmap=cmap)
    axes[row, col].set_title(f'Class: {label_index}')
    axes[row, col].axis('off')

plt.subplots_adjust(wspace=0.5, hspace=0.5)

plt.show()

input_layer = layers.Input(shape=(256,256, 3))

x = layers.Conv2D(
    filters=32,
    kernel_size=(3,3),
    strides=2,

```

```
padding="same"  
) (input_layer)  
x = layers.BatchNormalization(momentum=0.9)(x)  
x = layers.LeakyReLU(0.2)(x)  
x = layers.Dropout(rate=0.2)(x)
```

```
x = layers.Conv2D(  
64,  
3,  
strides=2,  
padding="same"  
) (x)  
x = layers.BatchNormalization(momentum=0.9)(x)  
x = layers.LeakyReLU(0.2)(x)  
x = layers.Dropout(rate=0.2)(x)
```

```
x = layers.Conv2D(  
128,  
3,  
strides=2,  
padding="same"  
) (x)  
x = layers.BatchNormalization(momentum=0.9)(x)  
x = layers.LeakyReLU(0.2)(x)  
x = layers.Dropout(rate=0.2)(x)
```

```
x = layers.Conv2D(  
256,  
3,  
strides=2,  
padding="same"
```

```

)(x)
x = layers.BatchNormalization(momentum=0.9)(x)
x = layers.LeakyReLU(0.2)(x)
x = layers.Dropout(rate=0.2)(x)

x = layers.Flatten()(x)

x = layers.Dense(units=100)(x)
x = layers.BatchNormalization(momentum=0.9)(x)
x = layers.LeakyReLU(0.2)(x)
x = layers.Dropout(rate=0.2)(x)

output_layer = layers.Dense(units=10, activation="softmax")(x)

model2 = models.Model(input_layer, output_layer)

model2.summary()
opt = optimizers.Adam(learning_rate=0.0005)
model2.compile(loss="categorical_crossentropy", optimizer=opt,
metrics=["accuracy"])
# save checkpoints to monitor best performing models

model_checkpoint_callback = callbacks.ModelCheckpoint(
    filepath="./checkpoint.keras",
    save_weights_only = False,
    save_freq = "epoch",
    monitor = "loss",
    mode = "min",
    save_best_only = True,
    verbose = 0
)

```

```

tensorboard_callback = callbacks.TensorBoard(log_dir="./logs")
history = model2.fit(
x_train,
y_train,
shuffle=True,
batch_size=32,
epochs=10,
validation_data=(x_val, y_val),
callbacks=[
model_checkpoint_callback,
tensorboard_callback,
]
)
train_loss = history.history["loss"]
train_accuracy = history.history["accuracy"]
val_loss = history.history["val_loss"]
val_accuracy = history.history["accuracy"]
def display_dense_layers_info(model):
"""
Menampilkan informasi tentang Fully Connected (Dense) Layers dari model.
Args:
model: Model Keras (contoh: model2)
"""
print("Fully Connected (Dense) Layers Information:")
print("-" * 50)

for i, layer in enumerate(model.layers):
# Periksa apakah layer adalah Dense
if isinstance(layer, keras.layers.Dense):
# Ambil nama layer, input, dan output

```

```

layer_name = layer.name
input_shape = layer.input_shape if hasattr(layer, "input_shape") else
layer.input.shape
output_shape = layer.output_shape if hasattr(layer, "output_shape") else
layer.output.shape
num_params = layer.count_params()

print(f"Layer {i}: {layer_name}")
print(f" Input Shape: {input_shape}")
print(f" Output Shape: {output_shape}")
print(f" Number of Parameters: {num_params}")
print("-" * 50)

# Tampilkan informasi Fully Connected Layers dari model CNN
display_dense_layers_info(model2)
import matplotlib.pyplot as plt
import numpy as np

def draw_neural_network(layer_sizes):
    """
    Membuat visualisasi jaringan saraf untuk Fully Connected Layers.
    Args:
    layer_sizes: List jumlah neuron pada setiap layer (contoh: [input, hidden1,
    hidden2, output])
    """
    # Atur warna untuk setiap layer
    colors = ['blue', 'orange', 'orange', 'green']

    # Set ukuran gambar
    plt.figure(figsize=(10, 6))

```

```

# Total jumlah layer
num_layers = len(layer_sizes)

# Posisi neuron dan koneksi
positions = []

for layer_idx, layer_size in enumerate(layer_sizes):
    # Posisi neuron pada layer ini
    x = np.full((layer_size,), layer_idx) # Sumbu X ditentukan oleh layer
    y = np.linspace(0, 1, layer_size)    # Sumbu Y proporsional

    positions.append((x, y))

# Plotkan neuron sebagai lingkaran
plt.scatter(x, y, s=400, color=colors[layer_idx], label=f"Layer {layer_idx+1}
({layer_size} neuron)")

# Tambahkan koneksi antar layer
for layer_idx in range(num_layers - 1):
    for i in range(layer_sizes[layer_idx]):
        for j in range(layer_sizes[layer_idx + 1]):
            # Garis antara neuron i di layer ke-l dan neuron j di layer ke-l+1
            plt.plot(
                [positions[layer_idx][0][i], positions[layer_idx + 1][0][j]],
                [positions[layer_idx][1][i], positions[layer_idx + 1][1][j]],
                'gray', lw=0.5, alpha=0.7
            )

# Tambahkan judul dan legenda
plt.title("Fully Connected Layers with Input and Output", fontsize=16)
plt.legend(loc="upper right", fontsize=10)

```



```

plt.axis("off")
plt.tight_layout()
plt.show()

# Tentukan ukuran layer [input, hidden1, hidden2, output]
layer_sizes = [5, 10, 5, 1]
draw_neural_network(layer_sizes)
model2.evaluate(x_val, y_val)
# plot learning curve

epochs= range(1,11)

plt.figure(figsize=(12,6))

plt.subplot(1,2,1)
plt.plot(epochs, train_loss, 'r-', 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.subplot(1,2,2)
plt.plot(epochs, train_accuracy, 'r-', label="Training Accuracy")
plt.plot(epochs, val_accuracy, 'b-', label="Validation Accuracy")
plt.title("Training and Validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()

plt.tight_layout()

```

```

plt.show()
# Fungsi untuk menampilkan gambar beserta prediksi
def display_predictions(images, preds, actuals, n=10, size=(15, 8)):
    """
    Menampilkan gambar beserta prediksi dan label aktual.
    images: Data gambar
    preds: Prediksi model (nama kelas)
    actuals: Label aktual (nama kelas)
    n: Jumlah gambar untuk ditampilkan
    size: Ukuran gambar plot
    """
    indices = np.random.choice(len(images), n, replace=False) # Pilih indeks
    gambar secara acak
    num_images_per_row = 5 # Gambar per baris
    num_rows = (n + num_images_per_row - 1) // num_images_per_row # Hitung
    jumlah baris

    fig, axes = plt.subplots(num_rows, num_images_per_row, figsize=size)
    axes = axes.flatten() # Flatten array axes untuk mempermudah indexing

    for i, index in enumerate(indices):
        img = images[index]
        pred = preds[index]
        actual = actuals[index]

        # Tampilkan gambar
        axes[i].imshow(img)
        axes[i].set_title(f'Pred: {pred}\nActual: {actual}', fontsize=10)
        axes[i].axis("off")

    # Hapus subplot kosong

```

```

for ax in axes[len(indices):]:
    ax.axis("off")

plt.tight_layout()
plt.show()

# Ambil prediksi dan label aktual
preds = model2.predict(x_val)
preds_single = [class_names[np.argmax(pred)] for pred in preds]
actual_single = [class_names[np.argmax(label)] for label in y_val]

# Tampilkan gambar dengan prediksi
display_predictions(x_val, preds_single, actual_single, n=10)
from sklearn.metrics import classification_report

# Prediksi pada data validasi
y_pred_probs = model2.predict(x_val) # Probabilitas prediksi
y_pred_classes = np.argmax(y_pred_probs, axis=-1) # Konversi ke kelas

# Konversi label sebenarnya dari one-hot encoding ke kelas
y_true_classes = np.argmax(y_val, axis=-1)

# Laporan klasifikasi
report = classification_report(y_true_classes, y_pred_classes,
                               target_names=class_names)

# Tampilkan laporan
print("Classification Report:\n")
print(report)
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

```

```

import seaborn as sns

# Dapatkan prediksi model
y_pred_probs = model2.predict(x_val) # Probabilitas prediksi
y_pred_classes = np.argmax(y_pred_probs, axis=-1) # Konversi ke kelas

# Konversi label sebenarnya dari one-hot encoding ke kelas
y_true_classes = np.argmax(y_val, axis=-1)

# Buat confusion matrix
cm = confusion_matrix(y_true_classes, y_pred_classes)

# Visualisasi dengan seaborn
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=class_names, yticklabels=class_names)
plt.title("Confusion Matrix", fontsize=16)
plt.xlabel("Predicted Label", fontsize=14)
plt.ylabel("True Label", fontsize=14)
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
2.

```

Lampiran 3 : RIWAYAT HIDUP PENELITI



Nama : Gusti Nuzlan Djafri
Nim : T3121124
Tempat Tanggal Lahir : Gorontalo, 13 November 2003
Email : gustinuzlan12@gmail.com

Riwayat Pendidikan:

1. Tahun 2015, menyelesaikan pendidikan di Sekolah Dasar Negeri 69 Kecamatan Kota Timur, Kota Gorontalo
2. Tahun 2018, menyelesaikan pendidikan di Madrasah Tsanawiyah Al-Huda Kota Gorontalo
3. Tahun 2021, menyelesaikan pendidikan di Sekolah Menengah Atas Negeri 1 Gorontalo, Kota Gorontalo
4. Tahun 2021, telah di terima menjadi mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
LEMBAGA PENELITIAN

Jl. Achmad Nadjamuddin No.17, Kampus Unisan Gorontalo Lt.1 Kota Gorontalo 96128
Website: lemlitunisan.ac.id, Email: lembagapenelitian@unisan.ac.id

Nomor : 117/PIP/B.04/LP-UIG/2024
Lampiran : -
Hal : Permohonan Izin Penelitian (Pengambilan Data)

Kepada Yth.,
UNIVERSITAS ICHSAN GORONTALO
di -
Tempat

Yang bertandatangan di bawah ini:

Nama : Dr. Rahmisyari, ST., SE., MM
NIDN : 0929117202
Pangkat Akademik : Lektor Kepala
Jabatan : Ketua Lembaga Penelitian Universitas Ichsan Gorontalo

Meminta kesediaannya untuk memberikan izin pengambilan data dalam rangka penyusunan **Proposal/Skripsi**, kepada:

Nama : Gusti Nuzlan Djafri
NIM : T3121124
Fakultas : Ilmu Komputer
Program Studi : Teknik Informatika
Judul Penelitian : Klasifikasi Jenis Patah Tulang pada Manusia Menggunakan Metode Convolutional Neural Network (CNN)
Lokasi Penelitian : UNIVERSITAS ICHSAN GORONTALO

Demikian surat ini saya sampaikan, atas bantuan dan kerjasamanya diucapkan banyak terima kasih.

Dikeluarkan di Gorontalo
Tanggal, 17/10/2024
Ketua Lembaga Penelitian

Dr. Rahmisyari, ST., SE., MM
NIDN: 0929117202



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO

FAKULTAS ILMU KOMPUTER

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001

Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT KETERANGAN PENELITIAN

Nomor : 029 /FIKOM-UIG/SKP/I/2025

Yang bertanda tangan dibawah ini :

N a m a : Irvan Abraham Salihi, M. Kom
Jabatan : Dekan Fakultas Ilmu Komputer

Dengan ini Menerangkan bahwa :

N a m a Mahasiswa : Gusti Nuzlan Djafri
N I M : T3121124
Program Studi : Teknik Informatika

Bahwa yang bersangkutan benar-benar telah melakukan penelitian tentang "Klasifikasi Jenis Patah Tulang pada Manusia Menggunakan Metode Convolutional Neural Network (CNN) " Guna untuk menyelesaikan Studi pada Program Studi Teknik Informatika Fakultas Ilmu Komputer, dan bersangkutan telah menyelesaikan penelitian Tersebut pada TGL 30 Januari 2025 sesuai dengan waktu yang telah di tentukan.

Demikian Surat Keterangan ini dibuat dan digunakan untuk seperlunya.

Gorontalo, 30 Januari 2025
Dekan,

Irvan A. Salihi, S.Kom., M.Kom
NIDN : 0928028101



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
UPT. PERPUSTAKAAN FAKULTAS
SK. MENDIKNAS RI NO. 84/D/0/2001**

Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No : 002/Perpustakaan-Fikom/II/2025

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota : Gusti Nuzla Djafri
No. Induk : T3121124
No. Anggota : M20252

Terhitung mulai hari, tanggal : Rabu, 05 Februari 2025, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.



Gorontalo, 05 Februari 2025

**Mengetahui,
Kepala Perpustakaan**

Apriyanto Alhamad, M.Kom

NIDN : 0924048601



KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
 Jl. Achmad Najamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI
No. 032/FIKOM-UIG/R/II/2025

Yang bertanda tangan di bawah ini :

Nama : Irvan Abraham Salihi, M.Kom
 NIDN : 0928028101
 Jabatan : Dekan Fakultas Ilmu Komputer

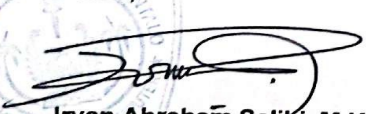
Dengan ini menerangkan bahwa :

Nama Mahasiswa : Gusti Nuzlan Djafri
 NIM : T3121124
 Program Studi : Teknik Informatika (S1)
 Fakultas : Fakultas Ilmu Komputer
 Judul Skripsi : Klafikasi Jenis Patah Tulang Pada Manusia
 Menggunakan Metode Convolutional Neural Network

Sesuai hasil pengecekan tingkat kemiripan skripsi melalui aplikasi **Turnitin** untuk judul skripsi di atas diperoleh hasil *Similarity* sebesar **13%**, berdasarkan Peraturan Rektor No. 32 Tahun 2019 tentang Pendeteksian Plagiat pada Setiap Karya Ilmiah di Lingkungan Universitas Ichsan Gorontalo dan persyaratan pemberian surat rekomendasi verifikasi calon wisudawan dari LLDIKTI Wil. XVI, bahwa batas kemiripan skripsi maksimal 30%, untuk itu skripsi tersebut di atas dinyatakan **BEBAS PLAGIASI** dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Mengetahui
 Dekan,


Irvan Abraham Salihi, M.Kom
 NIDN. 0928028101

Gorontalo, 08 Februari 2025
 Tim Verifikasi,


Zulfrianto S. Lamasigi, M.Kom
 NIDN. 0914089101

Terlampir :
 Hasil Pengecekan Turnitin

Fikom01 Unisan

KLASIFIKASI JENIS PATAH TULANG PADA MANUSIA MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWO...

-  FAKULTAS ILMU KOMPUTER
-  Fak. Ilmu Komputer
-  LL Dikti IX Turnitin Consortium

Document Details

Submission ID**tm:oid:::1:3147360932****Submission Date****Feb 6, 2025, 10:40 AM GMT+7****Download Date****Feb 6, 2025, 10:43 AM GMT+7****File Name****SKRIPSI_T3121124_GUSTI_NUZLAN_DJAFRI_gustinuzlan12gmail.pdf****File Size****2.8 MB****65 Pages****10,554 Words****66,803**

13% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

Bibliograp

hy Quoted Text

Top Sources

- 0%  Internet sources
- 11%  Publications
- 11%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

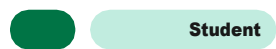
Top Sources

0%	Internet sources
11%	Publications
11%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

	Student		
	Udayana University	1%	
	Student		
	Universitas Brawijaya	<1%	
	Public		
	Heri Darmanto. "Pengenalan Spesies Ikan Berdasarkan Kontur Otolith Mengguna...	<1%	
	Student		
	Jefferson County School District No. R-1	<1%	
	Student		
	Universitas Islam Majapahit	<1%	
	Student		
	LL Dikti IX Turnitin Consortium	<1%	
	Public		
	"Advances in Neuro-Information Processing", Springer Nature, 2009	<1%	
	Student		
	Ajou University Graduate School	<1%	
	Public		
	Alfito Herdiansyah, Arief Hermawan, Sutarman. "Implementasi Algoritma Convo...	<1%	
	Student		
	Forum Komunikasi Perpustakaan Perguruan Tinggi Kristen Indonesia (FKPPTKI)	<1%	



Submitted on 1689585424656

<1%

Student

Bogazici University <1%

Public

Aang Alim Murtopo, Maulana Aditdya, Pingky Septiana Ananda, Gunawan Gunaw... <1%

Public

Novia Pramesti Aprilia, Theresia Herlina Rochadiani. "Image Captioning untuk Ga... <1%

Public

Didi Kurniawan, Dhani Ariatmanto. "IDENTIFIKASI VARIETAS BIBIT DURIAN MEN... <1%

Student

Harrisburg University of Science and Technology <1%

Student

Tarumanagara University <1%

Public

Gordianus Gampur, I Wayan Ordiyasa, Sri Hasta Mulyani. "Klasifikasi Jenis Pisang ... <1%

Public

Ivo Colanus Rally Drajana, Andi Bode. "Support Vector Machine Untuk Prediksi Pr... <1%

Student

Universitas Trunojoyo <1%

Public

P H Peckham. "Upper and lower motor neuron lesions in the upper extremity mu... <1%

Public

Xiaofeng Hu, Jianmin Chen, Xiaofei Zheng, Jianmei Li, Mingwei Zhou. "Establishm... <1%

Student

Syntax Corporation <1%

Student

UIN Sunan Ampel Surabaya

<1%

Student

UPN Veteran Jawa Timur

<1%

	Student	
Universitas Gunadarma		<1%
	Student	
Universitas Airlangga		<1%
	Student	
Universitas Budi Luhur		<1%
	Student	
Universitas Pelita Harapan		<1%
	Student	
Sriwijaya University		<1%
	Student	
University of Carthage		<1%
	Public	
Rindi Kusumawardani, Putu Dana Karningsih. "Detection and Classification of Ca...		<1%
	Student	
Universitas Maritim Raja Ali Haji		<1%



KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI

UNIVERSITAS ICHSAN GORONTALO

FAKULTAS ILMU KOMPUTER

SK MENDIKNAS NOMOR 84/D/O/2001

JL. Achmad Nadjamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

Berita Acara Perbaikan/Revisi Ujian Skripsi

Pada hari ini, Kamis 20 Februari 2025, Pukul 13.00-14.30 Wita. Telah dilaksanakan Ujian Skripsi mahasiswa Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Nama : Gusti Nuzlan Djafri
Nim : T3121124
Pembimbing I : Yasin Aril Mustofa, M.Kom
Pembimbing II : Sudirman S. Panna, M.Kom
Judul Skripsi : Klasifikasi Jenis Patah Tulang Pada Manusia Menggunakan Metode Convolutional Neural Network (Cnn)

Oleh Komite Seminar sebagai berikut :

No	Komite Seminar	Status	Tanda Tangan
1	Zohrahayaty, M.Kom	Ketua	
2	Rofiq Harun, M.Kom	Anggota	
3	Yusrianto Malago, M.Kom	Anggota	
4	Yasin Aril Mustofa, M.Kom	Anggota	
5	Sudirman S. Panna, M.Kom	Anggota	