



KERJA PRAKTIK - IF234603

PREDIKSI KECELAKAAN DENGAN MODEL BERBASIS KONVOLUSI

Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo Surabaya 60111
Periode: 30 Agustus 2024 – 30 November 2024

Oleh:

Richard Ryan

5025211141

Pembimbing Jurusan

Dr. Ahmad Saikhu, S.Si., M.T.

Pembimbing Lapangan

Dr. Eng. Muhammad Hilmil Muchtar Aditya Pradana, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2025



KERJA PRAKTIK - IF234603

PREDIKSI KECELAKAAN DENGAN MODEL BERBASIS KONVOLUSI

Departemen Teknik Informatika, Institut Teknologi Sepuluh
Nopember

Kampus ITS Sukolilo Surabaya 60111

Periode: 30 Agustus 2024 - 30 November 2024

Oleh:

Richard Ryan

5025211141

Pembimbing Jurusan

Dr. Ahmad Saikhu, S.Si., M.T.

Pembimbing Lapangan

Dr. Eng. Muhammad Hilmi Muchtar Aditya Pradana, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2025

DAFTAR ISI

DAFTAR ISI.....	iii
DAFTAR GAMBAR	vii
LEMBAR PENGESAHAN.....	ix
KATA PENGANTAR.....	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Manfaat	2
1.4. Rumusan Masalah.....	2
1.5. Lokasi dan Waktu Kerja Praktik.....	3
1.6. Metodologi Kerja Praktik.....	3
1.6.1. Perumusan Masalah.....	3
1.6.2. Studi Literatur.....	3
1.6.3. Pengumpulan, Eksplorasi, dan <i>Preprocessing</i> Data	3
1.6.4. Pelatihan Model Konvolusi	4
1.6.5. Evaluasi Model	4
1.6.6. Kesimpulan dan Saran	4
1.7. Sistematika Laporan.....	4
1.7.1. Bab I Pendahuluan.....	4
1.7.2. Bab II Profil Instansi.....	4
1.7.3. Bab III Tinjauan Pustaka	4

1.7.4.	Bab IV Implementasi.....	4
1.7.5.	Bab V Pengujian dan Evaluasi	5
1.7.6.	Bab VI Kesimpulan dan Saran	5
BAB II PROFIL INSTANSI		7
2.1.	Sejarah Instansi	7
2.2.	Visi Instansi	8
2.3.	Misi Instansi	8
2.4.	Struktur Instansi	9
2.5.	Laboratorium.....	11
2.6.	Lokasi.....	14
BAB III TINJAUAN PUSTAKA.....		15
3.1.	Action Prediction	15
3.2.	CNN.....	15
3.3.	MoViNet	16
BAB IV IMPLEMENTASI.....		19
4.1.	<i>Dataset</i>	19
4.2.	<i>Preprocessing</i>	20
4.3.	Augmentasi	21
4.4.	Training.....	22
4.5.	Evaluasi.....	23
BAB V PENGUJIAN DAN EVALUASI		25
5.1.	Tujuan Pengujian	25
5.2.	Kriteria Pengujian	25

5.3.	Skenarion Pengujian	25
5.4.	Hasil Pengujian	25
5.5.	Evaluasi Pengujian.....	28
BAB VI KESIMPULAN DAN SARAN.....		31
7.1.	Kesimpulan	31
7.2.	Saran	32
DAFTAR PUSTAKA.....		33
LAMPIRAN		35
LAMPIRAN 1. Augmentasi Video		35
LAMPIRAN 2. Pembuatan Model		36
LAMPIRAN 3. Interpolasi Berbobot		36
LAMPIRAN 4. Pengambilan Frame		37
BIODATA PENULIS.....		Error! Bookmark not defined.

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2. 1 Struktur organisasi Departemen Teknik Informatika ITS.....	11
Gambar 5. 1 Confusion matrix prediksi kecelakaan biner class..	26
Gambar 5. 2 Confusion matrix prediksi kecelakaan 4-class	27
Gambar 5. 3 Confusion matrix prediksi kecelakaan 7-class	28

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

Prediksi Kecelakaan dengan Model Berbasis Konvolusi

Oleh:

Richard Ryan

5025211141

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr. Ahmad Saikhu, S.Si.,
M.T.
NIP. 197107182006041001



(Pembimbing Departemen)

2. Dr. Eng. Muhammad Hilmil
Muchtart Aditya Pradana,
S.Kom., M.Sc.
NIP. 199205162024061001



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Prediksi Kecelakaan dengan Model Berbasis Konvolusi

Nama Mahasiswa : Richard Ryan
NRP : 5025211141
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Dr. Ahmad Saikhu, S.Si, M.T.,
Pembimbing Lapangan : Dr. Eng. Muhammad Hilmil
Mughtar Aditya Pradana, S.Kom,
M.Sc..

ABSTRAK

Advanced Driver Assistance Systems (ADAS) menjadi salah satu teknologi penting dalam meningkatkan keselamatan berkendara dengan memprediksi dan mendeteksi potensi kecelakaan. Penelitian ini bertujuan untuk mengembangkan model prediksi kecelakaan berbasis video dashcam menggunakan arsitektur MoViNet. Dataset yang digunakan adalah DADA-2000 yang telah diproses ulang untuk memperbaiki inkonsistensi, seperti kesalahan anotasi dan penghapusan data yang tidak relevan. Dataset dilabeli ulang menjadi empat skenario klasifikasi: biner, 4-class, 7-class, dan 16-class.

Metode yang digunakan meliputi preprocessing data, augmentasi menggunakan library Albumentations, dan pelatihan model dengan fine-tuning arsitektur MoViNet versi A2. Evaluasi model dilakukan menggunakan metrik recall, presisi, dan akurasi, serta confusion matrix untuk klasifikasi multi-class.

Hasil pengujian menunjukkan bahwa model memiliki tingkat recall yang tinggi pada klasifikasi biner, namun rendah dalam presisi karena seringnya kondisi normal diklasifikasikan sebagai kecelakaan. Pada klasifikasi multi-class, model

menunjukkan performa yang rendah, dengan akurasi yang tidak konsisten dalam membedakan jenis kecelakaan.

Penelitian ini menyimpulkan bahwa model MoViNet memiliki potensi dalam mendeteksi kecelakaan, namun masih memerlukan peningkatan, terutama dalam skenario multi-class. Saran untuk pengembangan meliputi eksplorasi arsitektur MoViNet lainnya dan penerapan variasi augmentasi data untuk meningkatkan kemampuan generalisasi model terhadap data baru.

Kata Kunci : Advanced Driver Assistance Systems (ADAS), Action Prediction, MoViNet

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Prediksi Kecelakaan dengan Model Berbasis Konvolusi.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Dr. Ahmad Saikhu, S.Si., M.T. selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Bapak Dr. Eng. Muhamad Hilmil Muchtar Aditya Pradana, S.Kom., M.Sc. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 20 Mei 2025

Richard Ryan

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kecelakaan jalan raya telah menjadi masalah global yang krusial, hal ini mengancam keselamatan publik dan stabilitas ekonomi (Ahmed et al., 2023). Menurut *World Health Organization* (WHO), kecelakaan lalu lintas, menyebabkan sekitar 1,2 juta kematian setiap tahun dan meninggalkan puluhan jutaan orang lainnya terluka atau cacat (World Health Organization, 2023). Insiden-insiden ini tidak hanya menyebabkan kehilangan nyawa, tetapi juga memberikan beban ekonomi yang berat bagi pemerintah, keluarga, dan individu, dengan perkiraan biaya mencapai 2.5% dari produk domestik bruto (PDB) negara-negara (Elvik, 2000).

Untuk mengatasi masalah ini, analisis prediktif muncul sebagai pendekatan yang menjanjikan untuk mencegah kecelakaan melalui identifikasi faktor risiko dan memberikan wawasan yang dapat ditindaklanjuti. Dengan memanfaatkan data historis dan teknik komputasi yang canggih, sistem prediktif dapat menjadi alat yang penting dalam manajemen keselamatan di jalan raya. Kemajuan terkini dalam bidang *deep learning*, khususnya *Convolutional Neural Network* (CNN), telah menunjukkan potensi besar dalam mengekstrak pola-pola bermakna dari data yang kompleks. Meskipun awalnya dirancang untuk pemrosesan gambar, CNN telah terbukti efektif di berbagai bidang, termasuk pemrosesan video, *Natural Language Processing* (NLP), pengenalan suara, dan analisis *time-series*. Kemampuan CNN untuk mempelajari fitur hierarkis secara otomatis membuatnya sangat cocok untuk tugas prediksi kecelakaan, yang melibatkan data dengan informasi spasial, temporal, dan kontekstual.

Kerja Praktik ini bertujuan untuk mengeksplorasi penerapan metode deep learning berbasis konvolusional dalam prediksi kecelakaan. Dengan menggunakan arsitektur berbasis konvolusi, studi ini bertujuan untuk meningkatkan akurasi prediksi kecelakaan. Pendekatan ini berpotensi untuk merevolusi strategi deteksi dini kecelakaan dan memberikan kontribusi pada terciptanya jalan raya yang lebih aman.

1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 4 sks dan membantu pengembangan mobil otonom melalui sistem prediksi kecelakaan menggunakan *dashcam*.

1.3. Manfaat

Manfaat yang diperoleh dari kerja praktik ini adalah dibuatnya metode prediksi dini kecelakaan dengan metode berbasis deep learning. Model yang dikembangkan dapat memprediksi kecelakaan sehingga dapat dilakukan tindakan proaktif untuk mencegah kecelakaan tersebut. Dengan tindakan proaktif tersebut, diharapkan jumlah kecelakaan yang terjadi dapat menurun sehingga beban ekonomi yang disebabkan oleh biaya medis dan infrastruktur dapat digunakan untuk hal lainnya

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

- 1) Apa metrik yang digunakan untuk mendapatkan performa model yang dikembangkan?

- 2) Bagaimana melatih model berbasis konvolusi untuk mendapatkan prediksi dini kecelakaan?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik dilakukan di Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember.

Adapun kerja praktik dimulai pada tanggal 30 September 2024 hingga 30 November 2024.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1. Perumusan Masalah

Pada tahap ini kami akan mengidentifikasi permasalahan yang ada dan seluruh kebutuhan yang terkait dengan permasalahan tersebut. Setelah itu akan dibentuk rumusan masalah yang akan diselesaikan.

1.6.2. Studi Literatur

Setelah rumusan masalah dibuat, akan dilakukan studi literatur untuk mendukung implementasi. Pada tahap ini akan dilakukan pencarian, pengumpulan informasi, dan pembelajaran topik yang berkaitan dengan implementasi model *deep learning* yang akan dikembangkan. Informasi ini diperoleh melalui *website* dan riset-riset terdahulu.

1.6.3. Pengumpulan, Eksplorasi, dan *Preprocessing* Data

Pada tahap ini akan dicari *dataset* yang relevan dengan topik penelitian. Setelah *dataset* ditemukan, dilakukan eksplorasi terhadap *dataset* tersebut. Eksplorasi meliputi jenis data yang digunakan, jumlah *dataset*, dan jenis label. Informasi yang didapat melalui tahap eksplorasi diproses terlebih dahulu sebelum *dataset* dapat digunakan untuk melatih model *deep learning* yang akan dikembangkan, tahap

preprocessing ini meliputi perubahan resolusi dan perubahan jumlah *channel* video.

1.6.4. Pelatihan Model Konvolusi

Setelah model di-*preprocess*, arsitektur *deep learning* dibangun. Setelah itu, *dataset* dapat diaugmentasi dan dimasukkan ke model untuk dilakukan pelatihan. Augmentasi *dataset* dilakukan dengan menambah jumlah data yang ada untuk mengurangi resiko terjadinya *overfitting* dengan melakukan variasi seperti pencerminan, rotasi, perubahan kontras, dan lain-lain.

1.6.5. Evaluasi Model

Performa model dievaluasi terhadap *dataset* pengujian berdasarkan metrik yang telah ditentukan sebelumnya seperti akurasi, presisi, *recall*, atau F1.

1.6.6. Kesimpulan dan Saran

Kesimpulan dan saran diambil untuk penelitian serupa di masa mendatang

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Instansi

Bab ini berisi gambaran umum Departemen Teknik Informatika ITS mulai dari sejarah, visi-misi, struktur, laboratorium, dan lokasi instansi.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Implementasi

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses eksplorasi data dan pelatihan model.

1.7.5. Bab V Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari model yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.6. Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL INSTANSI

2.1. Sejarah Instansi

Invasi teknologi informasi telah terasa dalam berbagai aspek kehidupan manusia dan akan menjadi tulang punggung pertumbuhan ekonomi negara, baik sekarang maupun di masa depan. Pemerintah sangat menyadari hal ini, sehingga sejak Repelita V yang lalu, mereka telah menetapkan bahwa pendidikan tinggi di bidang komputer dan informatika harus diprioritaskan, bersama dengan kursus di bidang lain seperti rekayasa, perilaku, manajemen, akuntansi, dan kesenian. Pendidikan tinggi dirancang dalam mempersiapkan Indonesia untuk era pembangunan industri dan teknologi informasi. Dengan demikian, pada tahun 1985, Direktorat Jenderal Pendidikan Tinggi pemerintah memutuskan untuk membuka program studi S1 baru di bidang ilmu teknologi komputer di empat universitas atau lembaga pendidikan yang memiliki program ITS. Pada awalnya, program di ITS disebut Program Studi Teknik Komputer. Namun, pada tahun 1993, namanya diubah menjadi Jurusan Teknik Komputer. Pada tahun 1996, Surat Keputusan Direktur Jenderal Pendidikan Tinggi Nomor 224/DIKTI/Kep/1996 dikeluarkan. Menurut Surat Keputusan Nomor 003/BAN-PT/Ak-X/S1/V/2006 dari Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) yang dikeluarkan pada tanggal 18 Mei 2006, Jurusan Teknik Informatika menerima nilai akreditasi A. Selain program Sarjana (S1), Jurusan Teknik Informatika juga menawarkan program Pasca Sarjana (S2). Program ini dimulai sejak tahun 1994 dan diizinkan oleh surat keputusan Direktur Jenderal Pendidikan Tinggi No. 2851/D/T/2001. Jurusan Teknik

Informatika juga memulai program Doktor (S3) pada tahun 2011.

2.2. Visi Instansi

Sejalan dengan visi ITS untuk menjadi perguruan tinggi dengan reputasi internasional dalam ilmu pengetahuan, teknologi, dan seni, terutama dalam membangun industri dan kelautan yang berwawasan lingkungan, Visi Departemen Informatika adalah menjadi inovator bidang informatika yang unggul di tingkat nasional dengan reputasi internasional serta berperan aktif dalam upaya memajukan dan mensejahterakan bangsa. Visi Departemen Teknik Informatika adalah menjadi lembaga pendidikan dan penelitian di bidang informatika yang unggul di tingkat nasional dan memiliki reputasi internasional.

2.3. Misi Instansi

Berikut adalah misi-misi yang dimiliki oleh Departemen Teknik Informatika ITS:

- 1) Menyelenggarakan proses pembelajaran yang berkualitas, dan memenuhi standar nasional maupun internasional.
- 2) Melaksanakan penelitian yang inovatif, bermutu, dan bermanfaat.
- 3) Meningkatkan pemanfaatan teknologi informasi dan komunikasi untuk masyarakat.
- 4) Menjalin kemitraan dengan berbagai lembaga, baik di dalam maupun di luar negeri.

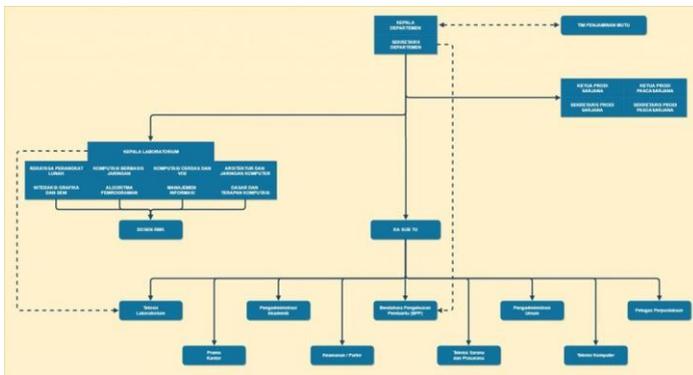
2.4. Struktur Instansi

Struktur organisasi pada Teknik Informatika dapat dilihat di Gambar 2.1 dengan daftar penjabat sebagai berikut:

- 1) Kepala dan Sekretaris Departemen
 - a. Kepala: Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
 - b. Sekretaris I: Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D.
 - c. Sekretaris II: Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
- 2) Kepala Program Studi
 - a. Kepala Pasca Sarjanan :Dr. Ahmad Saikhu, S.Si., M.T.
 - b. Kepala Program Studi Rekayasa Perangkat Lunak: Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
 - c. Kepala Program Studi Rekayasa Kecerdasan Artifisial: Dr. Anny Yuniarti, S.Kom., M.Comp.Sc.
- 3) Koordinator Tim Konsultasi Kemahasiswaan: Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.d
- 4) Kepala Laboratorium
 - a. Laboratorium Rekayasa Perangkat Lunak (RPL): Dr.Ir. Siti Rochimah, M.T.
 - b. Laboratorium Komputasi Berbasis Jaringan (KBJ): Tohari Ahmad, S.Kom, M.IT., Ph.D.
 - c. Laboratorium Komputasi Cerdas dan Visi (KCV): Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

- d. Laboratorium Teknologi Jaringan dan Keamanan Siber Cerdas (NETICS): Dr. Baskoro Adi P., S.Kom., M.Kom.
 - e. Laboratorium Grafika, Interaksi dan Game (GIGa): Imam Kuswardayan, S.Kom., M.T.
 - f. Laboratorium Algoritma & Pemrograman (AP): Dr. Dwi Sunaryono, S.Kom., M.Kom.
 - g. Laboratorium Manajemen Cerdas Informasi (MCI): Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
 - h. Laboratorium Pemodelan dan Komputasi Terapan (PKT): Victor Hariadi, S.Si., M.Kom.
- 5) Kasubag Tata Usaha: M. Chusaeni, S.T.
 - 6) Pengadministrasi Akademik
 - a. Prodi Sarjana: Yudi Mulyono, Inka Novita Dinia, S.Si., dan Rif'atus Sholichah
 - b. Prodi Pascasarjana: Lina Ambarwati, S.Kom. dan Kartika Dwi Yuniavita Rahmawati, S.T.
 - 7) Pengadministrasi Umum: Soegeng Santoso dan Supriyo
 - 8) Bendahara Pengeluaran Pembantu (BPP): Chafrida Devi dan Ayu Bahraeni Pramesti
 - 9) Teknisi Sarana dan Prasarana: Hari Pramono, S.E.
 - 10) Teknisi Komputer: Edy Lukito, S.T.
 - 11) Teknisi Laboratorium
 - a. RPL: Jumali, S.Sos.
 - b. KBJ: Jumali, S.Sos.
 - c. KCV: Ahmad Junaidi Abdillah, S.Kom.
 - d. NETICS: Kunto Hermono
 - e. GIGa: Gayuh Adi Rustanto, S.Kom.
 - f. AP: Gayuh Adi Rustanto, S.Kom.

- g. MCI: Gayuh Adi Rustanto, S.Kom.
 - h. PKT: Ahmad Junaidi Abdillah, S.Kom.
- 12) Teknisi Workshop
- a. LP1: Ahmad Junaidi Abdillah, S.Kom.
 - b. LP2: Edy Lukito, S.T.
- 13) Residensi S2/S3: Kunto Hermono
- 14) Pramu Kantor: Mochamad Suchron
- 15) Keamanan / Parkir: Purnomo



Gambar 2. 1 Struktur organisasi Departemen Teknik Informatika ITS

2.5. Laboratorium

Mahasiswa Departemen Teknik Informatika ITS memiliki akses ke fasilitas laboratorium yang terdiri dari laboratorium bidang minat dan laboratorium workshop. Jenis laboratorium bidang minat mencakup berbagai mata kuliah yang dapat dipelajari oleh mahasiswa Departemen Teknik Informatika, seperti berikut:

- 1) Laboratorium Rekayasa Perangkat Lunak

Keahlian dalam melakukan pengujian perangkat lunak, mengelola proyek perangkat lunak, mengurangi resiko kesalahan perangkat lunak, dan membuat perangkat lunak game adalah bidang minat di laboratorium ini.

2) Laboratorium Komputasi Cerdas dan Visi

Kemampuan lulusan dalam memanipulasi dan menganalisis data citra dalam berbagai bidang aplikasi (AI, biomedika, industri), menerapkan metode sistem cerdas dalam berbagai bidang aplikasi, memodelkan dan mengoptimalkan sistem nyata adalah bidang keahlian yang ditekankan di laboratorium ini.

3) Laboratorium Grafika, Interaksi, dan Game

Di bidang minat ini, laboratorium menawarkan keahlian yang menekankan kemampuan lulusan dalam mendesain, mengembangkan, mencatat proses pembuatan game yang standar, serta menggunakan game engine untuk membuat model tiga dimensi, pemrograman, dan aplikasi realitas virtual tiga dimensi.

4) Laboratorium Manajemen Cerdas Informasi

Laboratorium di bidang minat ini menawarkan bidang keahlian yang ditekankan pada kemampuan lulusan dalam menganalisis, mensintesis, dan mengevaluasi proses bisnis dan sistem informasi, memasukkan rekayasa pengetahuan ke dalam aplikasi, melakukan investigasi, pengujian, dan evaluasi kematangan dan kepatutan prosedur dan tata kelola teknologi informasi.

5) Laboratorium Komputasi Berbasis Jaringan

Di laboratorium ini, bidang keahlian yang ditekankan adalah kemampuan sarjana, magister, dan doktor dalam membangun infrastruktur jaringan yang aman, sistem grid, aplikasi standar jaringan, dan aplikasi multimedia berbasis jaringan.

6) Laboratorium Teknologi Jaringan dan Keamanan Siber Cerdas

Laboratorium bidang minat ini menawarkan bidang keahlian yang menekankan kemampuan lulusan dalam menerapkan keamanan jaringan dan berbagai arsitektur jaringan yang sesuai dengan standar teknologi terbaru.

7) Laboratorium Algoritma dan Pemrograman

Kemampuan untuk merancang dan menganalisa algoritma berdasarkan kaidah pemrograman yang kuat untuk menyelesaikan masalah secara efektif dan efisien, kemampuan untuk menggunakan model pemrograman yang ada mendasari berbagai bahasa pemrograman yang ada, kemampuan untuk memilih bahasa pemrograman untuk menghasilkan aplikasi yang sesuai, seperti mengembangkan sistem atau aplikasi berbasis kerangka kerja, dan peran yang sesuai di laboratorium ini adalah bidang keahlian yang ditekankan.

8) Laboratorium Pemodelan dan Komputasi Terapan

Pemodelan dan simulasi, peramalan sains, optimasi, dan komputasional saintifik adalah semua bidang di mana laboratorium ini menerima penelitian dan kerja sama industri.

Laboratorium workshop yang disediakan oleh Departemen Teknik Informatika adalah sebagai berikut:

1) Workshop Pemrograman 1

Laboratorium ini memiliki fasilitas unggulan dalam hal teknologi manajemen komputer dan fasilitas fisik representatif (PC, sistem suara video, jaringan). Selain itu, ruangnya dapat menampung sebanyak 54 orang.

2) Workshop Pemrograman 2

Laboratorium ini memiliki fasilitas unggulan dalam hal teknologi manajemen komputer dan fasilitas fisik representatif (PC, sistem suara video, jaringan). Selain itu, ruangnya dapat menampung sebanyak 54 orang.

3) Laboratorium Pascasarjana S2

Laboratorium ini memungkinkan mahasiswa program master untuk menyelesaikan tugas kuliah dan tesis, seperti studi literatur, uji coba aplikasi dan data, dan penulisan tesis. Mereka juga dapat menggunakan laboratorium bidang minat mereka sendiri.

4) Laboratorium Pascasarjana S3

Mahasiswa program doktor dapat menggunakan laboratorium ini untuk menyelesaikan tugas kuliah dan disertasi, seperti studi literatur, uji coba aplikasi dan data, dan penulisan disertasi. Laboratorium Pascasarjana S3 terdiri dari tiga lantai. Laboratorium S3 di lantai 1 digunakan oleh mahasiswa S3 yang belum mendaftar, Laboratorium Pascasarjana di lantai 3 digunakan oleh mahasiswa S3 yang sudah mendaftar, dan Laboratorium Kerjasama di lantai 3 digunakan oleh mahasiswa S3 yang bekerja sama. Mahasiswa program doktor juga dapat menggunakan laboratorium bidang minat mereka sesuai dengan bidang penelitian mereka.

2.6. Lokasi

Departemen Teknik Informatika, Kampus ITS
Sukolilo, Surabaya 60111.

BAB III

TINJAUAN PUSTAKA

3.1. Action Prediction

Action prediction adalah kemampuan untuk mengantisipasi dan mengenali tindakan yang sedang berlangsung sebelum tindakan tersebut selesai sepenuhnya (Monroy et al., 2021). *Action prediction* memainkan peran penting dalam perkembangan sosial-kognitif dan telah diterapkan di berbagai bidang, seperti prediksi tindakan pada bayi secara *realtime* dan dalam *computer vision* menggunakan *advanced deep learning*.

Generative adversarial networks telah digunakan untuk meningkatkan kemampuan membedakan fitur dalam *action prediction* awal pada video (Wang et al., 2019). Untuk *3D skeleton sequences*, *dilated convolutional networks* dengan *scale selection methods* telah menunjukkan hasil yang menjanjikan dalam *online action prediction* (J. Liu et al., 2020).

Salah satu penelitian terbaru mengusulkan *multi-task frameworks* yang menggabungkan *model action semantic prediction* dan *future human motion prediction*. Hasil dari penelitian ini menunjukkan peningkatan kinerja dengan memanfaatkan korelasi antara keduanya (X. Liu & Yin, 2022). Berbagai pendekatan ini menunjukkan upaya yang sedang berlangsung untuk memajukan kemampuan *action prediction* pada berbagai bidang.

3.2. CNN

Convolutional Neural Networks (CNN) adalah algoritma *deep learning* yang banyak digunakan untuk berbagai keperluan, termasuk *image classification*, *object detection*, dan *radiology* (Mohammadpour et al., 2022), (Yamashita

et al., 2018). CNN dirancang untuk secara otomatis mempelajari *spatial hierarchies of features* melalui *backpropagation* dengan meniru korteks visual manusia (Purwono et al., 2022), (Yamashita et al., 2018). Struktur dasar CNN terdiri dari beberapa *layer* yaitu *convolutional*, *pooling*, *fully connected*, dan *non-linear layers*. *Convolutional layer* menggunakan filter kernel untuk mengekstrak fitur fundamental. *Pooling layer* menggabungkan *convolutional layer* yang berurutan. Arsitektur CNN yang populer diantaranya *LeNet*, *AlexNet*, dan *VGGNet*. Kuo (Kuo, 2017) menginterpretasikan CNN sebagai implementasi dari *guided multilayer* RECOs (*rectified correlations on a sphere*). Terlepas dari keefektifannya, CNN memiliki beberapa tantangan dalam penerapannya terutama pada pada bidang radiologi termasuk dataset yang terbatas dan *overfitting* (Yamashita et al., 2018).

3.3. MoViNet

MoViNet (*Mobile Video Networks*) adalah *neural network architecture* yang dirancang untuk mengklasifikasikan video secara efisien secara *real time*. Arsitektur ini dirancang khusus untuk *action recognition* dalam video, dengan efisiensi yang memungkinkan implementasi pada perangkat dengan sumber daya terbatas seperti *smartphone* atau perangkat *IoT*. MoViNet menggabungkan fitur spasial dan temporal dari video untuk menghasilkan prediksi yang akurat (Hussain et al., 2023).

MoViNet dirancang untuk memungkinkan adanya *streaming inference*, artinya model dapat memproses video secara *frame-by-frame* untuk meminimalkan latensi tanpa harus memproses seluruh video secara bersamaan. Hal ini menjadikannya ideal untuk aplikasi berbasis *real-*

time seperti pengawasan, analisis video, dan *human-machine interaction*.

MoViNet terdiri dari tiga teknik utama yaitu *stream buffer*, *neural architecture search* (NAS), dan *ensembling*. *Stream buffer* memisahkan memori dari durasi klip video, memungkinkan 3D CNN untuk mengolah urutan video streaming dengan panjang beragam tanpa peningkatan penggunaan memori. NAS digunakan untuk merancang arsitektur 3D CNN yang efisien dan meningkatkan akurasi tanpa mengorbankan efisiensi. Serta *ensembling* merupakan teknik untuk meningkatkan akurasi lebih lanjut tanpa mengorbankan efisiensi.

MoViNet telah mencapai akurasi dan efisiensi tinggi pada berbagai dataset video *action recognition* seperti Kinetics, Moments in Time, dan Charades12. Misalnya, MoViNet-A5-Stream mencapai akurasi yang sama dengan X3D-XL pada Kinetics 600 dengan penggunaan FLOPs 80% lebih sedikit dan memori 65% lebih rendah (Kondratyuk et al., 2021). MoViNet membuka peluang baru untuk aplikasi kamera mobile, IoT, dan *autonomous vehicle* di mana pemrosesan *on-device* yang efisien dan akurat sangat penting.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Bab ini menjelaskan tahapan implementasi yang dilakukan pada kerja praktik. Tahapan ini mulai dari pengumpulan dan eksplorasi data, data *preprocessing*, augmentasi data, pelatihan model, dan evaluasi.

4.1. Dataset

DADA-2000 merupakan *dataset* yang berisi video *dashcam* terkait perhatian pengemudi dan kecelakaan. *Dataset* yang digunakan merupakan *dataset* DADA-2000 yang telah melalui tahap *preprocessing* terlebih dahulu. Pemrosesan ini dilakukan karena ditemukan inkonsistensi pada *dataset* DADA-2000 seperti kesalahan anotasi dan video berasal dari game (Pradana et al., 2022). *Preprocessing* yang dilakukan adalah pengambilan video-video dimana perekam terlibat dalam kecelakaan serta dilabeli ulang menjadi 4 jenis label seperti berikut:

1) 4-class

Label 4 class menjelaskan jenis kecelakaan yang dialami oleh perekam. Label ini meliputi tabrakan dengan pejalan kaki, tabrakan dengan kendaraan lain, tabrakan dengan artefak di jalan, atau keadaan normal.

2) 7-class

Label 7 class merupakan perincian lebih lanjut dari setiap jenis kecelakaan di 4-class. Label ini meliputi tabrakan dengan pejalan kaki, tabrakan dengan sepeda, tabrakan dengan sepeda motor, tabrakan dengan truk, tabrakan dengan mobil lain, self-accident, atau keadaan normal.

3) 16-class

Label 16 class merupakan perincian lebih lanjut dari keadaan normal di 7-class. Label ini meliputi pejalan kaki lewat, tabrakan dengan pejalan kaki, sepeda lewat,

tabrakan dengan sepeda, sepeda motor lewat, tabrakan dengan sepeda motor, truk lewat, tabrakan dengan truk, disalip oleh truk, mobil lewat, tabrakan dengan mobil, disalip oleh mobil, tabrakan dengan pembatas jalan, tabrakan dengan fasilitas jalan, self-accident, atau keadaan normal.

4) Miss end

Label *miss end* menjelaskan nomor *frame* dimulainya tabrakan. Label *miss end* menjadi negatif pada tiga kasus khusus yaitu:

a. Miss end = -1

Miss end akan dilabeli menjadi -1 apabila tabrakan hampir terjadi namun tidak sampai bersentuhan. Pada kasus ini video dianggap keadaan normal sebab kecelakaan tidak terjadi

b. Miss end = -2

Miss end akan dilabeli menjadi -2 apabila tidak terdapat indikasi apapun dari video bahwa akan terjadi tabrakan, hal ini dapat terjadi karena beberapa faktor seperti tabrakan terjadi dari samping yang tidak tertangkap oleh *dashcam*

c. Miss end = -3

Miss end akan dilabeli menjadi -3 apabila keadaan normal dan tidak ada risiko kecelakaan yang terjadi.

4.2. Preprocessing

Dataset melalui tahap *preprocessing* terlebih dahulu sebelum digunakan dalam pelatihan model. Terdapat 2 jenis *preprocessing* yang dilakukan yaitu perubahan resolusi video dan pembuatan label biner baru untuk klasifikasi.

1) Perubahan resolusi video

Data yang diambil memiliki resolusi $1584\text{px} \times 660\text{px}$. Model Movinet A2 yang digunakan memiliki performa paling baik ketika resolusi adalah $224\text{px} \times 224\text{px}$, sehingga setiap video yang akan digunakan akan terlebih dahulu diubah resolusinya menjadi $224\text{px} \times 224\text{px}$. Hal ini dilakukan dengan mengubah resolusi setiap *frame* video dengan menggunakan fungsi *resize_with_pad()* dari *class image* pada *library TensorFlow*. Fungsi ini mengubah resolusi *frame* dan menerapkan zero padding apabila ukuran awal lebih kecil dari ukuran yang diinginkan. *Frame* kemudian digabungkan kembali untuk mendapatkan video dengan resolusi $224\text{px} \times 224\text{px}$.

2) Pembuatan label biner

Prediksi kecelakaan secara biner dibuat untuk memprediksi apakah suatu kecelakaan akan terjadi atau tidak. Tahap ini dapat dilakukan dengan memberi label biner berdasarkan label 4-class setiap video. Apabila label 4-class dari video tersebut ≤ 2 , maka artinya terdapat kecelakaan yang terjadi sehingga mereka akan memiliki label biner 1. Sedangkan apabila label 4-class dari video tersebut = 3, maka artinya tidak terdapat kecelakaan yang terjadi sehingga label biner video tersebut adalah 0 .

4.3. Augmentasi

Augmentasi *dataset* diperlukan karena jumlah data sedikit dan untuk mengurangi risiko terjadinya *overfitting*. Augmentasi dilakukan melalui *library Albumentations*. Terdapat 4 augmentasi yang dilakukan yaitu pencerminan dengan probabilitas 50%, rotasi 90 derajat dengan

probabilitas 50%, pengubahan kontras secara acak dengan probabilitas 30%, serta pengubahan *hue* serta saturasi secara acak dengan batas 20 dan probabilitas 30%. Seluruh augmentasi diterapkan menggunakan *library Albumentation*. Penghitungan probabilitas augmentasi diterapkan ke seluruh video dan bukan pada setiap *frame* sehingga informasi temporal pada video dapat terjaga. Kode augmentasi ini menggunakan bahasa *Python* dan dapat dilihat pada Lampiran 1.

4.4. Training

Training dimulai dengan membangun model digunakan. Model yang digunakan yaitu model Movinet versi A2 yang di-*fine tune* berdasarkan *dataset* yang dimiliki. Proses *fine tuning* diterapkan ke seluruh *layer* model, tidak hanya pada *layer* klasifikasi. *Optimizer* yang digunakan yaitu Adam dengan *learning rate* 0.001. *Loss* yang digunakan bertipe *Sparse Categorical Crossentropy* karena tujuannya adalah klasifikasi. Kode untuk melakukan tahapan ini dapat dilihat pada Lampiran 2. Model dilatih sebanyak 10 epoch dengan batch 16.

Selama proses training, *dataset* diambil secara bertahap dari *data loader* untuk diproses terlebih dahulu. Rentang *frame* dipilih berdasarkan nilai label *miss end*. Apabila *miss end* adalah -1 atau -3, maka batasannya yaitu minimal terdapat 2 frame diantara frame awal dan frame akhir. Apabila *miss end* adalah -2 maka hanya 2 frame pertama yang diambil. Dan apabila *miss end* merupakan bilangan positif maka terdapat batasan pengambilan titik akhir yaitu nilai *miss end* dan minimal terdapat 2 frame antara titik awal dan titik akhir

pengambilan. Setelah itu, dipastikan bahwa setiap frame yang diambil beresolusi $224\text{px} \times 224\text{px}$. Kemudian dilakukan interpolasi berbobot hingga didapatkan 8 frame akhir. Kode untuk interpolasi berbobot dapat dilihat pada Lampiran 3. Apabila *dataset* berjenis *training*, maka akan diterapkan augmentasi terlebih dahulu. Karena label *miss end* = -1 berjumlah sedikit, maka proses pengambilan ini akan dilakukan 1-5x secara acak. Kode untuk pengambilan data secara keseluruhan dapat dilihat pada Lampiran 4.

4.5. Evaluasi

Evaluasi performa model akan dilakukan dengan menggunakan grafik yang dihasilkan selama proses *training* dan *recall* dari hasil klasifikasi terhadap *testing dataset*.

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas tahap pengujian dan evaluasi terhadap model prediksi kecelakaan menggunakan MoViNet yang telah diimplementasikan. Fokus pengujian adalah untuk mengukur performa model dalam mendeteksi kejadian kecelakaan (*recall*) serta mengevaluasi akurasi prediksi tipe kecelakaan untuk berbagai skenario klasifikasi.

5.1. Tujuan Pengujian

Pengujian bertujuan untuk mengukur tingkat *recall* model dalam mendeteksi kejadian kecelakaan dan mengevaluasi akurasi model pada prediksi tipe kecelakaan untuk skenario klasifikasi biner, *4-class*, dan *7-class*.

5.2. Kriteria Pengujian

Kriteria pengujian meliputi:

- 1) *Recall*: Mengukur kemampuan model untuk mendeteksi kejadian kecelakaan dengan sensitivitas tinggi, memastikan tidak ada kejadian penting yang terlewat.
- 2) Akurasi Prediksi Tipe Kecelakaan: Mengevaluasi kemampuan model dalam membedakan tipe kecelakaan berdasarkan ground truth, dengan analisis berbasis *confusion matrix*.

5.3. Skenarion Pengujian

Skenario pengujian mencakup langkah-langkah berikut:

- 1) Menggunakan dataset testing untuk mengukur tingkat *recall* model pada prediksi kejadian kecelakaan.
- 2) Membandingkan hasil prediksi model dengan label *ground truth* pada klasifikasi biner, *4-class*, dan *7-class* menggunakan *confusion matrix*.

5.4. Hasil Pengujian

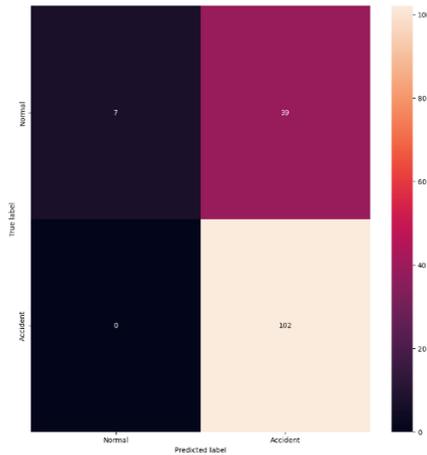
5.4.1. Recall Model pad Prediksi Kecelakaan

Hasil pengujian pada klasifikasi biner menunjukkan tingkat *recall* yang tinggi, dengan

semua kejadian kecelakaan berhasil terdeteksi. Namun, banyak data normal yang salah diklasifikasikan sebagai kecelakaan, sehingga presisi menjadi rendah.

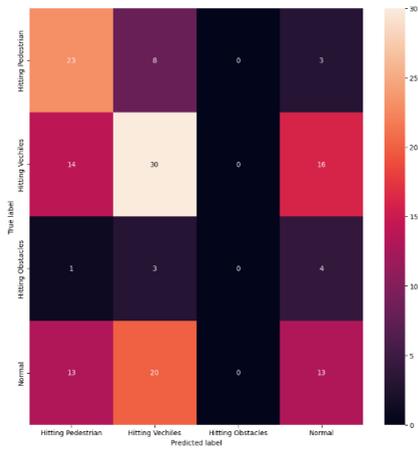
5.4.2. Akurasi Klasifikasi Tipe Kecelakaan

Evaluasi akurasi prediksi tipe kecelakaan divisualisasikan menggunakan *confusion matrix* berikut:



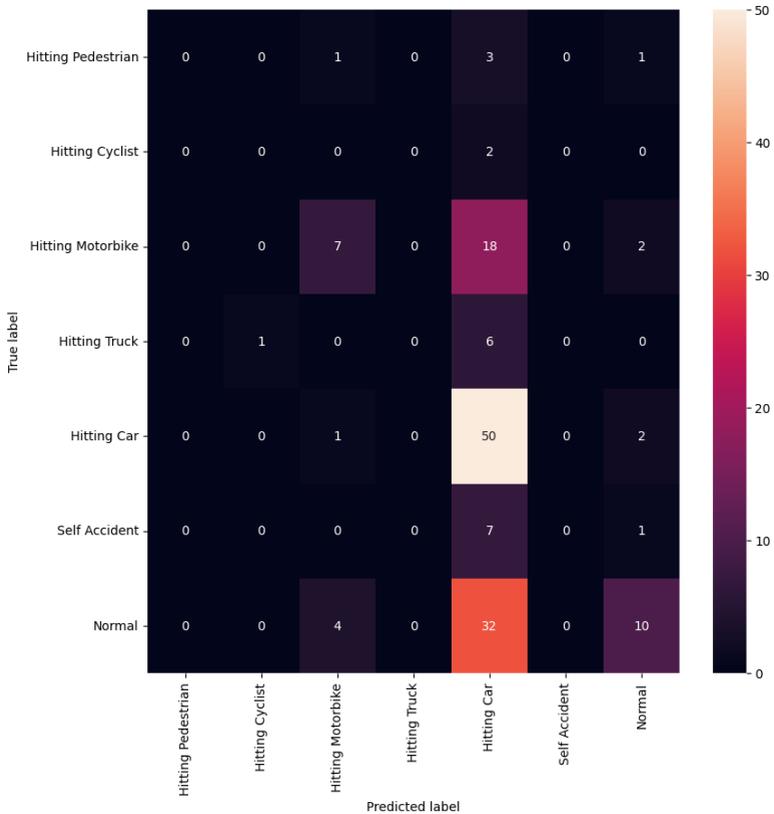
Gambar 5. 1 Confusion matrix prediksi kecelakaan biner class

Pada klasifikasi biner, semua kejadian kecelakaan berhasil diprediksi (*recall* tinggi), tetapi terdapat banyak data normal yang juga dianggap kecelakaan, menyebabkan presisi dan akurasi rendah.



Gambar 5. 2 Confusion matrix prediksi kecelakaan 4-class

Pada klasifikasi *4-class*, hasil prediksi model tersebar tidak merata di semua kategori. Model mengalami kesulitan dalam membedakan tipe kecelakaan dengan jelas.



Gambar 5. 3 Confusion matrix prediksi kecelakaan 7-class

Pada klasifikasi *7-class*, akurasi model sangat rendah. Prediksi model menunjukkan ketidakmampuan dalam menangkap karakteristik unik antar kategori kecelakaan.

5.5. Evaluasi Pengujian

Berdasarkan hasil pengujian:

1) Klasifikasi Biner

Model memiliki *recall* yang tinggi, sehingga andal dalam mendeteksi kejadian kecelakaan. Namun,

akurasi dan presisi rendah karena banyak data normal yang salah diklasifikasikan sebagai kecelakaan.

2) Klasifikasi *multi-class* (4-class dan 7-class)

Model menunjukkan performa yang rendah pada skenario klasifikasi dengan lebih banyak kategori. Hal ini menunjukkan bahwa model belum mampu menangkap perbedaan karakteristik antar tipe kecelakaan secara efektif.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini yaitu:

- 1) Evaluasi performa model dilakukan menggunakan beberapa metrik, termasuk *recall*, presisi, dan akurasi. Metrik-metrik ini memberikan gambaran menyeluruh tentang kemampuan model dalam mendeteksi kecelakaan. Hasil analisis menunjukkan bahwa model yang telah dikembangkan mampu memprediksi kejadian kecelakaan dengan tingkat *recall* yang tinggi pada klasifikasi biner. Tingginya nilai *recall* ini menunjukkan bahwa model memiliki sensitivitas yang baik dalam mengidentifikasi kejadian kecelakaan, sehingga dapat diandalkan dalam aplikasi yang memerlukan deteksi dini. Namun, meskipun banyak kecelakaan berhasil diprediksi, model juga cenderung mengklasifikasikan kondisi normal sebagai kecelakaan, yang mengakibatkan rendahnya presisi dan akurasi. Hal ini menjadi perhatian penting untuk pengembangan model lebih lanjut.
- 2) Performa model dalam memprediksi tipe kecelakaan masih belum optimal, terutama dalam skenario klasifikasi *multi-class* (*4-class* dan *7-class*). Akurasi yang rendah pada klasifikasi tipe kecelakaan menandakan bahwa model kesulitan dalam membedakan antara jenis kecelakaan yang berbeda. Hasil ini menunjukkan bahwa model memerlukan peningkatan, baik dari segi arsitektur maupun teknik pelatihan, untuk lebih baik menangkap perbedaan karakteristik antar tipe kecelakaan. Penelitian ini menekankan pentingnya eksplorasi lebih lanjut

mengenai variasi arsitektur dan teknik augmentasi data yang dapat meningkatkan kemampuan generalisasi model terhadap data baru.

7.2. Saran

Saran yang dapat diberikan dari penelitian ini yaitu:

- 1) Pengujian Versi Lain dari Arsitektur MoViNet: Untuk meningkatkan performa model, direkomendasikan untuk mencoba versi lain dari arsitektur MoViNet. Mengingat kemampuan model saat ini dalam memprediksi tipe kecelakaan masih belum optimal, eksplorasi arsitektur yang berbeda dapat membantu dalam menangkap perbedaan karakteristik antar tipe kecelakaan dengan lebih baik. Dengan mengoptimalkan cara model menangani informasi spasial dan temporal dari data video, diharapkan dapat meningkatkan akurasi klasifikasi dan presisi dalam mendeteksi jenis kecelakaan yang berbeda.
- 2) Variasi Teknik Augmentasi Data: Penambahan variasi teknik augmentasi data juga perlu dipertimbangkan. Saat ini, model menunjukkan performa yang baik dalam mendeteksi kecelakaan secara biner namun mengalami kesulitan dalam klasifikasi tipe kecelakaan. Dengan memperluas jenis augmentasi yang digunakan, model dapat dilatih dengan data yang lebih beragam, yang akan meningkatkan kemampuan generalisasi model terhadap data baru. Variasi dalam augmentasi dapat membantu model belajar lebih banyak dari data yang ada dan meningkatkan keakuratannya dalam mengklasifikasikan berbagai jenis kecelakaan.

DAFTAR PUSTAKA

- Ahmed, S. K., Mohammed, M. G., Abdulqadir, S. O., El-Kader, R. G. A., El-Shall, N. A., Chandran, D., Rehman, M. E. U., & Dhama, K. (2023). Road traffic accidental injuries and deaths: A neglected global health issue. *Health Science Reports*, 6(5), e1240. <https://doi.org/10.1002/HSR2.1240>
- Elvik, R. (2000). How much do road accidents cost the national economy? *Accident Analysis & Prevention*, 32(6), 849–851. [https://doi.org/10.1016/S0001-4575\(00\)00015-4](https://doi.org/10.1016/S0001-4575(00)00015-4)
- Hussain, T., Memon, Z. A., Qureshi, R., & Alam, T. (2023). EMO-MoviNet: Enhancing Action Recognition in Videos with EvoNorm, Mish Activation, and Optimal Frame Selection for Efficient Mobile Deployment. *Sensors 2023, Vol. 23, Page 8106*, 23(19), 8106. <https://doi.org/10.3390/S23198106>
- Kondratyuk, D., Yuan, L., Li, Y., Zhang, L., Tan, M., Brown, M., & Gong, B. (2021). MoViNets: Mobile Video Networks for Efficient Video Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 16015–16025. <https://doi.org/10.1109/CVPR46437.2021.01576>
- Kuo, C. C. J. (2017). The CNN as a Guided Multilayer RECOs Transform [Lecture Notes]. *IEEE Signal Processing Magazine*, 34(3), 81–89. <https://doi.org/10.1109/MSP.2017.2671158>
- Liu, J., Shahroudy, A., Wang, G., Duan, L. Y., & Kot, A. C. (2020). Skeleton-Based Online Action Prediction Using Scale Selection Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6), 1453–1467. <https://doi.org/10.1109/TPAMI.2019.2898954>
- Liu, X., & Yin, J. (2022). Multi-Head TrajectoryCNN: A New Multi-Task Framework for Action Prediction. *Applied Sciences 2022, Vol. 12, Page 5381*, 12(11), 5381. <https://doi.org/10.3390/APP12115381>
- Mohammadpour, L., Ling, T. C., Liew, C. S., & Aryanfar, A.

- (2022). A Survey of CNN-Based Network Intrusion Detection. *Applied Sciences* 2022, Vol. 12, Page 8162, 12(16), 8162. <https://doi.org/10.3390/APP12168162>
- Monroy, C., Chen, C. H., Houston, D., & Yu, C. (2021). Action prediction during real-time parent-infant interactions. *Developmental Science*, 24(3), e13042. <https://doi.org/10.1111/DESC.13042>
- Pradana, H., Dao, M. S., & Zettsu, K. (2022). Augmenting Ego-Vehicle for Traffic Near-Miss and Accident Classification Dataset Using Manipulating Conditional Style Translation. *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. <https://doi.org/10.1109/DICTA56598.2022.10034630>
- Purwono, Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Ul, H. M. (2022). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control System*, 2(4). <https://pubs2.ascee.org/index.php/IJRCS/article/view/888>
- Wang, D., Yuan, Y., & Wang, Q. (2019). Early Action Prediction with Generative Adversarial Networks. *IEEE Access*, 7, 35795–35804. <https://doi.org/10.1109/ACCESS.2019.2904857>
- World Health Organization. (2023). *Road traffic injuries*. <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>
- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional Neural Networks: An Overview and Application in Radiology. *Insights into Imaging*, 9(4), 611–629. <https://doi.org/10.1007/S13244-018-0639-9/FIGURES/15>

LAMPIRAN

LAMPIRAN 1. Augmentasi Video

Kode untuk melakukan augmentasi video terhadap *dataset* training

```
flip_transform = A.HorizontalFlip(p=0.5)
rotate_transform = A.RandomRotate90(p=0.5)
brightness_contrast_transform = A.RandomBrightnessContrast(p=0.3)
hue_saturation_transform = A.HueSaturationValue(hue_shift_limit=20, sat_shift_limit=30, val_shift_limit=20, p=0.3)

augmented_frames = []

flip_prob = random.random()
rotate_prob = random.random()
brightness_prob = random.random()
hue_saturation_prob = random.random()

for i, frame in enumerate(frames):
    aug_input = {"image": frame}
    if flip_prob < 0.5:
        augment = flip_transform(**aug_input)
    if rotate_prob < 0.5:
        augment = rotate_transform(**aug_input)
    if brightness_prob < 0.3:
        augment = brightness_contrast_transform(**aug_input)
    if hue_saturation_prob < 0.3:
        augment = hue_saturation_transform(**aug_input)

    augmented_frames.append(aug_input['image'])

return np.array(augmented_frames)
```

LAMPIRAN 2. Pembuatan Model

Kode untuk membuat arsitektur model dan load weights untuk di fine tune

```
tf.keras.backend.clear_session()

backbone = movinet.Movinet(model_id='a2')
backbone.trainable = True

loss_obj = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
optimizer = tf.keras.optimizers.Adam(learning_rate = 0.001)

model = movinet_model.MovinetClassifier(backbone=backbone, num_classes=600)
model.build([None, None, None, None, 3])

!wget https://storage.googleapis.com/tf_model_garden/vision/movinet/movinet_a2_base.tar.gz -O movinet_a2_base.tar.gz -q
!tar -xvf movinet_a2_base.tar.gz

checkpoint_dir = 'movinet_a2_base'
checkpoint_path = tf.train.latest_checkpoint(checkpoint_dir)
checkpoint = tf.train.Checkpoint(model=model)
status = checkpoint.restore(checkpoint_path)
status.assert_existing_objects_matched()

model = movinet_model.MovinetClassifier(
    backbone=backbone,
    num_classes=NUM_CLASSES
)
model.build([32, 8, 224, 224, 3])

model.compile(loss=loss_obj, optimizer=optimizer, metrics=['accuracy'])
```

LAMPIRAN 3. Interpolasi Berbobot

Kode untuk melakukan interpolasi berbobot pada frame sehingga dihasilkan 8 frame akhir

```
def interpolate_frames(frames, target_count=8):
    original_count = len(frames)
    if target_count == original_count:
        return frames

    new_indices = [i * (original_count - 1) / (target_count - 1) for i in range(target_count)]

    interpolated_frames = []
    for idx in new_indices:
        lower = int(np.floor(idx))
        upper = int(np.ceil(idx))
        weight = idx - lower

        if lower == upper:
            interpolated_frames.append(frames[lower])
        else:
            frame = cv2.addWeighted(frames[lower], 1 - weight, frames[upper], weight, 0)
            interpolated_frames.append(frame)

    return interpolated_frames
```

LAMPIRAN 4. Pengambilan Frame

Kode untuk pengambilan frame *dataset*

```
class FrameGenerator:
    def __init__(self, json_file, base_dirs, n_frames, resolution):
        with open(json_file, 'r') as f:
            self.metadata = json.load(f)

        self.base_dirs = base_dirs
        self.n_frames = n_frames
        self.resolution = resolution
        self.videos_by_type = {
            "training": [],
            "validating": [],
            "testing": []
        }

        for video_name, data in self.metadata.items():
            self.videos_by_type[data["data_type"]].append((
                video_name,
                data["classno"],
                data["miss_end"]
            ))

    def __call__(self, data_type="training"):
        video_list = self.videos_by_type.get(data_type, [])
        if data_type == "training":
            random.shuffle(video_list)

        for video_name, label, miss_end in video_list:
            video_path = os.path.join(random.choice(self.base_dirs), video_name)
            if not os.path.exists(video_path):
                continue

            n_times = random.randint(1,5) if miss_end == -1 else 1

            for _ in range(n_times):
                video_frames = frames_from_video_file(video_path, self.n_frames, (self.resolution, self.resolution), miss_end, (data_type == "training"))

                if video_frames is not None:
                    yield video_frames, label

def frames_from_video_file(video_path, n_frames, output_size=(224, 224), miss_end=-3, training=False):
    result = []
    src = cv2.VideoCapture(str(video_path))
    video_length = int(src.get(cv2.CAP_PROP_FRAME_COUNT))

    if (miss_end == -1) or (miss_end == -3):
        start = random.randint(0, video_length-2)
        end = random.randint(start+2, video_length)
    elif miss_end > 0:
        start = random.randint(0, miss_end-2)
        end = random.randint(start+2, min(miss_end, video_length))
    elif miss_end == -2:
        start = 0
        end = 1

    for frame_idx in range(start, end):
        src.set(cv2.CAP_PROP_POS_FRAMES, frame_idx)
        ret, frame = src.read()
        if not ret:
            break

        formatted_frame = format_frames(frame, output_size)
        result.append(formatted_frame.numpy())

    if len(result) > 1:
        result = interpolate_frames(result, target_count=n_frames)
    elif len(result) == 1:
        result = [result[0]] * n_frames

    src.release()

    if training:
        result = augment_frames(np.array(result), output_size)

    return np.array(result, dtype=np.float32)
```

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS I

Nama : Richard Ryan
Tempat, Tanggal Lahir : Surabaya, 26 Januari 2004
Jenis Kelamin : Laki-laki
Telepon : +62895621076823
Email : richard.ryan.rr0@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC, ITS
Angkatan : 2021
Semester : 8 (Delapan)