



TUGAS AKHIR - EC 184801

**KLASIFIKASI TINDAK KEJAHATAN DALAM VIDEO
DENGAN MENGGUNAKAN *3D CONVOLUTIONAL
NEURAL NETWORK (3D CNN)***

Muhammad Afa Rahmatullah
NRP 0721154000042

Dosen Pembimbing
Reza Fuad Rachmadi, ST., MT., Ph.D.
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC 184801

**KLASIFIKASI TINDAK KEJAHATAN DALAM VIDEO
DENGAN MENGGUNAKAN *3D CONVOLUTIONAL
NEURAL NETWORK (3D CNN)***

Muhammad Afa Rahmatullah
NRP 0721154000042

Dosen Pembimbing
Reza Fuad Rachmadi, ST., MT., Ph.D.
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



FINAL PROJECT - EC 184801

**CRIME VIDEO CLASSIFICATION USING 3D
CONVOLUTIONAL NEURAL NETWORK (3D CNN)**

Muhammad Afa Rahmatullah
NRP 0721154000042

Advisor
Reza Fuad Rachmadi, ST., MT., Ph.D.
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

Departement of Computer Engineering
Faculty of Intelligent Electrical and Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Klasifikasi Tindak Kejahatan Dalam Video Dengan Menggunakan 3D Convolutional Neural Network (3D CNN)**" adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Januari 2020



M. Aufa Rahmatullah
NRP. 0721154000042

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

**KLASIFIKASI TINDAK KEJAHATAN DALAM VIDEO DENGAN
MENGUNAKAN 3D CONVOLUTIONAL NEURAL NETWORK (3D
CNN)**

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: **Muhammad Aufa Rahmatullah (NRP: 0721154000042)**

Tanggal Ujian : 6 Januari 2020

Periode Wisuda : Maret 2020

Disetujui oleh:

Reza Fuad Rachmadi, ST., MT., Ph.D.
NIP: 198504032012121001

(Pembimbing I)

Dr. Supeno Mardi Susiki Nugroho, ST.,
MT.
NIP: 197003131995121001

(Pembimbing II)

Dr. Eko Mulyanto Yuniarno, ST., MT.
NIP: 196806011995121009

(Penguji I)

Prof. Dr. Ir. Yoyon Kusnendar Suprpto,
M.Sc.
NIP. 195409251978031001

(Penguji II)



Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, ST., MT.
NIP. 197003131995121001

ABSTRAK

Nama Mahasiswa : Muhammad Aufa Rahmatullah
Judul Tugas Akhir : Klasifikasi Tindak Kejahatan Dalam Video Dengan Menggunakan *3D Convolutional Neural Network (3D CNN)*
Pembimbing : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D.
2. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

Pada tempat – tempat yang terdapat banyak kerumunan orang, pengawasan keamanan sangatlah dibutuhkan karena semakin ramai tempat tersebut maka semakin besar juga kemungkinan terjadinya tindak kriminal. Definisi dari kerumunan orang itu sendiri adalah organisme yang kurang terstruktur dan perilakunya merupakan campuran dari beberapa perilaku dan tujuan individu yang heterogen atau bermacam – macam. Maka dari itu pada tugas akhir ini akan dikembangkan sebuah sistem yang bisa mengklasifikasi tindak kejahatan menggunakan 3D CNN (*3D Convolutional Neural Network*) dengan menggunakan dataset *UCF-Crime* yang berjumlah sebanyak 1250 video dan terpisah menjadi 14 kelas. Proses *training* menggunakan 3D CNN berjalan selama kurang lebih 4 hari dengan menggunakan 10 *epoch* dan *batch size* 12. Proses klasifikasi citra akan menggunakan lima macam model yang merupakan hasil modifikasi dari arsitektur C3D. Hasil perbandingan performa akurasi prediksi menunjukkan bahwa model pertama, kedua, ketiga, keempat, dan kelima masing - masing memiliki nilai akurasi sebesar 15,7%, 27,1%, 17,1%, 17,1%, dan 22,8%.

Kata Kunci: Kejahatan, Klasifikasi, *3D Convolutional Neural Network* (3D CNN).

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Muhammad Aufa Rahmatullah
Title : *Crime Video Classification Using 3D Convolutional Neural Network (3D CNN)*
Advisors : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D.
2. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

In places where there are large crowds, security surveillance is needed because the more crowded the place, the greater the possibility of crime. The definition of the crowd itself is a less structured organism and its behavior is a mixture of heterogeneous or diverse individual behaviors and goals. Therefore, in this thesis a system that can classify crime using 3D CNN (3D Convolutional Neural Network) will be developed by using the UCF-Crime dataset, totaling 1250 videos and separated into 14 classes. The process of training using 3D CNN runs for approximately 4 days using 10 epoch and 12 batch size. The image classification process will use five types of models that are a modification of the C3D architecture. The comparison of predictive accuracy shows that the first, second, third, fourth, and fifth models each have an accuracy value of 15.7%, 27.1%, 17.1%, 17.1%, 22,8%.

Keywords: Crime, Classification, 3D Convolutional Neural Network (3D CNN)

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadirat Allah Swt. atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Klasifikasi Tindak Kejahatan Dalam Video Dengan Menggunakan *3D Convolutional Neural Network* (3D CNN)**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Semua keluarga dan kedua orang tua tercinta dan kakak saya yang telah memberikan motivasi, dorongan spiritual, dan material dalam penyelesaian buku penelitian ini.
2. Bapak Kepala Departemen Teknik Komputer ITS Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
3. Bapak Reza Fuad Rachmadi, S.T., M.T., Ph.D., dan Bapak Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. yang senantiasa memberikan motivasi, bimbingan, dan arahan selama mengerjakan penelitian.
4. Bapak Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc., selaku dosen wali yang telah memberikan motivasi dan dukungan dalam pelaksanaan penelitian ini.
5. Bapak dan Ibu dosen pengajar Departemen Teknik Komputer ITS, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
6. Seluruh staf Departemen Teknik Komputer ITS yang telah membantu penulis dalam hal administrasi dan perlengkapan penelitian.
7. Seluruh mahasiswa Fakultas Teknologi Elektro angkatan 2013, 2014, 2015, 2016, dan seterusnya yang telah menjadi saudara terbaik selama masa studi saya di Surabaya.
8. Seluruh teman - teman EAG yang sudah menemani keseharian saya dan memberikan motivasi selama masa studi saya di Surabaya (Ian, Dito, Dika, Seno, Jerryco, Wildan, Mario, Oka, Kurnia, dan Abi).

9. Teman seperjuangan dalam mengerjakan penelitian berbasis CNN (Shahid, Aldo, dan Bagus).

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Januari 2020

Muhammad Afa Rahmatullah

DAFTAR ISI

Abstrak	i
Abstract	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
NUMENKLATUR	xiii
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan masalah	2
1.5 Sistematika Penulisan	2
2 TINJAUAN PUSTAKA	5
2.1 <i>Related Work</i>	5
2.1.1 <i>Learning Spatiotemporal Features with 3D Convolutional Networks</i>	5
2.1.2 <i>3D CNN for Human Action Recognition</i>	5
2.1.3 <i>Sentence Level Indonesian Sign Language Recognition Using 3D Convolutional Neural Network and Bidirectional Recurrent Neural Network</i>	6
2.2 <i>Machine Learning</i>	6
2.3 <i>Deep Learning</i>	7
2.4 <i>Convolutional Neural Network</i>	8
2.4.1 <i>Convolutional Layer</i>	9
2.4.2 <i>Fully-Connected Layer</i>	11
2.4.3 <i>Pooling Layer</i>	12

2.4.4	<i>Activation Function</i>	13
2.5	<i>3D Convolutional Neural Network</i>	15
2.6	<i>Computer Vision</i>	17
2.7	<i>Stochastic Gradient Descent</i>	17
3	DESAIN DAN IMPLEMENTASI SISTEM	19
3.1	Desain Sistem	20
3.2	Dataset	20
3.2.1	Mengubah Dataset Menjadi Frame	21
3.2.2	<i>Split</i>	22
3.3	<i>Training</i>	22
3.3.1	<i>Training Models</i>	23
3.4	Validasi	25
3.5	Evaluasi	25
4	PENGUJIAN DAN ANALISIS	27
4.1	<i>Training Data</i>	27
4.2	Pengujian	27
4.2.1	Pengujian Model Pertama	28
4.2.2	Pengujian Model Kedua	35
4.2.3	Pengujian Model Ketiga	42
4.2.4	Pengujian Model Keempat	49
4.2.5	Pengujian Model Kelima	56
4.3	Analisa	63
5	PENUTUP	67
5.1	Kesimpulan	67
5.2	Saran	68
	DAFTAR PUSTAKA	69
	Biografi Penulis	73

DAFTAR GAMBAR

2.1	<i>layer</i> yang terdapat pada CNN	8
2.2	<i>Convolutional Layer</i>	10
2.3	<i>Pooling layer</i>	12
2.4	Fungsi Aktivasi ReLU	14
2.5	Ilustrasi perbedaan 3D CNN dan 2D CNN	16
3.1	Diagram alir sistem	19
3.2	<i>Processing</i> dataset	21
3.3	Susunan <i>layer</i> pada model pertama	23
3.4	Susunan <i>layer</i> pada model kedua	24
3.5	Susunan <i>layer</i> pada model ketiga	24
3.6	Susunan <i>layer</i> pada model keempat	24
3.7	Susunan <i>layer</i> pada model kedua	25
4.1	Jumlah <i>layer</i> arsitektur C3D	27
4.2	Grafik nilai <i>loss training</i> model pertama	28
4.3	Grafik nilai <i>accuracy training</i> model pertama	29
4.4	<i>Confusion Matrix</i> nilai akurasi model pertama	31
4.5	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model pertama	32
4.6	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model pertama	33
4.7	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model pertama	34
4.8	Grafik nilai <i>loss training</i> model kedua	35
4.9	Grafik nilai <i>accuracy training</i> model kedua	36
4.10	<i>Confusion Matrix</i> nilai akurasi model kedua	37
4.11	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model kedua	39
4.12	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model kedua	40
4.13	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model kedua	41
4.14	Grafik nilai <i>loss training</i> model ketiga	42
4.15	Grafik nilai <i>accuracy training</i> model ketiga	43

4.16	<i>Confusion Matrix</i> nilai akurasi model ketiga	44
4.17	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model ketiga	46
4.18	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model ketiga	47
4.19	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model ketiga	48
4.20	Grafik nilai <i>loss training</i> model keempat	49
4.21	Grafik nilai <i>accuracy training</i> model keempat	50
4.22	<i>Confusion Matrix</i> nilai akurasi model keempat	51
4.23	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model keempat	53
4.24	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model keempat	54
4.25	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model keempat	55
4.26	Grafik nilai <i>loss training</i> model kelima	56
4.27	Grafik nilai <i>accuracy training</i> model kelima	57
4.28	<i>Confusion Matrix</i> nilai akurasi model kelima	58
4.29	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model kelima	60
4.30	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model kelima	61
4.31	Grafik hasil dari pengujian pada salah satu video pada <i>training</i> model kelima	62
4.32	Grafik rata - rata performa model pertama	63
4.33	Grafik rata - rata performa model kedua	64
4.34	Grafik rata - rata performa model ketiga	64
4.35	Grafik rata - rata performa model keempat	65
4.36	Grafik rata - rata performa model kelima	65

DAFTAR TABEL

3.1	Dataset UCF-Crime	22
4.1	Pengujian Pada Hasil <i>Training</i> Model Pertama . . .	30
4.2	Pengujian Pada Hasil <i>Training</i> Model Kedua	38
4.3	Pengujian Pada Hasil <i>Training</i> Model Ketiga	45
4.4	Pengujian Pada Hasil <i>Training</i> Model Keempat . . .	52
4.5	Pengujian Pada Hasil <i>Training</i> Model Kelima	59

Halaman ini sengaja dikosongkan

NUMENKLATUR

W	: Panjang / tinggi <i>input</i>
N	: Panjang / tinggi <i>filter</i>
P	: <i>Zero padding</i>
S	: <i>Stride</i>
$s(t)$: <i>Feature map</i> (satu dimensi)
x	: <i>Input</i> (satu dimensi)
w	: <i>Kernel</i> (satu dimensi)
t	: <i>Pixel</i> (satu dimensi)
$S(i, j)$: <i>Feature map</i> (dua dimensi)
(i, j)	: <i>Pixel</i> (dua dimensi)
$K(m, n)$: <i>Kernel</i> (dua dimensi)
I	: <i>Input</i> (dua dimensi)

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Ilmu dan teknologi pengolahan citra merupakan suatu ilmu yang bermanfaat dalam kehidupan. Pengolahan citra adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari camera)[4]. Beberapa manfaat pengolahan citra ialah untuk mendapatkan informasi berupa objek apa saja yang terdapat dalam sebuah gambar atau sebuah video. Salah satu penerapan pengolahan citra ialah pendeteksian objek tertentu dengan menggunakan kamera. Di Indonesia, pada tahun 2016 kasus kejahatan yang terjadi adalah sebanyak 357.197 kasus yang berarti dalam setiap 1 menit 28 detik terjadi kira – kira satu tindak kriminal dan dari 100.000 orang di Indonesia, 140 orang beresiko terkena tindak kejahatan. Pada tahun 2017 sebanyak 336.652 kasus kejahatan yang terjadi, yang berarti dari 100.00 orang, 129 orang beresiko terkena tindak kejahatan[5]. Walaupun terjadi penurunan, tapi penurunan yang terjadi tidak signifikan, sehingga masih sangat dibutuhkan tindakan – tindakan yang dapat mengurangi tingkat kejahatan di Indonesia. Salah satu metode yang saat ini sedang dikembangkan untuk mendeteksi dan mengenali object pada sebuah citra adalah *Convolutional Neural Network* (CNN). CNN adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra[6]. Sedangkan *3D Convolutional Neural Network* (3D CNN) adalah CNN yang dikhususkan untuk mengolah data 3 dimensi yang telah terbukti sangat baik dalam pengolahan citra video[7]. Oleh karena itu, pada tugas akhir ini akan dibuat sebuah sistem yang dapat menganalisa *anomaly* berupa

tindak - tindak kejahatan dalam sebuah video dengan *3D Convolutional Neural Network*.

1.2 Permasalahan

Perumusan masalah dalam penelitian ini adalah belum adanya sistem untuk mendeteksi tindak kejahatan yang terkomputerisasi dengan menggunakan 3D CNN.

1.3 Tujuan

Tujuan dari tugas akhir ini adalah untuk mengembangkan sebuah sistem yang dapat mendeteksi tindakan kejahatan menggunakan 3D CNN. Harapannya, dengan adanya sistem ini maka penganalisaan data dari kamera pengawas atau CCTV bisa dilakukan secara otomatis.

1.4 Batasan masalah

Untuk memfokuskan permasalahan yang diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut di antaranya adalah:

1. Kondisi yang akan di analisa adalah yang terlihat oleh kamera.
2. Data *input* yang diproses diambil dengan menggunakan kamera.
3. Tempat yang dianalisa adalah tempat yang membutuhkan keamanan.
4. Video yang digunakan adalah video yang tersedia untuk publik.
5. Proses pendeteksian pada intensitas cahaya yang cukup.

1.5 Sistematika Penulisan

Laporan penelitian tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu:

1. BAB I Pendahuluan
Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan, dan metodologi penelitian.

2. BAB II Tinjauan Pustaka

Bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait *Deep Learning*, *3D Convolutional Neural Network* (3D CNN), dan teori-teori penunjang lainnya.

3. BAB III Desain dan Implementasi Sistem

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan, langkah-langkah pemrosesan dataset dan proses *training*. Guna mendukung itu digunakanlah blok diagram atau *workflow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisis

Bab ini menjelaskan tentang hasil serta analisis yang didapatkan dari pengujian yang dilakukan.

Halaman ini sengaja dikosongkan

BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 *Related Work*

Berikut akan dijelaskan tentang beberapa penelitian yang berhubungan dengan tugas akhir ini.

2.1.1 *Learning Spatiotemporal Features with 3D Convolutional Networks*

Penelitian berjudul *Learning Spatiotemporal Features with 3D Convolutional Networks*[1] yang dilakukan oleh Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, dan Manohar Paluri ini mengusulkan pendekatan yang sederhana, namun efektif untuk pembelajaran fitur spasial temporal menggunakan 3D *Convolutional Networks* (3D ConvNets) yang dilatih pada dataset video yang berskala besar. Temuan yang dihasilkan dari penelitian ini meliputi: 1) 3D ConvNets lebih cocok untuk pembelajaran fitur spasial dibandingkan dengan 2D ConvNets; 2) Arsitektur homogen dengan kernel konvolusi $3 \times 3 \times 3$ di semua lapisan adalah salah satu arsitektur terbaik untuk ConvNets 3D; dan 3) Fitur-fitur yang telah dipelajari, yaitu C3D (*Convolutional 3D*), dengan klasifikasi linear sederhana mengungguli metode canggih pada 4 tolak ukur yang berbeda dan dapat dibandingkan dengan metode terbaik saat ini pada 2 tolak ukur lainnya. Penelitian ini mencapai akurasi 52,8% pada dataset UCF101.

2.1.2 *3D CNN for Human Action Recognition*

Penelitian berjudul *3D CNN for Human Action Recognition*[2] yang dilakukan oleh S. Neilu Boualia dan N. Essoukri ini bertujuan untuk membuat sistem *Human Action Recognition* berdasarkan model 3D CNN. Penelitian ini menerapkan model yang dikembangkan untuk mengenali aksi manusia dari dataset KTH dan JHMDB, dan mencapai kinerja terbaik dibandingkan dengan metode baseline. Penelitian ini mencapai akurasi rata - rata 78% pada dataset

KTH dan 90% pada dataset JHMDB.

2.1.3 Sentence Level Indonesian Sign Language Recognition Using 3D Convolutional Neural Network and Bidirectional Recurrent Neural Network

Dalam penelitian berjudul *Sentence Level Indonesian Sign Language Recognition Using 3D Convolutional Neural Network and Bidirectional Recurrent Neural Network*[3] yang dilakukan oleh M. Ariesta, F. Wiryana, S. Suharjito, A. Zahra ini, diusulkan sebuah SLR *Sign Language Recognition* tingkat kalimat menggunakan metode *deep learning* dengan menggabungkan *convolutional neural network* (CNN) dan *bidirectional-RNN* (Bi-RNN). Secara khusus, 3D CNN diimplementasikan untuk mengekstraksi fitur dari setiap frame video dan Bi-RNN diimplementasikan untuk mengekstraksi fitur unik dari perilaku sekuensial frame video, yang kemudian menghasilkan kalimat. Hasilnya menunjukkan bahwa hyperparameter yang digunakan, dalam hal ini Hyperparameter 1, mencapai hasil terbaik. Juga, penelitian ini menemukan bahwa jaringan yang lebih dalam tidak selalu menjamin hasil yang baik. Sejumlah besar dataset juga mempengaruhi kinerja sistem.

2.2 Machine Learning

Machine Learning (ML) merupakan sebuah cabang dari *Artificial Intelligence* (kecerdasan buatan) yang memungkinkan sistem komputer untuk belajar langsung dari contoh, data, dan pengalaman yang didapatkannya. Dengan membuat komputer mampu menjalankan tugas yang spesifik secara cerdas, ML dapat melakukan algoritma yang kompleks dengan cara mempelajari data yang diberikan tanpa perlu mengikuti program yang dibuat atau diinstruksikan secara manual oleh pengguna[8].

Ciri khas dari ML adalah adanya proses pembelajaran atau *training*. Oleh karena itu, ML membutuhkan data untuk dipelajari yang disebut sebagai data *training*. Setelah berhasil melakukan *training*, maka ML dapat melakukan proses klasifikasi dan prediksi terhadap data baru yang diberikan sesuai dengan hasil *training* yang telah dilakukan. Klasifikasi adalah metode dalam ML yang digunakan oleh mesin untuk memilah atau mengklasifikasikan ob-

jek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan. Beberapa contoh implementasi dari ML antara lain adalah *image processing*, *search recommendation*, *text and speech recognition*, dan *health diagnosis*[9].

Di dalam ML, terdapat tiga kategori pembelajaran[10], yaitu:

1. *Supervised Learning*

Supervised learning merupakan metode pembelajaran dimana data yang akan dipelajari sudah diberikan label yang menunjukkan klasifikasi atau kelas data tersebut berada. Model yang dihasilkan adalah model prediksi dari data yang telah diberi label.

2. *Unsupervised Learning*

Unsupervised learning merupakan metode pembelajaran dimana data yang akan dipelajari tidak memiliki label, sehingga komputer perlu melakukan analisis dan klasifikasi berdasarkan data yang ada.

3. *Reinforcement Learning*

Reinforcement learning merupakan metode pembelajaran dimana diberlakukan *reward* pada hasil pembelajaran yang positif dan *punishment* pada hasil pembelajaran yang negatif. Komputer tidak diberitahu apa yang harus dilakukan, namun memiliki tujuan untuk mendapatkan *reward* yang maksimal sehingga memaksa komputer untuk belajar secara terus-menerus.

2.3 Deep Learning

Deep Learning (DL) adalah salah satu bidang yang muncul dari penelitian ML. DL memungkinkan model komputasi yang tersusun dari beberapa lapisan pemrosesan untuk mempelajari representasi dari data dengan berbagai level abstraksi. DL menemukan struktur sulit yang terdapat di dalam kumpulan data yang besar dengan menggunakan algoritma *backpropagation*. Struktur yang didapatkan menunjukkan parameter internal apa yang harus diubah oleh mesin agar dapat menghitung representasi di setiap *layer* ber-

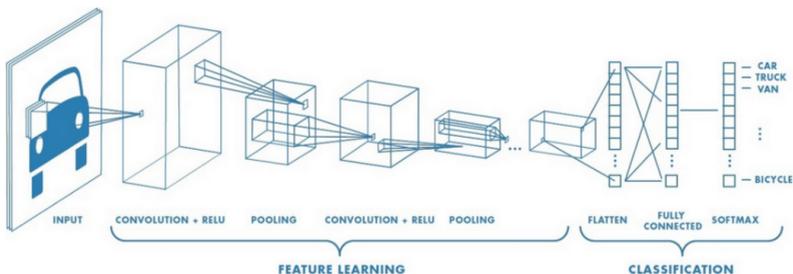
dasarkan representasi dari *layer* sebelumnya[11].

DL melakukan pendekatan dalam penyelesaian masalah dengan menggunakan konsep hierarki. Dengan konsep tersebut, komputer mampu mempelajari sebuah konsep yang kompleks dengan menggabungkan konsep-konsep yang lebih sederhana[12].

2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah variasi dari *Multilayer Perceptron* yang terinspirasi dari jaringan saraf manusia. Pemberian nama *convolutional neural network* mengindikasikan bahwa jaringan tersebut menggunakan operasi matematika yang disebut konvolusi. Konvolusi sendiri adalah sebuah operasi linier. CNN merupakan pengembangan dari *artificial neural network* yang saat ini diklaim sebagai model terbaik untuk memecahkan masalah seputar *object recognition* dan *detection*.

Secara teknis, CNN adalah arsitektur yang bisa di *training* dan terdiri dari beberapa tahap. Input dan output dari masing - masing tahap berupa *array* yang disebut *feature map* atau peta fitur. *Output* dari masing-masing tahap adalah *feature map* hasil pengolahan dari semua lokasi pada input. Struktur CNN dibangun dari tiga jenis layer utama yaitu *convolution layer*, *pooling layer*, dan *activation function*[13]. Susunan *layer* yang terdapat pada CNN dapat dilihat pada gambar 2.1.



Gambar 2.1: *layer* yang terdapat pada CNN[14].

2.4.1 Convolutional Layer

Convolutional Layer melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel(kotak kuning)pada citra disemua *offset* yang memungkinkan seperti yang ditunjukkan pada gambar 2.2. Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input*. Secara umum operasi konvolusi dapat dituliskan dengan persamaan 2.1.

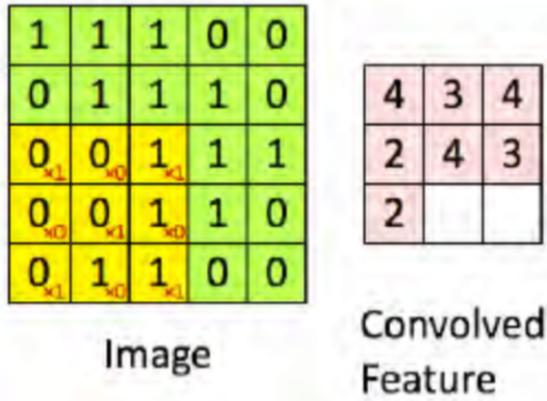
$$s(t) = (x \times w)(t) \quad (2.1)$$

Pada fungsi $s(t)$ menghasilkan *output* tunggal yaitu *feature map*. Argumen pertama berupa *input* yang merupakan x dan argumen kedua yang merupakan w sebagai *kernel* atau filter. Jika melihat *input* sebagai citra dua dimensi, maka (t) bisa diasumsikan sebagai sebuah piksel dan menggantinya dengan i dan j . Oleh karena itu, untuk operasi konvolusi dengan lebih dari satu dimensi dapat digunakan persamaan 2.2.

$$S_{(i,j)} = (K \times I)_{(i,j)} = \sum \sum I_{(i-m,j-n)} K_{(m,n)} \quad (2.2)$$

Persamaan 2.2 merupakan perhitungan dalam operasi konvolusi dengan i dan j sebagai piksel dari sebuah citra. Perhitungannya bersifat komutatif dan muncul ketika $K_{(m,n)}$ sebagai *kernel*-nya serta I sebagai *input* dan *kernel* yang dapat dibalik relatif terhadap *input*. Operasi konvolusi dapat dilihat sebagai perkalian matriks antara citra *input* dan filter dimana *output*-nya dapat dihitung dengan *dot product*.

Parameter yang menentukan berapa jumlah pergeseran filter disebut "*Stride*". Jika nilai stride adalah 1, maka *convolutioal filter* akan bergeser sebanyak 1 *pixels* secara horizontal lalu vertical. Semakin kecil *stride* maka akan semakin detail informasi yang akan



Gambar 2.2: Convolutional Layer

didapatkan dari sebuah *input*, namun membutuhkan proses komputasi yang lebih jika dibandingkan dengan *stride* yang besar. Perlu diperhatikan bahwa dengan menggunakan *stride* bernilai kecil tidak selalu akan mendapatkan performa yang bagus. Persamaan 2.3 digunakan untuk menghitung dimensi dari *feature map*.

$$Output = \frac{W - N + 2P}{S} + 1 \quad (2.3)$$

Padding atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input. Dimensi *output* dari *convolutional layer* selalu lebih kecil dari *input*-nya (kecuali penggunaan 1×1 filter dengan jumlah *stride* 1). *Output* ini akan digunakan kembali sebagai input dari convolutional layer selanjutnya, sehingga informasi akan banyak yang terbuang. Dengan menggunakan *padding* dapat mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis, sehingga penggunaan *convolutional layer* yang lebih dalam dapat menghasilkan lebih banyak *features* yang di-*extract*[16]. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Bobot pada

layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan *input*.

2.4.1.1 *Stride*

Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah satu, maka *feature map* akan bergeser sebanyak 1 piksel secara horizontal lalu vertikal. Semakin kecil *stride* yang digunakan, maka semakin detail informasi yang didapatkan dari sebuah *input*, namun membutuhkan komputasi lebih jika dibandingkan dengan *stride* yang besar.

2.4.1.2 *Padding*

Padding atau *zero padding* adalah parameter yang menentukan jumlah piksel (berisi nilai nol) yang akan ditambahkan di setiap sisi dari *input*. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *feature map*. Penggunaan *padding* dapat digunakan untuk mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang drastis sehingga dapat dilakukan ekstraksi fitur yang lebih mendalam.

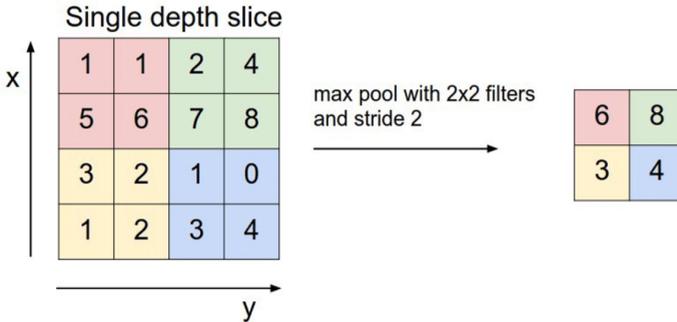
2.4.1.3 *Loss Function*

Loss function merupakan fungsi yang menggambarkan kerugian yang terkait dengan semua kemungkinan yang dihasilkan oleh sebuah model. *Loss function* ini bekerja ketika model pembelajaran memberikan kesalahan yang harus diperhatikan. *Loss function* yang baik memberikan nilai *error* yang rendah.

2.4.2 *Fully-Connected Layer*

Layer tersebut adalah *layer* yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap neuron pada *convolutional layer* perlu ditransformasi menjadi data satu dimensi atau *vector* terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel. *Fully connected layer* hanya dapat diimplementasikan di akhir jaringan.

2.4.3 Pooling Layer



Gambar 2.3: Pooling layer[16].

Fungsi dari *pooling* ini adalah untuk mereduksi input secara *spasial* (mengurangi jumlah parameter) dengan operasi *down-sampling*. Dalam pooling layer terdapat dua macam pooling yang biasa digunakan yaitu *average pooling* dan *max pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max pooling* adalah nilai maksimal. Lapisan *pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume output pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *overfitting*. Berdasarkan gambar 2.3 menunjukkan proses dari *max-pooling*. Output dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan dimensi citra awal. Lapisan *pooling* diatas akan beroperasi pada setiap irisan kedalaman volume input secara bergantian. Jika dilihat dari Gambar 2.3 operasi *max pooling* menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4, dari masing-masing 4 angka pada *input* operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2[15].

2.4.3.1 *Global Average Pooling*

Global Average Pooling menghasilkan satu *feature map* untuk masing-masing kategori dalam sebuah klasifikasi di lapisan *mlpconv* terakhir. Sebagai ganti dari *full-connected layer* yang seharusnya ditambahkan diatas *feature map*, *Global Average Pooling* mengambil rata - rata dari setiap *feature map* dan memasukkannya langsung ke *softmax*. Satu keuntungan dari *Global Average Pooling* adalah tidak ada parameter yang perlu di optimalisasi di dalam *Global Average Pooling* bisa menghindari *overfitting*[17].

2.4.4 *Activation Function*

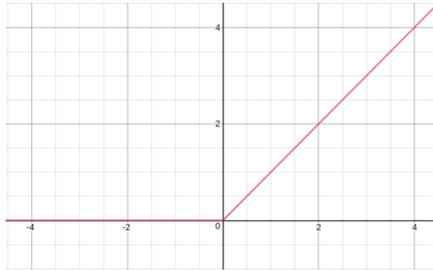
Activation function atau fungsi aktivasi merupakan sebuah *node* yang ditambahkan di akhir *output* dari setiap *neural network*. Fungsi aktivasi juga dikenal sebagai *transfer function* yang digunakan untuk menentukan *output* dari *neural network*. Fungsi aktivasi dibagi menjadi dua tipe yaitu linier dan *non*-linier. Pada arsitektur CNN, fungsi aktivasi terletak pada perhitungan akhir *output feature map*, sesudah proses perhitungan konvolusi atau *pooling*, untuk menghasilkan suatu pola fitur. Beberapa macam fungsi aktivasi yang sering digunakan dalam penelitian antara lain fungsi *softmax*, sigmoid, tanh, dan *Rectified Linear Unit* (ReLU).

2.4.4.1 *Rectified Linear Unit (ReLU)*

Pada dasarnya fungsi ReLU melakukan *threshold* dari 0 hingga tidak terhingga (*infinity*). Fungsi ini menjadi salah satu fungsi aktivasi yang populer saat ini. Pada fungsi ini input dari neuron - neuron berupa bilangan negatif akan diterjemahkan kedalam nilai 0, dan jika masukan bernilai positif maka output dari neuron adalah nilai aktivasi itu sendiri. Fungsi aktivasi ini memiliki kelebihan yaitu dapat mempercepat proses konfigurasi yang dilakukan dengan *Stochastic Gradient Descent* (SGD). Kelemahan ReLU yaitu fungsi aktivasi ini bisa menjadi rapuh pada proses training dan bisa membuat unit tersebut mati. Grafik fungsi aktivasi ReLU ditunjukkan pada Gambar 2.4[16]

2.4.4.2 *Exponential Linear Unit (ELU)*

Exponential Linear Unit (ELU) adalah jenis fungsi aktivasi lain yang diusulkan oleh Clevert et al pada tahun 2015 dan fungsi



Gambar 2.4: Fungsi Aktivasi ReLU[16].

aktivasi ini digunakan untuk mempercepat proses *training deep neural network*. Keuntungan utama ELU adalah mereka dapat mengurangi masalah gradien yang hilang dengan menggunakan identitas untuk nilai - nilai positif dan juga meningkatkan karakteristik *learning*. ELU memiliki nilai negatif yang memungkinkan untuk mendorong aktivasi unit rata - rata mendekati nol sehingga mengurangi kompleksitas komputasi sehingga meningkatkan kecepatan *training*. ELU merupakan alternatif yang baik untuk ReLU karena mengurangi pergeseran bias dengan mendorong aktivasi rata-rata nol selama proses *training*[18].

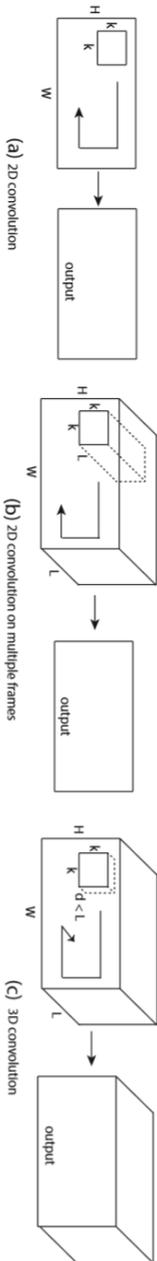
2.4.4.3 *Softmax*

Fungsi aktivasi Softmax adalah contoh lain dari fungsi aktivasi yang digunakan dalam komputasi saraf atau *neural computing*. Fungsi aktivasi ini digunakan untuk menghitung probabilitas distribusi dari vektor bilangan real. Fungsi aktivasi Softmax menghasilkan output yang merupakan kisaran nilai antara 0 dan 1, dengan jumlah probabilitas sama dengan 1. Fungsi Softmax digunakan dalam model *multi-class* dimana ia mengembalikan probabilitas setiap kelas, dengan kelas target memiliki probabilitas tertinggi. Fungsi aktivasi Softmax sebagian besar muncul di hampir semua lapisan output dari arsitektur *deep learning*. Perbedaan utama antara fungsi aktivasi Sigmoid dan Softmax adalah fungsi aktivasi Sigmoid digunakan dalam klasifikasi biner sedangkan fungsi aktivasi Softmax digunakan untuk klasifikasi multivarian[18].

2.5 3D Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu metode yang saat ini sedang dikembangkan untuk mendeteksi dan mengenali object pada sebuah image. CNN mengambil operasi linear matematika antara matriks yang disebut konvolusi. CNN memiliki banyak *layer*, termasuk *convolutional layer*, *pooling layer*, dan *fully-connected layer*. Pada 3D CNN, proses konvolusi dan penggabungan (*pooling*) dilakukan secara *spatio-temporal* (terhadap ruang dan waktu), yang pada CNN hanya dilakukan secara *spatial* (terhadap ruang). Sehingga 3D CNN bisa diaplikasikan ke citra video[19], [1].

Dalam 3D CNN, operasi konvolusi dan penyatuan (*pooling*) dilakukan secara *spatio-temporal* sedangkan pada 2D CNN, mereka hanya dilakukan secara spasial. Gambar 2.5 mengilustrasikan perbedaannya, konvolusi 2D yang diterapkan pada gambar akan menghasilkan gambar, konvolusi 2D yang diterapkan pada banyak gambar juga menghasilkan gambar. Oleh karena itu, 2D CNN kehilangan informasi temporal input tepat setelah setiap operasi konvolusi. Hanya konvolusi 3D yang menjaga informasi temporal dari input yang menghasilkan volume output. Hal itu berlaku juga untuk *pooling* 2D dan 3D.



Gambar 2.5: Ilustrasi perbedaan 3D CNN dan 2D CNN[1].

2.6 *Computer Vision*

Computer vision adalah suatu pembelajaran menganalisis citra dan video oleh komputer untuk memperoleh hasil sebagaimana yang bisa dilakukan manusia. Pada hakikatnya, *computer vision* mencoba meniru cara kerja sistem visual manusia. Manusia melihat objek dengan indra penglihatan (mata), lalu citra objek diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya. Hasil interpretasi ini kemudian digunakan untuk pengambilan keputusan seperti menghindari pejalan kaki ketika sedang mengendarai sebuah mobil atau memanggil nama seseorang ketika mengenali wajahnya. Pada komputer, hal ini dilakukan dengan melakukan penangkapan citra atau video melalui kamera, lalu dilakukan proses analisis terhadap gambar tersebut. Hasil analisis digunakan untuk melakukan keputusan-keputusan yang dibuat berdasarkan kondisi citra atau video yang ditangkap oleh kamera. *Computer vision* dibuat agar dapat membantu manusia melakukan proses pengamatan dan pengambilan keputusan yang sulit jika dilakukan dalam kondisi yang spesifik[20].

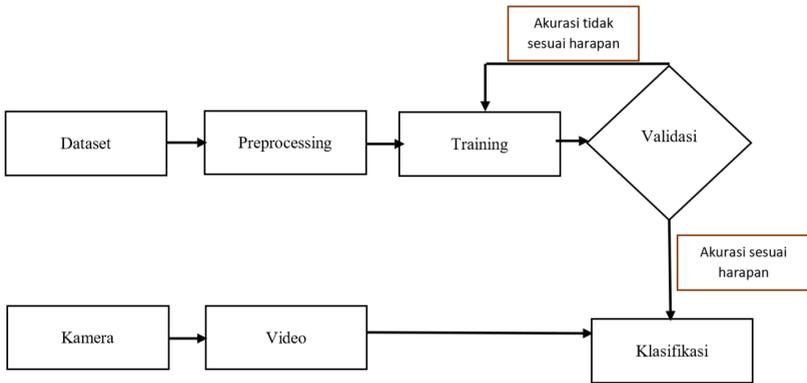
2.7 *Stochastic Gradient Descent*

Gradient Descent adalah sebuah metode optimasi yang biasa digunakan pada berbagai bidang *Machine Learning*. Metode ini adalah metode yang paling sering digunakan untuk optimasi Jaringan Saraf Tiruan. *Gradient Descent* bekerja dengan prinsip mengoptimasi secara efisien sebuah fungsi objektif. Pada sebuah fungsi yang didefinisikan oleh kumpulan parameter-parameter, *Gradient Descent* memulai dari nilai awal parameter-parameter tersebut yang kemudian secara iteratif bergerak menuju nilai parameter yang meminimalisasi nilai fungsi tersebut. Pada Jaringan Saraf Tiruan, *Gradient Descent* digunakan untuk menemukan nilai bobot optimal agar didapatkan nilai fungsi biaya yang kecil, di mana nilai fungsi biaya merepresentasikan tingkat kesalahan (error) pada sistem. Pencarian nilai bobot dilakukan dengan cara merubah bobot secara bertahap ketika proses *training*. Lamanya pencarian bobot optimal menentukan waktu pelatihan dari sistem Jaringan Saraf Tiruan. Ketika tantangan datang berupa permintaan waktu

proses yang minimal dengan hasil akurasi yang maksimal, banyak bermunculan metode baru untuk mengoptimalkan proses pencarian nilai optimum tersebut. Salah satu turunan dari *Gradient Descent* adalah *Stochastic Gradient Descent* (SGD). Algoritme SGD memilih *data point* pada setiap *step* untuk menghitung derivatif. SGD secara acak mengambil satu *data point* dari seluruh kumpulan data di setiap iterasi untuk mengurangi perhitungan dengan sangat besar[21].

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM



Gambar 3.1: Diagram alir sistem

Tujuan dari tugas akhir ini adalah untuk mendeteksi tindak - tindak kejahatan yang terekam oleh kamera berbasis *3D Convolutional Neural Network*. Proses kerja dari sistem ini ditunjukkan pada gambar 3.1. Berdasarkan pada gambar 3.1, salah satu proses yang dilakukan adalah *anomaly detection*. Tujuan dari *anomaly detection* adalah untuk mengetahui adanya pergerakan yang tidak normal berupa tindak kejahatan di dalam sebuah video. Agar bisa mencapai hal tersebut, dibutuhkan dataset. Dataset yang digunakan untuk *training* dalam sistem ini berupa berbagai macam video perbuatan kriminal yang diubah menjadi sekumpulan *frame*. Setelah mendapatkan dataset, langkah selanjutnya adalah pembuatan *split* yang berfungsi untuk memisahkan dataset yang digunakan untuk *training* dan *testing* atau validasi. Metode yang digunakan pada sistem ini adalah 3D CNN yang menggunakan 5 arsitektur yang berbeda - beda berdasarkan pada metode C3D yang telah dimodifikasi. Hasil dari *training* tersebut adalah *weight* yang bisa digunakan untuk mendeteksi *anomaly* dari sebuah video. Proses selanjutnya adalah identifikasi, yaitu proses dimana sistem meng-

identifikasi *anomaly* yang terdeteksi di dalam sebuah video sesuai dengan *class* dari dataset yang digunakan.

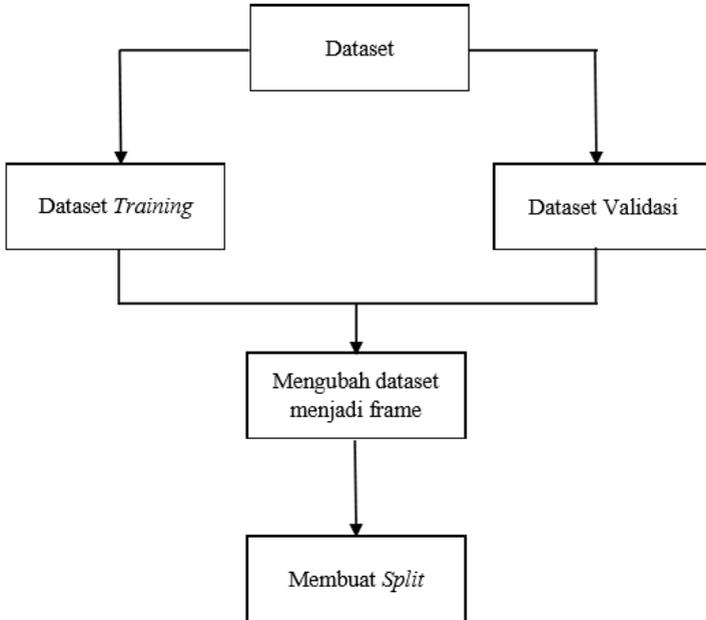
3.1 Desain Sistem

Secara garis besar, terdapat empat proses yang ada pada bagian ini atau seperti yang digambarkan pada Gambar 3.2.

1. Dataset.
Dataset yang digunakan pada kelas ini adalah dataset *UCF-Crime* yang berisi 1250 video.
2. *Preprocessing* data.
Sebelum bisa digunakan, dataset perlu diproses terlebih dahulu. Dataset yang akan digunakan akan diubah menjadi sekumpulan *frame* dan dibuat *split* untuk memisahkan data yang akan digunakan sebagai input *training* dan validasi
3. *Training*.
Training merupakan sebuah proses pelatihan model 3D CNN menggunakan dataset *UCF-Crime*. *Training* pada sistem ini akan dilakukan dengan menggunakan 5 model yang berbeda - beda.
4. Evaluasi.
Model yang telah melewati proses *training* kemudian akan dievaluasi untuk menilai performanya. Evaluasi model akan melakukan pengujian performasi terhadap model hasil training.

3.2 Dataset

Sistem ini dibuat dengan menggunakan dataset UCF-Crime yang berisi 15 jenis *anomaly*, yaitu *Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, Road Accidents, Robbery, Shooting, Shoplifting, Stealing, Vandalism, Testing Normal Videos*, dan *Training Normal Videos* yang jumlahnya bisa dilihat pada tabel 3.1. Video - video yang terdapat didalam dataset tersebut merupakan kumpulan video yang didapat dari *YouTube* dan *LiveLeak* dengan menggunakan *keyword* pencarian yang bervariasi dan menggunakan bahasa yang berbeda - beda untuk setiap *anomaly*. Video yang termasuk dalam salah satu kondisi berikut: diedit secara manual, video usil(*prank*), tidak direkam dengan menggunakan CCTV, video berita, video kompilasi, dan video yang *anomaly*-nya tidak terlihat dengan jelas tidak akan dimasukkan kedalam dataset. Se-



Gambar 3.2: *Processing* dataset

telah dataset didapat, agar dapat digunakan untuk proses *training*, maka dataset harus diubah menjadi sekumpulan *frame* dan diberi *split*. Selanjutnya, dataset akan diproses sesuai dengan gambar 3.2.

3.2.1 Mengubah Dataset Menjadi Frame

Karena dataset yang bisa digunakan adalah gambar, dataset berupa video yang telah didapat diubah dengan menggunakan program yang sudah dibuat sebelumnya menjadi sekumpulan *frame*. Masing - masing video yang ada pada dataset ini memiliki 30*frame*/detik. Sesuai dengan tabel 3.1, dataset berupa video yang berjumlah sebanyak 1250 video, menjadi kurang lebih 5.000.000 *frame*.

Tabel 3.1: Dataset UCF-Crime

Nama <i>Anomaly</i>	Jumlah Video	Jumlah <i>Frame</i>
<i>Abuse</i>	50	193.497
<i>Arrest</i>	50	297.395
<i>Arson</i>	50	271.903
<i>Assault</i>	50	129.946
<i>Burglary</i>	100	471.206
<i>Explosion</i>	50	252.404
<i>Fighting</i>	50	258.895
<i>Road Accidents</i>	150	260.873
<i>Robbery</i>	150	422.594
<i>Shooting</i>	50	147.473
<i>Shoplifting</i>	50	324.367
<i>Stealing</i>	100	467.416
<i>Testing Normal Videos</i>	150	648.905
<i>Training Normal Videos</i>	150	784.492
<i>Vandalism</i>	50	147.162
Total	1250	5.078.528

3.2.2 *Split*

Split yang digunakan berupa file berformat .txt yang berisi nama folder - folder video yang akan dipanggil sebagai *input* untuk proses *training* dan validasi. *Split* yang digunakan pada sistem ini memanggil tiap 16 *frame* yang kemudian digunakan sebagai 1 data *input* pada proses *training*. *Split* juga memberikan penomoran yang menunjukkan jumlah jenis video atau kelas yang digunakan sebagai input *training*.

3.3 *Training*

Proses *training* yang akan dilakukan pada penelitian ini akan dilakukan dengan menggunakan algoritme *Stochastic Gradient Descent* (SGD). Pada proses *training* ini, digunakan data *train* yang sudah diubah menjadi sekumpulan *frame* yang kemudian akan di-*resize* menjadi berukuran 128×171 lalu digunakan untuk proses *training* sehingga membentuk sebuah pola yang hasilnya berbentuk bobot atau *weight*. *Training* akan dilakukan dengan menggunakan

5 model *3D Convolutional Neural Networks* atau 3D CNN yang berbeda - beda yang merupakan hasil modifikasi dari model arsitektur C3D. Pada proses *training* ini perlu dilakukan penentuan ukuran *batch size*, dan *epoch* yang akan dijelaskan sebagai berikut :

1. *Batch Size*

Penentuan *batch size* berdasarkan jumlah sampel data yang ingin disebarakan ke seluruh neural network dalam setiap iterasi (*train steps*). Pada proses *training* kali ini ukuran *batch size* yang digunakan berjumlah 12, yang berarti setiap satu *step* akan disebarakan 12 data *training* ke *neural network*.

2. *Epoch*

Epoch merupakan satu set putaran *training steps*. Beberapa *epoch* diperlukan untuk pelatihan sebuah neural network sehingga didapatkan kesalahan (*loss*) mendekati nol. Jumlah *epoch* pada proses *training model* ini sejumlah 10 kali.

3.3.1 Training Models

Pada penelitian ini, akan digunakan 5 macam model yang merupakan hasil modifikasi dari arsitektur C3D. Hasil dari proses *training* keempat model ini akan diuji coba dan dibandingkan untuk mengetahui model mana yang terbaik.

3.3.1.1 Model Pertama

Conv1	Pool1	Conv2	Pool2	Conv3	Pool3	Conv4	Pool4	Conv5	Pool5	fc6	fc7	Softmax
64		128		128		256		256		2048	2048	

Gambar 3.3: Susunan *layer* pada model pertama

Model pertama yang digunakan merupakan model yang paling mirip dengan model dari arsitektur C3D. sesuai dengan Gambar 3.3, model ini terdiri dari 5 *3D Convolution Layer* yang masing - masing memiliki *kernel* berukuran $3 \times 3 \times 3$ dan *stride* $1 \times 1 \times 1$. Masing - masing *3D Convolution Layer* diikuti dengan *3D max pooling layer* dengan ukuran $2 \times 2 \times 2$ dan *stride* $2 \times 2 \times 2$ kecuali untuk *3D max pooling layer* pertama yang memiliki ukuran $2 \times 2 \times 1$ dan *stride* $2 \times 2 \times 1$. Masing - masing *fully-connected layer*

memiliki 4096 *output*. Model ini menggunakan *activation function* ReLU dan *softmax* yang diletakkan sebagai *layer* terakhir.

3.3.1.2 Model Kedua

Conv1 64	Pool1	Conv2 128	Pool2	Conv3 128	Pool3	Conv4 256	Pool4	Conv5 256	Pool5	fc6 2048	fc7 2048	Softmax
-------------	-------	--------------	-------	--------------	-------	--------------	-------	--------------	-------	-------------	-------------	---------

Gambar 3.4: Susunan *layer* pada model kedua

Model kedua tidak jauh berbeda dengan model pertama, sesuai dengan Gambar 3.4. Model ini juga terdiri dari 5 *3D Convolution Layer*, 5 *3D max pooling layer*, dan 2 *fully-connected layer* yang semuanya memiliki ukuran *kernel* dan *stride* yang sama dengan model pertama. Hanya saja model ini tidak menggunakan *activation function* ReLU, melainkan ELU.

3.3.1.3 Model Ketiga

Conv1 64	Pool1	Conv2 128	Pool2	Conv3 128	Pool3	Conv4 256	Pool4	Conv5 256	3D GAP	fc6 2048	fc7 2048	Softmax
-------------	-------	--------------	-------	--------------	-------	--------------	-------	--------------	--------	-------------	-------------	---------

Gambar 3.5: Susunan *layer* pada model ketiga

Model ketiga yang ditunjukkan pada gambar 3.5 tidak berbeda jauh dengan model pertama yang sudah dijelaskan sebelumnya. Yang membedakan model ini dari model pertama adalah model ini tidak menggunakan *3D max pooling* melainkan *3D Global Average Pooling*.

3.3.1.4 Model Keempat

Conv1 64	Pool1	Conv2 128	Pool2	Conv3 128	Pool3	Conv4 256	Pool4	Conv5 256	3D GAP	fc6 2048	fc7 2048	Softmax
-------------	-------	--------------	-------	--------------	-------	--------------	-------	--------------	--------	-------------	-------------	---------

Gambar 3.6: Susunan *layer* pada model keempat

Model keempat yang ditunjukkan pada gambar 3.6 memiliki

susunan *layer* yang sama dengan susunan *layer* pada model ketiga, hanya saja model keempat ini tidak menggunakan ReLU melainkan ELU.

3.3.1.5 Model Kelima

Conv1	Pool1	Conv2	Pool2	Conv3	Pool3	Conv4	Pool4	Conv5	Pool5	fc6	fc7	Softmax
64		128		128		256		256		2048	2048	

Gambar 3.7: Susunan *layer* pada model kedua

Model terakhir yang ditunjukkan pada gambar 3.7 memiliki susunan *layer* dan *activation function* yang sama dengan model kedua, hanya saja model ini menggunakan 4 *frame* untuk setiap satu *input* data. Sedangkan model yang lainnya menggunakan 16 *frame*.

3.4 Validasi

Setelah proses *training* selesai, proses selanjutnya adalah melakukan evaluasi terhadap hasil *training* tersebut. Evaluasi ini bertujuan untuk melakukan pengecekan terhadap proses training model yang telah dilakukan. Untuk memvalidasinya digunakan metode *cross validation*. Metode ini merupakan metode statistik untuk mengevaluasi dan membandingkan algoritma learning. Dalam *cross validation*, data train dan data validasi disilangkan dalam putaran berturut-turut sehingga setiap node data memiliki peluang untuk divalidasi.

3.5 Evaluasi

Setelah proses validasi selesai, proses selanjutnya adalah melakukan evaluasi terhadap hasil *training* tersebut. Evaluasi ini bertujuan untuk melakukan pengecekan terhadap proses training model yang telah dilakukan. Untuk meengevaluasi model digunakan metode *voting* secara manual yang akan dilakukan terhadap kelas-kelas yang terdeteksi pada video uji coba. Kelas yang paling banyak terdeteksi pada setiap video uji cobalah yang merupakan hasil klasifikasi video tersebut. Hasil dari evaluasi dari tiap model akan dibandingkan untuk mengetahui model mana yang performanya terbaik. Performa setiap model akan dapat diketahui akurasiya menggunakan *confusion matrix*.

Halaman ini sengaja dikosongkan

BAB 4

PENGUJIAN DAN ANALISIS

Pada penelitian ini, dipaparkan hasil pengujian serta analisis dari desain sistem dan implementasi. Terdapat lima macam pengujian yang dilakukan pada penelitian ini. Data yang digunakan dalam pengujian diperoleh dari video yang sudah disediakan di dalam dataset UCF-Crime. Pengujian dilakukan dalam beberapa bagian sebagai berikut :

4.1 *Training Data*

Proses *training* data dilakukan sebanyak 5 kali dengan menggunakan *3D Convolutional Neural Network* dengan 5 model yang berbeda - beda yang merupakan arsitektur C3D sesuai gambar 4.1 yang telah dimodifikasi dengan menggunakan python. Training dilakukan dengan menggunakan GPU NVIDIA GEFORCE GTX 1070 6GB. Proses training berlangsung selama kurang lebih 4 hari dengan menggunakan 10 epoch, *learning rate* 0.003 yang akan dibagi 10 setiap 4 *epoch*, dan batch size 12 dengan *class* berjumlah 14. Saat proses ini berlangsung, semua input (*dataset*) di-*resize* menjadi 128×171 .



Gambar 4.1: Jumlah *layer* arsitektur C3D[1].

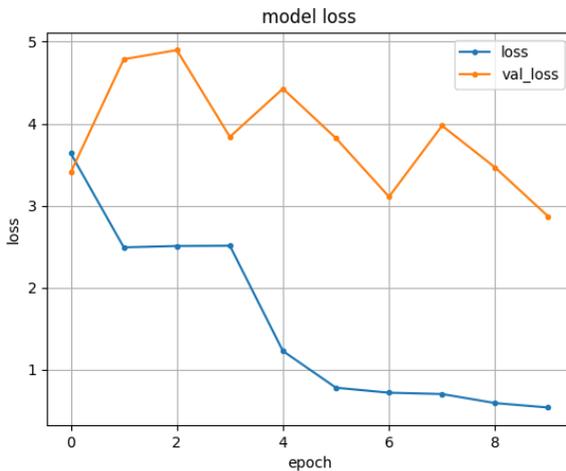
4.2 Pengujian

Pada pengujian ini, akan digunakan masing - masing 5 video uji coba pada setiap kelas yang ada untuk proses klasifikasi menggunakan program yang sudah dibuat sebelumnya. Hasil dari proses berupa nama kelas serta probabilitasnya dan file .txt yang berisikan nama kelas - kelas yang terdeteksi pada tiap video. Kemudian akan dilakukan pengecekan akurasi secara manual dengan cara melakukan *voting* terhadap hasil prediksi pada tiap video yang tertulis di dalam file .txt yang merupakan hasil dari pengujian. Klasifikasi dilakukan berdasarkan kelas - kelas berikut:

1. *Abuse.*
2. *Arrest.*
3. *Arson.*
4. *Assault.*
5. *Burglary.*
6. *Explosion.*
7. *Fighting.*
8. *Road Accidents.*
9. *Robbery.*
10. *Shooting.*
11. *Shoplifting.*
12. *Stealing.*
13. *Vandalism.*
14. *Normal Videos.*

4.2.1 Pengujian Model Pertama

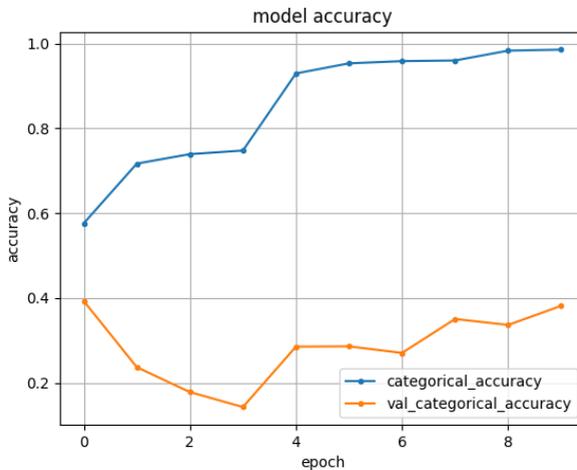
Pengujian pertama dilakukan dengan menggunakan *weight* yang didapatkan dari hasil *training* menggunakan model pertama. Lang-



Gambar 4.2: Grafik nilai *loss training* model pertama.

kah pertama untuk melakukan pengujian terhadap model pertama yaitu dengan melakukan analisa *cross-validation history* saat proses *training*. Dari grafik Gambar 4.2 menunjukkan bahwa nilai *loss* pada proses training ini terus mengalami penurunan.

Dari grafik Gambar 4.3 menunjukkan bahwa nilai akurasi pada proses *training* pertama terus mengalami kenaikan. Dapat disimpulkan bahwa pada proses *training* tidak mengalami *underfitting* maupun *overfitting*.



Gambar 4.3: Grafik nilai *accuracy training* model pertama.

Gambar 4.4 menunjukkan *confusion matrix* yang menunjukkan semua jumlah prediksi benar dan salah dari setiap video yang digunakan sebagai pengujian pada setiap kelas yang menggunakan hasil proses *training* dari model pertama. Dari *confusion matrix* tersebut bisa kita lihat bahwa model ini berhasil memprediksi beberapa video pengujian dengan benar, yaitu pada kelas *burglary* sebanyak 1 video, kelas *normal* sebanyak 4 video, kelas *road accidents* sebanyak 2 video, kelas *robbery* sebanyak 2 video, kelas *shoplifting* sebanyak 1 video, dan kelas *stealing* sebanyak 1 video.

Tabel 4.1 menunjukkan jumlah prediksi yang benar pada pengujian yang menggunakan masing - masing 5 video setiap kelas sebagai input. Dari tabel tersebut bisa kita lihat bahwa model ini berhasil mendapatkan beberapa prediksi sesuai yaitu pada kelas *burglary*, *normal*, *road accidents*, *robbery*, *shoplifting*, dan *stealing*.

Tabel 4.1: Pengujian Pada Hasil *Training* Model Pertama

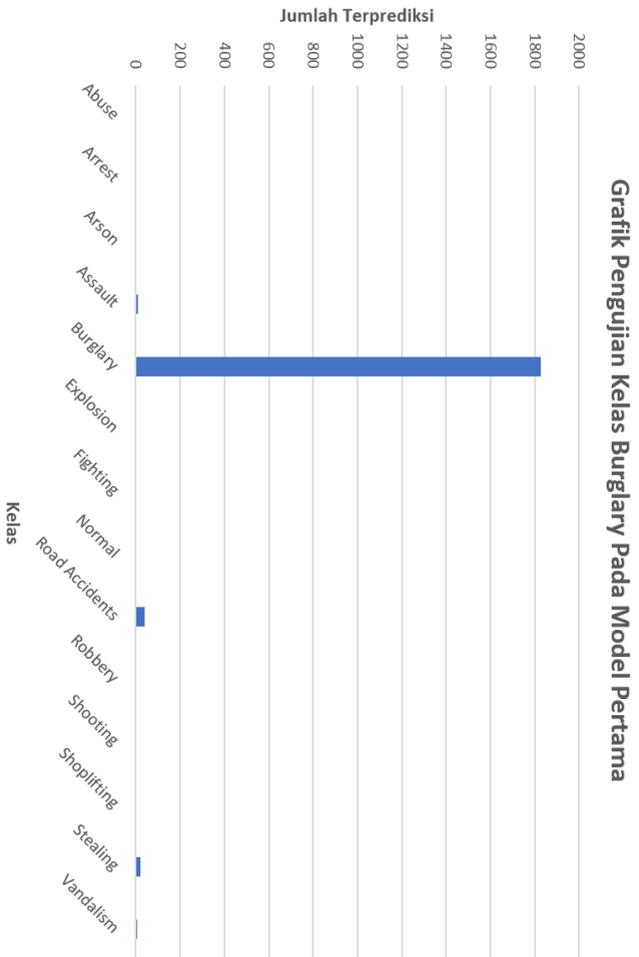
Kelas	Prediksi Sesuai	Persentase
<i>Abuse</i>	0/5	0%
<i>Arrest</i>	0/5	0%
<i>Arson</i>	0/5	0%
<i>Assault</i>	0/5	0%
<i>Burglary</i>	1/5	20%
<i>Explosion</i>	0/5	0%
<i>Fighting</i>	0/5	0%
<i>Normal</i>	4/5	80%
<i>Road Accidents</i>	2/5	40%
<i>Robbery</i>	2/5	40%
<i>Shooting</i>	0/5	0%
<i>Shoplifting</i>	1/5	20%
<i>Stealing</i>	1/5	20%
<i>Vandalism</i>	0/5	0%
Total	11/70	15,7%

Berikut disertakan beberapa grafik hasil pengujian pada salah satu video dari beberapa kelas pada model ini yang ditunjukkan pada gambar 4.5, gambar 4.6, dan gambar 4.7.

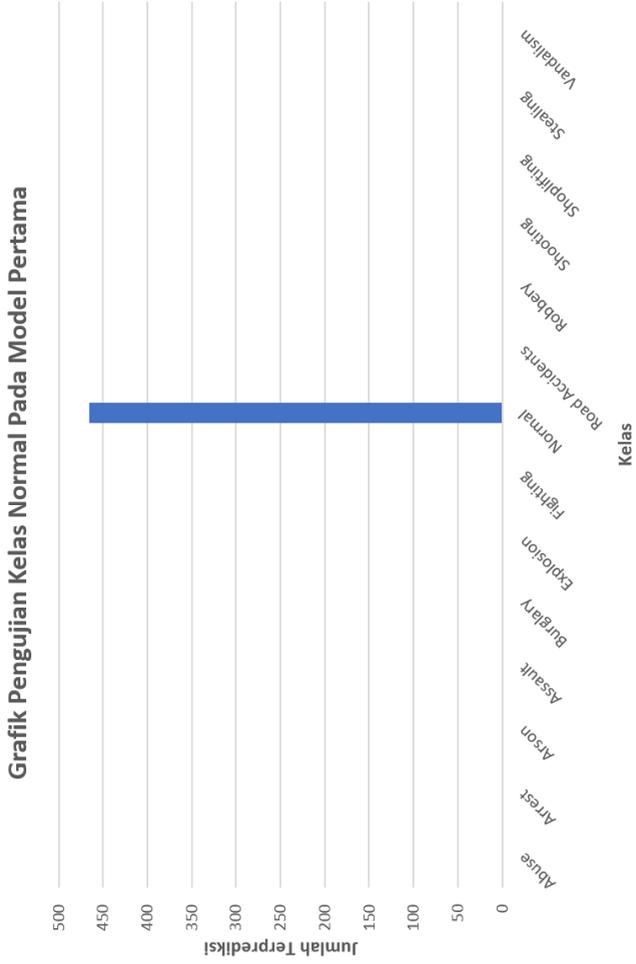
P R E D I C T E D

	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	Normal	Road Accidents	Robbery	Shooting	Shoplifting	Stealing	Vandalism
A	0	0	0	0	0	0	0.4	0.2	0	0.4	0	0	0	0
C	0.2	0	0	0	0	0	0	0.2	0.2	0.2	0	0	0	0.2
T	0.6	0	0	0	0	0	0	0	0.2	0.2	0	0	0	0
U	0	0	0	0	0	0	0	0.2	0.2	0.4	0	0.2	0	0
A	0.2	0	0.2	0	0.2	0	0	0	0.4	0	0	0	0	0
L	0	0	0.2	0	0	0	0	0.2	0.2	0	0	0.2	0	0.2
	0	0.2	0	0	0	0	0	0.4	0.4	0	0	0	0	0
	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0.2
	0	0.2	0	0	0	0.2	0	0.2	0.4	0	0	0	0	0
	0	0	0	0	0	0.2	0	0.2	0	0.4	0.2	0	0	0
	0	0	0.2	0	0	0	0.2	0.2	0.4	0	0	0	0	0
	0	0	0	0	0	0	0.2	0	0	0.6	0	0.2	0	0
	0	0	0	0	0	0	0	0	0.4	0	0	0	0.2	0.4
	0	0	0	0	0	0	0.4	0.2	0	0.2	0	0	0.2	0

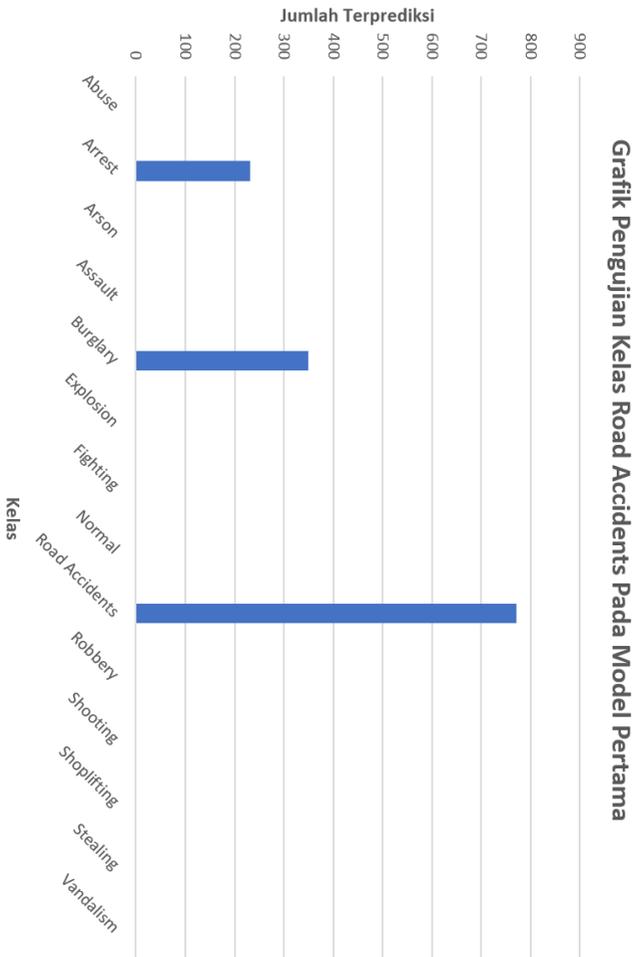
Gambar 4.4: *Confusion Matrix* nilai akurasi model pertama



Gambar 4.5: Grafik hasil dari pengujian pada salah satu video pada *training* model pertama.



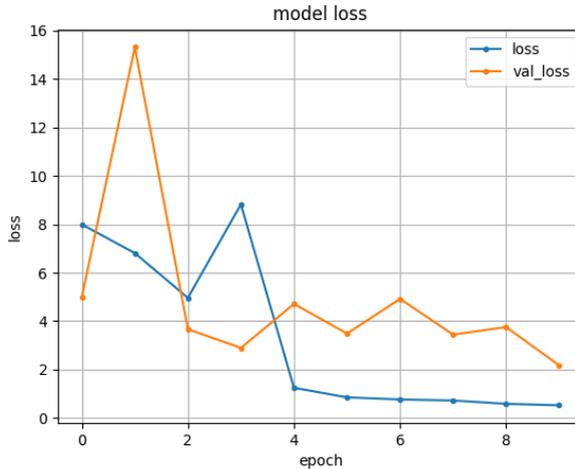
Gambar 4.6: Grafik hasil dari pengujian pada salah satu video pada *training* model pertama.



Gambar 4.7: Grafik hasil dari pengujian pada salah satu video pada *training* model pertama.

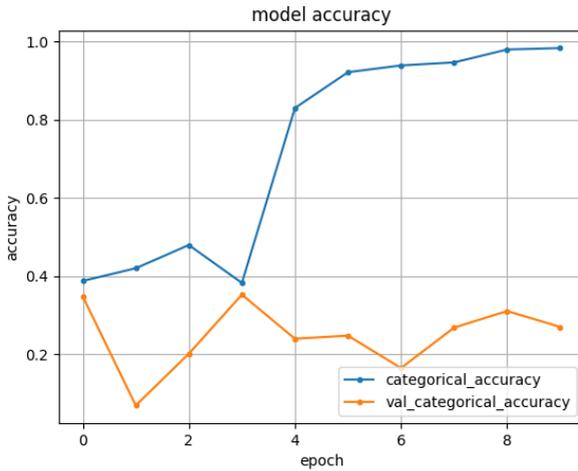
4.2.2 Pengujian Model Kedua

Pada pengujian model kedua, model yang digunakan untuk *training* tidak beda jauh dengan pengujian pertama, hanya saja *activation function* yang digunakan diubah dari ReLU menjadi ELU. Sama seperti model pertama, akan dilakukan pengujian menggunakan analisa *cross-validation*. Dari grafik Gambar 4.8 dapat dilihat bahwa nilai *loss* pada proses *training* kedua terus mengalami penurunan.



Gambar 4.8: Grafik nilai *loss training* model kedua.

Dari grafik Gambar 4.9 menunjukkan bahwa nilai akurasi pada proses *training* kedua terus mengalami kenaikan. Dapat disimpulkan bahwa pada proses *training* tidak mengalami *underfitting* maupun *overfitting*.



Gambar 4.9: Grafik nilai *accuracy training* model kedua.

Gambar 4.10 menunjukkan *confusion matrix* yang menunjukkan semua jumlah prediksi benar dan salah dari setiap video yang digunakan sebagai pengujian pada setiap kelas yang menggunakan hasil proses *training* dari model kedua. Dari *confusion matrix* tersebut bisa kita lihat bahwa model ini berhasil memprediksi beberapa video pengujian dengan benar, yaitu pada kelas *arrest* sebanyak 2 video, kelas *arson* sebanyak 2 video, kelas *burglary* sebanyak 2 video, kelas *explosion* sebanyak 2 video, kelas *normal* sebanyak 4 video, kelas *road accidents* sebanyak 1 video, kelas *Robbery* sebanyak 2 video, kelas *shoplifting* sebanyak 2 video, dan kelas *stealing* sebanyak 2 video.

P R E D I C T E D

	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	Normal	Road Accidents	Robbery	Shooting	Shoplifting	Stealing	Vandalism
A	0	0	0	0.2	0	0	0.2	0.2	0	0.2	0	0	0.2	0
C	0	0.4	0	0	0	0	0	0.4	0.2	0	0	0	0	0
T	0	0	0.4	0	0.4	0	0	0	0	0.2	0	0	0	0
U	0	0	0	0	0	0	0.2	0	0.2	0.4	0	0	0	0.2
A	0	0	0.2	0	0.4	0	0.2	0	0	0	0	0	0	0.2
L	0.4	0	0	0	0	0.4	0	0	0.2	0	0	0	0	0
	0.2	0	0	0	0	0	0	0.2	0	0.2	0	0.2	0.2	0
	0	0.2	0	0	0	0	0	0.8	0	0	0	0	0	0
	0	0.2	0	0	0	0.2	0	0.2	0.2	0	0	0	0.2	0
	0	0.2	0	0	0	0.2	0	0.2	0	0.4	0	0	0	0
	0.2	0	0	0	0.2	0	0	0.4	0	0	0	0	0.2	0
	0.2	0	0	0.2	0.2	0	0	0	0	0	0	0.4	0	0
	0	0.2	0	0	0	0	0	0	0.4	0	0	0	0.4	0
	0	0	0	0	0.2	0	0.2	0	0.2	0.2	0	0	0.2	0

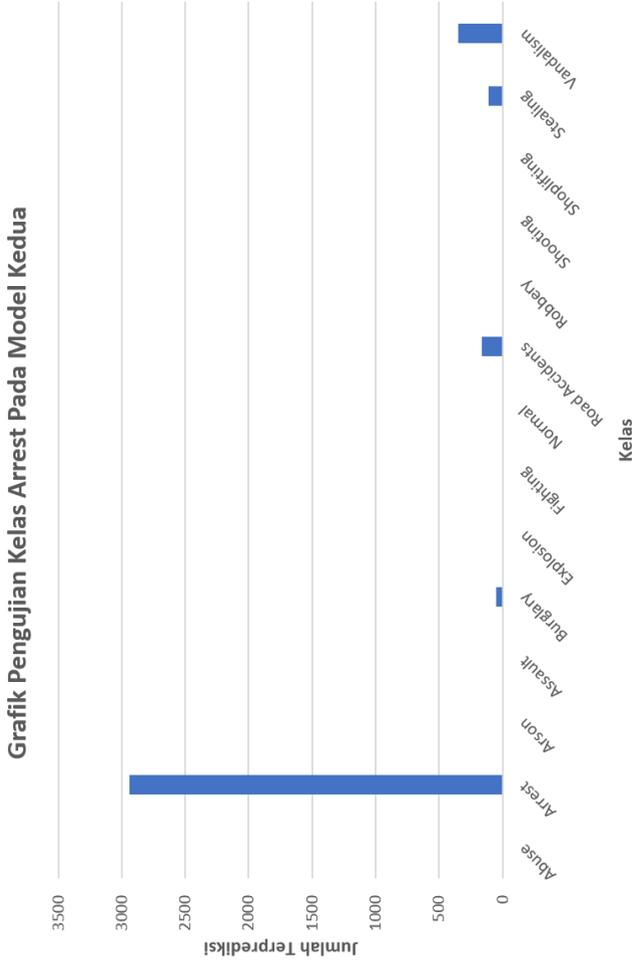
Gambar 4.10: Confusion Matrix nilai akurasi model kedua

Tabel 4.2 menunjukkan jumlah prediksi yang benar pada pengujian yang menggunakan masing - masing 5 video setiap kelas sebagai input. Dari tabel tersebut bisa kita lihat bahwa model ini berhasil mendapatkan beberapa prediksi sesuai yaitu pada kelas *arrest*, *arson*, *burglary*, *explosion*, *normal*, *road accidents*, *robbery*, *shoplifting*, dan *stealing*.

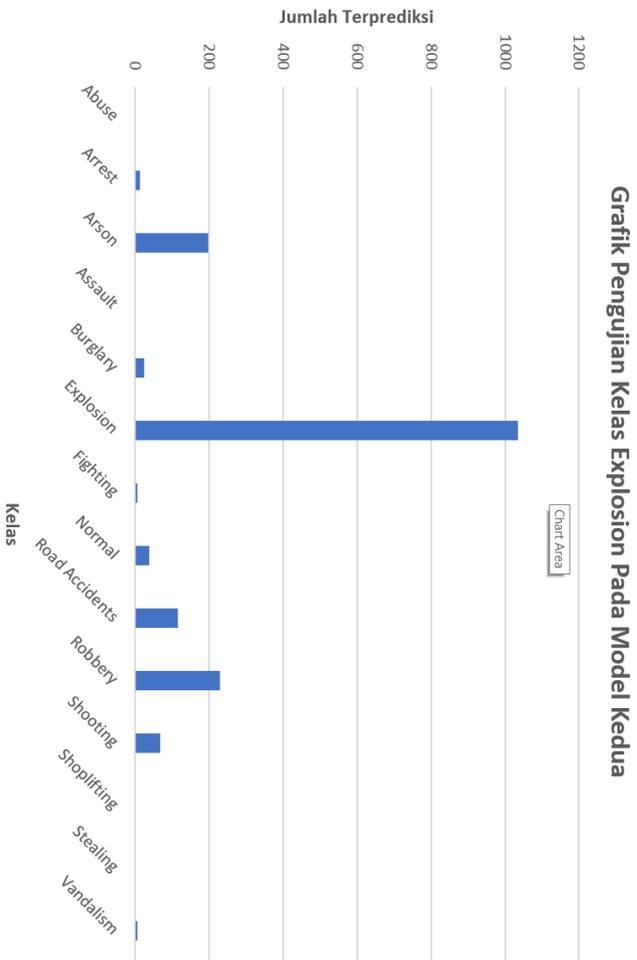
Tabel 4.2: Pengujian Pada Hasil *Training* Model Kedua

Kelas	Prediksi Sesuai	Persentase
<i>Abuse</i>	0/5	0%
<i>Arrest</i>	2/5	40%
<i>Arson</i>	2/5	40%
<i>Assault</i>	0/5	0%
<i>Burglary</i>	2/5	40%
<i>Explosion</i>	2/5	40%
<i>Fighting</i>	0/5	0%
<i>Normal</i>	4/5	80%
<i>Road Accidents</i>	1/5	20%
<i>Robbery</i>	2/5	40%
<i>Shooting</i>	0/5	0%
<i>Shoplifting</i>	2/5	40%
<i>Stealing</i>	2/5	40%
<i>Vandalism</i>	0/5	0%
Total	19/70	27,1%

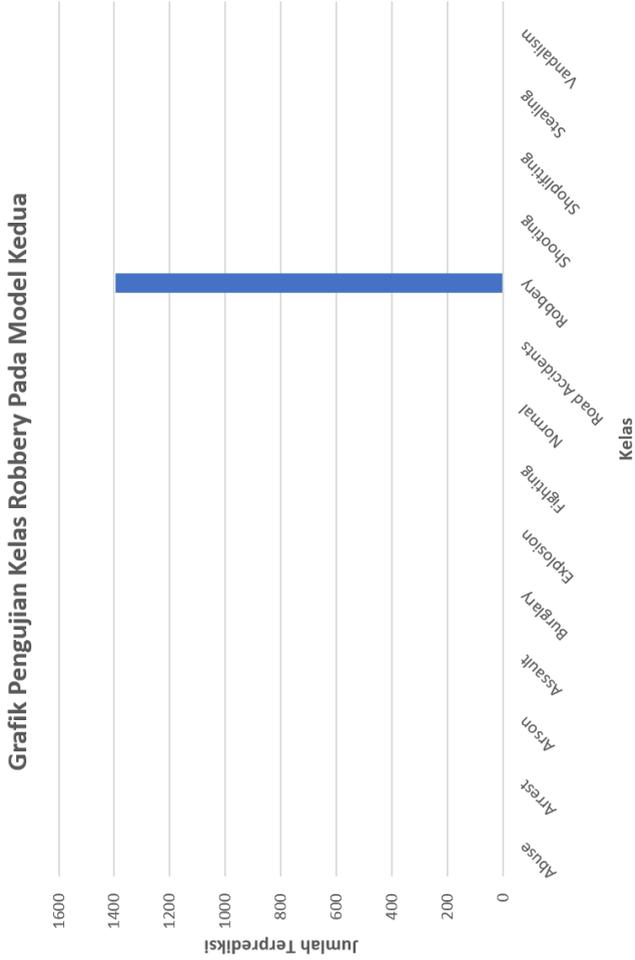
Berikut disertakan beberapa grafik hasil pengujian pada salah satu video dari beberapa kelas pada model ini yang ditunjukkan pada gambar 4.11, gambar 4.12, dan gambar 4.13.



Gambar 4.11: Grafik hasil dari pengujian pada salah satu video pada *training* model kedua.



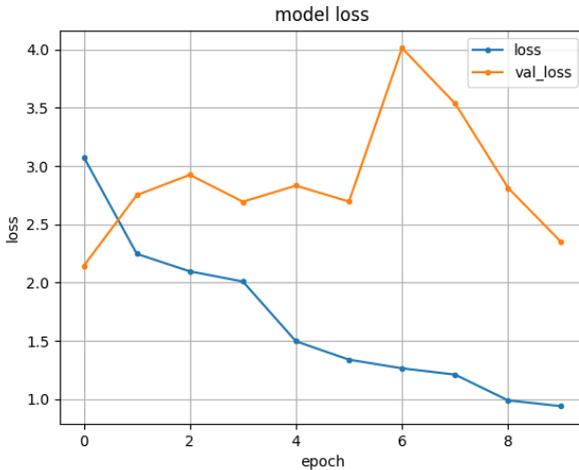
Gambar 4.12: Grafik hasil dari pengujian pada salah satu video pada *training* model kedua.



Gambar 4.13: Grafik hasil dari pengujian pada salah satu video pada *training* model kedua.

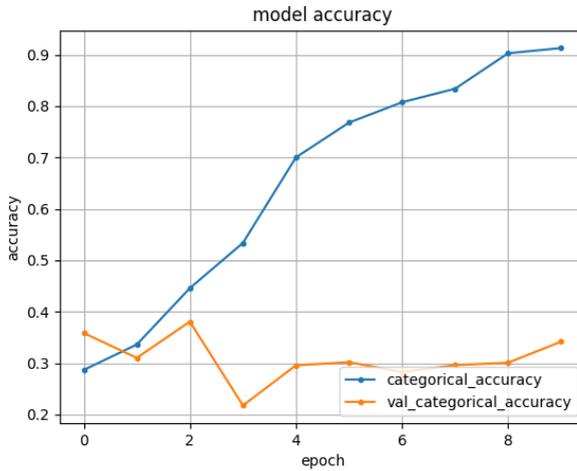
4.2.3 Pengujian Model Ketiga

Pada pengujian model ketiga, model menggunakan *global average pooling* sebagai pengganti dari *max pooling* dan menggunakan *activation function* ReLU. Sama seperti model - model sebelumnya, akan dilakukan pengujian menggunakan analisa *cross-validation*. Pada grafik Gambar 4.14 dapat dilihat bahwa nilai *loss* pada proses *training* ketiga terus mengalami penurunan.



Gambar 4.14: Grafik nilai *loss training* model ketiga.

Pada grafik Gambar 4.15 menunjukkan bahwa nilai akurasi pada proses *training* ketiga terus mengalami kenaikan. Dapat disimpulkan bahwa pada proses *training* tidak mengalami *underfitting* maupun *overfitting*.



Gambar 4.15: Grafik nilai *accuracy training* model ketiga.

Gambar 4.16 menunjukkan *confusion matrix* yang menunjukkan semua jumlah prediksi benar dan salah dari setiap video yang digunakan sebagai pengujian pada setiap kelas yang menggunakan hasil proses *training* dari model ketiga. Dari *confusion matrix* tersebut bisa kita lihat bahwa model ini berhasil memprediksi beberapa video pengujian dengan benar, yaitu pada kelas *burglary* sebanyak 1 video, kelas *normal* sebanyak 4 video, kelas *road accidents* sebanyak 1 video, kelas *Robbery* sebanyak 1 video, kelas *shoplifting* sebanyak 2 video, dan kelas *stealing* sebanyak 3 video.

	P R E D I C T E D													
	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	Normal	Road Accidents	Robbery	Shooting	Shoplifting	Stealing	Vandalism
Abuse	0	0	0	0	0	0	0	0	0.2	0.6	0	0	0.2	0
Arrest	0	0	0	0	0.2	0	0	0.4	0.2	0.2	0	0	0	0
Arson	0.2	0	0	0	0.2	0	0	0	0.2	0	0	0	0.4	0
Assault	0	0	0	0	0	0	0	0	0.6	0	0	0.4	0	0
Burglary	0	0	0.2	0	0.2	0	0	0	0	0	0	0	0.6	0
Explosion	0	0	0	0	0	0	0	0.4	0.2	0	0	0	0	0.4
Fighting	0	0.2	0.2	0	0.2	0	0	0.2	0.2	0	0	0	0	0
Normal	0	0	0	0	0	0	0	0.8	0.2	0	0	0	0	0
Road Accidents	0	0	0	0	0	0.2	0	0	0.2	0.2	0	0	0.2	0.2
Robbery	0.2	0	0	0	0	0	0	0.2	0	0.2	0	0	0	0.4
Shooting	0	0	0	0	0	0	0	0.2	0.2	0.2	0	0	0.2	0.2
Shoplifting	0	0	0	0	0	0	0.2	0	0.2	0.2	0	0.4	0	0
Stealing	0	0	0	0	0	0	0	0	0.2	0.2	0	0	0.6	0
Vandalism	0	0	0	0	0.4	0	0	0.2	0.2	0.2	0	0	0	0

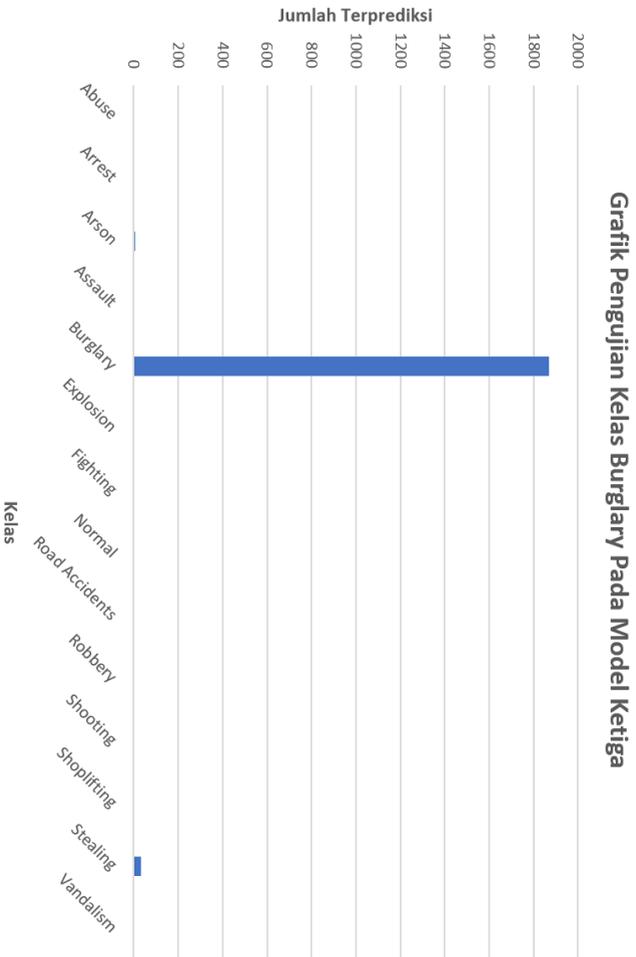
Gambar 4.16: Confusion Matrix nilai akurasi model ketiga

Pada Tabel 4.3 menunjukkan jumlah prediksi yang benar pada pengujian yang menggunakan masing - masing 5 video pada setiap kelas sebagai input. Dari tabel tersebut bisa kita lihat bahwa model ini berhasil mendapatkan beberapa prediksi sesuai yaitu pada kelas *burglary*, *normal*, *road accidents*, *robbery*, *shoplifting*, dan *stealing*.

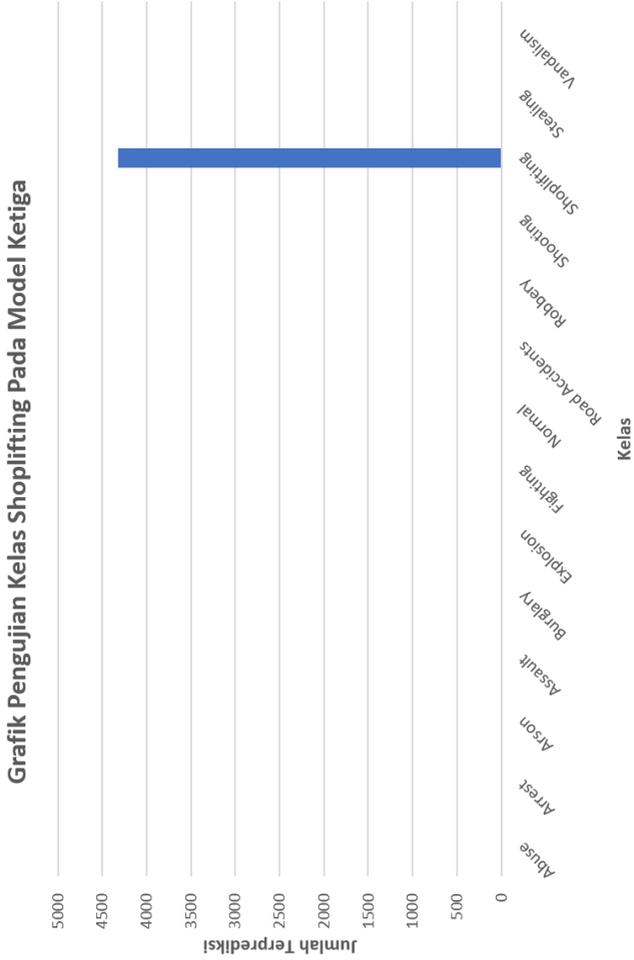
Tabel 4.3: Pengujian Pada Hasil *Training* Model Ketiga

Kelas	Prediksi Sesuai	Persentase
<i>Abuse</i>	0/5	0%
<i>Arrest</i>	0/5	0%
<i>Arson</i>	0/5	0%
<i>Assault</i>	0/5	0%
<i>Burglary</i>	1/5	20%
<i>Explosion</i>	0/5	0%
<i>Fighting</i>	0/5	0%
<i>Normal</i>	4/5	80%
<i>Road Accidents</i>	1/5	20%
<i>Robbery</i>	1/5	20%
<i>Shooting</i>	0/5	0%
<i>Shoplifting</i>	2/5	40%
<i>Stealing</i>	3/5	60%
<i>Vandalism</i>	0/5	0%
Total	12/70	17,1%

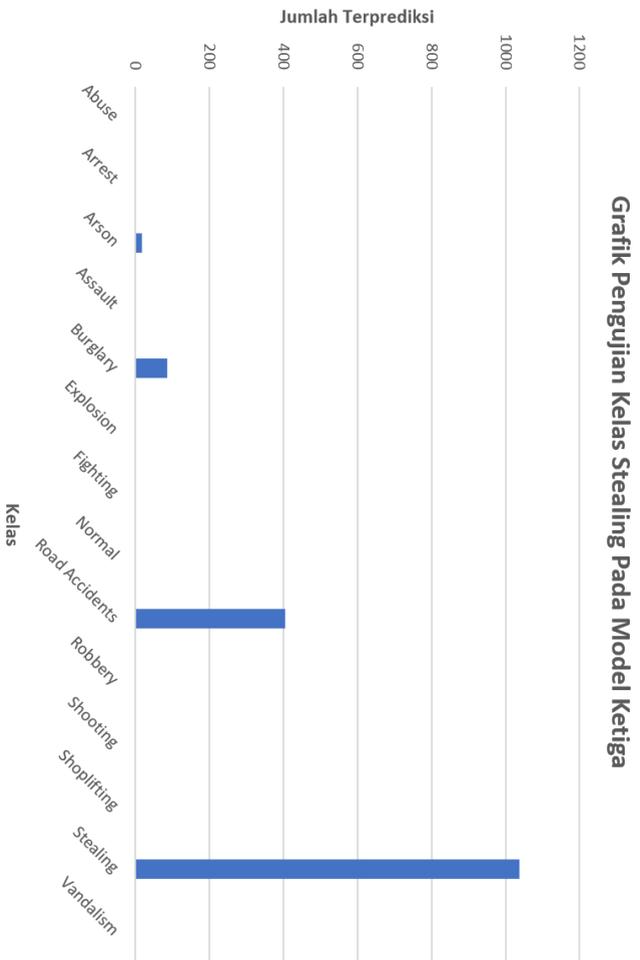
Berikut disertakan beberapa grafik hasil pengujian pada salah satu video dari beberapa kelas pada model ini yang ditunjukkan pada gambar 4.17, gambar 4.18, dan gambar 4.19.



Gambar 4.17: Grafik hasil dari pengujian pada salah satu video pada *training* model ketiga.



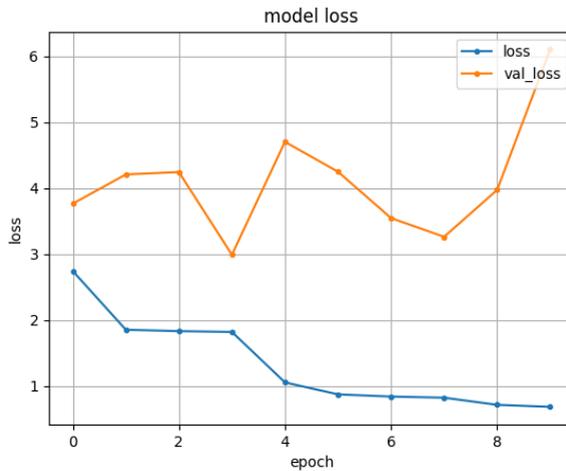
Gambar 4.18: Grafik hasil dari pengujian pada salah satu video pada *training* model ketiga.



Gambar 4.19: Grafik hasil dari pengujian pada salah satu video pada *training* model ketiga.

4.2.4 Pengujian Model Keempat

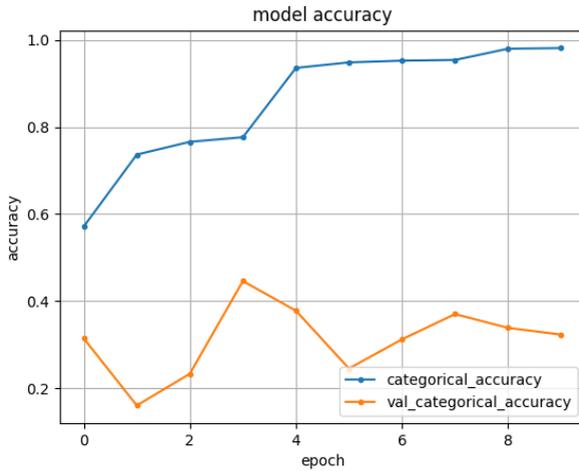
Pada pengujian model keempat, model tidak jauh berbeda dari model yang digunakan pada pengujian ketiga, hanya saja model ini menggunakan *activation function* ELU. Sama seperti model - model sebelumnya, akan dilakukan pengujian menggunakan analisa *cross-validation*. Dari grafik 4.20 dapat dilihat bahwa nilai *loss* pada proses *training* keempat terus mengalami penurunan.



Gambar 4.20: Grafik nilai *loss training* model keempat.

Dari grafik 4.21 menunjukkan bahwa nilai akurasi pada proses *training* keempat terus mengalami kenaikan. Dapat disimpulkan bahwa pada proses *training* tidak mengalami *underfitting* maupun *overfitting*.

Gambar 4.22 menunjukkan *confusion matrix* yang menunjuk-



Gambar 4.21: Grafik nilai *accuracy training* model keempat.

an semua jumlah prediksi benar dan salah dari setiap video yang digunakan sebagai pengujian pada setiap kelas yang menggunakan hasil proses *training* dari model keempat. Dari *confusion matrix* tersebut bisa kita lihat bahwa model ini berhasil memprediksi beberapa video pengujian dengan benar, yaitu pada kelas *abuse* sebanyak 1 video, kelas *arson* sebanyak 1 video, kelas *burglary* sebanyak 1 video, kelas normal sebanyak 4 video, kelas *road accidents* sebanyak 2 video, kelas *Robbery* sebanyak 2 video, dan kelas *stealing* sebanyak 1 video.

P R E D I C T E D

	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	Normal	Road Accidents	Robbery	Shooting	Shoplifting	Stealing	Vandalism
A	0.2	0	0	0	0	0	0	0.2	0	0.4	0	0	0.2	0
C	0	0	0	0	0	0	0	0.4	0.6	0	0	0	0	0
T	0	0	0.2	0	0.4	0	0	0	0.2	0.2	0	0	0	0
U	0	0.2	0	0	0	0	0	0	0.4	0.4	0	0	0	0
A	0.2	0	0.2	0	0.2	0	0	0	0.2	0	0	0	0.2	0
L	0.2	0.2	0.2	0	0	0	0	0.2	0.2	0	0	0	0	0
	0	0.2	0	0	0	0	0	0.8	0	0	0	0	0	0
	0.2	0	0	0	0.2	0	0	0	0.4	0.2	0	0	0	0
	0	0	0.2	0	0	0	0	0.2	0	0.4	0	0	0	0.2
	0	0	0	0	0.2	0	0.2	0.2	0.4	0	0	0	0	0
	0.2	0	0	0	0.2	0	0	0	0	0.6	0	0	0	0
	0	0	0	0	0.4	0	0	0.2	0.2	0	0	0	0.2	0
	0.2	0	0	0	0.2	0	0	0.2	0	0.2	0	0	0.2	0

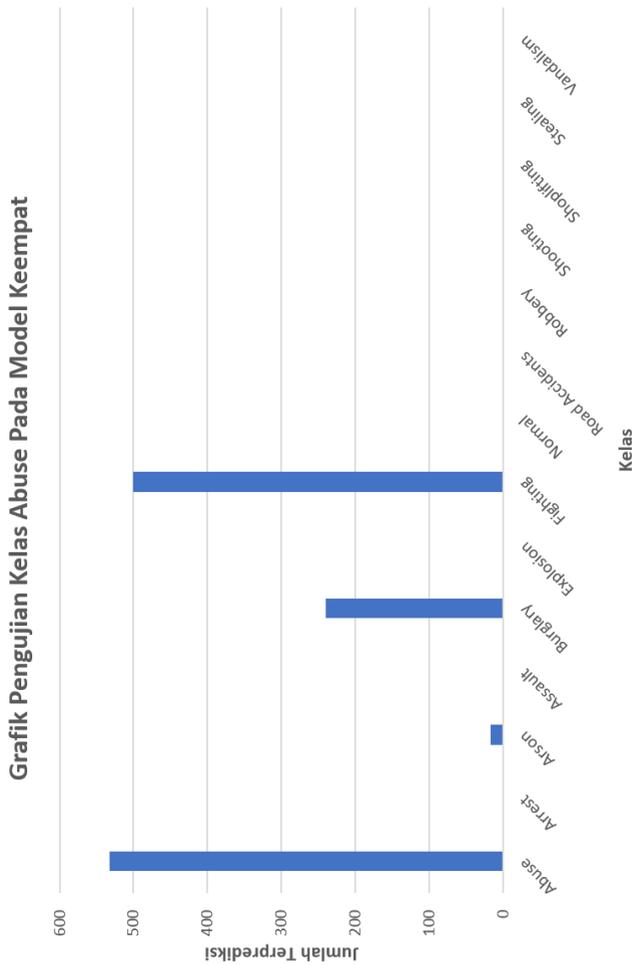
Gambar 4.22: Confusion Matrix nilai akurasi model keempat

Pada Tabel 4.4 menunjukkan jumlah prediksi yang benar pada pengujian yang menggunakan masing - masing 5 video pada setiap kelas sebagai input. Dari tabel tersebut bisa kita lihat bahwa model ini berhasil mendapatkan beberapa prediksi sesuai yaitu pada kelas *abuse*, *arson*, *burglary*, *normal*, *road accidents*, *robbery*, dan *stealing*.

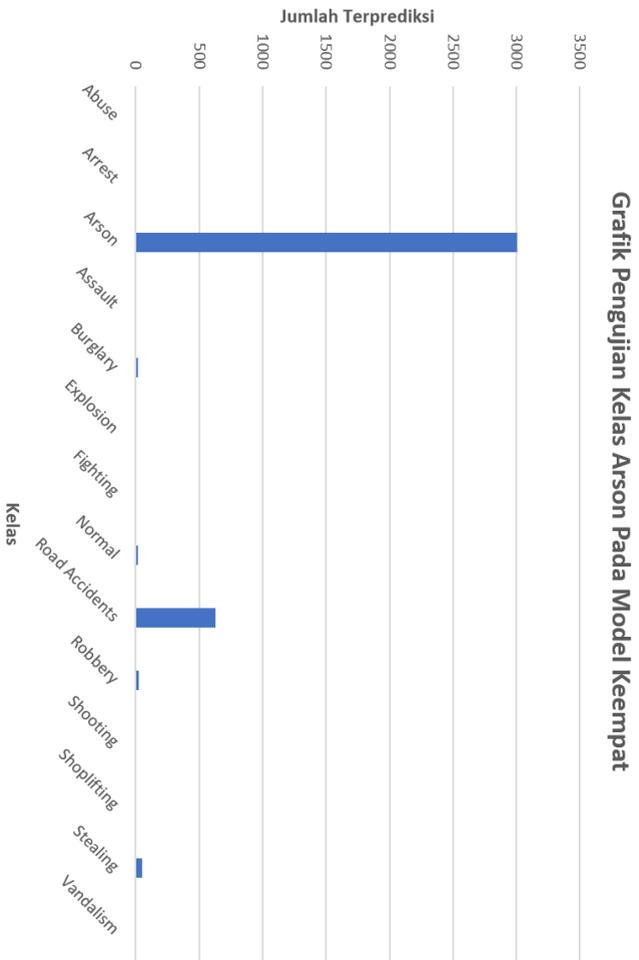
Tabel 4.4: Pengujian Pada Hasil *Training* Model Keempat

Kelas	Prediksi Sesuai	Persentase
<i>Abuse</i>	1/5	20%
<i>Arrest</i>	0/5	0%
<i>Arson</i>	1/5	20%
<i>Assault</i>	0/5	0%
<i>Burglary</i>	1/5	20%
<i>Explosion</i>	0/5	0%
<i>Fighting</i>	0/5	0%
<i>Normal</i>	4/5	80%
<i>Road Accidents</i>	2/5	40%
<i>Robbery</i>	2/5	40%
<i>Shooting</i>	0/5	0%
<i>Shoplifting</i>	0/5	0%
<i>Stealing</i>	1/5	20%
<i>Vandalism</i>	0/5	0%
Total	12/70	17,1%

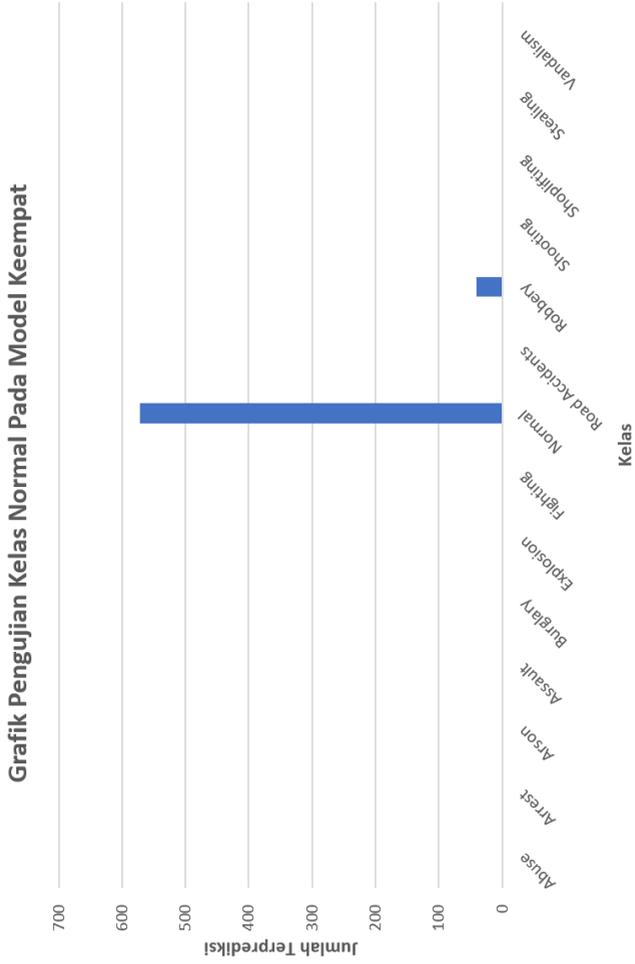
Berikut disertakan beberapa grafik hasil pengujian salah satu video dari beberapa kelas pada model ini yang ditunjukkan pada gambar 4.23, gambar 4.24, dan gambar 4.25.



Gambar 4.23: Grafik hasil dari pengujian pada salah satu video pada *training* model keempat.



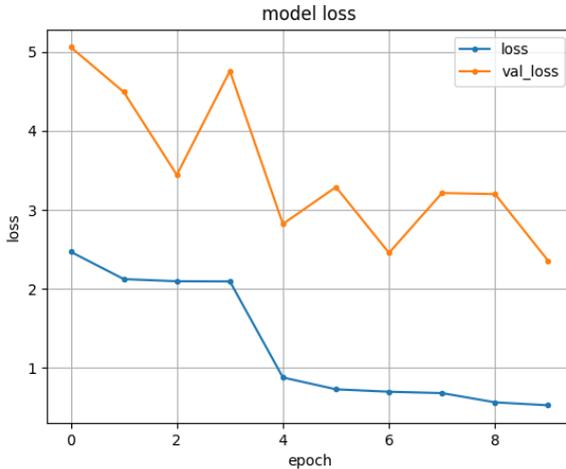
Gambar 4.24: Grafik hasil dari pengujian pada salah satu video pada *training* model keempat.



Gambar 4.25: Grafik hasil dari pengujian pada salah satu video pada *training* model keempat.

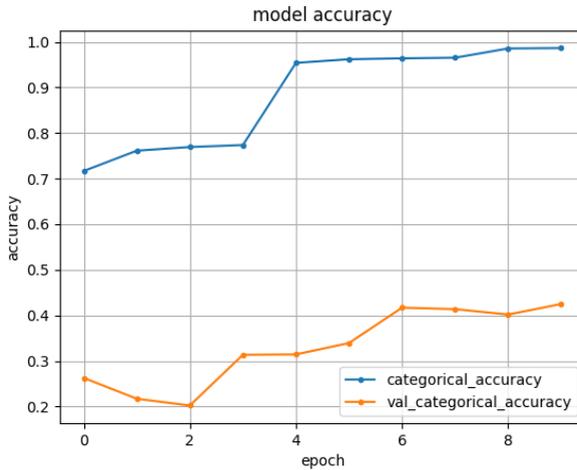
4.2.5 Pengujian Model Kelima

Pada pengujian model kelima, model yang digunakan untuk *training* tidak beda jauh dengan pengujian kedua, hanya saja model ini menggunakan 4 *frame* untuk setiap satu *input* data, sedangkan model yang lainnya menggunakan 16 *frame*. Sama seperti model lainnya, akan dilakukan pengujian menggunakan analisa *cross-validation*. Dari grafik Gambar 4.26 dapat dilihat bahwa nilai *loss* pada proses *training* kelima terus mengalami penurunan.



Gambar 4.26: Grafik nilai *loss training* model kelima.

Dari grafik Gambar 4.27 menunjukkan bahwa nilai akurasi pada proses *training* kelima terus mengalami kenaikan. Dapat disimpulkan bahwa pada proses *training* tidak mengalami *underfitting* maupun *overfitting*.



Gambar 4.27: Grafik nilai *accuracy training* model kelima.

Gambar 4.28 menunjukkan *confusion matrix* yang menunjukkan semua jumlah prediksi benar dan salah dari setiap video yang digunakan sebagai pengujian pada setiap kelas yang menggunakan hasil proses *training* dari model kedua. Dari *confusion matrix* tersebut bisa kita lihat bahwa model ini berhasil memprediksi beberapa video pengujian dengan benar, yaitu pada kelas *arrest* sebanyak 1 video, kelas *arson* sebanyak 2 video, kelas *burglary* sebanyak 1 video, kelas *explosion* sebanyak 1 video, kelas *normal* sebanyak 4 video, kelas *road accidents* sebanyak 3 video, kelas *Robbery* sebanyak 2 video, kelas *shoplifting* sebanyak 1 video, dan kelas *stealing* sebanyak 1 video.

	P R E D I C T E D													
	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	Normal	Road Accidents	Robbery	Shooting	Shoplifting	Stealing	Vandalism
Abuse	0	0	0	0	0	0	0.2	0.2	0	0.6	0	0	0	0
Arrest	0	0.2	0	0	0	0	0	0.4	0.4	0	0	0	0	0
Arson	0	0	0.4	0	0.4	0	0	0	0.2	0	0	0	0	0
Assault	0	0	0	0	0	0	0	0	0.6	0	0	0.4	0	0
Burglary	0	0	0.2	0	0.2	0	0	0	0.4	0	0	0	0.2	0
Explosion	0	0.2	0.2	0	0	0.2	0	0.2	0.2	0	0	0	0	0
Fighting	0.2	0	0	0	0	0	0	0.4	0	0.2	0	0	0.2	0
Normal	0	0	0	0	0	0.2	0	0.8	0	0	0	0	0	0
Road accidents	0	0	0	0	0	0	0	0.2	0.6	0	0	0	0	0.2
Robbery	0	0	0	0	0	0	0.2	0.2	0	0.4	0.2	0	0	0
Shooting	0.2	0.2	0	0	0	0	0	0.2	0	0	0	0	0.4	0
Shoplifting	0	0	0	0	0	0	0	0.6	0	0.2	0	0.2	0	0
Stealing	0	0	0	0	0	0	0	0	0.6	0.2	0	0	0.2	0
Vandalism	0	0	0	0	0.2	0.2	0	0.2	0	0.2	0	0	0.2	0

Gambar 4.28: *Confusion Matrix* nilai akurasi model kelima

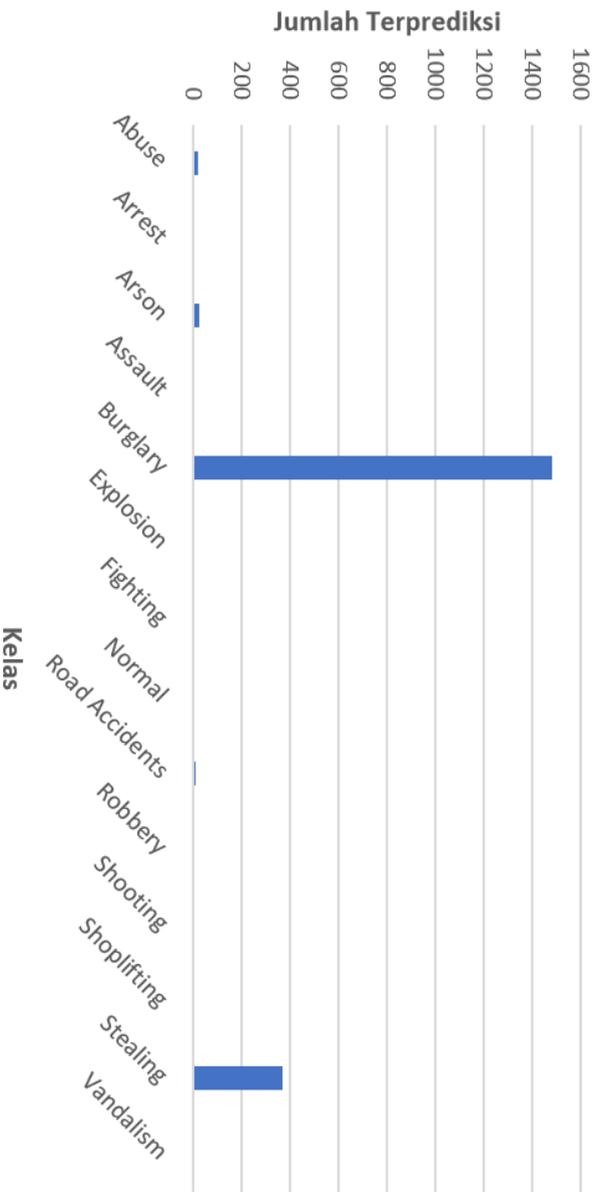
Tabel 4.5 menunjukkan jumlah prediksi yang benar pada pengujian yang menggunakan masing - masing 5 video setiap kelas sebagai input. Dari tabel tersebut bisa kita lihat bahwa model ini berhasil mendapatkan beberapa prediksi sesuai yaitu pada kelas *arrest*, *arson*, *burglary*, *explosion*, *normal*, *road accidents*, *robbery*, *shoplifting*, dan *stealing*.

Tabel 4.5: Pengujian Pada Hasil *Training* Model Kelima

Kelas	Prediksi Sesuai	Persentase
<i>Abuse</i>	0/5	0%
<i>Arrest</i>	1/5	20%
<i>Arson</i>	2/5	40%
<i>Assault</i>	0/5	0%
<i>Burglary</i>	1/5	40%
<i>Explosion</i>	1/5	40%
<i>Fighting</i>	0/5	0%
<i>Normal</i>	4/5	80%
<i>Road Accidents</i>	3/5	60%
<i>Robbery</i>	2/5	40%
<i>Shooting</i>	0/5	0%
<i>Shoplifting</i>	1/5	40%
<i>Stealing</i>	1/5	40%
<i>Vandalism</i>	0/5	0%
Total	16/70	22,8%

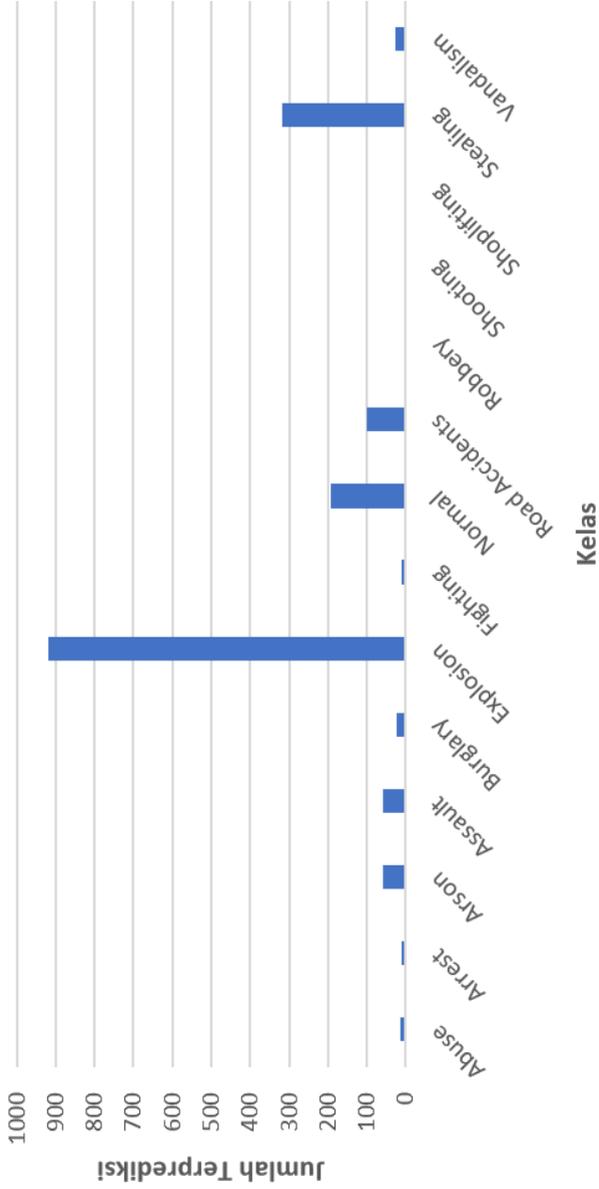
Berikut disertakan beberapa grafik hasil pengujian pada salah satu video dari beberapa kelas pada model ini yang ditunjukkan pada gambar 4.29, gambar 4.30, dan gambar 4.31.

Grafik Pengujian Kelas Burglary Pada Model Kelima

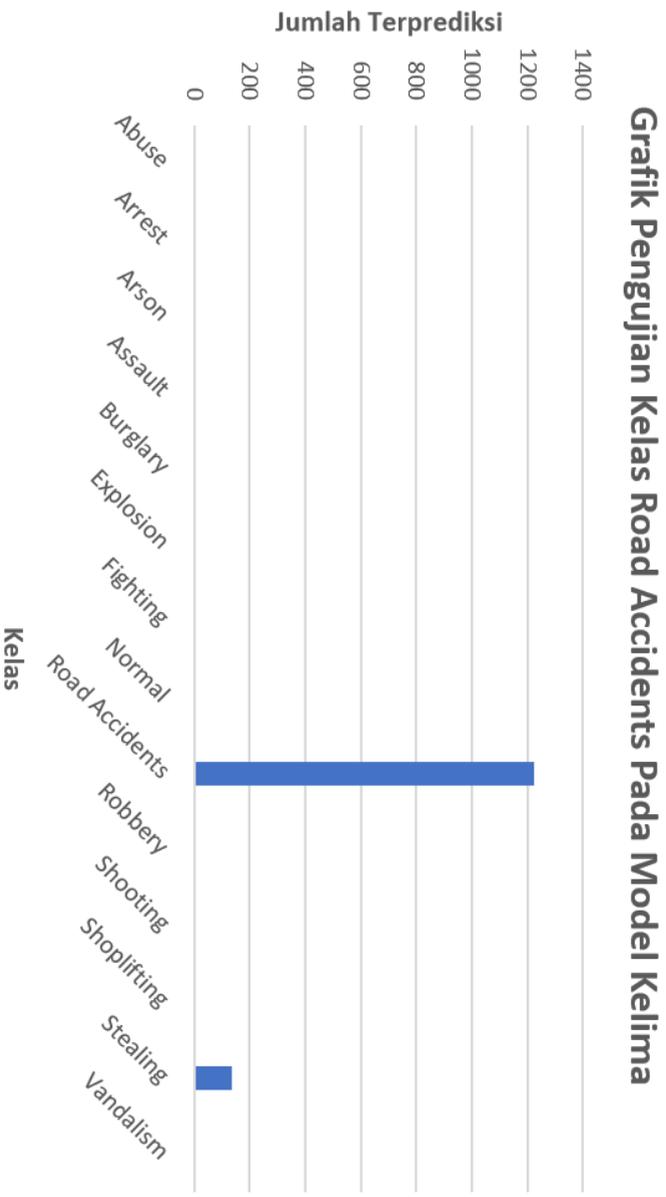


Gambar 4.29: Grafik hasil dari pengujian pada salah satu video pada *training* model kelima.

Grafik Pengujian Kelas Explosion Pada Model Kelima



Gambar 4.30: Grafik hasil dari pengujian pada salah satu video pada *training* model kelima.

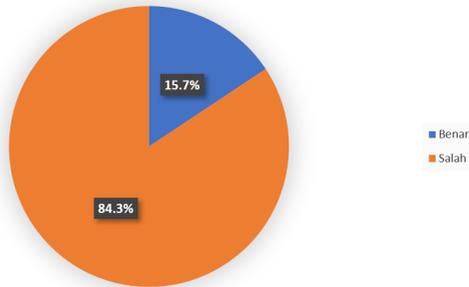


Gambar 4.31: Grafik hasil dari pengujian pada salah satu video pada *training* model kelima.

4.3 Analisa

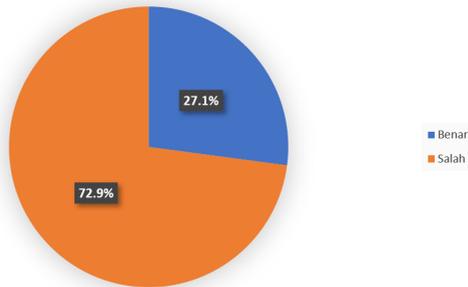
Setelah dilakukan uji coba performa dengan melakukan *voting* terhadap hasil pengujian secara manual dari setiap model, didapatkan nilai rata - rata performa dari tiap model. Prediksi benar pada model pertama, sesuai dengan grafik Gambar 4.32 adalah sekitar 15,7%, sedangkan sesuai dengan grafik Gambar 4.33, prediksi benar pada model kedua sekitar 27,1%, prediksi benar pada model ketiga sesuai dengan grafik Gambar 4.34 sekitar 17,1%, pada model keempat sekitar 17,1% sesuai dengan grafik Gambar 4.35, dan pada model kelima sekitar 22,8% sesuai dengan grafik Gambar 4.36. Dari hasil tersebut bisa disimpulkan bahwa model kedua memiliki performa terbaik, walaupun memiliki nilai *validation accuracy* terendah sesuai dengan grafik Gambar 4.9 dibandingkan dengan model lainnya.

Grafik Rata - Rata Performa Model Pertama



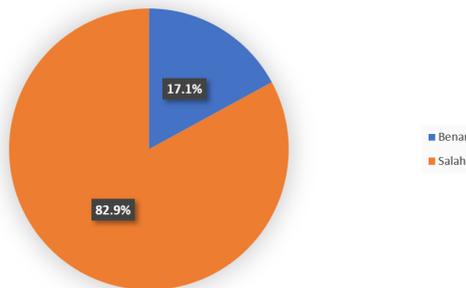
Gambar 4.32: Grafik rata - rata performa model pertama.

Grafik Rata - Rata Performa Model Kedua



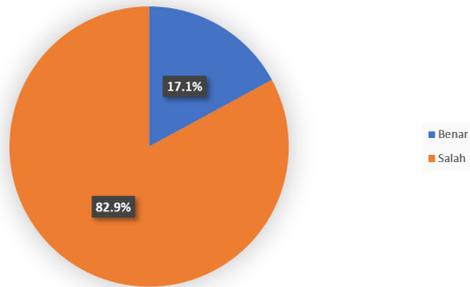
Gambar 4.33: Grafik rata - rata performa model kedua.

Grafik Rata - Rata Performa Model Ketiga



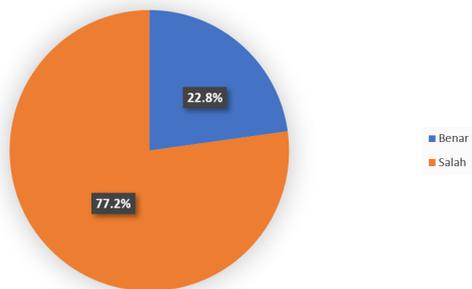
Gambar 4.34: Grafik rata - rata performa model ketiga.

Grafik Rata - Rata Performa Model Keempat



Gambar 4.35: Grafik rata - rata performa model keempat.

Grafik Rata - Rata Performa Model Kelima



Gambar 4.36: Grafik rata - rata performa model kelima.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Berdasarkan hasil *training*, sesuai dengan gambar 4.3 model pertama mendapatkan nilai akhir *validation accuracy* terbesar, yaitu sekitar 38,1%. Tapi, berdasarkan hasil perbandingan performa akurasi prediksi yang dilakukan secara manual, menunjukkan bahwa model kedua mendapatkan nilai akurasi sebesar 27,1% sesuai dengan tabel 4.2.
2. Berdasarkan tabel 4.1, tabel 4.2, 4.3, tabel 4.4, dan tabel 4.5 masih banyak kelas kelas yang belum bisa diklasifikasi. Hal ini mungkin disebabkan sudut pandang kamera terhadap aksi kejahatan yang kurang bagus. Untuk mendapatkan hasil yang maksimal, tindakan kejahatan yang terjadi di dalam video yang terekam tidak boleh terhalang objek lain atau harus terlihat secara jelas.
3. Berdasarkan hasil pengujian yang ditunjukkan pada tabel 4.1, tabel 4.2, tabel 4.3, tabel 4.4, dan tabel 4.5 masih banyak kelas - kelas yang sama sekali belum bisa terprediksi secara benar. Mungkin ini disebabkan karena masih rendahnya akurasi yang didapatkan dari semua model yang seharusnya masih bisa ditingkatkan lagi.
4. Berdasarkan hasil pengujian pada tabel 4.1, tabel 4.2, tabel 4.3, tabel 4.4, dan tabel 4.5 kelas normal merupakan kelas yang paling mudah terprediksi dengan benar. Sepertinya, kelas tersebut memiliki tingkat prediksi yang tinggi. Setiap kelas berhasil memprediksi video pengujian kelas normal dengan akurasi sebesar 80%.
5. Berdasarkan hasil pengujian yang ditunjukkan pada tabel 4.1, tabel 4.2, tabel 4.3, tabel 4.4, dan tabel 4.5 masih banyak kelas - kelas yang sama sekali belum bisa terprediksi secara benar. Menurut pengamatan saya, latar tempat pada video yang digunakan sebagai dataset bisa mempengaruhi hasil klasifikasi.

Sebagai contoh, bila pada dataset kelas *abuse* terjadi pada tempat yang terdapat banyak kursinya, maka saat pengujian pada video kelas lain yang pada tempat kejadian terdapat banyak kursi, video tersebut bisa diklasifikasikan sebagai kelas *abuse* yang menyebabkan kesalahan pada hasil klasifikasi.

5.2 Saran

Untuk pengembangan lebih lanjut pada penelitian tugas akhir ini, terdapat beberapa beberapa saran yang dapat dilakukan, antara lain:

1. Mencoba meningkatkan *epoch* yang digunakan dalam proses *training*.
2. Melakukan *trimming* pada dataset yang digunakan untuk meningkatkan performansi klasifikasi tindak kejahatan.
3. Melakukan penelitian dengan mengubah model yang sudah ada untuk menemukan model yang memiliki performansi lebih bagus.
4. Melanjutkan penelitian ini ke tahap mengklasifikasikan tindak kejahatan secara *real-time*

DAFTAR PUSTAKA

- [1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in Proceedings of the IEEE international conference on computer vision, pp. 4489–4497, 2015. (Dikutip pada halaman 5, 15, 16, 27).
- [2] S. Neili Boualia and N. Essoukri, “3d cnn for human action recognition,” 05 2018. (Dikutip pada halaman 5).
- [3] M. Ariesta, F. Wiryana, S. Suharjito, and A. Zahra, “Sentence level indonesian sign language recognition using 3d convolutional neural network and bidirectional recurrent neural network,” 09 2018. (Dikutip pada halaman 6).
- [4] R. Kusumanto and A. N. Tompunu, “pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi rgb,” Semantik, vol. 1, no. 1, 2011. (Dikutip pada halaman 1).
- [5] S. S. P. dan Keamanan, Statistik Kriminal 2018. Badan Pusat Statistika, 2018. (Dikutip pada halaman 1).
- [6] I. Putra, Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. PhD thesis, Institut Teknologi Sepuluh Nopember, 2016. (Dikutip pada halaman 1).
- [7] C. Dupont, L. Tobias, and B. Luvison, “Crowd-11: A dataset for fine grained crowd behaviour analysis,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 9–16, 2017. (Dikutip pada halaman 1).
- [8] D. Sharma and N. Kumar, “A review on machine learning algorithms, tasks and applications,” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 10, 2017. (Dikutip pada halaman 6).

- [9] A. Ahmad, “Mengenal artificial intelligence, machine learning, neural network, dan deep learning,” no. October, 2017. (Dikutip pada halaman 7).
- [10] J. DeMuro, “What is a neural network?.” <https://www.techradar.com/news/what-is-a-neural-network>, 2018. Diakses: 2019-04-06. (Dikutip pada halaman 7).
- [11] Y. Bengio et al., “Learning deep architectures for ai,” Foundations and trends® in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009. (Dikutip pada halaman 8).
- [12] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016. (Dikutip pada halaman 8).
- [13] S. R. Dewi et al., “Deep learning object detection pada video menggunakan tensorflow dan convolutional neural network,” 2018. (Dikutip pada halaman 8).
- [14] “Convolutional neural network.” <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>. Accessed: 2019-11-13. (Dikutip pada halaman 8).
- [15] T. Nurhikmat et al., “Implementasi deep learning untuk image classification menggunakan algoritma convolutional neural network (cnn) pada citra wayang golek,” 2018. (Dikutip pada halaman 12).
- [16] S. Sena, “Pengenalan deep learningpart 7: Convolutional neural network(cnn),” 2017. (Dikutip pada halaman 10, 12, 13, 14).
- [17] M. Lin, Q. Chen, and S. Yan, “Network in network,” arXiv preprint arXiv:1312.4400, 2013. (Dikutip pada halaman 13).
- [18] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” arXiv preprint arXiv:1811.03378, 2018. (Dikutip pada halaman 14).

- [19] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in 2017 International Conference on Engineering and Technology (ICET), pp. 1–6, IEEE, 2017. (Dikutip pada halaman 15).
- [20] K. Umam and B. S. Negara, “Deteksi obyek manusia pada basis data video menggunakan metode background subtraction dan operasi morfologi,” Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi, vol. 2, no. 2, pp. 31–40, 2016. (Dikutip pada halaman 17).
- [21] A. V. Srinivasan, “Stochastic gradient descent clearly explained.” <https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31>. Accessed: 2019-12-29. (Dikutip pada halaman 18).

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS



Muhammad Aufa Rahmatullah, lahir pada 11 Februari di Jakarta. Penulis merupakan anak kedua dari dua bersaudara. Saat ini penulis tinggal di Kota Depok, Provinsi Jawa Barat. Pada tahun 2009 menyelesaikan pendidikan di SD Pemuda Bangsa. Tahun 2012 lulus dari SMP Negeri 4 Depok. Dan pada tahun 2015 lulus dari SMA Negeri 3 Depok. Penulis sedang menjalani studi di Departemen Teknik Komputer, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Selama menjadi mahasiswa di ITS, penulis pernah menjadi staff departemen Lingkar Kampus dalam Himpunan Mahasiswa Teknik Elektro (HIMATEKTRO) ITS pada tahun 2016 - 2017 dan menjadi staff ahli dalam departemen dan himpunan yang sama pada tahun 2017 - 2018. Penulis memiliki hobi bermain *game* dan membaca buku. Penulis berhasil menyelesaikan tugas akhir dengan judul "**Klasifikasi Tindak Kejahatan Dalam Video Dengan Menggunakan 3D Convolutional Neural Network (3D CNN)**". Bagi pembaca yang memiliki kritik, saran, atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui surel aufa0298@gmail.com.

Halaman ini sengaja dikosongkan