



**SISTEM PENANGGULANGAN KECELAKAAN  
NASIONAL  
(SIP KENA)**

**DIUSULKAN OLEH:**

**Ibnu Kasyful Haq ; 18102089**

**Febri Arista Rahadian P. ; 18102085**

**Fadhilah Gusti S. ; 18102084**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
BANYUMAS**

**2020**

## **KATA PENGANTAR**

Assalamu'alaikum wr. wb.

Kecelakaan merupakan salah satu permasalahan di Indonesia yang patut untuk kita waspadi dan akselerasi tingkat keefektifitasan penanganannya. Kami, tim Arista, dengan sedikit ilmu yang kami miliki, ingin mencoba melakukan sumbangsih untuk negara kami tercinta, dengan membuat kernel CNN untuk melakukan deteksi otomatis kejadian-kejadian berbahaya yang ada di jalan raya, yaitu, kecelakaan tanpa adanya kebakaran, kecelakaan yang mengakibatkan kebakaran, dan terakhir ambulans.

Kami harap sistem kami dapat diimplementasikan secara masif dengan funding yang cukup suatu hari kelak, sehingga dapat bermanfaat untuk banyak orang di Indonesia dan menyelamatkan banyak nyawa.

Kami ucapkan terima kasih kepada Allah S.W.T yang telah menganugerahi kami kesempatan untuk hidup dan berbuat kebaikan untuk sesama. Tak lupa kepada orang tua kami, semoga Tuhan selalu merahmati mereka di mana mereka berada. Dan juga segenap pihak yang tidak bisa kami jabarkan satu per satu di sini.

Sukses untuk panitia Gemastik. Semoga kebaikan selalu dilimpahkan kepada semua yang berbuat kebaikan.

Wassalamu'alaikum wr. wb.

## **ABSTRAK**

Penanganan korban kecelakaan lalu lintas merupakan salah satu hal komunal di Indonesia yang mengakibatkan keselamatan korban sangat bergantung atas inisiatif masyarakat sekitar tempat kejadian perkara untuk melakukan pertolongan pertama. Pada penelitian ini, kami menawarkan sebuah sistem bernama Sip Kena (Sistem Penanggulangan Kecelakaan Nasional) untuk meningkatkan keefektifitasan penanganan korban kecelakaan lalu lintas secara masif dan terotomatisasi memanfaatkan teknologi deep learning. Sistem yang kami ajukan dapat mendeteksi 4 jenis kondisi yang umum ditemui di jalan raya secara otomatis: jalan raya aman, kecelakaan tanpa kebakaran, kecelakaan dengan kebakaran, dan keberadaan mobil ambulans. Kami memanfaatkan Mask Regional Convolutional Neural Network (Mask RCNN) untuk mendeteksi keberadaan mobil di jalan raya, untuk kemudian kami kenali kondisi mobil tersebut menggunakan kernel Convolutional Neural Network (CNN) yang telah kami buat. Dikarenakan keterbatasan sumber daya dalam melakukan training kernel CNN, maka kami berinovasi untuk memprediksi arsitektur kernel CNN terbaik yang dirasa tepat untuk mendeteksi kasus-kasus yang kami miliki dengan memanfaatkan Ridge Regression. Sebagai hasilnya, kami dapati program yang kami buat dapat mendeteksi kecelakaan tanpa kebakaran, kecelakaan dengan kebakaran, dan keberadaan mobil ambulans dengan akurasi masing-masing mencapai 60%, 92.25%, 70.37%.

## DAFTAR ISI

<b>Kata Pengantar</b>	1
<b>Abstrak</b>	2
<b>Daftar Isi</b>	3
<b>Daftar Gambar</b>	4
<b>Daftar Tabel</b>	5
<b>BAB I : Pendahuluan</b>	
1.1. Latar Belakang	6
1.2. Tujuan dan Manfaat	7
1.3. Batasan yang Digunakan	7
<b>BAB II : Metode</b>	
2.1. Preprocessing	8
2.2. Desain dan Implementasi Penambangan Data	13
<b>BAB III : Hasil</b>	
3.1. Hasil Simulasi	17
3.2. Analisis dan Inovasi	21
<b>BAB IV : Kesimpulan</b>	24
<b>Dokumentasi</b>	25
<b>Daftar Pustaka</b>	28

## DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Mask RCNN	8
Gambar 2.2 Folder Dataset	9
Gambar 2.3 Augmentasi Mobil Biasa	11
Gambar 2.4 Augmentasi Mobil Kecelakaan	11
Gambar 2.5 Augmentasi Mobil Terbakar	12
Gambar 2.6 Augmentasi Mobil Ambulans	12
Gambar 2.7 Model Skema A	13
Gambar 2.8 Ilustrasi Hidden Layer Convolutional Network Skema A	14
Gambar 2.9 Model Skema B	14
Gambar 2.10 Ilustrasi Hidden Layer Convolutional Network Skema B	15
Gambar 2.11 Model Skema C	15
Gambar 2.12 Ilustrasi Hidden Layer Convolutional Network Skema C	16
Gambar 3.1 Akurasi Skema A	17
Gambar 3.2 Akurasi Skema B	18
Gambar 3.3 Akurasi Skema C	20
Gambar 3.4 Flowchart Sistem Penanggulangan Kecelakaan Nasional	23
Gambar Ambulans	25
Gambar Jalan Raya	25
Gambar Kebakaran	26
Gambar Kecelakaan	26
Gambar Pada Saat Pengerjaan Sistem	27

## DAFTAR TABEL

Gambar 3.1 Akurasi skema A epoch ke-66 akurasi training 90%	17
Gambar 3.2 Akurasi skema A epoch ke-148 akurasi training 100%	18
Gambar 3.3 Akurasi skema B epoch ke-81 akurasi training 90%	19
Gambar 3.4 Akurasi skema B epoch ke-196 akurasi training 100%	19
Gambar 3.5 Akurasi skema C epoch ke-67 akurasi training 100%	20
Gambar 3.6 Akurasi skema C epoch ke-186 akurasi training 100%	21
Gambar 3.7 Prediksi Akurasi Arsitektur CNN menggunakan Ridge Regression	22

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Kecelakaan di Indonesia untuk tahun 2020 ini masuk di angka 1.049.000 juta kejadian dengan rincian korban meninggal dunia 57.000 jiwa, korban dengan luka ringan 1.273.000 jiwa dan korban dengan luka berat 9000 jiwa [1]. Setiap jam rata-rata 3 orang meninggal akibat kecelakaan jalan di Indonesia dengan faktor penyebab 61% karena faktor manusia, 9% karena faktor kendaraan, dan 30% karena faktor prasarana & lingkungan [2].

Peristiwa kecelakaan terjadi, biasanya terdapat 3 hal yang terlibat yaitu Korban dan pelaku, polisi, dan para petugas medis. Apabila kecelakaan dengan suasana jalan raya ramai, masyarakat mampu memanggil bantuan atau bahkan mereka membantu dengan memberikan pertolongan pertama sebelum petugas medis datang sedangkan kalau suasana jalan raya sepi tidak ada masyarakat maka bantuan akan terlambat datang dapat mengakibatkan korban meninggal di tempat ataupun ditinggal lari sama pelaku yang menabrak nya. Sedangkan CCTV di Indonesia sudah banyak terpasang di setiap perempatan lampu merah dari itu sangat disayangkan apabila tidak di tambahkan AI (Artificial Intelligence). Pemantauan peristiwa melalui CCTV di jalan raya meliputi mobil biasa yang melintas di jalan, mobil yang mengalami kecelakaan baik tunggal maupun dengan pengemudi yang lain, mobil dalam kondisi terbakar[3]. Sip Kena hadir untuk memberikan solusi untuk menjaga kesejahteraan rakyat NKRI secara otomatis dan menyeluruh dengan menggunakan metode *Deep Learning*.

*Deep Learning* adalah bagian dari pembelajaran mesin yang berbasis jaringan syaraf tiruan dengan banyak *hidden layers* yang memiliki kemampuan untuk mempelajari representasi atau fitur data secara otomatis[4] [5] [6]. Deep learning memiliki kemampuan yang sangat baik dalam computer vision, salah satunya adalah penerapan algoritma *Convolutional Neural Network (CNN)*[7] [8] [9] [10]. Algoritma CNN menjadi populer di dalam teknik *Deep Learning* karena memiliki faktor penting seperti menghilangkan kebutuhan untuk ekstraksi fitur secara manual, dapat dilatih kembali untuk tugas-tugas dalam pengenalan suatu objek baru yang memungkinkan untuk membangun di jaringan yang sudah ada sebelumnya, serta algoritma CNN juga

mempunyai banyak sekali kemungkinan arsitektur model di dalamnya, baik itu 1 layer, 2 layer, hingga n layer [11][12][13].

Mask R-CNN salah satu algoritma yang diturunkan dari CNN. Dia bisa mendeteksi objek-objek tertentu. Di sini kami menggunakan Mask RCNN dengan dataset dari COCO [14].

Pada penelitian kami kali ini, kami melakukan inovasi pencarian struktur arsitektur terbaik untuk mensupport kernel CNN kami agar dapat mendeteksi kejadian-kejadian dan objek-objek yang ingin kami awasi dengan akurasi setinggi-tingginya menggunakan *Algoritma Ridge Regression* [15] [16]. Hal ini kami lakukan karena keterbatasan sumber daya komputer dan waktu untuk melakukan training deep learning secara cepat. Sehingga dengan cara yang kami ajukan, dengan sumber daya sedikit, kami mencoba membuktikan bahwa komputer rumahan pun bisa menghasilkan kernel CNN yang mumpuni dan luar biasa tidak kalah dibanding menggunakan super komputer.

## **1.2. Tujuan dan manfaat**

Tujuan dari Sistem Penanggulangan Kecelakaan Nasional ini adalah :

1. Untuk mendeteksi kecelakaan mobil di jalan raya.
2. Untuk mendeteksi kebakaran yang terjadi akibat kecelakaan.
3. Untuk mempercepat tim medis untuk datang ke tempat terjadinya kecelakaan.

Manfaat dari Sistem Penanggulangan Kecelakaan Nasional ini adalah untuk meningkatkan keefektifitasan penanganan korban kecelakaan di jalan raya secara masif dan otomatis pada level nasional, sehingga dapat mengurangi korban jiwa akibat kecelakaan mobil di jalan raya.

## **1.3. Batasan yang digunakan**

1. Dataset merupakan video kecelakaan dan video mobil ambulans yang diambil dari Youtube
2. Kasus kecelakaan hanya yang terjadi pada mobil yang bertabrakan dan mobil yang terbakar akibat kecelakaan.
3. Model objek mobil ambulans yang dipergunakan untuk training data adalah mobil ambulans dari Indonesia.
4. Sistem yang kami buat dibangun menggunakan python.



## BAB II

### METODE

Metode Penambangan Data yang kami lakukan pada pembuatan Sistem Penanggulangan Kecelakaan Nasional (Sip Kena) adalah *Preprocessing*, *Processing*, dan *Postprocessing*.

#### 2.1. Preprocessing

Pada *Preprocessing* kami mengumpulkan data sebanyak 4 kategori yaitu gambar mobil yang tidak mengalami kecelakaan, mobil yang mengalami kecelakaan, mobil yang terbakar akibat kecelakaan, dan juga mobil ambulans.

##### 2.1.1 Pengumpulan Video

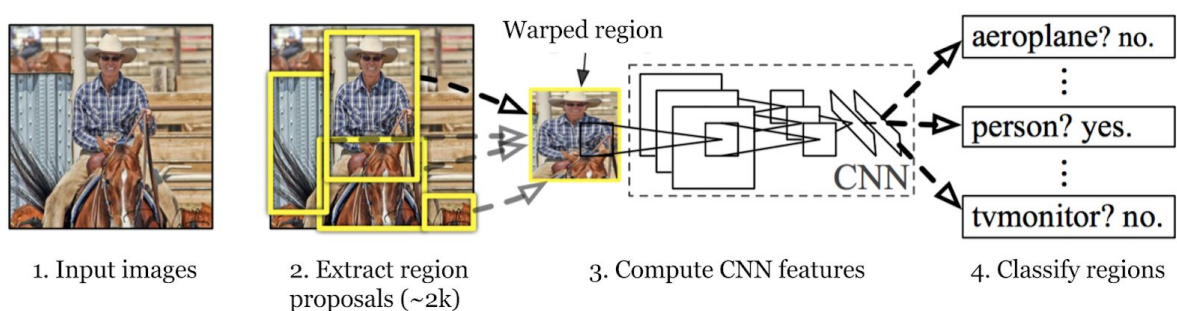
Data diambil dari video yang berada di Youtube yang selanjutnya video tersebut kami potong per-frame.

##### 2.1.2 Pemotongan Frame

Setelah dipotong per-frame kemudian frame tersebut dipotong per-objek menggunakan Mask RCNN.

##### 2.1.3 Mask RCNN

Berikut adalah ilustrasi dari Mask RCNN



**Gambar 2.1 Ilustrasi Mask RCNN (Sumber gambar: Girshick et al., 2014)**

Mask RCNN adalah sebuah update dari RCNN (versi sebelumnya) dengan memasukkan mask untuk memprediksi branch paralel dengan label kelas dan bounding box prediction branch.

Pada fase pertama mask RCNN gambar pertama akan dipotong menjadi 2000 bagian yang kecil-kecil. Potongan ini akan dipergunakan sebagai basis untuk mencari kesamaan-kesamaan area-area yang membentuk suatu objek tertentu. Objek-objek ini kemudian kita sebut sebagai segmentasi citra. Setiap segmen citra akan dikirimkan ke sebuah kernel CNN untuk di deteksi jenis objeknya. Pada penelitian kali ini kami menggunakan kernel Mask RCNN berbasiskan dataset dari MS COCO yang berisikan 90 kategori [17][18].

#### **2.1.4 Teori CNN**

Convolutional Neural Network (CNN) merupakan salah satu jenis neural network yang biasa digunakan dalam memproses semua data yang berbentuk image. CNN terdiri dari beberapa lapisan yang terinspirasi dari mata kucing dimana mata kucing terdiri dari beberapa layer untuk menghasilkan sebuah persepsi visual.

Lapisan yang ada di CNN adalah sebagai berikut :

- **Convolutional Layer**

Layer konvolusi adalah layer yang dibuat mengikuti struktur mata pada kucing, di mana terdapat banyak ragam matriks filter yang dipergunakan untuk memfilter fitur-fitur tersembunyi yang mungkin ada pada citra.

- **Pooling**

Pooling layer adalah lapisan yang meresize dimensi gambar agar lebih kecil sehingga mempercepat komputasi dan mengatasi overfitting. Pooling terdiri dari Max Pooling, Min Pooling, dan Average Pooling.

- **Flatten**

Flatten adalah sebuah proses membentuk ulang matriks gambar menjadi sebuah vektor yang panjang agar bisa kita gunakan sebagai input pada fase penentuan klasifikasi.

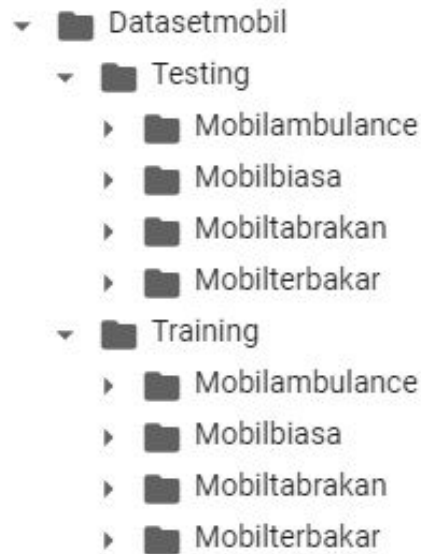
- **Dense**

Pada fase ini, fitur-fitur yang sudah memanjang setelah keluar dari fase flatten, maka dilakukanlah pengerucutan fitur hingga menjadi beberapa kelas saja sebagai output penentuan klasifikasi gambar.

### 2.1.5 Pemotongan Objek

Dari 14 video yang dikumpulkan, kami hanya memotong objek yang berupa mobil. Total objek mobil yang kami dapat adalah sebanyak 175. Objek-objek ini kami simpan berdasarkan kategorinya masing-masing.

Berikut susunan folder dataset kami.

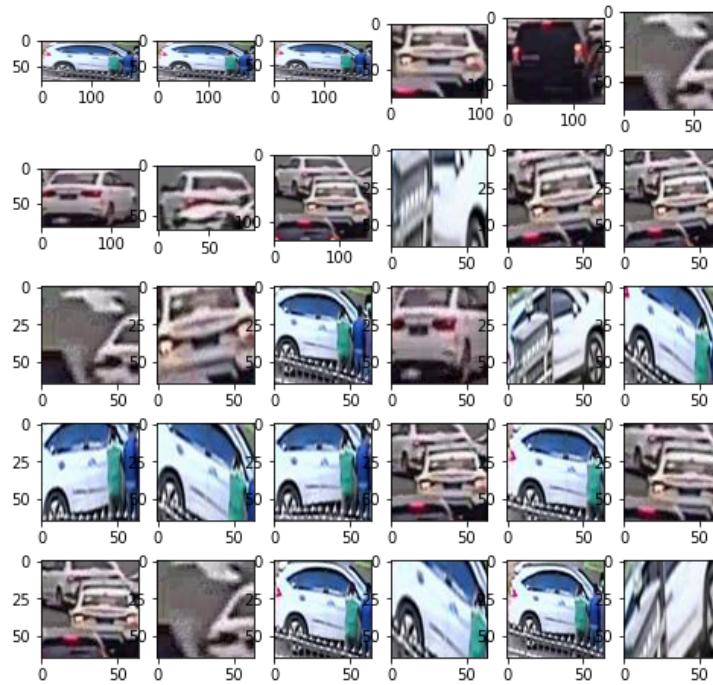


**Gambar 2.2 Folder Dataset**

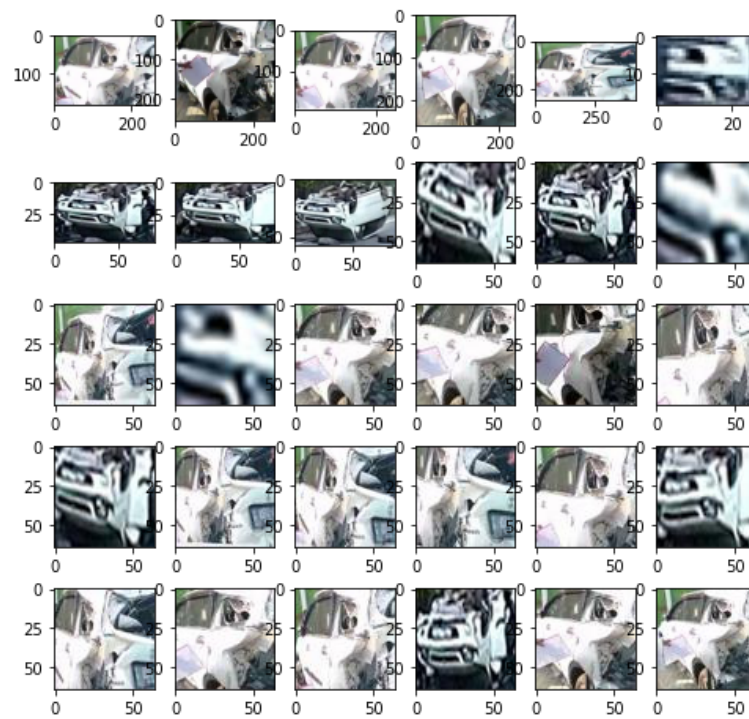
Bisa dilihat pada gambar di atas kami mengelompokkan objek mobil menjadi 4 kategori yaitu mobil biasa, mobil tabrakan, mobil terbakar, dan mobil ambulans. Pada kategori mobil biasa, mobil tabrakan, dan mobil terbakar masing-masing terdiri dari 40 gambar. 10 gambar pada folder testing dan 30 gambar pada folder training. Sedangkan pada kategori mobil ambulans terdiri dari 55 gambar, dengan 15 gambar pada folder testing dan 40 gambar pada folder training.

### 2.1.6 Augmentasi dan Rasio Training Test

Dataset yang telah kita kumpulkan hanya terdapat 40 gambar mobil biasa, 40 gambar mobil kecelakaan, 40 gambar mobil terbakar, dan 55 gambar mobil ambulans. Jumlah data tersebut masih kurang mencukupi untuk mendapatkan performa yang optimal. Untuk itu kami perlu meng-augmentasi data tersebut agar bisa di test dengan rate train test 70:30, dengan total data setiap kategori masing-masing 700 banding 300 gambar. Augmentasi pada data berupa gambar dapat berupa rotate, flip, crop, dll [19][20]. Berikut merupakan contoh dari augmentasi pada keempat kategori yang akan di test.



**Gambar 2.3 Augmentasi Mobil Biasa**



**Gambar 2.4 Augmentasi Mobil Kecelakaan**



**Gambar 2.5 Augmentasi Mobil Terbakar**



**Gambar 2.6 Augmentasi Mobil Ambulans**

Data yang telah di augmentasi kemudian masuk ke fase *Processing* menggunakan CNN dengan 4 *convolutional layer*. Kemudian kami melakukan testing akurasi per kategori.

Dari data yang telah diproses dan di testing akurasi, kami memprediksi konfigurasi layer terbaik agar mendapatkan probabilitas deteksi paling maksimal menggunakan Ridge Regression [21].

## 2.2. Desain dan Implementasi Penambahan Data

### 2.2.1 Skema A

Berikut adalah arsitektur CNN untuk skema A. Dari tabel dibawah, dapat diketahui bahwa skema ini menerima input gambar berukuran 64x64 pixel. Skema ini memiliki 4 layer konvolusi, dengan jumlah filter masing-masing layer berukuran 16, 32, 64 dan 128. Adapun konfigurasi max pooling di setiap akhir layer konvolusi adalah (2,2). Kami menggunakan aktivasi ReLU pada hidden layer, diakhiri dengan aktivasi Softmax pada layer pengambil keputusan terakhir. Adapun untuk optimalisasi, kami menggunakan SGD (stochastic gradient descent).

Model: "Sequential\_1" (Skema A)|

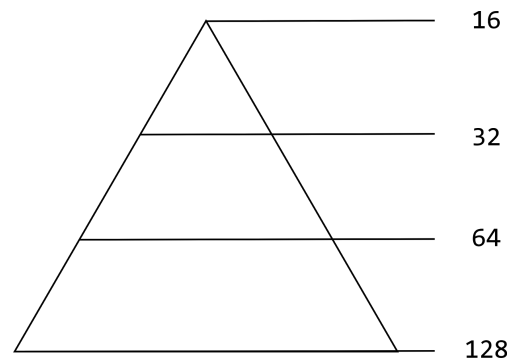
Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 62, 62, 16)	448
max_pooling2d_4 (MaxPooling2)	(None, 31, 31, 16)	0
conv2d_5 (Conv2D)	(None, 29, 29, 32)	4640
max_pooling2d_5 (MaxPooling2)	(None, 14, 14, 32)	0
conv2d_6 (Conv2D)	(None, 12, 12, 64)	18496
max_pooling2d_6 (MaxPooling2)	(None, 6, 6, 64)	0
conv2d_7 (Conv2D)	(None, 4, 4, 128)	73856
max_pooling2d_7 (MaxPooling2)	(None, 2, 2, 128)	0
flatten_1 (Flatten)	(None, 512)	0
dense_2 (Dense)	(None, 100)	51300
dense_3 (Dense)	(None, 4)	404

Total params:149,144

Trainable params:149,144

Non-trainable params: 0

**Gambar 2.7 Model Skema A**



**Gambar 2.8 Ilustrasi Hidden Layer Convolutional Network Skema A**

### 2.2.2 Skema B

Berikut adalah arsitektur CNN untuk skema B. Dari tabel dibawah, dapat diketahui bahwa skema ini menerima input gambar berukuran 64x64 pixel. Skema ini memiliki 4 layer konvolusi, dengan jumlah filter masing-masing layer berukuran 128, 64, 32, dan 16. Adapun konfigurasi max pooling di setiap akhir layer konvolusi adalah (2,2). Kami menggunakan aktivasi ReLU pada hidden layer, diakhiri dengan aktivasi Softmax pada layer pengambil keputusan terakhir. Adapun untuk optimalisasi, kami menggunakan SGD (stochastic gradient descent).

Model: "Sequential" (Skema B)|

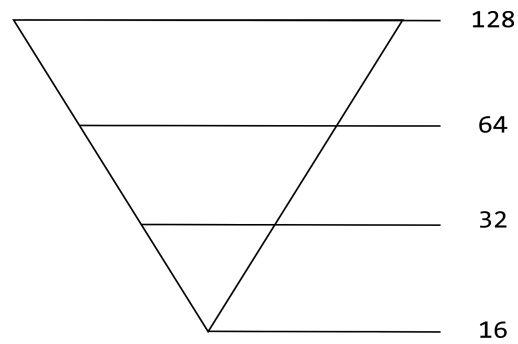
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 128)	3584
max_pooling2d(MaxPooling2D)	(None, 31, 31, 128)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	73792
max_pooling2d_1(MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 32)	18464
max_pooling2d_2(MaxPooling2D)	(None, 6, 6, 32)	0
conv2d_3 (Conv2D)	(None, 4, 4, 16)	4624
max_pooling2d_3(MaxPooling2D)	(None, 2, 2, 16)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 100)	6500
dense_1 (Dense)	(None, 4)	404

Total params: 107,368

Trainable params: 107,368

Non-trainable params: 0

**Gambar 2.9 Model Skema B**



**Gambar 2.10 Ilustrasi Hidden Layer Convolutional Network Skema B**

### 2.2.3 Skema C

Berikut adalah arsitektur CNN untuk skema C. Dari tabel dibawah, dapat diketahui bahwa skema ini menerima input gambar berukuran 64x64 pixel. Skema ini memiliki 4 layer konvolusi, dengan jumlah filter masing-masing layer berukuran 64, 64, 64, dan 64. Adapun konfigurasi max pooling di setiap akhir layer konvolusi adalah (2,2). Kami menggunakan aktivasi ReLU pada hidden layer, diakhiri dengan aktivasi Softmax pada layer pengambil keputusan terakhir. Adapun untuk optimalisasi, kami menggunakan SGD (stochastic gradient descent).

Model: "Sequential" (Skema C)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 31, 31, 64)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_3 (Conv2D)	(None, 4, 4, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 100)	25700
dense_1 (Dense)	(None, 4)	404

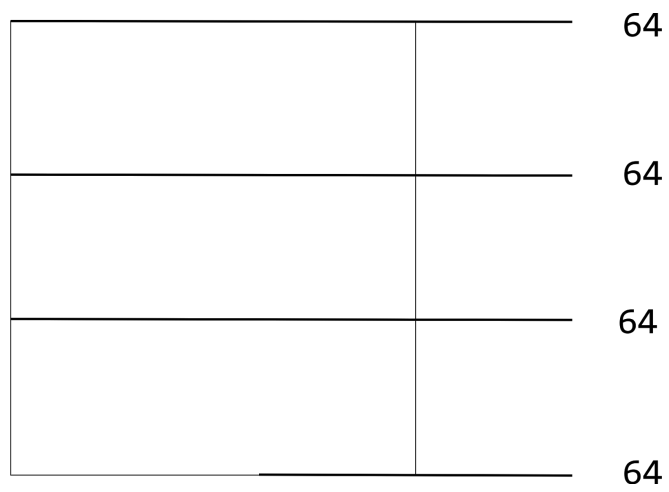
Total params: 138,680

Trainable params: 138,680

Non-trainable params: 0

**Gambar 2.11 Model Skema C**





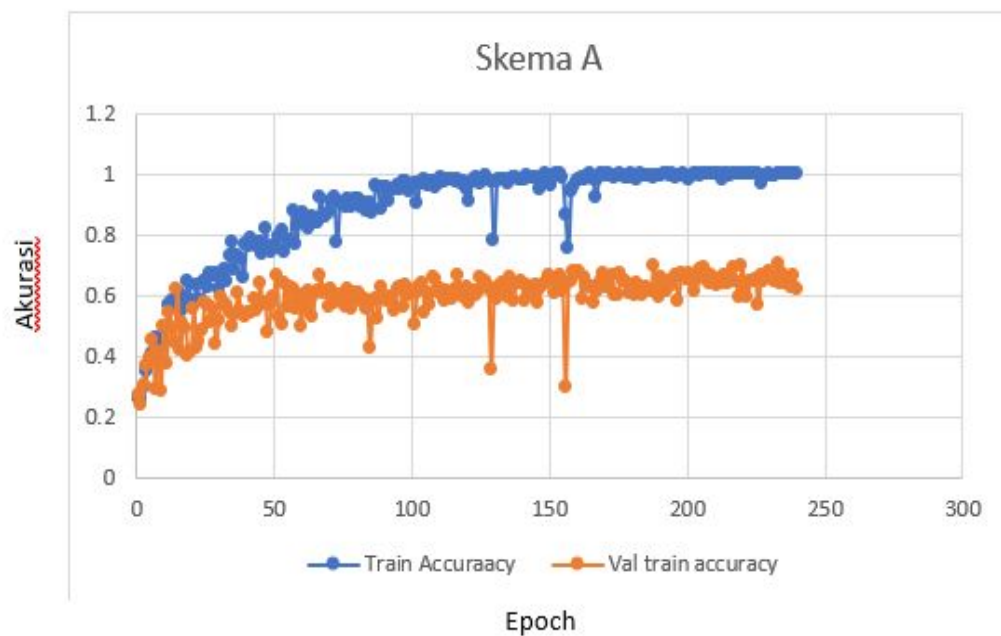
**Gambar 2.12 Ilustrasi Hidden Layer Convolutional Network Skema C**

## BAB III

### HASIL

#### 3.1. Hasil Simulasi

##### 3.1.1 Skema A



Gambar 3.1 Akurasi Skema A

Pada skema ini akurasi training kisaran 90% didapatkan ketika berada pada epoch ke-66. Pada epoch ini kami melakukan testing menggunakan dataset testing kami didapatkan akurasi untuk setiap kelas sebagai berikut:

**Tabel 3.1. Akurasi skema A epoch ke-66 akurasi training 90%**

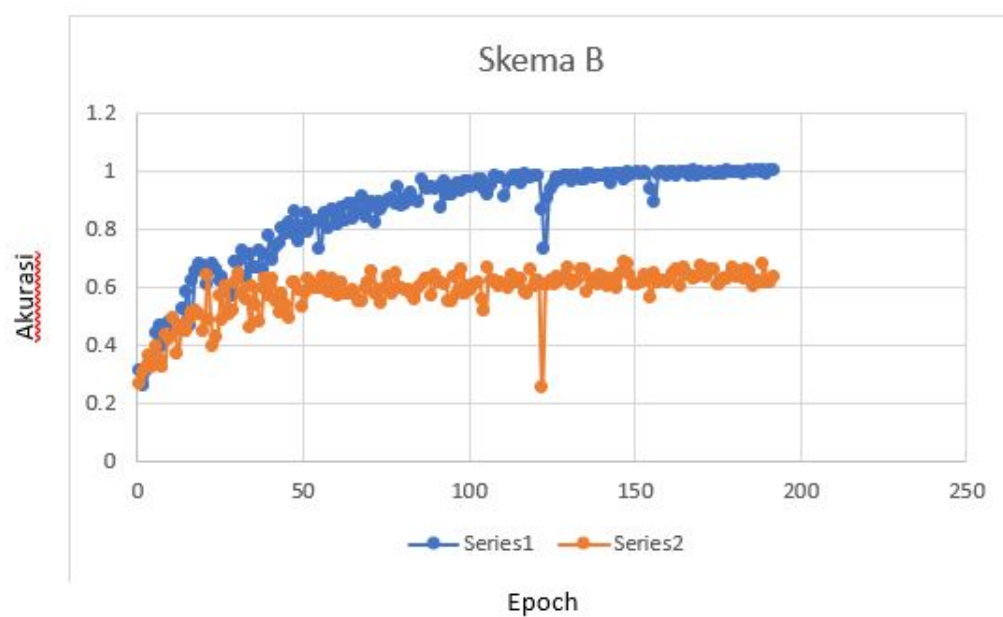
Kategori	Akurasi
Mobil ambulans	70.37%
Mobil Biasa	25.48%
Mobil Tabrakan	60%
Mobil Terbakar	92,25%

Pada skema ini akurasi training 100% didapatkan ketika berapa pada epoch ke-148. Pada epoch ini kami melakukan testing menggunakan dataset testing kami didapatkan akurasi untuk setiap kelas sebagai berikut:

**Tabel 3.2. Akurasi skema A epoch ke-148 akurasi training 100%**

Kategori	Akurasi
Mobil ambulans	75,30%
Mobil Biasa	42,585
Mobil Tabrakan	36,12%
Mobil Terbakar	96,12%

### 3.1.2 Skema B



**Gambar 3.2 Akurasi Skema B**

Pada skema ini akurasi training kisaran 90% didapatkan ketika berada pada epoch ke-81. Pada epoch ini kami melakukan testing menggunakan dataset testing kami didapatkan akurasi untuk setiap kelas sebagai berikut:

**Tabel 3.3. Akurasi skema B epoch ke-81 akurasi training 90%**

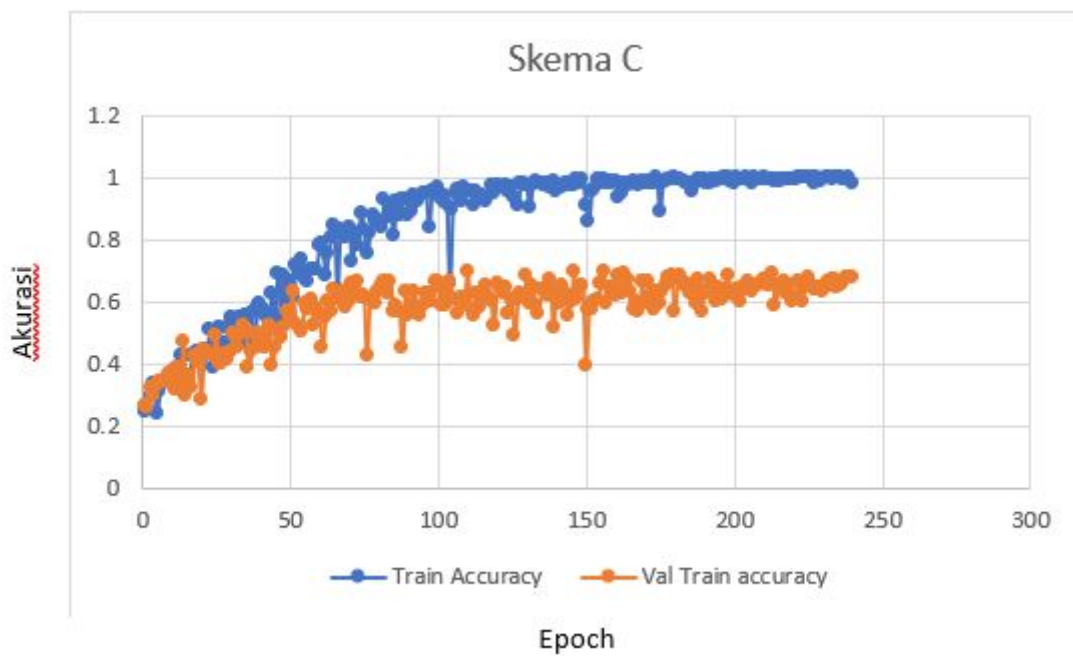
Kategori	Akurasi
Mobil ambulans	55,44%
Mobil Biasa	59,35%
Mobil Tabrakan	27,09%
Mobil Terbakar	38,06%

Pada skema ini akurasi training 100% didapatkan ketika berapa pada epoch ke-196. Pada epoch ini kami melakukan testing menggunakan dataset testing kami didapatkan akurasi untuk setiap kelas sebagai berikut:

**Tabel 3.4. Akurasi skema B epoch ke-196 akurasi training 100%**

Kategori	Akurasi
Mobil ambulans	68,87%
Mobil Biasa	24,19%
Mobil Tabrakan	78,70%
Mobil Terbakar	58,70%

### 3.1.3 Skema C



Gambar 3.3 Akurasi Skema C

Pada skema ini akurasi training kisaran 90% didapatkan ketika berada pada epoch ke 67. Pada epoch ini kami melakukan testing menggunakan dataset testing kami didapatkan akurasi untuk setiap kelas sebagai berikut:

**Tabel 3.5. Akurasi skema C epoch ke-67 akurasi training 90%**

Kategori	Akurasi
Mobil ambulans	84,25%
Mobil Biasa	17,74%
Mobil Tabrakan	32,90%
Mobil Terbakar	86,45%

Pada skema ini akurasi training 100% didapatkan ketika berapa pada epoch ke-186. Pada epoch ini kami melakukan testing menggunakan dataset testing kami didapatkan akurasi untuk setiap kelas sebagai berikut:

**Tabel 3.6 Akurasi skema C epoch ke-186 akurasi training 100%**

Kategori	Akurasi
Mobil ambulans	76,85%
Mobil Biasa	47,09%
Mobil Tabrakan	30,96%
Mobil Terbakar	97,41%

Dari ketiga skema diatas maka sebenarnya terjadi overfitting pada kernel kami.

### 3.2 Analisis dan Inovasi

Setelah banyak percobaan yang telah kami lakukan, kami menemukan bahwa akurasi mobil biasa tergolong rendah. Maka dari itu di sini kami melakukan inovasi tambahan di tengah terbatasnya sumber daya yang kami miliki. Kami menggunakan Ridge Regression untuk memprediksi tingkat akurasi suatu arsitektur CNN yang belum pernah kami train sama sekali.

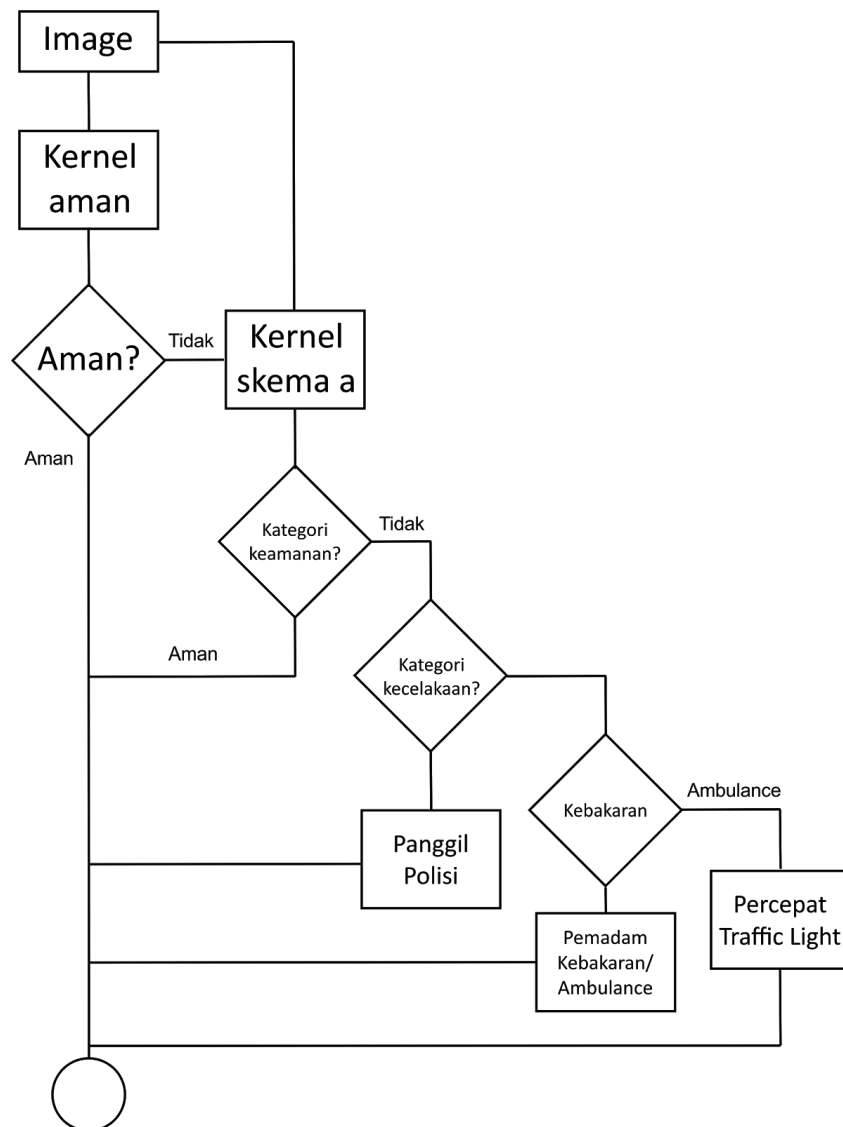
Pertama, kami mengumpulkan seluruh performansi setiap arsitektur CNN yang pernah kami buat kemudian kami mentrainingnya menggunakan Ridge Regression dengan kombinasi input data yaitu filter dan epoch yang kami sandingkan dengan kombinasi output yaitu akurasi pada arsitektur tersebut. Setelah dilakukan training kemudian kami mencoba mencari tahu akurasi suatu arsitektur CNN yang belum pernah kami jalankan sama sekali menggunakan kernel Ridge Regression

Berdasarkan kernel Ridge Regression, kami dapatkan prediksi akurasi hingga 109% pada filter layer konvolusi (65536, 32768, 16384, 8192) pada epoch ke 81 yang mana tidak akan benar-benar terjadi di dunia nyata, tapi setidaknya sudah memberikan gambaran performansi dengan rincian sebagai berikut:

**Tabel 3.7. Prediksi Akurasi Arsitektur CNN menggunakan Ridge Regression**

Filter layer konvolusi	Epoch	Prediksi Akurasi
65536, 32768, 16384, 8192	81	109%
32768, 16384, 8192, 4096	81	74,75%
16384, 8192, 4096, 2048	81	57,46%
8192, 4096, 2048, 1024	81	48,82%
4096, 2048, 1024, 512	81	44,49%
2048, 1024, 512, 256	81	42,33%
1024, 512, 256, 128	81	41,25%
512, 256, 128, 64	81	40,71%

Sehingga dari hasil diatas dapat kami hasilkan gambaran flowchart sistem sebagai berikut:



Gambar 3.4 Flowchart Sistem Penanggulangan Kecelakaan Nasional



## **BAB IV**

### **KESIMPULAN**

Dari penelitian ini didapatkan dua macam kernel yang dapat kami pergunakan sebagai kandidat kuat untuk membangun sistem Sip Kena. Satu kernel spesial yang kami ajukan struktur arsitekturnya menggunakan Ridge Regression dengan konfigurasi filter konvolusi [65536, 32768, 16384, 8192], khusus untuk membedakan kondisi jalan raya aman atau tanpa kecelakaan. Adapun satu kernelnya lagi adalah kernel yang bisa kami pergunakan untuk membedakan 3 jenis kategori khusus untuk penanganan kecelakaan dengan konfigurasi filter konvolusi [16, 32, 64, 128].

Dengan ini, kami dapatkan akurasi deteksi sebagai berikut:

1. Kecelakaan mobil tanpa kebakaran sebesar 60%.
2. Kecelakaan mobil dengan kebakaran sebesar 92,25%.
3. Kehadiran mobil ambulans di jalan sebesar 70,37%.

Dengan penelitian ini kami harap kami dapat mengembangkan sistem kami dan menyempurnakan kernel yang kami buat menggunakan komputer dengan spesifikasi yang memungkinkan untuk mentraining kernel dengan arsitektur di atas.

## DOKUMENTASI

Berikut adalah gambaran dari Sistem Penanggulangan Kecelakaan Nasional jika sudah berjalan:

1. Gambar dibawah ini merupakan gambaran jika ada ambulans. Maka ketika ambulans ada di sebuah perlintasan lampu merah, maka jika lampu lalu lintas dari arah ambulans maka waktu lampu merah akan berjalan lebih cepat.



**Gambar Ambulans**

2. Gambar dibawah ini merupakan gambaran jika tidak terjadi kecelakaan, kebakaran akibat kecelakaan, dan ambulans yang sedang melintasi jalan raya.



**Gambar Jalan Raya**

3. Gambar dibawah ini merupakan gambaran jika terjadi kebakaran akibat kecelakaan di jalan raya.



**Gambar Kebakaran**

4. Gambar dibawah merupakan gambaran ketika terjadi kecelakaan di jalan raya.



**Gambar Kecelakaan**

5. Gambar di bawah merupakan saat pengerjaan project oleh Arista Team.



**Gambar Pada Saat Pengerjaan Sistem**

## DAFTAR PUSTAKA

[1]

[http://bappeda.jogjapro.go.id/dataku/data\\_dasar/index/548-data-kecelakaan-dan-pelanggaran-lalu-lintas?id\\_skpd=39](http://bappeda.jogjapro.go.id/dataku/data_dasar/index/548-data-kecelakaan-dan-pelanggaran-lalu-lintas?id_skpd=39)

[2]

[https://kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel\\_gpr](https://kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel_gpr)

[3] K. B. Lee and H. S. Shin, "An Application of a Deep Learning Algorithm for Automatic Detection of Unexpected Accidents Under Bad CCTV Monitoring Conditions in Tunnels," 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), Istanbul, Turkey, 2019, pp. 7-11, doi: 10.1109/Deep-ML.2019.00010.

[4] J. Akbar, M. Shahzad, M. I. Malik, A. Ul-Hasan and F. Shafait, "Runway Detection and Localization in Aerial Images using Deep Learning," 2019 Digital Image Computing: Techniques and Applications (DICTA), Perth, Australia, 2019, pp. 1-8, doi: 10.1109/DICTA47822.2019.8945889.

[5] A. Kim and J. Lee, "Data Analyst Platform Using Open Source Based Deep Learning Engine," 2018 International Conference on Platform Technology and Service (PlatCon), Jeju, 2018, pp. 1-6, doi: 10.1109/PlatCon.2018.8472770.

[6] A. Srivastava et al., "Deep Learning for Detecting Diseases in Gastrointestinal Biopsy Images," 2019 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2019, pp. 1-4, doi: 10.1109/SIEDS.2019.8735619.

[7] H. Yanagisawa, T. Yamashita and H. Watanabe, "A study on object detection method from manga images using CNN," 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, 2018, pp. 1-4, doi: 10.1109/IWAIT.2018.8369633.

- [8] Y. Heryadi and H. L. H. S. Warnars, "Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM," 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), Phuket, 2017, pp. 84-89, doi: 10.1109/CYBERNETICSCOM.2017.8311689.
- [9] Y. Luan and S. Lin, "Research on Text Classification Based on CNN and LSTM," 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2019, pp. 352-355, doi: 10.1109/ICAICA.2019.8873454.
- [10] M. Sardogan, A. Tuncer and Y. Ozen, "Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm," 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, 2018, pp. 382-385, doi: 10.1109/UBMK.2018.8566635.
- [11] V. Atliha and D. Šešok, "Comparison of VGG and ResNet used as Encoders for Image Captioning," 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2020, pp. 1-4, doi: 10.1109/eStream50540.2020.9108880.
- [12] A. A. Almisreb, N. Jamil and N. M. Din, "Utilizing AlexNet Deep Transfer Learning for Ear Recognition," 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), Kota Kinabalu, 2018, pp. 1-5, doi: 10.1109/INFRKM.2018.8464769.
- [13] C. Chen and F. Qi, "Single Image Super-Resolution Using Deep CNN with Dense Skip Connections and Inception-ResNet," 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, 2018, pp. 999-1003, doi: 10.1109/ITME.2018.00222.
- [14] [https://medium.com/@umerfarooq\\_26378/from-r-cnn-to-mask-r-cnn-d6367b196cfd](https://medium.com/@umerfarooq_26378/from-r-cnn-to-mask-r-cnn-d6367b196cfd)
- [15] C. Lin and C. Lin, "Using Ridge Regression Method to Reduce Estimation Uncertainty in Chlorophyll Models Based on Worldview Multispectral Data," IGARSS 2019 - 2019 IEEE

International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 1777-1780, doi: 10.1109/IGARSS.2019.8900593.

[16] H. Luo and Y. Liu, "A prediction method based on improved ridge regression," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2017, pp. 596-599, doi: 10.1109/ICSESS.2017.8342986.

[17] A. Ben-Cohen, E. Klang, M. M. Amitai, J. Goldberger and H. Greenspan, "Anatomical data augmentation for CNN based pixel-wise classification," 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, 2018, pp. 1096-1099, doi: 10.1109/ISBI.2018.8363762.

[18] K. Fujita, M. Kobayashi and T. Nagao, "Data Augmentation using Evolutionary Image Processing," 2018 Digital Image Computing: Techniques and Applications (DICTA), Canberra, Australia, 2018, pp. 1-6, doi: 10.1109/DICTA.2018.8615799.

[19] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger and H. Greenspan, "Synthetic data augmentation using GAN for improved liver lesion classification," 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, 2018, pp. 289-293, doi: 10.1109/ISBI.2018.8363576.

[20] D. Jeong, H. Kang, D. Kim and J. Lee, "Mask-RCNN based object segmentation and distance measurement for Robot grasping," 2019 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea (South), 2019, pp. 671-674, doi: 10.23919/ICCAS47443.2019.8971673.

[21] Y. Zhou, "Irregular ridge regression: Improvements based on computer simulation," 2011 International Conference on Electrical and Control Engineering, Yichang, 2011, pp. 3955-3957, doi: 10.1109/ICECENG.2011.6057468.