



Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN)

Febri Hafifah, Sayuti Rahman, Munjiat Setiani Asih

Fakultas Teknik dan Komputer, Teknik Informatika, Universitas Harapan Medan, Medan, Indonesia
Email: ¹febrihafifah2000@gmail.com, ²masay.ram@email.com, ³munjiat.stth@gmail.com

Abstrak—Kemacetan lalu lintas merupakan masalah utama yang terjadi di kota-kota besar di Indonesia. Hal ini dapat menimbulkan berbagai dampak negatif seperti pemborosan bahan bakar, pemborosan waktu dan polusi udara. Oleh karena itu, pemerintah membagi jenis jalan raya dan hanya membolehkan truk muatan besar melintas pada jalan arteri. Sehingga perlu adanya smart city untuk mengimplementasikan kebijakan pemerintah agar mengatasi dampak yang ditimbulkan oleh kemacetan lalu lintas. Klasifikasi jenis kendaraan yang melintas pada jalan raya perlu dilakukan agar tidak terjadi pelanggaran kendaraan diluar spesifikasi yang diperbolehkan untuk memasuki jalan raya tertentu. Klasifikasi jenis kendaraan menggunakan metode Convolution Neural Network (CNN). Arsitektur yang digunakan berupa arsitektur existing CNN atau arsitektur CNN yang sudah ada yaitu googlenet dan shufflenet. Kami melakukan fine tuning terhadap Googlenet dan Shufflenet untuk menghasilkan akurasi yang maksimal. Dataset yang digunakan adalah data yang diambil dari beberapa titik kamera CCTV beberapa kota di Indonesia pada bulan Juli 2021. Metode yang diajukan memiliki kemampuan dalam mengklasifikasikan jenis kendaraan dengan akurasi Googlenet sebesar 95,88% dan Shufflenet 96,48%. Dengan demikian, diharapkan dapat memberi kontribusi pada peneliti untuk mengembangkan CNN yang lebih baik agar dapat diimplementasikan untuk kepentingan lalu lintas jalan raya di Indonesia dimasa yang akan datang.

Kata Kunci: Klasifikasi Jenis Kendaraan; Convolution Neural Network (CNN); Googlenet; Shufflenet; Fine Tuning

Abstract—Traffic congestion is a major problem that occurs in big cities in Indonesia. This can cause various negative impacts such as waste of fuel, waste of time, and air pollution. Therefore, the government divides the types of highways and only allows large cargo trucks to pass on arterial roads. So it is necessary to have a smart city to implement government policies in order to overcome the impacts caused by traffic congestion. Classification of the types of vehicles that pass on the highway needs to be done so that there are no vehicle violations outside the specifications that are allowed to enter certain highways. Classification of vehicle types using the Convolution Neural Network (CNN) method. The architecture used is in the form of an existing CNN architecture or an existing CNN architecture, namely googlenet and shufflenet. We fine tune Googlenet and Shufflenet to get maximum accuracy. The dataset used is data taken from several CCTV camera points in several cities in Indonesia in July 2021. The proposed method can classify vehicle types with an accuracy of 95.88% Googlenet and 96.48% Shufflenet. Thus, it is hoped that it can contribute to researchers to develop a better CNN so that it can be implemented for the benefit of road traffic in Indonesia in the future.

Keywords: Car Classification; Convolutional Neural Network (CNN); Googlenet; Shufflenet; Fine Tuning

1. PENDAHULUAN

Kemacetan lalu lintas saat ini merupakan masalah utama yang terjadi di kota-kota besar di Indonesia. Kemacetan lalu lintas merupakan suatu keadaan dimana jumlah kendaraan yang melintas lebih banyak sehingga melebihi kapasitas jalan. Pada umumnya di jalan raya sudah membuat rambu peringatan jalan untuk kendaraan berdasarkan fungsi dan kelasnya yaitu jalan arteri ditujukan untuk kendaraan utama dengan ciri-ciri perjalanan jarak jauh, kecepatan rata-rata tinggi, dan jumlah jalan masuk dibatasi secara efisien dan jalan kolektor ditujukan untuk kendaraan pengumpul/pembagi dengan ciri-ciri perjalanan sedang, kecepatan rata-rata sedang dan jumlah jalan masuk dibatasi. Tetapi saat ini, banyak kendaraan yang melanggar lalu lintas dengan tidak berjalan sesuai dengan jalurnya (Suyuti, 2012).

Seiring berjalannya waktu, kemacetan lalu lintas menjadi masalah yang cukup serius. Hal ini dapat menimbulkan berbagai dampak negatif seperti pemborosan bahan bakar, pemborosan waktu, kemacetan lalu lintas dan polusi udara. Sehingga perlu adanya *smart city* untuk mengatasi dampak yang ditimbulkan oleh kemacetan lalu lintas (Hariani et al., 2017). Konsep *Smart City* dapat digambarkan sebagai infrastruktur perkotaan yang berkembang dengan bantuan mengintegrasikan beberapa solusi *Information and Communication Technology (ICT)* dan *Internet of Things (IoT)* dengan cara yang aman dan tujuan akhir adalah mengelola aset kota. *Smart City* menjadi isu yang sangat penting dalam aspek perkembangan kota – kota. Salah satu konsep perencanaan pembentukan *Smart City* adalah *Smart Transportation*. *Smart Transportation* merupakan bagian dari *Smart City* yang paling penting untuk meningkatkan ekonomi perkotaan. Selain untuk menghemat waktu dan biaya, informasi lalu lintas pun dapat dengan mudah didapatkan oleh pengguna jalan. *Smart Transportation* juga berperan penting untuk melaporkan apa saja yang terjadi di jalan raya (Shukla et al., 2016).

Upayah untuk mengurangi masalah kemacetan lalu lintas yaitu dengan menetapkan jalur-jalur khusus yang hanya boleh dilalui kendaraan roda empat atau lebih. Teknologi seperti visi komputer dapat digunakan untuk mengenali jenis kendaraan yang melewati jalan, sehingga kegiatan pengawasan lalu lintas dapat diserahkan pada aplikasi berbasis komputer. Salah satu metode yang sering digunakan dalam visi komputer adalah deep learning. *Deep Learning* sangat efektif dalam menyelesaikan masalah, salah satunya dalam mengeksploitasi *Convolution Neural Network (CNN)*.

Pada penelitian sebelumnya dibuat oleh Nur Fadlia dan Rifki Kosasih dengan judul *Klasifikasikan Jenis Kendaraan Menggunakan Metode Convolution Neural Network (CNN)*. Menghasilkan tingkat akurasi mencapai 94,4% pada pelatihan dan 73,3% pada tahap pengujian (Fadlia & Kosasih, 2019). Pada penelitian ini dilakukan klasifikasi jenis kendaraan dengan metode *Convolution Neural Network (CNN)* dengan memanfaatkan *Existing CNN*. *Existing CNN* yang akan digunakan dalam penelitian ini antara lain seperti *Googlenet* dan *Shufflenet* yang telah terbukti berhasil

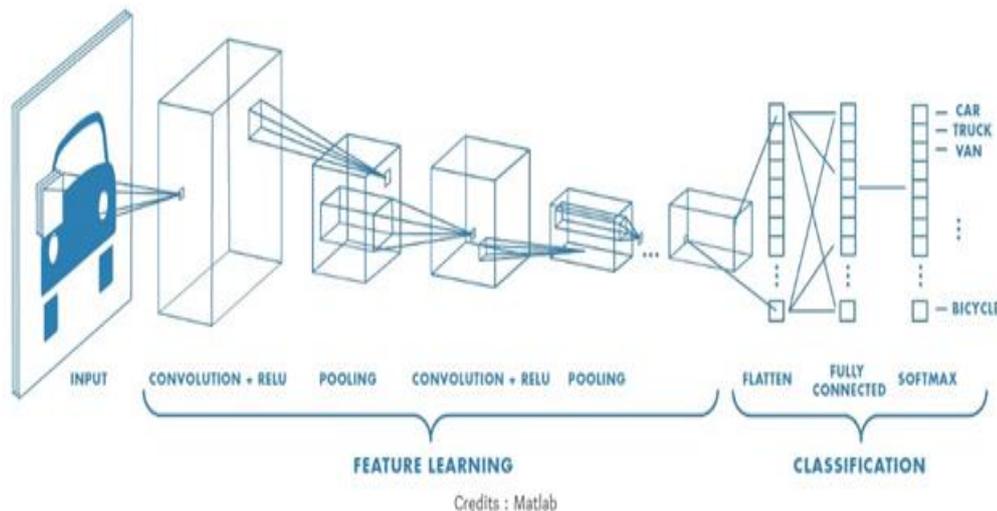
mengklasifikasikan objek dengan baik. *Googlenet* adalah *arsitektur* dari CNN yang dikembangkan google dan telah melakukan *training* dengan jutaan gambar, sedangkan *shufflenet* adalah *arsitektur CNN* yang sangat efisien dalam komputasi, dirancang untuk perangkat seluler dengan daya komputasi 10-150 MFLOP (*mega floating-Point operations per second*). *Existing CNN* ini didesain untuk mengklasifikasi 1000 objek pada *dataset* yang memerlukan komputasi dan ruang penyimpanan yang besar.

Fine tuning merupakan teknik memanfaatkan existing CNN untuk kelas yang lebih sedikit. *Fine tuning* juga dapat meningkatkan akurasi existing CNN (S Rahman et al., 2021; Sayuti Rahman et al., 2020). Pada penelitian ini kami melakukan *fine tuning* terhadap *googlenet* dan *shufflenet* untuk klasifikasi jenis kendaraan. *fine tuning* dilakukan pada jumlah *output fully connected* dengan mengubah 1000 kelas menjadi empat kelas sesuai kebutuhan.

2. METODOLOGI PENELITIAN

2.1 Convolution Neural Network (CNN)

Convolution neural network (CNN) merupakan salah satu algoritma bagian dari *deep learning*. *CNN* termasuk kedalam jenis *deep neural network* karena kedalaman *jarangan* yang tinggi dan banyak diaplikasikan pada data citra. *CNN* mampu melakukan perhitungan matematis pada *input* yang terdiri dari banyak lapisan tersembunyi. Lapisan tersembunyi mengelola *input* gambar dan mengirimkan hasil yang diproses di bagian *output*. *CNN* memiliki lapisan konvolusi yang berguna untuk mengekstraksi fitur dan bagian klasifikasi untuk mengklasifikasi fitur kedalam kelas yang ditentukan pada pelatihan. Fase pelatihan membutuhkan lebih banyak waktu untuk menyelesaikan dan membutuhkan perangkat komputasi yang mumpuni. namun proses prediksi CNN cukup cepat dan akurat. Jaringan *arsitektur convolutional neural network* disajikan pada gambar 1 berikut (Nurhikmat, 2018).



Gambar 1. Arsitektur Convolutional Neural Network

Berdasarkan gambar 1, dapat dilihat CNN memiliki dua bagian utama yaitu featur learning yang terdiri dari lapisan konvolusi, pooling dan fungsi aktivasi. Bagian classification terdiri dari flatten, fully connected dan fungsi aktivasi. Berikut penjelasan dari beberapa bagian CNN.

1. Convolution Layer

Convolution layer yaitu merupakan bagian dari *arsitektur jaringan CNN*. Pada tahap ini dilakukan *operasi konvolusi* terhadap citra dan filter sehingga menghasilkan peta fitur baru. Konvolusi ini merupakan *proses* yang mendasari *jaringan arsitektur CNN*. *Konvolusi* merupakan istilah dari matematis dimana pengaplikasian sebuah fungsi pada *output* fungsi lain secara berulang. *Operasi konvolusi* menerapkan fungsi *output* sebagai *feature Map* dari *input citra*. *Input* dan *output* dapat dilihat sebagai dua argumen bernilai *real*. (Putra, 2016). Pada tabel 1 merupakan piksel dari citra yang di input, kemudian pada tabel 2 yaitu filter 3 x 3 yang akan di kalikan ke citra input sesuai dengan ukuran filternya.

Tabel 1. Citra input

5	0	1	8	4
7	2	4	0	3
9	6	0	1	7
0	1	2	5	3
0	0	4	8	0

Tabel 2. Filter 3 x 3

0	1	0
1	0	1
0	1	0

Cara kerja konvolusi dengan bergesernya filter 3 x 3 dari sudut kiri atas dari gambar dan bergeser ke kanan, setelah semua baris di lalui piksel akan turun ke baris selanjutnya, hingga semua piksel dilalui kemudian disimpan dalam matriks baru. Seperti terlihat pada tabel 3.

Tabel 3. Convolution Layer

5	0	1	8	4	5	0	1	8	4	5	0	1	8	4
7	2	4	0	3	7	2	4	0	3	7	2	4	0	3
9	6	0	1	7	9	6	0	1	7	9	6	0	1	7
0	1	2	5	3	0	1	2	5	3	0	1	2	5	3
0	0	4	8	0	0	0	4	8	0	0	0	4	8	0

5	0	1	8	4	5	0	1	8	4	5	0	1	8	4
7	2	4	0	3	7	2	4	0	3	7	2	4	0	3
9	6	0	1	7	9	6	0	1	7	9	6	0	1	7
0	1	2	5	3	0	1	2	5	3	0	1	2	5	3
0	0	4	8	0	0	0	4	8	0	0	0	4	8	0

5	0	1	8	4	5	0	1	8	4	5	0	1	8	4
7	2	4	0	3	7	2	4	0	3	7	2	4	0	3
9	6	0	1	7	9	6	0	1	7	9	6	0	1	7
0	1	2	5	3	0	1	2	5	3	0	1	2	5	3
0	0	4	8	0	0	0	4	8	0	0	0	4	8	0

Setelah dilakukan proses konvolusi, kemudian dilakukan perkalian antara citra input dan filter 3 x 3, hasil dari perkalian tersebut akan menghasilkan *feature map* yang akan dihitung menggunakan rumus konvolusi.

$$C_{ij} = A \times P_1 + B \times P_2 + C \times P_3 + D \times P_4 + E \times P_5 + F \times P_6 + G \times P_7 + H \times P_8 + I \times P_9$$

Keterangan :

C_{ij} = Nilai *feature map* yang dihasilkan proses konvolusi

A-I = Nilai matriks gambar input

P_{1-9} = Nilai matriks *feature detector*

Nilai *feature map* yang didapatkan dari hasil penjumlahan akan dijadikan kedalam matriks seperti pada Tabel 4

Tabel 4. Hasil feature map

20	3	16
12	13	12
8	10	14

2. Pooling Layer

Pooling layer adalah proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, tujuan dari *pooling layer* yaitu untuk meningkatkan *invariansi posisi* dari fitur. Kemudian *max pooling* membagi *output* dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun *matriks citra* yang telah direduksi seperti yang ditunjukkan pada tabel 5. (Putra, 2016). Kemudian hasil dari nilai maksimum pada *max pooling* ditunjukkan pada tabel 6

Tabel 5. Proses max pooling

20	3	16	20	3	16
12	13	12	12	13	12
8	10	14	8	10	14

20	3	16	20	3	16
12	13	12	12	13	12
8	10	14	8	10	14

Tabel 6. Hasil max pooling

20	16
13	14

3. Fully Connected Layer

Fully connected layer yaitu layer akan mengambil seluruh *neuron* pada layer sebelumnya (*convolutional layer* dan *max pooling layer*) dan menghubungkannya ke setiap *single neuron* yang ada. Fully connected layer adalah layer yang biasanya digunakan dalam penerapan *multilayer perceptron (MLP)* dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Seperti terlihat pada tabel 7 (Putra, 2016).

Tabel 7. Hasil Fully connected

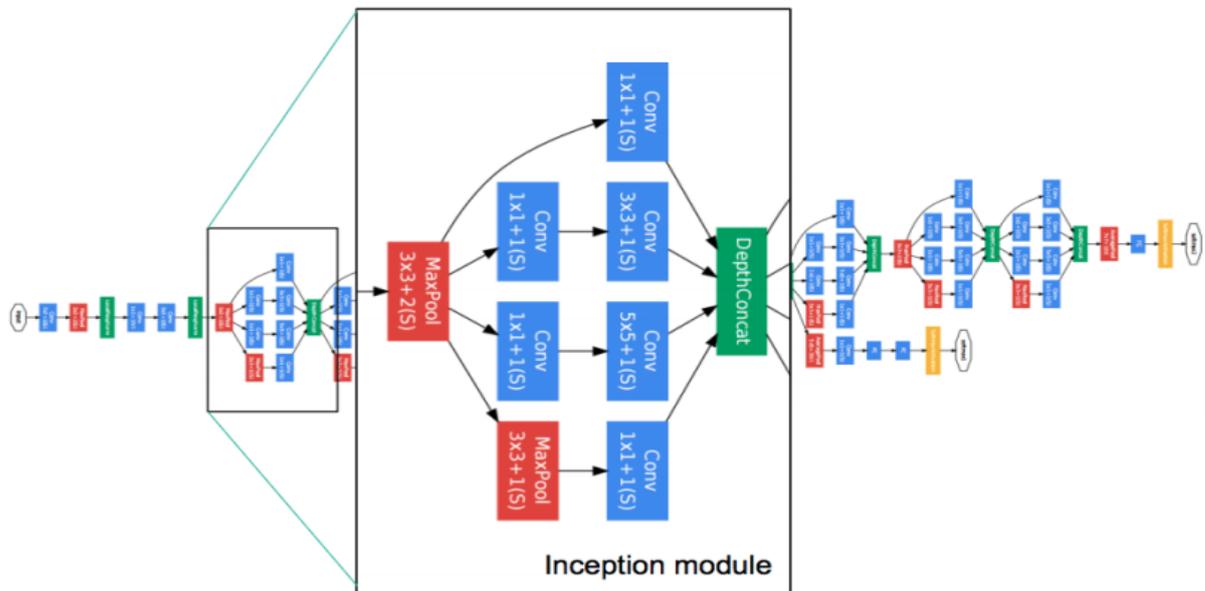
20
16
14
13

2.2 Existing Convolutional Neural Network (CNN)

Existing CNN merupakan bagian dari *arsitektur CNN* yang telah dibuat oleh peneliti terdahulu, Existing CNN dibuat untuk bisa mengklasifikasikan suatu objek dengan tingkat *akurasi* yang cukup memuaskan. Beberapa *existing CNN* yang banyak dikenal diantaranya adalah *googlenet* dan *shufflenet*.

1. Googlenet

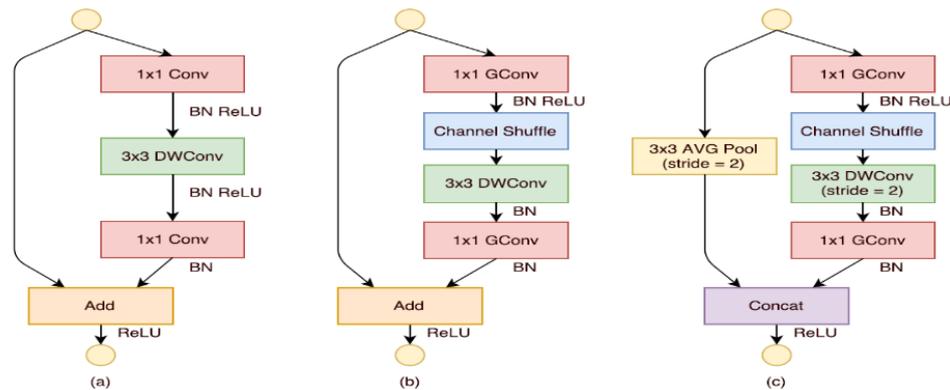
Googlenet memiliki karakteristik dengan menggunakan *jaringan inception* lebih dari satu dan lebih luas, yang berfungsi untuk meningkatkan *akurasi*. Googlenet (*Inception V1*) merupakan sebuah model dan *arsitektur* berdasarkan modifikasi *arsitektur CNN*. Googlenet mampu meminimalkan tingkat kesalahan menjadi 6,67% yang mengungguli alexnet sebesar 15,3%. Metode ini sangat dekat dengan kemampuan kinerja manusia. Berdasarkan penelitian menyatakan bahwa diperlukan beberapa pelatihan manusia untuk mengalahkan keakuratan dari *googlenet*. Googlenet menggunakan *normalisasi batch* untuk mengelola *input* secara bertahap, dan distorsi gambar untuk menyesuaikan ukuran sesuai kebutuhan. Kemudian *googlenet* melewati *fase inception* terlebih dahulu, kemudian pada saat layer kelima, beberapa *fitur* akan melakukan *fully connected neuron*. *Fitur* yang lain akan tetap melanjutkan *proses ekstraksi* dengan *convolution*. (Sabilla, 2020)



Gambar 2. Googlenet

2. Shufflenet

Jaringan *shufflenet* di desain untuk *small network* dengan prinsip *desain bottleneck*, kemudian diaplikasikan untuk *mobile device* dengan daya komputasi yang terbatas, *jaringan* ini memiliki *arsitektur* yang efisien dengan cara menambahkan *fitur map channels* untuk menyajikan lebih banyak data yang dapat bekerja pada *jaringan* yang sangat kecil. *Shufflenet* adalah *arsitektur CNN* yang sangat efisien dalam komputasi, dirancang untuk perangkat seluler dengan daya komputasi 10-150 MFLOP (*mega floating-Point operations per second*). *Shufflenet* menggunakan *konvolusi grup Pointwise* dan *channel shuffle* untuk mengurangi biaya komputasi sambil menjaga *akurasi* (Zhang et al., 2018).



Gambar 3. Shufflenet

3. HASIL DAN PEMBAHASAN

Ada beberapa *proses* yang akan dilakukan yaitu mengambil data masukkan berupa hasil *screenshot* dari CCTV dan data yang sudah tersedia di penelitian sebelumnya, untuk mengklasifikasi data citra. Kemudian dilakukan pemotongan citra atau *cropping* untuk menghilangkan latar yang tidak gunakan untuk *proses arsitektur Existing CNN*. Pengumpulan data dilakukan untuk mendapatkan citra yang akan digunakan pada *proses* pelatihan dan pengujian model *arsitektur Existing CNN*. Selanjutnya data akan *diproses* dengan mengubah ukuran citra menjadi ukuran yang sama, kemudian dilakukan pembagian data menjadi dua, yaitu *data latih (training)* dan *data uji (testing)*, setelah *diproses* masuk pada tahap perancangan *arsitektur Existing CNN* dalam tahapan ini digunakan untuk melatih data dalam mengenali objek. Tahap selanjutnya melakukan pelatihan model dengan menggunakan *data latih* yang dikumpulkan sebelumnya. Kemudian dilakukan untuk menguji tingkat *akurasi* dari *arsitektur Existing CNN*.

2.3 Fine Tuning

Fine tuning adalah *proses* yang mengambil model yang telah dilatih untuk suatu tugas yang diberikan dan kemudian menyetel atau mengubah model untuk membuatnya melakukan tugas serupa. *Proses Fine Tuning* yaitu *jaringan* akan tersimpan yang sebelumnya dilatih pada kumpulan data besar, biasanya *Fine Tuning* melakukan tugas pada *klasifikasi gambar* dengan skala besar. Dengan asumsi tugas asli mirip dengan tugas baru, menggunakan *jaringan saraf tiruan* yang telah dirancang dan dilatih memungkinkan untuk mengambil keuntungan dari apa yang telah dipelajari model tanpa harus mengembangkannya dari awal. Ketika membangun sebuah model dari awal, biasanya harus mencoba banyak pendekatan melalui trial-and-error (Gando et al., 2016).

3.1 Dataset

Dataset merupakan kumpulan objek yang berasal dari data dan dikelola menjadi sebuah informasi. Data yang digunakan berupa gambar atau citra yang diambil dari data yang sudah tersedia di penelitian sebelumnya dan data mentah yang di rekam dari CCTV jalan raya. CCTV yang diambil dari berbagai daerah yaitu di balmera, semangi, semarang, tanjung mulia. Data yang di gunakan untuk *data uji* sebanyak 633 citra, dan *data latih* sebanyak 1479 citra dengan kualitas gambar yang baik dapat dilihat pada Tabel 8. Seluruh data berjumlah 2112 citra. Hasil pengambilan data akan dikelompokkan ke dalam beberapa *class* yang sesuai untuk kebutuhan data *training*.

Tabel 8. Keseluruhan Dataset

No	Dataset	Jumlah Data
1.	Data latih	1479
2.	Data uji	633

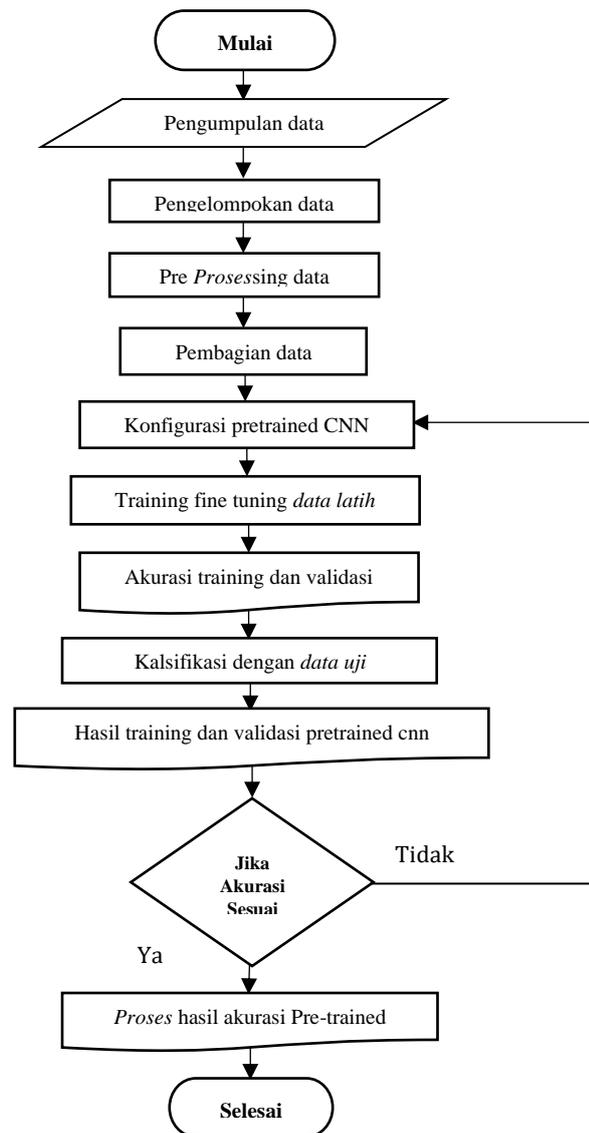
Pada keseluruhan data yang berjumlah 2112 terdapat 4 *class* kendaraan, 4 *class* tersebut terdiri dari mobil, truk 2 sumbu, truk 3 sumbu, dan truk 4 sumbu. Pada tahap ini masing-masing *dataset* pada *class* dibagi menjadi *data latih* dan *data uji* yang digunakan pada penelitian ini dapat dilihat pada Tabel 9.

Tabel 9. Pembagian Dataset per Class

No	Jenis Class	Jumlah data	Data latih	Data uji
1	Mobil	1040	728	312
2	Truk 2 sumbu	575	403	172
3	Truk 3 sumbu	280	196	84
4	Truk 4 sumbu	217	152	65

3.2 Alur Perancangan Sistem

Alur perancangan sistem yaitu dimana alur tersebut akan menjelaskan sistem secara luas, yang memperlihatkan masukan, proses dan keluaran dari sistem yang akan dirancang, disajikan pada gambar 4.



Gambar 4. Alur Tahap Penelitian

Adapun penjelasan masing-masing tahap yang terdapat pada Gambar 4 sebagai berikut:

1. Pengumpulan Data
Pada tahapan pengumpulan data yang perlu di siapkan yaitu *dataset*, *dataset* yang digunakan berupakan gambar atau citra yang diambil dari data yang sudah tersedia di penelitian sebelumnya, dan data mentah yang direkam dari *cctv* jalan raya, data diambil melalui rekaman kemudian di *screenshot*.
2. Pengelompokan data
Setelah data dikumpulkan, data akan dikelompokkan berdasarkan jenisnya atau classnya seperti mobil, truk 2 sumbu, truk 3 sumbu, truk 4 sumbu.
3. Pre-processing data
Tahapan *pre-processing* data, tahapan ini merupakan tahap pengolahan citra dengan mengubah ukuran citra kendaraan menjadi ukuran yang sama bertujuan untuk menghasilkan citra yang lebih baik
4. Pembagian data
Selanjutnya dilakukan pembagian data menjadi 2 bagian yaitu *data latih* dan *data uji*, Proses data latih yang merupakan proses untuk melatih jaringan saraf tiruan, kemudian sistem membaca data yang di input dan melakukan preprosesing serta ekstraksi fitur untuk mengambil fitur yang akan digunakan sebagai data input, kemudian dilakukan konfigurasi *existing CNN*.
5. Training fine tuning data latih

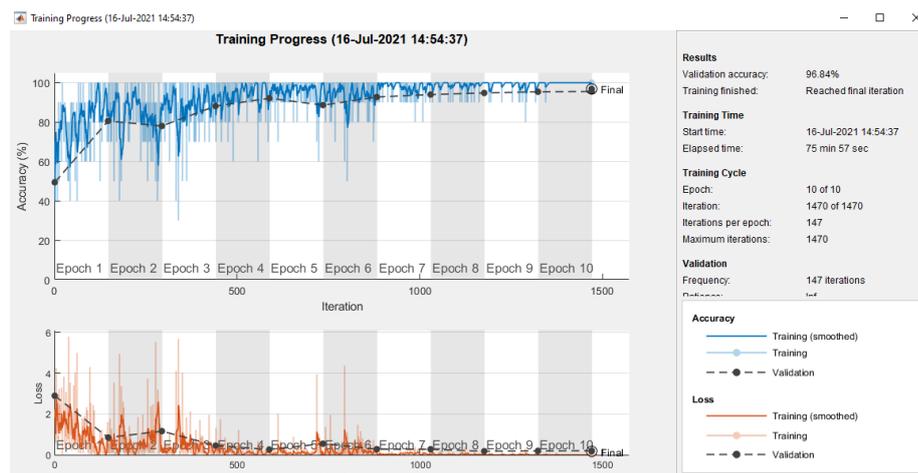
Training fine tuning data latih, tahap ini dilakukan *proses ngetraining ulang data latih* untuk mendapatkan hasil yang lebih baik, setelah di *fine tuningkan proses* akan memunculkan grafik tingkat *akurasi training* dan *validasi*. Kemudian pada *data uji* dilakukan *pengklasifikasian* jenis kendaraan dalam mengenali *karakter* dan mengambil tingkat *akurasi* pada *proses pelatihan jaringan*

6. Hasil *training* dan *validasi existing CNN*

Pada tahap selanjutnya akan menampilkan hasil *training* dan *validasi existing cnn* dengan menunjukkan berapa persen tingkat *akurasi* untuk mengenali data yang di *input* dan hasil keluarannya mampu mengenali atau mendeteksi golongan jenis kendaraan. Jika *akurasi* sesuai maka *proses akurasi existing CNN* akan selesai, jika tingkat *akurasi* tidak sesuai maka *proses* akan kembali ke *konfigurasi pretrained cnn*.

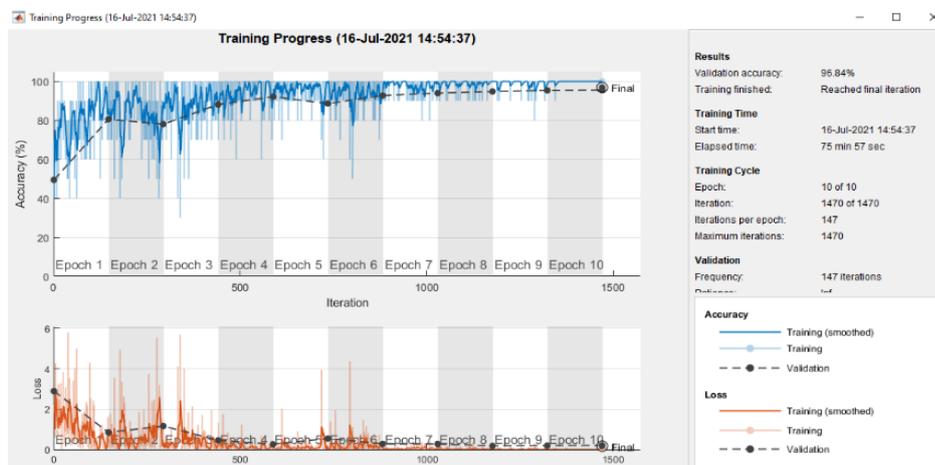
3.3 Pelatihan Existing CNN

Dataset pelatihan berjumlah 70% dan *dataset* pengujian berjumlah 30% dari data keseluruhan. Hasil dari pelatihan *jaringan googlenet* dengan *epoch* 10 memiliki *iterasi* 1470 dengan *iterasi per epochnya* 147. Waktu yang dibutuhkan untuk pelatihan *jaringan* yaitu 75 menit dengan banyaknya data gambar yang dilatih sebanyak 2112 gambar berupa mobil dan truk, pada *proses* pelatihan *jaringan googlenet akurasi* yang dicapai sebesar 95,88% seperti terlihat pada Gambar 5 berikut.



Gambar 5. Proses Pelatihan Googlenet

Proes pelatihan *jaringan shufflenet*, sama halnya dengan pelatihan *jaringan googlenet*, dimana *epoch* 10 dan *iterasinya* 1470, yang membedakan hanya waktu yang dibutuhkan untuk melatih *jaringan shufflenet*, dibutuhkan waktu 105 menit. pada *proses* pelatihan *jaringan shufflenet akurasi* yang dicapai sebesar 96,84% terlihat pada gambar 6. Semakin naik nilai *epoch* yang dibuat, semakin tinggi hasil *akurasi* yang didapatkan.



Gambar 6. Proses Pelatihan Shufflenet

3.4 Pengujian Sistem

Proses pengujian dilakukan untuk mengetahui kerja *jaringan*. Pengujian dilakukan terhadap *data uji*. Dari hasil pengujian dapat dinilai kemampuan *jaringan* dalam mengenali data yang di *input*. Gambar 7 merupakan contoh hasil pengujian sistem. Sistem mampu mengkalasifikasikan gambar tersebut dan mengenali jenisnya dengan benar, dari gambar tersebut sistem mengenali dengan akurasi yang tinggi.



Gambar 7. Tampilan Hasil Proses

Pada gambar 10 dapat mengetahui seberapa besar keberhasilan *aplikasi* tersebut mengenali jenis gambar yang di *input*. Pada Tabel dibawah ini merupakan hasil dari pengujian deteksi jenis kendaraan.

Tabel 10. Hasil Pengujian Sistem

Gambar	Akurasi	Hasil Klasifikasi
	<i>Googlenet</i> : 99,95% <i>Shufflenet</i> : 99,99%	Mobil
	<i>Googlenet</i> : 99,98% <i>Shufflenet</i> : 99,99%	Mobil
	<i>Googlenet</i> : 88,46% <i>Shufflenet</i> : 90,37%	Truk 2 Sumbu
	<i>Googlenet</i> : 98,84% <i>Shufflenet</i> : 99,89%	Truk 2 Sumbu
	<i>Googlenet</i> : 99,26% <i>Shufflenet</i> : 99,92%	Truk 3 Sumbu

Gambar	Akurasi	Hasil Klasifikasi
 <p>Citra RGB</p>	<p>Googlenet : 90,99% Shufflenet : 99,91%</p>	Truk 3 Sumbu
 <p>Citra RGB</p>	<p>Googlenet : 99,98% Shufflenet : 97,02%</p>	Truk 4 Sumbu
 <p>Citra RGB</p>	<p>Googlenet : 99,87% Shufflenet : 99,98%</p>	Truk 4 Sumbu

Berdasarkan hasil dari pelatihan *Existing cnn*, data hasil uji yang telah dilakukan pada identifikasi jenis mobil berdasarkan bentuk, menggunakan metode *convolutional neural network* yang dibangun menggunakan arsitektur *googlenet* dan *shufflenet* dapat diperoleh nilai akurasi dengan rata-rata 95,88% dan 96,84%, akurasi tersebut bisa dihitung menggunakan rumus dibawah ini.

$$\text{Persentase akurasi} = \frac{\text{jumlah citra data uji benar}}{\text{jumlah keseluruhan citra data uji}} \times 100\%$$

$$\text{akurasi googlenet} = \frac{607}{633} \times 100\% = 95,88$$

$$\text{akurasi shufflenet} = \frac{613}{633} \times 100\% = 96,84$$

4. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan, maka dapat diambil kesimpulan bahwa pendeteksian jenis kendaraan sangat dibutuhkan untuk dapat di *aplikasikan* berbagai bidang, salah satunya mengatur lalu lintas dengan cara mendeteksi bentuk fisik atau jenis dari kendaraan yang lewat sehingga mengurangi kemacetan. Perbandingan tingkat *akurasi* antara pelatihan *arsitektur existing Convolution Neural Network (CNN) googlenet* dan *shufflenet* dalam mengklasifikasikan 4 (empat) kelas jenis kendaraan, dimana *shufflenet* mampu mengenali jenis kendaraan pada data gambar dengan *akurasi* rata-rata 96,84% dan *googlenet* mampu mengenali jenis mobil dengan *akurasi* rata-rata 95,88%. Berdasarkan tingkat *akurasi* rata-rata pada penelitian ini didapat bahwa *arsitektur existing Convolution Neural Network (CNN) Shufflenet* adalah *arsitektur* terbaik karena memiliki nilai *akurasi* rata-rata tertinggi yaitu 96,84%.

UCAPAN TERIMAKASIH

Penelitian ini merupakan penelitian dasar visi komputer yang dikerjakan di Laboratorium Komputer Fakultas Teknik Universitas Harapan Medan. Terimakasih kepada rekan-rekan mahasiswa dan dosen yang telah memberikan saran pada penelitian ini. Terima kasih juga kepada pimpinan Fakultas Teknik dan Komputer Universitas Harapan Medan yang telah memberikan keleluasaan para peneliti dalam menggunakan fasilitas laboratorium.

REFERENCES

Fadlia, N., & Kosasih, R. (2019). Klasifikasi Jenis Kendaraan Menggunakan Metode Convolutional Neural Network (Cnn). *Jurnal Ilmiah Teknologi Dan Rekayasa*, 24(3), 207–215. <https://doi.org/10.35760/tr.2019.v24i3.2397>

Gando, G., Yamada, T., Sato, H., Oyama, S., & Kurihara, M. (2016). Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs. *Expert Systems with Applications*, 66, 295–301.

Hariani, P., Safina, L., & Syarifuddin, J. (2017). City Smart Transportation Sebagai Strategi Medan Menuju Smart City. *Jurnal*



Pembangunan Perkotaan, 5(2), 50–58.

- Nurhikmat, T. (2018). *Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek*.
- Putra, I. (2016). *Klasifikasi citra menggunakan convolutional neural network (CNN) pada caltech 101*. Institut Teknologi Sepuluh Nopember.
- Rahman, S., Ramli, M., Arnia, F., Muharar, R., & Sembiring, A. (2021). Performance analysis of mAlexnet by training option and activation function tuning on parking images. *IOP Conference Series: Materials Science and Engineering*, 1087(1), 012084. <https://doi.org/10.1088/1757-899X/1087/1/012084>
- Rahman, Sayuti, Ramli, M., Arnia, F., Sembiring, A., & Muharar, R. (2020). Convolutional Neural Network Customization for Parking Occupancy Detection. *2020 International Conference on Electrical Engineering and Informatics (ICELTICs)*, 1–6.
- Sabilla, I. A. (2020). *Arsitektur Convolutional Neural Network (CNN) untuk Klasifikasi Jenis dan Kesegaran Buah pada Neraca Buah*. Institut Teknologi Sepuluh Nopember.
- Shukla, S., Balachandran, K., & Sumitha, V. S. (2016). A framework for smart transportation using Big Data. *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, 1–3.
- Suyuti, R. (2012). Implementasi” Intelligent Transportation System (ITS)” untuk Mengatasi Kemacetan Lalu Lintas di DKI Jakarta. *Konstruksia*, 3(2).
- Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6848–6856.