

**IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK TERHADAP
TRANSPORTASI TRADISIONAL MENGGUNAKAN KERAS**

(Studi Kasus : Data Citra Transportasi Tradisional Andong, Becak dan Pedati)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Program Studi Statistika



Arfian

14 611 057

**JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2018**

HALAMAN PERSETUJUAN PEMBIMBING

TUGAS AKHIR

Judul Implementasi *Convolutional Neural Network*
Terhadap Transportasi Tradisional Menggunakan
Keras

Nama Mahasiswa Arfian

Nomor Mahasiswa 14 611 057

TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK
DIUJIKAN

Yogyakarta, 03 April 2018

Pembimbing

البرج الاستاذة الانسية

(Dr. RB. Fajriya Hakim, S.Si., M.Si.)

HALAMAN PENGESAHAN

TUGAS AKHIR

IMPLEMENTASI *CONVOLUTIONAL NEURAL NETWORK* TERHADAP TRANSPORTASI TRADISIONAL MENGGUNAKAN KERAS

(Studi Kasus : Data Citra Transportasi Tradisional Andong, Pedati dan Becak)

Nama Mahasiswa : Arfian
Nomor Mahasiswa : 14 611 057

TUGAS AKHIR INI TELAH DIUJIKAN
PADA TANGGAL 18 April 2018

Nama Penguji

Tanda tangan

Andrie Pasca Hendradewa, S.T., M.T.

Ayundyah Kesumawati, S.Si., M.Si.

Dr. RB. Fajriya Hakim, S.Si., M.Si.

Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



(Drs. Aliwar, M.Sc., Ph.D.)

KATA PENGANTAR



Assalamu'alaikum Warahmatullaahi Wabarakaatuh

Alhamdulillah rabbil'aalamiin, puji syukur kehadiran Allah SWT, yang tiada henti melimpahkan nikmat, taufiq dan hidayah-Nya, sehingga penulis dapat menyelesaikan penulisan tugas akhir ini. Shalawat salam tak lupa penulis haturkan ke junjungan nabi besar Muhammad SAW, keluarganya, sahabatnya, dan semua umat yang mengikuti petunjuk beliau hingga akhir zaman.

Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana jurusan statistika di Universitas Islam Indonesia. Tugas akhir yang berjudul "*Implementasi Convolutional Neural Network Terhadap Transportasi Tradisional Menggunakan Keras*" ini selain disusun guna memenuhi persyaratan untuk menyelesaikan studi jenjang strata satu di Jurusan Statistika Universitas Islam Indonesia, juga untuk memberikan wawasan dan mengenalkan penerapan ilmu statistika dalam hal pengumpulan dan pengolahan data.

Perlu disadari bahwa pelaksanaan penyusunan tugas akhir ini tidak lepas dari bimbingan, dorongan dan bantuan baik materi maupun non materi dari berbagai pihak. Oleh karena itu perkenankanlah penyusun menghaturkan ucapan terimakasih kepada:

1. Allah SWT dan Rasulullah SAW yang telah memberikan nikmat Iman dan Islam serta karunia-Nya kepada penulis sehingga dilancarkan dalam pelaksanaan tugas akhir.
2. Kedua orang tua tercinta, Ayah Umar Usman dan Emak Azizah (Almh) yang telah mencurahkan kasih sayangnya, memberikan doa, batuan moril maupun materil, serta kakak dan abang terkasih, Apriyani dan Julizar yang selalu memberikan do'a dan motivasi, sehingga dapat memberikan semangat yang pantang menyerah.
3. Bapak Drs. Allwar, M.Sc., Ph.D. Selaku dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia berserta jajaran yang telah

membantu memudahkan saya dalam proses perkuliahan dan proses penyelesaian tugas akhir ini.

4. Bapak Dr. RB. Fajriya Hakim, M.Si. selaku Pembimbing dan Kepala Program Studi Statistika FMIPA UII yang selalu sabar membimbing saya selama mengerjakan tugas akhir ini.
5. Seluruh dosen pengajar dan staff prodi Statistika, terimakasih atas bekal ilmu dan bantuannya dalam proses belajar, semoga menjadi amal kebaikan Bapak/Ibu sekalian.
6. Sendhyka Cakra Pradana dan Moh. Khualid Yusuf yang bersedia mengajarkan penulis terkait metode penelitian yang digunakan dan juga setia menjadi sahabat penulis sampai saat ini.
7. Keluarga besar INVISIO, sebagai tim promosi dan publikasi prodi Statistika UII, terimakasih atas kebersamaan, kekeluargaan, kelucuan, kekompakan, dan pelajaran berharga lainnya, semoga INVISIO tambah maju, dan dapat membuat prodi Statistika UII menjadi lebih hebat.
8. Sahabat setia, sekaligus sahabat seperjuangan, Grup Kita, Sulhaerati, Rima Junidar, Adwi Guntur, Reza Yusuf Bahtiar, Yayan Dwi Septian, Rizki Dwi Noviantika, dan Fitria Nuralif, kalian semua istimewa, semoga tali silaturahmi di antara kita tidak pernah putus.
9. Mbak Asmadhini, selaku keluarga, sahabat, kakak yang telah memberi nasehat, arahan, dan bimbingan dalam hal akademik selama kuliah.
10. Keluarga Besar IPMKRKA-Y, terimakasih sebagai tempat belajar, bermain dan bersosialisasi dengan orang banyak.
11. Keluarga Kontrakan Bujang Dare Sengak Agung Lesmana, Priyo, M Firdaus, Heri Ramdhan, dan Ayu Febrianti yang selalu memberikan keceriaan dan hiburan kepada penulis.
12. Semua pihak yang telah mendukung dan ikut membantu penulis, terima kasih.

Semoga dukungan dan bantuan dari semua pihak senantiasa mendapat balasan dari Allah SWT. Penulis menyadari bahwa tugas akhir ini masih memiliki kekurangan di dalamnya. Hal tersebut karena keterbatasan ilmu dan

pengetahuan yang dimiliki penulis semata. Penulis menerima kritik dan saran yang membangun demi perbaikan tugas akhir ini. Penulis berharap agar penelitian ini dapat bermanfaat dan memberikan khasanah pengetahuan bagi penulis, pembaca, maupun penelitian di masa depan.

Wassalamu'alaikum Warahmatullaahi Wabarakaatuh

Yogyakarta, 03 April 2018

Arfian

DAFTAR ISI

HALAMAN PERSETUJUAN PEMBIMBING	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN	xi
PERNYATAAN.....	xii
INTISARI	xiii
ABSTRACT.....	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	6
BAB III LANDASAN TEORI.....	12
3.1. Transportasi.....	12
3.1.1 Fungsi dan Manfaat Transportasi... ..	12
3.1.2 Jenis-Jenis Transportasi dan Alat Transportasi.....	13
3.1.3 Transportasi Tradisional dan Modern.....	14
3.2. Citra	16
3.2.1 Representasi Citra Digital.....	16
3.2.2 Pengolahan Citra Digital	17
3.3. <i>Preprocessing</i> Citra.....	19
3.3.1 Penghilang <i>noise</i> dengan filter spasial.....	19
3.4. <i>Machine Learning</i>	20
3.5. <i>Deep Learning</i>	21

3.6.	<i>Deep Feedforward Networks</i>	23
3.7.	<i>Convolutional Neural Network</i>	24
3.8.	Operasi Konvolusi	24
3.9.	Arsitektur Jaringan CNN	25
3.9.1.	<i>Feature Extraction Layer</i>	26
3.9.2	<i>Fully-Connected Layer (FC Layer)</i>	29
3.10.	<i>Keras</i>	29
BAB IV METODOLOGI PENELITIAN		30
4.1.	Populasi dan Sampel Penelitian	30
4.2.	Teknik Pengumpulan Data	30
4.3.	Variabel dan Definisi Operasional Variabel	30
4.4.	Metode Analisis Data	31
4.5.	Tahap Penelitian	31
BAB V ANALISIS DAN PEMBAHASAN		32
5.1.	Implementasi Pengumpulan Data	32
5.2.	Penggunaan <i>Fatkun Batch Image Downloader</i>	32
5.3.	<i>Load</i> Datase	33
5.4.	Pembuatan Data Latih dan Data Uji	34
5.5.	<i>Risize</i> dan <i>Combine</i>	35
5.6.	<i>Reorder Dimension</i> dan <i>Response</i>	36
5.7.	Model	37
5.8.	<i>Fit</i> Model	40
5.9.	<i>Evaluation</i> dan <i>Prediction</i> Data Latih	42
5.10.	<i>Evaluation</i> dan <i>Prediction</i> Data Latih	42
5.11.	<i>Data Test</i>	43
BAB VI PENUTUP		45
6.1.	Kesimpulan	45
6.1.	Saran	46
DAFTAR PUSTAKA		47
LAMPIRAN		49

DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian sebelumnya dengan penelitian yang penulis lakukan	8
Tabel 3.1 Perbandingan Metode Convolutional Neural Network dan Background Subtraction	23
Tabel 4.1 Definisi Operasional Variabel.....	30
Tabel 5.1 Perbandingan data latih dan data uji	34
Tabel 5.2 Hasil akurasi data latih dan data uji pada model.....	41

DAFTAR GAMBAR

Gambar 3.1 Representasi Citra Digital Dalam Bentuk Matriks.....	16
Gambar 3.2 Contoh Citra Digital	17
Gambar 3.3 Langkah-Langkah Dalam Pemrosesan Citra Digital.....	17
Gambar 3.4 Filter dengan Ukuran 3x3.....	20
Gambar 3.5 Perbandingan Pemrograman <i>Tradision</i> dengan <i>Machine Learning</i> ..	20
Gambar 3.6 Contoh Jaringan <i>Deep Feedforward</i>	23
Gambar 3.7 Contoh Jaringan <i>Convolutional Neural Network</i>	24
Gambar 3.8 MLP	26
Gambar 3.9 <i>Convolutional Layer</i>	26
Gambar 3.10 <i>Feature Map</i>	27
Gambar 3.11 <i>Pooling Layer</i>	28
Gambar 4.1 <i>Flowchart</i> Penelitian	31
Gambar 5.1 Citra dalam Google Image	32
Gambar 5.2 Data Citra Dari <i>Software Fatkun Batch Image Downloader</i>	33
Gambar 5.3 <i>Script</i> Pemanggilan Data.....	33
Gambar 5.4 <i>Script</i> Pemanggilan Data Latih Dan Data Uji	35
Gambar 5.5 <i>Script Resize dan Combine</i>	35
Gambar 5.6 <i>Combine</i> Dataset Latih Dan Uji	36
Gambar 5.7 <i>Script Reorder dimension, Response, dan One hot encoding</i>	37
Gambar 5.8 <i>Script</i> Penentuan Model	38
Gambar 5.9 <i>Output</i> Model software R Studio.....	39
Gambar 5.10 <i>Script Fit</i> Model	40
Gambar 5.11 Nilai Akurasi dan <i>Loss Fit</i> Model	40
Gambar 5.12 <i>Plot Fit</i> Model	41
Gambar 5.13 Hasil Dari Evaluasi dan Prediksi Data Latih.....	42
Gambar 5.14 Hasil Dari Evaluasi dan Prediksi Data Uji	43
Gambar 5.15 Citra Data <i>Test</i>	43
Gambar 5.16 Hasil Prediksi Data <i>Test</i>	44

DAFTAR LAMPIRAN

Lampiran 1 <i>Script R Studio CNN Library Keras</i>	49
---	----

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 03 April 2018



Arfian

IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK TERHADAP TRANSPORTASI TRADISIOANAL MENGGUNAKAN KERAS

(Studi Kasus : Data Citra Transportasi Tradisional Andong, Becak dan Pedati)

Arfian

Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Islam Indonesia

INTISARI

Transportasi merupakan urat nadi Pembangunan Nasional yang sangat dibutuhkan dalam kehidupan sehari-hari sebagai penunjang dalam memenuhi kebutuhan konsumsi, produksi, dan distribusi. Perkembangan akan teknologi juga berdampak bagi kemajuan alat transportasi yang digunakan oleh masyarakat. Berbagai sarana transportasi seperti Bus, Taxi, Andong, becak, pedati atau gerobak sapi sampai sepeda banyak digunakan oleh masyarakat untuk melakukan pergerakan atau perpindahan. Andong, becak dan pedati merupakan transportasi tradisional yang biasanya digunakan masyarakat untuk mengelilingi tempat wisata, pasar, dan lain-lain. Dalam mengenali bentuk serta jenis andong, becak dan pedati, kita bisa mempelajari dari gambar yang ada di internet. Kesamaan bentuk ketiga transportasi tidak dapat dikenali oleh citra digital sehingga sulit untuk melakukan klasifikasi terhadap ketiga objek yang digunakan. Problem dalam visi komputer belum mampu menduplikasi kemampuan manusia dalam memahami informasi citra, agar komputer dapat melakukan hal yang sama selayaknya manusia. Oleh karena itu, tujuan penelitian ini adalah ingin mengetahui implementasi convolution neural network (CNN) pada transportasi tradisional berbasis citra digital dengan bantuan library keras pada software R Studio. Prediksi setiap klasifikasi citra data test Andong, Pedati, dan Becak memiliki akurasi yang seluruhnya benar dan tepat. Pada hasil kelas prediksi menunjukkan akurasi dari hasil data test seluruhnya benar. Hasil tersebut dapat dikatakan bahwa penggunaan metode Covolutional Neural Network (CNN) menggunakan Keras bisa diimplementasikan terhadap citra transportasi tradisional Andong, Pedati, dan Becak.

Kata kunci : *Convolutional Neural Network, , Library Keras, Citra Transportasi Tradisional.*

IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK ON TRADITIONANAL TRANSPORTATION WITH KERAS

(Case Study : Image Data Traditioanal Transportation Andong, Becak and
Pedati)

Arfian

Department of Statistics, Faculty of Mathematics and Natural Sciences
Islamic University of Indonesia

ABTRACT

Transportation is a vein of National Development that is needed in daily life as a support in meeting the needs of consumption, production, and distribution. The development of technology will also affect the progress of transportation used by the community. Various means of transportation such as Bus, Taxi, Andong, rickshaw, cart or cattle cart until the bike is widely used by the community to do the movement or displacement. Andong, pedicab and pedicab is a traditional transportation that is usually used by the community to surround the sights, markets, and others. In recognizing the shape and type of andong, pedicab and pedicab, we can learn from the images on the internet. The similarity of the three forms of transport can not be recognized by digital images making it difficult to classify the three objects used. The problem in computer vision has not been able to duplicate human capability in understanding image information, so that computer can do the same thing as human. Therefore, the purpose of this research is to know the implementation of Convolution Neural Network (CNN) on traditional trasportation based on digital image with the help of library Keras in software R Studio. Prediction of each image data classification Andong, Pedati, and Becak have accurate and correct accuracy. In the predicted class results show the accuracy of the results of the test data is entirely correct. The result can be said that the use of convolutional neural network (CNN) method using Hard can be implemented to the traditional transportation image Andong, Pedati, and Becak.

Keywords : *Convolutional Neural Network, Library Keras, Traditional Transportation Image.*

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Transportasi merupakan urat nadi Pembangunan Nasional yang sangat dibutuhkan dalam kehidupan sehari-hari sebagai penunjang dalam memenuhi kebutuhan konsumsi, produksi, dan distribusi. Didukung dengan kemudahan dalam mengakses transportasi, baik transportasi tradisional maupun modern. Dalam penggunaannya transportasi tradisional lebih menggunakan tenaga manusia dan hewan, sedangkan transportasi modern lebih menggunakan bahan bakar. Perkembangan akan teknologi juga berdampak bagi kemajuan alat transportasi yang digunakan oleh masyarakat. Berbagai alat transportasi banyak menjamur diberbagai kota di Indonesia. Berbagai sarana transportasi seperti Bus, Taxi, Andong, becak, pedati atau gerobak sapi sampai sepeda banyak digunakan oleh masyarakat untuk melakukan pergerakan atau perpindahan. Transportasi tradisional yang masih menjadi sarana bagi masyarakat dari dulu hingga saat ini adalah Andong/delman, Becak dan Pedati/Gerobak Sapi.

Andong, becak dan pedati merupakan transportasi tradisional yang biasanya digunakan masyarakat untuk mengelilingi tempat wisata, pasar, dan lain-lain. Dalam mengenali bentuk serta jenis andong, becak dan pedati, kita bisa mempelajari dari gambar yang ada di internet. Pengenalan ini dilakukan agar masyarakat tahu bahwa bentuk dan ciri khas dari transportasi tradisional seperti Andong, Becak, dan Pedati merupakan ciri khas kebudayaan dari bangsa Indonesia tanpa merubah bentuk dan memodifikasi jenis dari transportasi tersebut. Pada saat ini ada beberapa transportasi tradisional yang telah dimodifikasi sehingga menghilangkan ciri khas dari transportasi tradisional misalnya Bentor atau sering disebut dengan becak motor yang merupakan modifikasi dari becak dengan menggunakan tenaga mesin sehingga menimbulkan permasalahan transportasi. Modifikasi bentor membuat bingung masyarakat akan penggunaan SIM dan Tanda Nomor Kendaraan Bermotor (TNKB) pada Bentor. Kebanyakan bentor tidak

menggunakan Tanda Nomor Kendara Bermotor (TNKB) sedangkan penggunaan motor harus menggunakan nomor kendaraan sesuai dengan peraturan dalam berkendara. Sehingga dengan pengenalan ini pemerintah dapat membuat kebijakan terhadap pengguna bentor dan penggunaan SIM serta nomor kendaraan agar ciri khas dari becak tidak hilang sebagai ciri khas transportasi tradisional di Indonesia. Banyak cara dalam pengenalan yang bisa dilakukan dengan menggunakan klasifikasi objek citra digital dari transportasi tradisional. Problem dalam visi komputer belum mampu menduplikasi kemampuan manusia dalam memahami informasi citra, agar komputer dapat melakukan hal yang sama selayaknya manusia.

Proses *feature engineering* penggunaannya sangat terbatas hanya berlaku pada dataset tertentu tanpa kemampuan generalisasi pada jenis citra. Hal ini dikarenakan ada perbedaan antar citra lainnya seperti perbedaan sudut pandang, perbedaan skala, perbedaan kondisi pencahayaan, deformasi objek, dan lainnya. Salah satu pendekatan yang digunakan adalah Jaringan Syaraf Tiruan (JST) yang terinspirasi dari syaraf pada manusia. Konsep ini yang kemudian dikembangkan dengan menggunakan *Deep Learning*. (Suartika, 2016)

Deep Learning telah menjadi suatu topik yang hangat diperbincangkan di dunia *Machine Learning* karena kapabilitasnya yang signifikan dalam memodelkan berbagai data kompleks seperti citra dan suara. Metode *Deep Learning* yang memiliki hasil paling signifikan didalam pengenalan citra yaitu *Convolutional Neural Network* (CNN). Karena metode CNN berusaha meniru cara pengenalan citra pada visual cortex yang sama dengan manusia sehingga mampu mengolah informasi yang sama.

Di dalam penelitian ini metode CNN yang digunakan untuk mengklasifikasikan objek citra andong, becak dan pedati. Difokuskan pada pembuatan model dalam melakukan ekstraksi fitur dan klasifikasi objek penelitian. Model yang digunakan 3 kelas yaitu membedakan transportasi tradisional dengan kategori 1 Andong, kategori 2 Becak dan kategori 3 Pedati, serta menentukan pengaruh jumlah data latih dan uji dalam metode CNN.

Sehingga dilakukan penelitian mengenai implementasi *Convolutional Neural Network* pada transportasi tradisional berbasis citra digital.

1.2 Rumusan Masalah

Pada penelitian ini, permasalahan yang diselesaikan adalah sebagai berikut:

1. Bagaimana *Convolutional Neural Network* dapat diimplementasikan terhadap pengenalan transportasi tradisional andong, becak dan pedati?
2. Bagaimana menentukan model dan jumlah filter yang digunakan dalam *convolutional neural network* untuk pengenalan transportasi tradisional andong, becak dan pedati?
3. Bagaimana dampak pengurangan jumlah data latih pada data *test* terhadap akurasi dari CNN?

1.3 Batasan Masalah

Permasalahan yang dibahas pada penelitian ini memiliki ruang lingkup yang luas, sehingga diberikan batasan-batasan sebagai berikut:

1. Penelitian ini berfokus pada pengembangan model klasifikasi objek transportasi tradisional andong, becak dan pedati.
2. Penelitian ini tidak membahas tentang kelayakan transportasi.
3. Data latih dan uji merupakan gambar transportasi tradisional andong, becak, dan pedati masing-masing gambar sebanyak 45 citra yang kumpulan melalui *google*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini sebagai berikut:

1. Mengetahui implementasi *convolution neural network* (CNN) pada transportasi tradisional berbasis citra digital.
2. Mengetahui pemodelan dan jumlah filter yang digunakan dalam *convolutional neural network* untuk pengenalan transportasi tradisional andong, becak dan pedati.

3. Mengetahui pengaruh jumlah data *test* terhadap akurasi dari *convolution neural network* (CNN).

1.5 Manfaat Penelitian

Manfaat dari penelitian ini yang diharapkan antara lain:

1. Mampu mengetahui implementasi *convolution neural network* (CNN) pada pengenalan transportasi tradisional berbasis citra digital.
2. Mampu mengetahui pemodelan dan jumlah filter yang digunakan dalam *convolutional neural network* untuk pengenalan transportasi tradisional andong, becak dan pedati.
3. Mampu mengetahui pengaruh jumlah data *test* terhadap akurasi dari *convolution neural network* (CNN).

1.6 Sistemasi Penulisan

Tugas akhir ini berisi 6 (enam) bab dengan rincian setiap bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang uraian singkat dari penelitian seperti latar belakang, rumusan masalah, batasan masalah, keaslian penelitian, tujuan penelitian, manfaat penelitian, metode penelitian dan sistemasi penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini memaparkan penelitian-penelitian terdahulu yang berhubungan dengan permasalahan yang diteliti dan menjadi acuan konseptual.

BAB III LANDASAN TEORI

Pada bab ini akan dibahas tentang teori-teori dan konsep yang berhubungan dengan penelitian yang dilakukan dan mendukung dalam pemecahan masalahnya. Selain itu, bab ini juga memuat teori-teori dalam pelaksanaan pengumpulan dan pengolahan data serta saat melakukan penganalisaan.

BAB IV METODOLOGI PENELITIAN

Bab ini memaparkan populasi dan sampel, variabel penelitian, jenis dan sumber data, metode analisis data, dan tahapan penelitian.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai analisa yang dilakukan terhadap hasil pengumpulan, pengolahan dan analisa data yang diperoleh dari hasil penelitian.

BAB VI PENUTUP

Pada bab ini akan dibahas mengenai kesimpulan yang diperoleh dari hasil penelitian dan analisa data yang telah dilakukan serta saran-saran yang dapat diterapkan dari hasil pengolahan data yang dapat menjadi masukan yang berguna kedepannya.

BAB II

TINJAUAN PUSTAKA

Pentingnya penelitian terdahulu untuk membantu penulis sebagai kajian keterkaitan dengan penelitian yang saat ini dilakukan, agar terhindar dari duplikasi yang dilakukan oleh penelitian-penelitian sebelumnya dengan yang dilakukan penulis. Tujuan dari bab ini untuk membantu penulis menunjukkan penelitian yang dilakukan sangat bermanfaat dan penting sehingga dapat diketahui kontribusi penulis dalam ilmu pengetahuan. Berikut ini beberapa penelitian yang telah dilakukan sebelum-sebelumnya berkaitan dengan data dan metode yang berkaitan dengan penelitian penulis. Berikut beberapa jurnal dan penelitian yang di jadikan sebagai acuan penulis sebagai berikut.

Hanni Suprhitani (2017) pada penelitian pemertahanan transportasi tradisional di Era Modern terhadap becak kayuh di Maliobor, Yogyakarta. Pada penelitian ini menggunakan teori identitas sosial secara pendekatan interview atau komunikasi langsung dengan komunitas becak kayuh di Malioboro, Yogyakarta berkaitan dengan dampak dari modernisasi. Dampak modernisasi terhadap becak kayuh adalah berkurangnya jumlah penumpang, semakin sedikitnya orang yang memanfaatkan becak kayuh untuk mengangkut barang, para langganan becak kayuh pun telah banyak beralih ke transportasi modern.

I wayan S, Arya Y, dan Rully S (2016) pada penilitan Klasifikasi Citra Menggunakan *Convolutional Nerual Network* (CNN) pada *Caltech 101* melakukan klasifikasi citra menggunakan *feedforward* dan selanjutnya dilakukan pembelajaran metode *backpropagation*. Penelitian menggunakan 5 kategori unggas, yaitu : *Emu*, *Flamingo*, *Ibis*, *Pigeon* dan *Rooster* yang terdiri dari 150 citra. Selain itu ketegori lainnya adalah *Cougar*, *Crocodile* dan *Face*. Klasifikasi citra yang dilakukan menghasilkan presentase kebenaran antara 20% sampai 50%.

Andy Harkoyo, Sholeh Hadi Pramono, M. Aziz Muslim (2016) pada penelitian Pengenalan Karakter Plat Motor Kendaraan Bermotor Berbasis Citra dengan Menggunakan Metode *Canny* dan Algoritma *Backpropagation* membahas tentang kelemahan sistem pendataan kendaraan umum yang masih menggunakan

cara manual. Penggunaan cara manual menyebabkan kesalahan dalam pencatatan kendaraan dan waktu yang dibutuhkan dalam proses akan sangat lama. Dalam mengatasi hal ini dapat menggunakan metode pengenalan Tanda Nomor Kendara Bermotor (TNKB) yang melekat pada kendaraan. Pengenalan dilakukan dengan identifikasi plat nomor kendaraan pada system yang dilakukan didalam penelitian. Filter canny digunakan untuk mendapatkan citra yang baik dalam pengambilan citra karakter. Karakter disampling dengan ukuran 12X7 untuk dikonversi kedalam bentuk biner sebagai input neuron dalam jaringan syaraf tiruan arsitektur Multi Layer Perceptron yang digunakan menggunakan 3 Layer dengan jumlah masing masing node 84, 50, 36. Jaringan syaraf tiruan dilatih dengan algoritma Backpropagation dengan parameter *learning rate* 0.3 dan momentum 0.9. Proses training akan dihentikan apabila iterasi mencapai nilai maksimal 10.000 atau MSE (*Mean Square Error*) 0.0001. Dari hasil pengujian yan dilakukan sistem mampu mengenali karakter angka 100% sedangkan huruf 86,87% jadi kehandalan sistem mengenali karakter secara keseluruhan adalah 94,29%.

Rismiyati (2016) pada penelitian Implementasi *Convolutional Neural Network* (CNN) untuk Sortasi Mutu Salak Ekpor Berbasis Citra Digital, proses yang dilakukan selama penelitian adalah pengumpulan data, proses *preprocessing* klasifikasi dan pengujian. *Preprocessing* dengan menggunakan metode *blurring* dan *thresholding* metode Otsu adalah telah mampu menghasilkan citra yang berisi *region of interest* (ROI) dengan *noise* yang minimal. Ukuran yang tepat untuk filter metode *blurring* adalah 25x25. Nilai yang lebih kecil dari ukuran tersebut tidak mampu mengurangi *noise* pada latar belakang citra. Klasifikasi dilakukan dengan *Convolutional Neural Network* (CNN), dimana untuk mendapatkan akurasi yang terbaik parameter-parameter yang ada harus diujikan. Terdapat dua model yang digunakan yaitu model dua kelas dan empat kelas. Percobaan juga dilakukan menggunakan ukuran data latih yang beragam untuk melihat pengaruh jumlah data latih terhadap akurasi. Pengujian dilakukan dengan metode *stratified cross validation* untuk mengukur akurasi berdasarkan *cconfusion matrix*. Akurasi terbaik untuk model dua kelas adalah 81,45% dengan menggunakan satu lapisan konvolusi, limas belas filter dengan ukuran masing-

masing 3x3x3, dan jumlah *neuron* pada lapisan tersembunyi sebanyak 100. Sementara itu, klasifikasi empat kelas mendapatkan akurasi terbaik sebesar 67,4% dengan 18 filter berukuran 3x3x3 dan jumlah *neuron* 150.

Penelitian dari Kefin P D (2017) tentang Implementasi *Deep Learning* Menggunakan *Convolutional Neural Network* (CNN) Untuk Klasifikasi Citra Candi Berbasis GPU memiliki hasil pengujian yang optimal terhadap citra candi menunjukkan akurasi sebesar 98,99% pada training set dan 85,57% pada *test set* dengan waktu pelatihan mencapai 389,14 detik. Sehingga dapat disimpulkan bahwa teknik *Deep Learning* dengan *Convolutional Neural Network* (CNN) mampu melakukan klasifikasi citra candi dengan sangat baik.

Barners B A C (2014) pada penelitiannya tentang Aplikasi Jaringan Syaraf Tiruan Untuk Penentuan Kelas Kemasakan Buah Tomat Berdasarkan Warna Citra menggunakan parameter warna dalam format RGB (*Red, Green, Blue*). Data RGB tomat yang diperoleh dari pengambilan citra tomat yang disimpan dalam 3 variasi suhu 6,15 dan 28 derajat *Celcius*. Setelah dilakukan pelatihan jaringan *Backpropagation* dengan tingkat pembelajaran (*learning rate*) 0,3 diperoleh arsitektur jaringan dengan 2 layer tersembunyi terdiri dari 8 *neuron* pada layer pertama dan 10 *neuron* pada layer yang kedua dan memiliki 3 bias. Nilai MSE yang didapatkan adalah 0,00422. Hasil penelitian ini menunjukkan bahwa instrumen yang dirancang dapat menentukan kelas kemasaka tomat dengan tingkat keakuratan 93%.

Pada **Tabel 2.1** menjadi perbandingan penelitian sebelumnya dan penelitian yang akan dilakukan oleh peneliti.

Tabel 2.1 Perbandingan penelitian sebelumnya dengan penelitian yang penulis lakukan

Judul Penelitian	Penelitian, tahun	Metode	Tujuan
Pemertahanan Transportasi Tradisional di Era	Hasni Suprihatin, 2017	Teori Identitas Sosial	Untuk mengetahui dampak

Modern Terhadap Komunitas Becak Kayuh di Malioboro, Yogyakarta			modernisasi terhadap komunitas becak kayuh di Malioboro, Yogyakarta
Klasifikasi Citra Menggunakan <i>Convolutional Neural Network</i> (CNN) pada Caltech 101	I wayan S, Arya Y, dan Rully S., 2016	<i>Convulotional Neural Network</i> (CNN)	Membuktikan bahwa klasifikasi menggunakan metode CNN relatif handal terhadap perubahan parameter yang dilakukan. Dengan menggunakan data <i>training</i> yang baik dan optimal, maka subset dari data <i>training</i> tersebut juga akan menghasilkan klasifikasi yang baik.
Pengenalan Karakter Plat Motor Kendaraan Bermotor Berbasis Citra	Andy Harkoyo, Sholeh Hadi Pramono, M. Aziz Muslim, 2016	<i>Canny</i> dan <i>Backpropagation</i>	Untuk mengatasi kelemahan sistem pendataan kendaraan umum yang masih

dengan Menggunakan Metode <i>Canny</i> dan Algoritma <i>Backpropagation</i>			menggunakan cara manual
Implementasi <i>Convolutional Neural Network</i> Untuk Sortasi Mutu Salak Ekspor Berbasis Citra Digital	Rismiyati, 2016	<i>Convolutional Neural Network</i> (CNN)	Menerapkan <i>Convolutional Neural Network</i> (CNN) untuk penentuab kelayakan buah salak untuk tujuan ekspor berdasarkan citra salak
Implementasi <i>Deep Learning</i> Menggunakan <i>Convolutional Neural Network</i> Untuk Klasifikasi Citra Candi Berbasis GPU	Kefin P D, 2017	<i>Convolutional Neural Network</i> (CNN)	Melakukan klasifikasi citra digital candi menggunakan metode CNN
Aplikasi Jaringan Syaraf Tiruan Untuk Penentuan Kelas Kemasakan Buah Tomat Berdsarkan Warna Citra	Barner Bogabriel A C, 2014	ANN, <i>Backpropagation</i>	Mengembakan instrumen sederhana untuk menentukan kelas kemasakan tomat berdasarkan warna citra.

			Parameter warna yang digunakan dalam format RGB.
--	--	--	--

BAB III

LANDASAN TEORI

3.1 Transportasi

Transportasi adalah perpindahan orang atau barang dari satu tempat ketempat yang lainnya atau tempat asal ke tempat tujuan dengan menggunakan wahana digerakkan manusia, hewan atau mesin (Zulfiar Sani,2010:2). Tujuan orang menggunakan alat transportasi adalah agar lebih cepat dan lebih mudah dalam perpindahan orang atau barang dari tempat asal ke tempat tujuannya. Fungsi transportasi ini tidak hanya dilihat secara perorangan tapi juga dilihat dari kepentingan masyarakat luas. (Eva Dewi Purita, 2013)

3.1.1 Fungsi dan Manfaat Transportasi

Fungsi dan manfaat transportasi diklasifikasi menjadi beberapa bagian penting. Transportasi memiliki fungsi yang terbagi menjadi dua yaitu melancarkan arus barang dan manusia serta menunjang perkembangan pembangunan (*the promoting sector*). (Radar Planologi, 2015) Sedangkan manfaat transportasi menjadi tiga klasifikasi:

a. **Manfaat Ekonomi**

Kegiatan ekonomi bertujuan memenuhi kebutuhan manusia dengan menciptakan manfaat. Transportasi adalah salah satu jenis kegiatan yang menyangkut peningkatan kebutuhan manusia dengan mengubah letak geografis barang dan orang sehingga akan menimbulkan adanya transaksi.

b. **Manfaat Sosial**

Transportasi menyediakan berbagai kemudahan, diantaranya:

1. Pelayanan untuk perorangan atau kelompok
2. Pertukaran atau penyampaian informasi
3. Perjalanan untuk bersantai
4. Memendekkan jarak
5. Memencarkan penduduk

c. Manfaat politik

Transportasi menciptakan persatuan, pelayanan lebih luas, keamanan negara, mengatasi bencana, dan lain-lain.

d. Manfaat kewilayahan

Memenuhi kebutuhan penduduk di kota, desa, atau pedalaman terutama yang berkaitan dengan sirkus dan mobilitas serta perangsangan pembangunan.

3.1.2 Jenis-Jenis Transportasi dan Alat Transportasi

Jenis-jenis transportasi terbagi menjadi 3 yaitu:

1. Transportasi Darat

Alat transportasi darat dipilih berdasarkan faktor-faktor seperti jenis dan spesifikasi kendaraan, jarak perjalanan, tujuan perjalanan, ketersediaan alat transportasi, ukuran kota dan kepadatan permukiman, faktor sosial-ekonomi,. Contoh model transportasi darat adalah kendaraan bermotor, kereta api, gerobak yang ditarik oleh hewan (kuda, sapi, kerbau), atau manusia.

2. Transportasi Air

Transportasi air (sungai, danau, dan laut) sama seperti halnya transportasi darat. Contoh dari transportasi air seperti kapal, tongkang, perahu, dan rakit.

3. Transportasi Udara

Alat transportasi udara dapat menjangkau tempat-tempat yang tidak dapat ditempuh dengan alat transportasi darat dan air, disamping mampu bergerak lebih cepat dan mempunyai lintasan yang lurus, serta praktis bebas hambatan. Contohnya pesawat terbang, helicopter, balon udara, dan lain-lain.

3.1.3 Transportasi Tradisional dan Modern

1. Transportasi Tradisional

a. Becak

Becak yang berasal dari bahasa Hokkien yang berarti “kereta kuda” adalah suatu alat transportasi darat beroda tiga yang banyak ditemukan di Indonesia dan juga di Asia. Kapasitas normal becak umumnya adalah dua orang penumpang dan seorang pengemudi. (Jamziah,2016)

b. Delman/Andong

Delaman adalah transportasi tradisional yang beroda dua, tiga atau empat yang tidak menggunakan mesin tetapi menggunakan kuda sebagai penggantinya, variasi alat transportasi yang menggunakan kuda antara lain adalah kereta perang, kereta kencana, dan kereta kuda. Nama kendaraan ini berasal dari nama penemunya, yaitu Charles Theodore Delman, seorang litografer dan insinyur pada masa Hindia Belanda. Orang Belanda sendiri menyebutnya dengan nama dos-à-dos (punggung pada punggung, arti harfiah bahasa Perancis), yaitu sejenis kereta yang posisi duduk penumpangnya saling memunggungi. Istilah dos-à-dos ini kemudian oleh pribumi Batavia disingkat lagi menjadi “Sado”.

c. Pedati

Pedati adalah kendaraan atau alat transportasi yang memiliki dua atau empat roda yang digunakan sebagai sarana transportasi mengangkut orang, maupun mengangkut barang-barang. Pedati dapat ditarik oleh hewan seperti kuda, sapi, kambing, dan zebu. Pedati umumnya ada di wilayah pedesaan Indonesia. (Ok Nusantara, 2017)

2. Transportasi Modern

a. Kereta Api

Kereta api adalah bentuk transportasi rel yang terdiri dari serangkaian kendaraan yang didorong sepanjang jalur kereta api untuk mengangkut kargo atau penumpang. Gaya gerak disediakan oleh lokomotif yang terpisah atau motor individu dalam beberapa unit. Meskipun propulsi historis mesin uap mendominasi, bentuk-bentuk modern yang paling umum adalah mesin diesel dan listrik lokomotif, yang disediakan oleh kabel overhead atau rel tambahan. Sumber energi lain termasuk kuda, tali atau kawat, gravitasi, pneumatik, baterai, dan turbin gas. Rel kereta api biasanya terdiri dari dua, tiga atau empat rel, dengan sejumlah monorel dan *guideways maglev* dalam campuran. Kata 'train' berasal dari bahasa Perancis Tua *trahiner*, dari bahasa Latin *trahere* 'tarik, menarik'. (Jamziah,2016)

b. Kapal Laut

Kapal Laut adalah alat transportasi yang berfungsi untuk mengangkut penumpang dan barang di laut atau sungai, seperti halnya sampan atau perahu yang lebih kecil.

c. Pesawat Terbang

Pesawat terbang adalah alat transportasi udara berupa pesawat udara yang lebih berat dari udara, *bersayap tetap*, dan dapat terbang dengan tenaga sendiri. Secara umum istilah pesawat terbang sering juga disebut dengan pesawat udara atau kapal terbang atau cukup pesawat dengan tujuan pendefinisian yang sama sebagai kendaraan yang mampu terbang di atmosfer atau udara.

d. Mobil

Mobil adalah alat transportasi darat yang digerakkan oleh tenaga mesin, beroda empat atau lebih (jumlahnya selalu genap), umumnya menggunakan bahan bakar minyak (bensin atau solar)

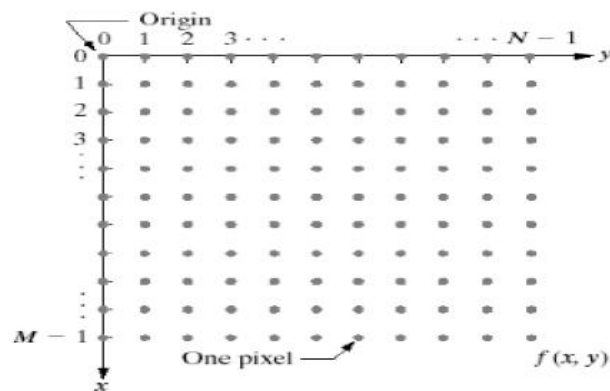
untuk menghidupkan mesinnya. Mobil kependekan dari *otomobil* yang berasal dari bahasa Yunani 'autos' (sendiri) dan Latin 'movére' (bergerak).

3.2 Citra

Citra bisa didefinisikan sebagai suatu fungsi dua dimensi, $f(x,y)$, dimana x dan y adalah koordinat spasial, dan $f(x,y)$ adalah nilai pada koordinat (x,y) yang sering disebut intensitas. (Gonzales dan Wood, 2008). Citra digital adalah citra $f(x,y)$ yang telah digitalisasi baik dari segi koordinat area maupun pada nilai intensitasnya. Sebuah citra digital terdiri dari sejumlah elemen yang disebut *picture element* atau *pixel*.

3.2.1 Representasi Citra Digital

Setiap nilai *pixel* diwakili oleh dua buah bilangan bulat untuk menunjukkan lokasi dalam bidang citra. Sebagai contoh, koordinat $(0,0)$ digunakan untuk pojok kiri atas citra dan koordinat $(m-1, n-1)$ digunakan untuk pojok kanan bawah dalam citra yang berukuran $m \times n$ *pixel*. Selain itu, citra digital juga bisa direpresentasikan dalam bentuk matriks seperti pada gambar 3.1 berikut.



Gambar 3.1 Representasi Citra Digital Dalam Bentuk Matriks

Gambar 3.1 menunjukkan citra dipandang sebagai dua dimensi dengan jumlah kolom N dan jumlah baris M . Masing-masing dari nilai intensitas *pixel* dianggap sebagai elemen matriks dua dimensi. Contoh dari

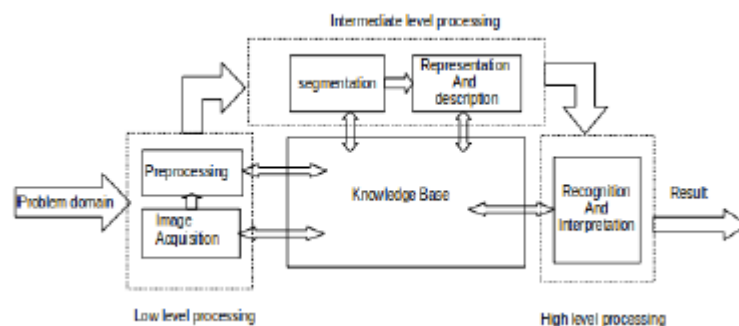
citra digital yang memiliki ukuran 200x200 *pixel* dengan nilai intensitas *pixel* pada koordinat $(x=100, y=100) = 35$ sebagai berikut.



Gambar 3.2 Contoh Citra Digital

3.2.2 Pengolahan Citra Digital

Citra merupakan salah satu komponen multimedia yang memegang peranan penting dalam menyajikan informasi visual. Pengolahan citra (*image processing*) bertujuan untuk memperbaiki kualitas citra sesuai dengan kebutuhan pengguna. Teknik-teknik pengolahan citra mampu menghasilkan citra lain atau menghasilkan fitur dari citra input. Pada gambar berikut menunjukkan langkah-langkah dalam pengolahan citra digital.



Gambar 3.3 Langkah-Langkah Dalam Pemrosesan Citra Digital

Dari **Gambar 3.3** menunjukkan beberapa langkah dalam pengolahan citra digital yang terdiri dari:

1. **Image acquisition atau Akurasi Citra** langkah pertama pada pengolahan citra adalah akurasi citra (*image aquisition*) yang merupakan proses untuk menangkap atau mengambil citra yang diperlukan dengan menggunakan sensor pencitraan seperti kamera, scanner, dan lain-lain.
2. **Image Preprocessing**, setelah citra diperoleh langkah kedua yang umum dilakukan adalah *image preprocessing*. Salah satu bentuk *preprocessing* adalah *image enhancement* yaitu teknik untuk menjelaskan detail pada citra atau fitur yang diinginkan dari sebuah citra. Contoh dari *image enhancement* adalah memperbaiki kontras agar citra terlihat lebih baik. Dalam penggunaannya, teknik yang dipakai dalam *enhancement* tergantung pada kebutuhan pengguna. Sebagai contoh, citra untuk kebutuhan visual menggunakan teknik *enhancement* yang berbeda jika dibandingkan dengan citra untuk pengenalan wajah.
3. **Segmentasi**, membagi sebuah citra menjadi bagian-bagian pembentuknya. Segementasi dilakukan sampai objek yang diinginkan dalam suatu aplikasi terpisah dari objek aslinya. Segmentasi yang akurat mampu meningkatkan tingkat kesuksesan dari sebuah sistem pengenalan citra.
4. **Representasi dan Deskripsi**, proses representasi dan deskripsi adalah proses yang dilakukan setelah proses segmentasi. *Output* dari proses segmentasi biasanya berupa data pixel kasar, baik untuk menyatakan batas dari suatu region atau data pixel pembentuk sebuah region. Proses selanjutnya adalah mengubah data pixel tersebut menjadi bentuk yang bisa disimpan dan diproses oleh komputer jika dibutuhkan. Hal pertama yang harus diputuskan adalah bagaimana representasi dari data tersebut. Dengan kata lain, representasi berarti bagaimana menyatakan data

pixel tersebut dalam bentuk data yang mampu diolah oleh komputer. Sedangkan proses deskripsi atau seleksi fitur merupakan proses untuk mengekstrak atribut-atribut yang menghasilkan informasi kuantitatif yang diinginkan atau yang merupakan fitur untuk membedakan citra dari satu kelas ke kelas yang lain.

5. **Recognition atau Pengenalan**, merupakan proses untuk memberikan label pada objek sesuai dengan fitur-fitur yang dimiliki oleh objek tersebut. Metode pengenalan objek dibagi menjadi beberapa kategori, antara lain adalah metode *statistical classifier*. Metode *statistical classifier* data latih digunakan untuk melakukan pembelajaran terhadap model yang digunakan untuk klasifikasi. Metode ini terbagi menjadi dua yaitu *generative learning* dan *discriminative learning*. Dalam metode *generative learning*, dataset digunakan untuk membuat model terhadap kelas-kelas yang ada. Contoh *generative learning* adalah *Naive Bayes Classifier*. Sementara itu dalam *discriminative learning*, tidak diperlukan perlakuan pemodelan terlebih dahulu terhadap kelas-kelas yang ada. Akan tetapi dilakukan training secara langsung untuk menentukan bobot yang ideal bagi masing-masing fitur. Contoh untuk *discriminative learning* adalah *Neural Network*.

3.3 Preprocessing Citra

Preprocessing citra dilakukan untuk menyiapkan citra untuk diproses lebih lanjut, baik untuk kebutuhan ekstraksi fitur maupun kebutuhan klasifikasi. Beberapa teknik *preprocessing* yang umum digunakan antara lain penghilang *noise* dengan filter spasial, segmentasi, *cropping* atau pemotongan *region of interest* dan perubahan ukuran citra.

3.3.1 Penghilang *noise* dengan filter spasial

Noise removal atau penghilang *noise* adalah proses mengurangi atau menghilangkan data yang tidak diinginkan. Salah satu metode untuk menghilangkan *noise* adalah dengan melakukan penghalusan citra

(*smoothing*) pada domain spasial. Penghalusan citra dilakukan dengan melakukan konvolusi dengan *smoothing* mask.

Prinsip dasar dalam melakukan *smoothing* adalah menggantikan nilai intensitas pada masing-masing pixel dengan nilai rata-rata intensitas pixel yang berada pada jangkauan filter mask. Proses ini menghasilkan citra dengan transisi yang lebih halus.

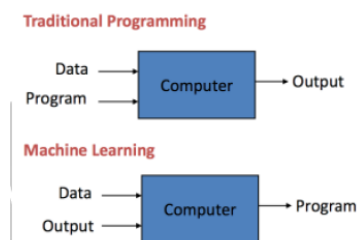
Salah satu mask yang digunakan untuk melakukan penghalusan citra ditunjukkan pada gambar berikut.

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Gambar 3.4 Filter dengan Ukuran 3x3

3.4 *Machine Learning*

Machine Learning adalah serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara merepresentasikan data-data tersebut dengan algoritma pembelajaran. *Machine Learning* dapat membuat komputer memprogram diri mereka sendiri. Jika pemrograman adalah pekerjaan untuk membuat otomatis, maka *Machine Learning* mengotomatisasi proses otomatis. Berikut gambaran umum *Machine Learning* dibandingkan dengan pemrograman secara tradisional.



Gambar 3.5 Perbandingan Pemrograman Tradisional Dengan *Machine Learning*

Dari gambar diatas dapat dilihat bahwa pemrograman secara tradisional data dan programan dijalankan di komputer untuk menghasilkan *output*. Sedangkan pada *Machine Learning* data dan *output* dijalankan di komputer untuk membuat sebuah program.

Ada banyak algoritma *Machine Learning* yang dikembangkan setiap tahunnya. Setiap algoritma pembelajaran mesin memiliki tiga komponen penting antara lain:

- a. **Representasi:** bagaimana merepresentasikan pengetahuan. Contohnya termasuk *Decision tree*, *Neural Network*, *Support Vector Machine* dan lain-lain.
- b. **Evaluasi:** cara mengevaluasi prediksi dan hipotesis. Contohnya meliputi *Mean Squared Error*, *Cost function* dan lain-lain.
- c. **Optimasi:** cara program dari model dihasilkan dan proses pencarian parameter terbaik. Misalnya *Convex Optimization* dan *Gradient Descent*

Selain dari algoritma pembelajaran ada empat jenis cara pembelajaran pada *Machine Learning*, yakni:

- a. *Supervised Learning*: Data pembelajaran mencakup keluaran yang sudah ditentukan.
- b. *Unsupervised Learning*: Data pembelajaran tidak mencakup keluaran yang ditentukan.
- c. *Semi-supervised Learning*: Data pembelajaran mencakup beberapa keluaran yang ditentukan.
- d. *Reinforcement Learning*: Pemberian hadiah dari setiap serangkaian tindakanyang dilakukan.

3.5 *Deep Learning*

Deep Learning merupakan salah satu bidang dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *Deep Learning* memberikan arsitektur yang sangat kuat untuk *Supervised Learning*. Dengan menambahkan lebih banyak lapisan

maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik.

Pada *Machine Learning* terdapat teknik untuk menggunakan ekstrak fitur dari data pelatihan dan algoritma pembelajaran khusus mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi. Aplikasi konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditanggguhkan pada algoritma *Machine Learning* yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip ini terus berkembang hingga *Deep Learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *Computer vision*, *Speech recognition*, dan *Natural Language Processing*.

Feature Engineering adalah salah satu fitur dari *Deep Learning* untuk mengekstrak pola yang berguna dari data akan memudahkan model untuk membedakan kelas. *Feature Engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga. Algoritma yang digunakan pada *Feature Engineering* dapat menemukan pola umum yang penting untuk membedakan antara kelas.

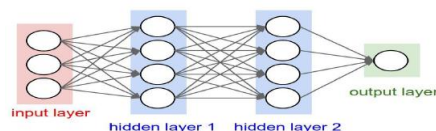
Pada tahun 2012 muncul terobosan yang baru lagi pada *deep learning*. *Convolutional Neural Network* dengan arsitektur tertentu yang dipadukan dengan berbagai trik *drop out regularization*, pemanfaatan *Rectified Linear Unit* (ReLU) sebagai fungsi aktivasi, data *augmentation* mampu mencapai terobosan pada *large scale image classification (ImageNet)* yang memiliki 1000 kategori objek dan lebih dari 1 juta gambar. Model yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia *Deep Learning* penggunaan GPU sudah sangatlah umum. Dalam melakukan pengolahan citra ada beberapa metode yang digunakan selain *convolutional neural network* yaitu *Background Subtraction*. Perbandingan kelebihan dan kekurangan metode CNN dan *Background Subtraction* seperti pada tabel berikut.

Tabel 3.1 Perbandingan Metode Convolutional Neural Network dan Background Subtraction

	Convolutional Neural Network	Backgraound Subtraction
Kelebihan	Data <i>augmentation</i> mampu mencapai terobosan pada <i>large scale image classification</i> (ImageNet) yang memiliki 1000 kategori objek dan lebih dari 1 juta gambar. Serta mampu mengklasifikasi gambar melalui video.	untuk mendeteksi objek bergerak pada video dari kamera statis (<i>stationary camera</i>)
Kekurangan	Proses pelatihan model yang lama	Bayangan yang ditimbulkan akibat cahaya yang mengenai oyek penelitian akan terdeteksi sebagai obyek di luar penelitian saat dilakukan pengurangan citra dan hanya bisa dilakukan dengan video

3.6 Deep Feedforward Networks

Bisa dikatakan bahwa *Deep Learning Feedforward Networks* merupakan ekstensi dari model standar jaringan syaraf tiruan yang terinspirasi dari cara bekerja neuron pada otak manusia. Disebut juga dengan *Feedforward* karena cara kerjanya dimana sebuah input x mengalir maju dengan mediasi komputasi atau fungsi f untuk menghasilkan *output* y . Gambar 3.6 merupakan gambaran dasar arsitektur jaringan syaraf.

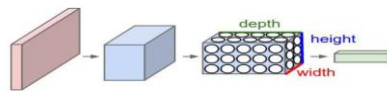


Gambar 3.6 Contoh Jaringan Deep Feedforward

Sumber: cs231n.github.io

3.7 Convolutional Neural Network

Merupakan jaringan syaraf yang dikhususkan untuk memproses data yang memiliki grid. Sebagai contoh adalah berupa citra dua dimensi. Nama konvolusi sendiri merupakan operasi aljabar linear yang mengkalikan matriks dari filter pada citra yang akan di proses. Proses ini disebut lapisan konvolusi dan merupakan salah satu jenis dari banyak lapisan bisa dimiliki dalam satu jaringan. Lapisan konvolusi merupakan lapisan utama yang paling penting untuk digunakan. Jenis lapisan lain yang biasa digunakan adalah *Pooling Layer*, yakni lapisan yang digunakan untuk mengambil nilai maksimal atau nilai rata-rata dari bagian-bagian piksel pada citra. Gambaran umum *Convolutional Neural Network* seperti berikut.



Gambar 3.7 Contoh Jaringan Convolutional Neural Network (CNN)

Sumber: cs231n.github.io

Pada gambar diatas menunjukkan setiap lapisan *input* yang dimasukkan memiliki volume yang berbeda da mewakili dengan kedalaman, tinggi dan lebar. Setiap besaran yang didapatkan tergantung dari hasil filtrasi dari lapisan sebelumnya dan juga banyak filter yang digunakan. Model jaringan seperti ini sudah terbukti sangat ampuh dalam menangani permasalahan klasifikasi citra.

3.8 Operasi Konvolusi

Operasi konvolusi adalah operasi pada dua fungsi argument bernilai nyata (Goodfellow, Bengio, dan Courville, 2016). Operasi ini menerapkan fungsi output sebagai *Feature Map* dari *input* citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. Secara formal operasi konvolusi dapat ditulis dengan rumus berikut.

$$s(t) = (x * w)(t) \quad (3.1)$$

Fungsi $s(t)$ memberikan *output* tunggal berupa *Feature Map*, argument pertama adalah *input* yang merupakan x dan argument kedua w sebagai kernel

atau filter. Jika kita melihat *input* sebagai citra dua dimensi, maka kita bisa mengasumsikan t sebagai pixel dan menggantinya dengan i dan j . Operasi untuk konvolusi ke *input* dengan lebih dari satu dimensi dapat ditulis sebagai berikut.

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n) \quad (3.2)$$

Persamaan diatas adalah perhitungan dasar dalam operasi operasi konvolusi dimana i dan j adalah piksel dari citra. Perhitungannya bersifat komutatif dan muncul saat K sebagai kernel, I sebagai *input* dan kernel yang dapat dibalik relative terhadap input. Sebagai alternatif, operasi konvolusi dapat dilihat sebagai perkalian matriks antaran citra masukan dan kernel dimana keluarannya dapat dihitung dengan *dot product*.

3.9 Arsitektur Jaringan CNN

Pada *neural network*, neuron dikumpulkan dalam lapisan-lapisan (*layers*) yang disebut dengan lapisan neuron (*neuron layers*). Neuron-neuron pada satu lapisan dihubungkan dengan lapisan sebelumnya dan sesudahnya, kecuali pada lapisan *input* dan *output*. Lapisan diantara lapisan biasa disebut dengan lapisan tersembunyi atau *hidden layer*. Perambatan juga bisa dilakukan pada arah sebaliknya tergantung pada algoritma pembelajaran yang dipakai.

Faktor terpenting dalam menentukan sifat suatu neuron adalah pola bobot dan fungsi aktivitas dari neuron tersebut. Pada setiap lapisan yang sama, neuron-neuron memiliki fungsi aktivitas yang sama.

Arsitektur *neural network* biasa dibagi berdasarkan jumlah lapisannya menjadi:

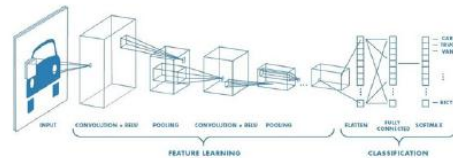
1. Jaringan dengan lapisan tunggal (*Single layer network*).

Jaringan dengan lapisan tunggal adalah sebuah jaringan dengan semua *input* terhubung langsung pada *output* tanpa melewati lapisan tersembunyi.

2. Jaringan dengan banyak lapisan (*multi-layer networks*)

Jaringan dengan lapis banyak memiliki satu atau lebih lapisan tersembunyi di antara lapisan *input* dan *output*.

Arsitektur dari CNN juga terbagi menjadi 2 bagian besar yaitu *Feature Extraction Layer* dan *Fully-Connected Layer (MLP)*. Seperti pada gambar 3.8 berikut.

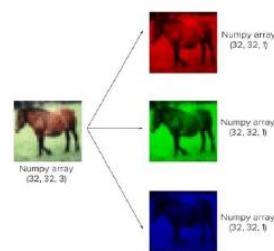


Gambar 3.8 MLP

3.9.1 Feature Extraction Layer

Proses pada bagian ini adalah melakukan “Encoding” dari sebuah gambar menjadi *feature* yang berupa angka-angka yang mempresentasikan gambar tersebut. *Feature extraction layer* terdiri dari dua bagian, *Convolutional Layer* dan *Pooling Layer*. Ada beberapa riset/paper yang tidak menggunakan *pooling*.

1. Convolutional Layer (Conv. Layer)

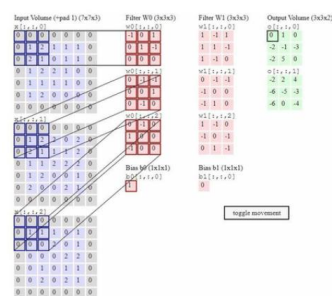


Gambar 3.9 Convolutional Layer

Gambar 3.9 menunjukkan RGB (*Red, Green, Blue*) gambar berukuran 32×32 *pixel* yang sebenarnya adalah multidimensional array dengan ukuran 32×32 *pixel* (3 adalah jumlah *channel*). *Convolutional layer* terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixel*). Sebagai contoh , *layer* pertama pada *feature extraction*

layer adalah *conv. layer* dengan ukuran $5 \times 5 \times 3$. Panjang 5 *pixel*, tinggi 5 *pixel*, dan tebal/jumlah 3 buah sesuai dengan *channel* dari gambar tersebut.

Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*. proses dari *feature map* seperti pada gambar berikut.



Gambar 3.10 Feature Map

2. Stride

Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah 1, maka *conv. filter* akan bergeser sebanyak 1 *pixel* secara *horizontal* lalu *vertical*. Pada ilustrasi diatas, *stride* yang digunakan adalah 2. Semakin kecil *stride* maka akan semakin detail informasi yang kita dapatkan dari sebuah *input*, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar.

3. Padding

Padding atau *zero padding* adalah parameter menentukan jumlah *pixel* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari *input*. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *conv. layer* (*feature map*).

Dengan menggunakan *padding*, kita akan dapat mengukur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis. Sehingga kita bisa

menggunakan *conv. layer* yang lebih dalam sehingga lebih banyak *feature* yang berhasil di-extract.

Meningkatkan performa model karena *conv. layer* akan fokus pada informasi yang sebenarnya yaitu yang berada diantara *zero padding* tersebut. Pada ilustrasi diatas, dimensi dari *input* sebenarnya adalah 5x5, jika dilakukan *convolution* dengan filter 3x3 dan *stride* sebesar 2, maka akan didapatkan *feature map* dengan ukuran 2x2. Namun jika ditambahkan *zero padding* sebanyak 1, maka *feature map* yang dihasilkan berukuran 3x3 (lebih banyak informasi yang dihasilkan). Untuk menghitung dimensi dari *feature map* kita bisa gunakan rumus sebagai berikut.

$$Output = \frac{W+N+2P}{s} + 1$$

W = Panjang/Tinggi *Input*

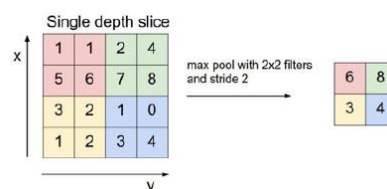
N = Panjang/Tinggi Filter

P = Zero Padding

S = Stride

4. Pooling Layer

Pooling layer biasanya berada setelah *conv. layer*. Pada prinsipnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*. *Pooling* yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. Proses *pooling layer* seperti pada gambar berikut.



Gambar 3.11 Pooling Layer

Tujuan dari penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*downsampling*), sehingga mempercepat

komputasi karena parameter yang harus *diupdate* semakin sedikit dan mengatasi *overfitting* (Samuel Sena/medium.com).

3.9.2 *Fully-Connected Layer (FC Layer)*

Feature map yang dihasilkan dari *feature extraction* masih berbentuk *multidimensional array*, sehingga harus melakukan “*flatten*” atau *reshape feature map* menjadi sebuah *vector* agar bisa digunakan sebagai *input* dari *fully-connected layer*.

Lapisan *Fully-connected* adalah lapisan dimana semua *neuron* aktivitas dari lapisan sebelumnya terhubung semua dengan *neuron* di lapisan selanjutnya seperti halnya jaringan syaraf tiruan bisa. Setiap aktivitas dari lapisan sebelumnya perlu diubah menjadi data satu dimensi sebelum dapat dihubungkan ke semua *neuron* di lapisan *Fully-Connected*.

Lapisan *Fully-Connected* biasanya digunakan pada metode *Multi* lapisan *Perceptron* dan bertujuan untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan antar lapisan *Fully-Connected* dan lapisan konvolusi biasa adalah *neuron* di lapisan konvolusi terhubung hanya ke daerah tertentu pada input. Sementara lapisan *Fully-Connected* memiliki *neuron* yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk *dot*, sehingga fungsinya tidak begitu berbeda. (Kefin Pudi D, 2013)

3.10 *Keras*

Keras adalah open source network library yang di tuliskan dalam *python*. *Keras* mampu berjalan menggunakan MXnet, Deeplearning4j, Tensorflow, Microsoft Cognitive Toolkit atau Theano. *Keras* direncanakan untuk melakukan eksperimen cepat dengan deep neural network, hal ini berfokus pada ukuran minimal, modular dan dapat diperluas kembali. *Keras* dikembangkan sebagai bagian dari upaya penelitian proyek ONEIROs dan penulis serta pengelolanya adalah Francois Chollet. Pada tahun 2017, tim *Google Tensorflow* memutuskan untuk mendukung *Keras* dalam *TensorFlow's core library*.

BAB IV

METODE PENELITIAN

4.1 Populasi dan Sampel Penelitian

Populasi yang digunakan dalam penelitian ini adalah data citra atau gambar transportasi tradisional andong, becak, dan pedati masing-masing sebanyak 40 data citra dengan total data 120 yang terdapat pada situs *website google*, yang diambil dalam halaman *Google Image* dengan *keyword* andong, becak dan pedati.

4.2 Teknik Pengumpulan Data dan Analisis data

Proses pengambilan data dari situs *google* dilakukan dengan menggunakan *fatkun batch image downlaoader* yaitu proses pengumpulan gambar dari *google*. Proses ini dilakukan dengan menggunakan *google crome*. Studi kasus yang dianalisis menggunakan metode *Convolutional Neural Network (CNN) library Keras* menggunakan *software Rstudio*.

4.3 Variabel dan Definisi Operasional Variabel

Varibale yang didgunakan dalam penelitian ini dapat dilihat pada **Tabel 4.1** berikut.

Tabel 4.1 *Definisi Operasional Variabel*

Variabel	Definisi Operasional Variabel
Citra Andong	Citra transportasi andong merupakan transportasi tradisional yang beroda duacatau empat yang tidak menggunakan mesin tetapi menggunakan kuda sebagai penggantinya.
Citra Becak	Citra transportasi becak merupakan transportasi tradisional beroda tiga menggunakan tenaga manusia dalam menjalankannya.

Citra Pedati	Citra transportasi pedati merupakan transportasi tradisional yang memiliki dua dapat ditarik oleh hewan seperti sapi.
--------------	---

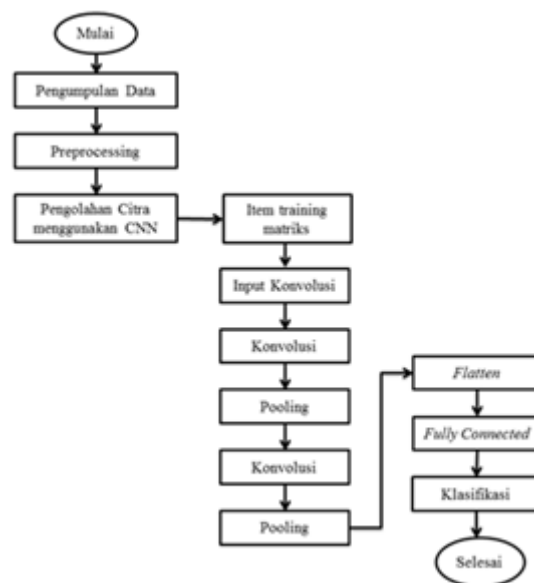
4.4 Metode Analisis Data

Proses analisis dalam penelitian ini menggunakan bantuan *software fatkun batch image downloader* dan *Rstudio*. Ada beberapa metode yang digunakan dalam penelitian ini, diantaranya sebagai berikut:

1. *Image downloader*, digunakan untuk mengumpulkan data citra transportasi andong, pedati dan becak menggunakan *software fatkun batch image downloader*.
2. *Image preprocessing*, digunakan untuk menyiapkan dataset citra yang akan dianalisis
3. *Convolutional Neural Network (CNN) library Keras* adalah metode yang digunakan untuk menentukan model dataset citra dan pengaruh data *test* terhadap akurasi.

4.5 Tahapan Penelitian

Tahapan pada penelitian ini digambarkan dalam *flowchart* seperti berikut.



Gambar 4.1 Flowchart Penelitian

BAB V

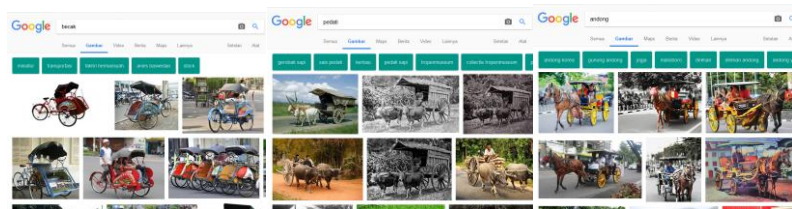
ANALISIS DAN PEMBAHASAN

5.1 Implementasi Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan *software fatkun batch image downloader* yang digunakan didalam browser *google chrome*. Pada kasus ini, pengambilan data dari situs web www.google.com dibagian *image search*. Data yang diambil adalah data transportasi tradisional berupa data citra transportasi andong, becak, dan pedati dengan jumlah sampel sebanyak 40 data citra setiap kategorinya. Menggunakan *google image* akan lebih memudahkan peneliti mendapatkan citra yang ada pada *google*, selanjutnya gambar yang telah di search pada *google image* akan di *download* dengan cepat menggunakan *fatkun batch image downloader*.

5.2 Penggunaan *fatkun batch image downloader*

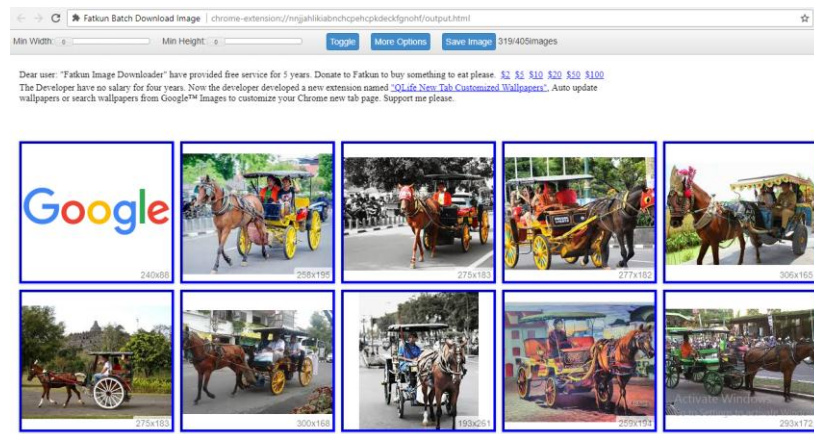
Software fatkun batch image downloader digunakan untuk mengumpulkan citra dengan mudah dan cepat. Langkah yang digunakan peneliti adalah melakukan pencarian objek yang digunakan untuk penelitian yaitu Andong, Becak, dan Pedati dalam *google image* dan memilih data yang sesuai dengan yang diinginkan oleh peneliti menimbang dari kedalaman gambar atau format RGB (*Red, Green, dan Blue*).



Gambar 5.1 Citra Dalam Google Image

Setelah melakukan pencarian objek pada *google image*, selanjutnya peneliti mengumpulkan data citra dengan menggunakan *Software fatkun batch image downloader* setelah terkumpul data citra yang diinginkan sebelum disave peneliti

memilih citra yang memiliki kedalaman gambar yang sama dan gambar yang mudah untuk terbaca untuk penelitian ini, kemudian di *save* dalam satu folder dan diberi nama untuk setiap dataset citra secara berurutan sesuai dengan masing-masing kategori benda.



Gambar 5.2 Data Citra Dari Software Fatkun Batch Image Downloader

5.3 Load Dataset

Data yang telah di siapkan dalam satu folder selanjutnya dilakukan pengolahan menggunakan *software R Studio*. Pada proses ini peneliti membutuhkan *library packages keras* dan *packages EBImages*. *Packages library keras* digunakan untuk menjalankan metode yang digunakan oleh peneliti sedangkan, *library packages EBImages* digunakan agar data citra yang akan diolah menggunakan *software R Studio* dapat digunakan dan ditampilkan. *Script* yang digunakan untuk menampilkan data sebagai berikut.

```
#input data
setwd("E://datafix/bismillah")
# Read Images
images <- list.files()
images

summary(images)

list_of_images = lapply(images, readImage)
list_of_images
display(list_of_images[[15]])
```

Gambar 5.3 Script Pemanggilan Data

Pada gambar diatas untuk memasukan gambar pada *software R Studio* menggunakan cara `setwd` pada folder data yang telah disiapkan dan berisi data citra yang memiliki ukuran *pixel* yang berbeda-beda dengan kedalaman atau format citra yang sama yaitu RGB (*Red, Green, Blue*). Sebaiknya pada folder yang telah disiapkan hanya berisi dataset yang akan digunakan karena jika terdapat file lainnya maka akan terpanggil juga. Selanjutnya data citra yang terdapat didalam folder di *load* dengan cara `list.files`, sehingga seluruh data citra yang terdapat didalam folder akan terpanggil kedalam *software R Studio*. Untuk memasukan seluruh data citra terbaca kedalam *software R Studio* dapat menuliskan *script images* dan melihat *summary* yang ada dalam folder dataset dengan cara *script summary(images)*.

Selain itu, untuk memastikan gambar yang sudah masuk dalam *software* maka dilakukan *lapplay* menggunakan *readImage* sehingga peneliti dapat memastikan seluruh data citra dapat dilihat dalam *software*. *Script display* untuk menampilkan salah satu dari data citra sesuai dengan yang diinginkan.

5.4 Pembuatan Data Latih dan Data Uji

Data latih digunakan algoritma klasifikasi untuk membentuk sebuah model *classifier*, model ini merupakan representasi pengetahuan yang akan digunakan untuk prediksi kelas data baru yang belum pernah ada, semakin besar data latih yang akan digunakan, maka akan semakin baik machine dalam memahami pola data. Data uji yang ada digunakan untuk mengetahui sejauh mana *classifier* berhasil melakukan klasifikasi dengan tepat. Data yang digunakan sebagai data latih dan data uji yang memiliki kelas, dengan jumlah perbandingan 80%:20% untuk data latih dan data uji. Perbandingan jumlah data latih dan data uji dapat dilihat pada **tabel 5.1** untuk data citra yang digunakan.

Tabel 5.1 Perbandingan data latih dan data uji

Data Citra	Jumlah	Data Latih (80%)	Data Uji (20%)
Andong	40	32	8

Becak	40	32	8
Pedati	40	32	8
Jumlah	120	96	24

Data latih dan data uji yang telah di uat selanjutnya diaplikasikan dengan R Studio. Seperti pada gambar berikut.

```
#create train
train <- list_of_images[c(1:32,41:72,81:112)]
str(train)
display(train[[8]])

#create test
test <- list_of_images[c(33:40,73:80,113:120)]
test
display(test[[9]])

par(mfrow = c(8,12))
for (i in 1:96) plot(train[[i]])
```

Gambar 5.4 Script Pemanggilan Data Latih Dan Data Uji

Pada data cira yang telah dikumpulkan diberi nama berdasarkan kategori benda agar mempermudah pengurutan dan pembuatan data latih dan data uji. Untuk membuat data latih seperti yang terlihat pada gambar diatas menggunakan *script train* dan memasukan urutan dari dataset citra setiap kategori benda yang digunakan. Dalam penelitian ini peneliti menggunakan data latih citra andong sebanyak 32 sesuai dengan urutan 1 sampai 32, dan dilanjutkan dengan kategori lainnya yang memiliki jumlah sama dan secara berurutan.

Sedangkan dalam pembuatan data uji script yang digunakan adalah test sama hal nya dengan data latih pada data uji juga dataset citra diurutkan sesuai dengan kategori benda dengan jumlah masing-masing 8 data citra.

5.5 *Risize dan Combine*

Pada proses ini dilakukan untuk menyamakan urutan dari seluruh data citra yang akan diteliti, dengan melakukan proses ini maka proses untuk menganalisis akan lebih mudah dan cepat dikarenakan ukuran citra sesuai dengan kemampuan dari *software R Studio*.

```
# Resize & combine
for (i in 1:96) {train[[i]] <- resize(train[[i]], 32, 32)}
for (i in 1:24) {test[[i]] <- resize(test[[i]], 32, 32)}

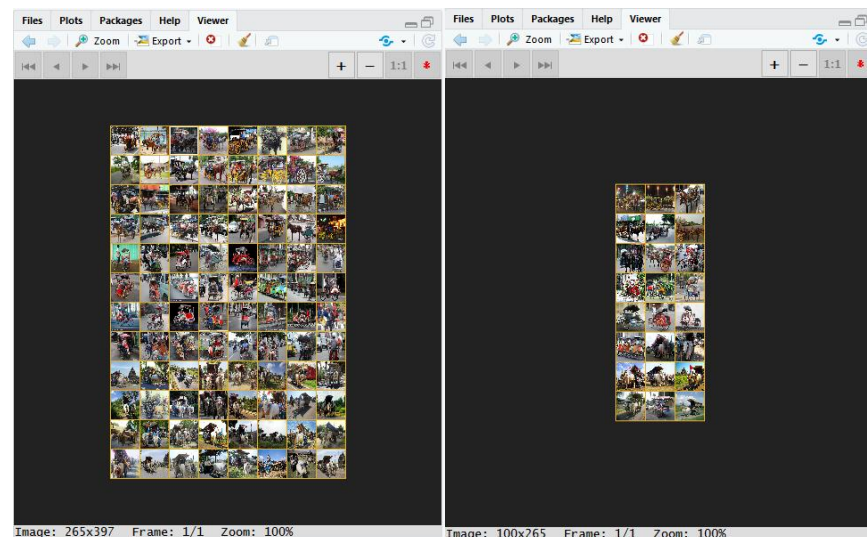
train <- combine(train)
str(train)
x <- tile(train, 8)
display(x, title='Pictures')

test <- combine(test)
y <- tile(test, 3)
display(y, title = 'Pics')
```

Gambar 5.5 Script *Resize dan Combine*

Pada gambar diatas menunjukkan proses *resize dan combine* terhadap data yang digunakan. Pada proses *resize* untuk membuat ukuran yang sama dari data citra latih dan uji masing-masing menjadi ukuran 32x32 *pixel* , selain itu dataset citra dibuat dalam bentuk format gambar yang sama RGB (*Red, Green, Blue*) yang berarti seluruh dataset citra yang digunakan peneliti memiliki warna bukan *grayscale*. Perubahan ukuran dataset citra menandakan setiap citra siap untuk dilakukan analisis menggunakan metode *convolutional neural network*.

Sedangkan, untuk fungsi dari *combine* untuk mengkombinasikan dataset latih maupun uji dari hasil pengubahan citra. Sehingga dapat mengetahui bahwa perubahan ukuran citra dilakukan dengan sempurna pada seluruh dataset citra seperti pada gambar berikut.



Gambar 5.6 *Combine Dataset Latih Dan Uji*

Pada **Gambar 5.6** menunjukkan hasil dari kombinasi dataset citra yang digunakan. Pada data latih menggunakan jumlah kolom sebanyak 8, sedangkan

pada data uji menggunakan jumlah kolom 3. Dataset citra dari keseluruhan kombinasi sudah memiliki ukuran 32x32 *pixel*.

5.6 *Reorder Dimension dan Response*

Proses *reorder dimension* dan *reponse* berfungsi agar penempatan dimensi citra dapat terbaca oleh program dalam urutan yang digunakan. Program menginginkan kombinasi jumlah dataset citra latih dan uji, ukuran piksel dan format RGB yang digunakan. Sehingga dilakukan *reorder dimension* agar data citra dapat dianalisis.

Response berfungsi untuk memberikan pelebelaan dataset citra yang digunakan berdasarkan klasifikasi dataset citra agar pembentukan matriks dapat dibaca dengan mudah dan tepat.

One hot encoding digunakan untuk melakukan kategori dataset citra latih dan uji berdasarkan label yang telah dibuat. *Script* yang digunakan oleh peneliti sebagai beriku.

```
# Reorder dimension
train <- aperm(train, c(4, 1, 2, 3))
test <- aperm(test, c(4, 1, 2, 3))
str(train)

# Response
trainy <- c(rep(0,32),rep(1,32),rep(2,32))
testy <- c(rep(0,8),rep(1,8),rep(2,8))

# One hot encoding
trainLabels <- to_categorical(trainy)
testLabels <- to_categorical(testy)
```

Gambar 5.7 *Script Reorder dimension, Response, dan One hot encoding*

Pada gambar diatas menunjukkan *script* yang digunakan pada proses ini, *script reorder dimension* untuk membuat dataset citra latih sebanyak 96 dengan piksel 32x32 dan format RGB. Sedangkan, pada *script* untuk data uji dataset yang digunakan sebanyak 24 dengan format ukuran 32x32 piksel dan format citra RGB.

Response melakukan klasifikasi berdasarkan citra dataset latih dan uji, untuk data latih klasifikasi dengan menggunakan pelebelaan yaitu Andong dengan lebel 0, Becak dengan label 1, dan Pedati dengan label 2 dengan masing-masing klasifikasi dataset sebanyak 32. Sedangkan pada data uji dilakukan klasifikasi

yang sama dengan melakukan pelebelan seperti pada latih tetapi pada data uji menggunakan dataset sebanyak 8 citra. Setelah melakukan pelebelan maka dilakukan kategorisasi berdasarkan data latih dan data uji.

5.7 Model

Pada proses pembuatan model memuat beberapa *layer* yaitu *layer* konvolusi, *layer pooling*, *layer dropout*, *layer flatten* dan *layer dense*. Selain *layer-layer* yang digunakan dalam proses cnn terdapat pula fungsi aktivasi, dalam penelitian ini menggunakan aktivasi *Rule*. Pada penelitian ini *script* penentuan model yang digunakan seperti berikut.

```
# Model
model <- keras_model_sequential()

model %>%
  layer_conv_2d(filters = 32,
                kernel_size = c(3,3),
                activation = 'relu',
                input_shape = c(32, 32, 3)) %>%
  layer_conv_2d(filters = 32,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_max_pooling_2d(pool_size = c(2,2)) %>%
  layer_dropout(rate = 0.25) %>%
  layer_conv_2d(filters = 64,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_conv_2d(filters = 64,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_max_pooling_2d(pool_size = c(2,2)) %>%
  layer_dropout(rate = 0.25) %>%
  layer_flatten() %>%
  layer_dense(units = 256, activation = 'relu') %>%
  layer_dropout(rate=0.25) %>%
  layer_dense(units = 3, activation = 'softmax') %>%

compile(loss = 'categorical_crossentropy',
        optimizer = optimizer_sgd(lr = 0.01,
                                   decay = 1e-6,
                                   momentum = 0.9,
                                   nesterov = T),
        metrics = c('accuracy'))
summary(model)
```

Gambar 5.8 Script Penentuan Model

Pada gambar diatas citra berupa angka menjalin konvolusi pertama, citra yang berukuran 32x32 piksel memiliki multidimensional *array* berukuran 32x32x3 karena dalam format RGB. Kernel yang memiliki ukuran 3x3 dengan tebal 3 sesuai dengan *channel* dari data citra, dan *filter* yang digunakan sebanyak 32. Peneliti menggunakan kernel 3x3 agar lebih *detail* dalam mendapatkan informasi dari setiap citra yang digunakan. Setelah mendapatkan hasil operasi

maka model akan melakukan fungsi aktivasi dan *Pooling* data. Proses *Pooling* layer berfungsi untuk mengurangi dimensi dari *feature map*.

Hasil dari proses konvolusi berupa *feature map* yang digunakan untuk proses konvolusi seterusnya secara berulang. Langkah selanjutnya dilakukan *flatten feature map* dalam bentuk vektor untuk dilakukan proses *fully-connected layer* untuk menghasilkan klasifikasi dari citra. *Output* model yang dihasilkan dari *software R Studio* sebagai berikut.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 30, 30, 32)	896
conv2d_2 (Conv2D)	(None, 28, 28, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
dropout_1 (Dropout)	(None, 14, 14, 32)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	18496
conv2d_4 (Conv2D)	(None, 10, 10, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_2 (Dropout)	(None, 5, 5, 64)	0
flatten_1 (Flatten)	(None, 1600)	0
dense_1 (Dense)	(None, 256)	409856
dropout_3 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 3)	771
Total params: 476,195		
Trainable params: 476,195		
Non-trainable params: 0		

Gambar 5.9 Output Model software R Studio

Pada **Gambar 5.9** menunjukkan hasil dari model yang telah didapatkan, hasil dari model berdasarkan arsitektur yang telah dibuat dengan berbagai jumlah parameter yang dihasilkan dalam model. Hasil *inputan* model dimasukan dalam rumus *output shape* sehingga dapat diketahui *output shape* dari setiap *layer*.

$I/p = 32 \times 32$, *filter* = 32 *kernel size* 3x3, *padding* = 0, *stride* = 1

$C1 = 30 \times 30$, *filter* = 32 *kernel size* 3x3, *padding* = 0, *stride* = 1

$C2 = 28 \times 28$, *filter* = 32 *kernel size* 3x3, *padding* = 0, *stride* = 1

Pada *pooling layer* menggunakan dimensi 14x14 dikarenakan *pool size* = 2x2 sehingga *pixel* 28 pada C2 dibagi 2.

$P1 = 14 \times 14 \times 32$

Output shape dari *dropout* mempunyai nilai sama dengan *pooling layer*

$D1 = 14 \times 14 \times 32$

Kemudian masuk kembali pada *convolution layer* ke 3 dengan menggunakan dimensi pada *dropout* 1 untuk di olah menjadi C3. Pada C3 peneliti menggunakan *filter* 64 dengan *kernel* 3x3 sehingga didapatkan *output shape* 12x12x64.

C3 = 12x12x64, *filter* = 64 *kernel size* 3x3, *padding* = 0, *stride* = 1

C4 = 10x10x64, *filter* = 64

P2 = 2x2. 5x5x64

D2 = 5x5x64

5.8 Fit Model

Proses penentuan model dari data citra latih dan data uji untuk mengetahui hasil dari nilai akurasi data citra latih dan data citra uji, selain nilai akurasi, akan terdapat nilai *loss*. Model ini terbentuk dari arsitektur algoritma dalam cnn yang telah disiapkan sebelumnya.

```
# Fit model
history <- model %>%
  fit(train,
      trainLabels,
      epochs = 30,
      batch_size = 32,
      validation_split = 0.2,
      validation_data = list(test, testLabels))
plot(history)
```

Gambar 5.10 Script Fit Model

Pada penentuan model menggunakan perintah yang di jalankan seperti gambar diatas peneliti menggunakan beberapa perintah yaitu: *epoch* berfungsi dalam perulangan model dalam menentukan akurasi dan *loss*, dalam penelitian ini peneliti telah melakukan beberapa percobaan dengan menggunakan nilai *epoch* 400, *epoch* 60, *epoch* 30 dan *epoch* 50 agar mendapatkan nilai akurasi mendekati 1. Dari percobaan yang telah dilakukan oleh peneliti ternyata dengan menggunakan *epoch* 30 kali perulangan disertai nilai *Batch size* menggunakan 32 dan selain itu peneliti menggunakan *validation split* sebesar 0.2 didapatkan nilai akurasi mendekati 1. Validasi data citra menggunakan data uji sebanyak 24 data. Hasil dari pemodelan yang didapatkan dengan nilai akurasi dan *loss* dari data latih dan uji seperti pada gambar berikut:

```

Train on 96 samples, validate on 24 samples
Epoch 1/30
96/96 [=====] - 20s 210ms/step - loss: 1.1063 - acc: 0.3125
- val_loss: 1.1009 - val_acc: 0.3750
Epoch 2/30
96/96 [=====] - 1s 15ms/step - loss: 1.0977 - acc: 0.4271 -
- val_loss: 1.0986 - val_acc: 0.2083
Epoch 3/30
96/96 [=====] - 1s 11ms/step - loss: 1.0988 - acc: 0.3958 -
- val_loss: 1.0951 - val_acc: 0.3333
Epoch 4/30
96/96 [=====] - 1s 11ms/step - loss: 1.0801 - acc: 0.4792 -
- val_loss: 1.0900 - val_acc: 0.2917
...
Epoch 28/30
96/96 [=====] - 1s 12ms/step - loss: 0.6596 - acc: 0.6771 -
- val_loss: 0.9128 - val_acc: 0.6250
Epoch 29/30
96/96 [=====] - 1s 12ms/step - loss: 0.7215 - acc: 0.6250 -
- val_loss: 0.8211 - val_acc: 0.6250
Epoch 30/30
96/96 [=====] - 1s 12ms/step - loss: 0.6304 - acc: 0.7292 -
- val_loss: 0.7389 - val_acc: 0.7500

```

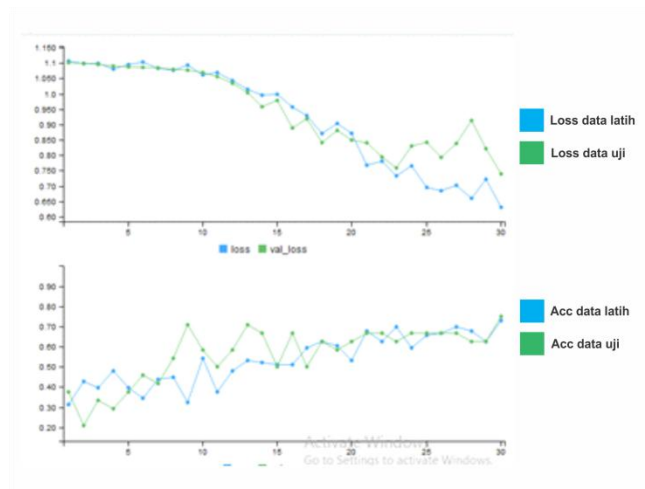
Gambar 5.11 Nilai Akurasi Dan Loss Fit Model

Berdasarkan hasil *output* pada gambar diatas *sampel* dari proses *run* model seperti pada **Tabel 5.2** berikut:

Tabel 5.2 Hasil akurasi data latih dan data uji pada model

Epoch	Data Latih		Data Uji	
	Acc	Loss	Val Acc	Val Loss
1	0.3125	1.1063	0.3750	1.1009
2	0.4271	1.0977	0.2083	1.0986
3	0.3958	1.0988	0.3333	1.0951
4	0.4792	1.0801	0.2917	1.0900

28	0.6771	0.6596	0.6250	0.9128
29	0.6250	0.7215	0.6250	0.8211
30	0.7292	0.6304	0.7500	0.7389



Gambar 5.12 *Plot Fit Model*

Seperti yang terlihat pada gambar diatas *plot* yang dihasilkan dari *fit model* menunjukkan akurasi dari data latih dan data uji *relative* meningkat dari awal sampai ke *epoch* 30. Sedangkan pada nilai *loss* data latih dan data uji dari epoch pertama sampai ke *epoch* 30 terus mengalami penurunan.

5.9 *Evaluation dan Prediction Data Latih*

Nilai evaluasi dan prediksi didapatkan dari hasil model yang telah dibuat sebelumnya. Nilai model yang didapatkan dari dataset citra latih sebanyak 32 citra dari masing-masing kategori Andong, Pedati, dan Becak.

```
> # Evaluation & Prediction - train data
> model %>% evaluate(train, trainLabels)
96/96 [=====] - 1s 6ms/step
$loss
[1] 0.5250072

$acc
[1] 0.8229167

> pred <- model %>% predict_classes(train)
> table(Predicted = pred, Actual = trainy)
      Actual
Predicted 0  1  2
      0 25  3  2
      1  5 28  4
      2  2  1 26
```

Gambar 5.13 *Hasil Dari Evaluasi Dan Prediksi*

Berdasarkan pada gambar diatas menunjukkan hasil validitas data latih memiliki nilai *loss* sebesar 0.52500072 dan nilai akurasi yang didapatkan sebesar 0.8228167 dari range nilai 1.0. Nilai yang didapatkan dari hasil *fit model*

sebelumnya, pada tabel prediksi dapat dilihat pada citra andong mengalami 7 kesalahan dengan 25 citra yang diprediksi besar dan pada citra becak mengalami 4 kesalahan dengan 28 citra yang diprediksi sesuai dengan citra yang diinginkan. Sedangkan pada klasifikasi citra pedati mengalami 6 kesalahan dengan 26 citra yang diprediksi sesuai yang diinginkan.

5.10 Evaluation dan Prediction Data Uji

Nilai evaluasi dan prediksi didapatkan dari hasil model yang telah dibuat sebelumnya. Nilai model yang didapatkan dari dataset citra uji sebanyak 32 citra dari masing-masing kategori Andong, Becak, dan Pedati.

```
> # Evaluation & Prediction - test data
> model %>% evaluate(test, testLabels)
24/24 [=====] - 0s 8ms/step
$loss
[1] 0.7389246

$acc
[1] 0.75

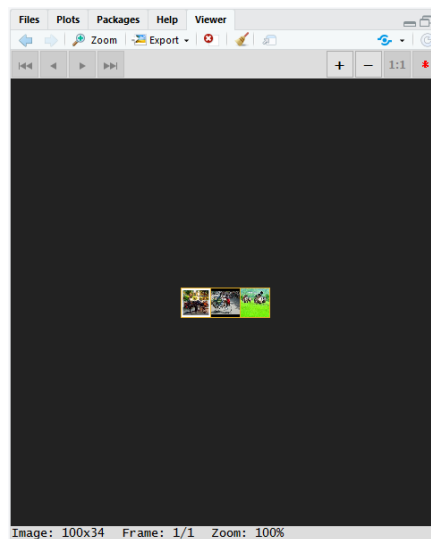
> pred <- model %>% predict_classes(test)
> table(Predicted = pred, Actual = testy)
      Actual
Predicted 0 1 2
         0 6 2 0
         1 1 5 1
         2 1 1 7
```

Gambar 5.14 Hasil Dari Evaluasi Dan Prediksi

Berdasarkan pada gambar diatas menunjukkan hasil validitas data uji memiliki nilai *loss* sebesar 0.7389246 dan nilai akurasi yang didapatkan sebesar 0.75 range nilai 0 sampai 1. Nilai yang didapatkan dari hasil *fit model* sebelumnya, pada tabel prediksi dapat dilihat pada citra andong mengalami 2 kesalahan dan pada citra becak mengalami 3 kesalahan dengan 5 citra yang diprediksi sesuai dengan citra yang diinginkan. Sedangkan pada klasifikasi citra pedati mengalami 1 kesalahan dengan 7 citra yang diprediksi sesuai yang diinginkan.

5.11 Data Test

Setelah didapatkan sebelumnya peneliti memasukan data baru untuk dilakukan tahap uji coba. Pada tahap ini peneliti menggunakan 3 data yang akan di uji coba yaitu masing-masing dari kategori 1 citra Andong, 1 citra Becak, dan 1 citra Pedati. Citra dari data uji coba yang lakukan seperti pada gambar berikut.



Gambar 5.15 *Citra Data Test*

Pada gambar diatas menunjukka data baru yang telah melewati proses *preprocessing* citra. Dari data baru akan dilakukan data *test*, peneliti melakukan uji coba model terhadap data *test* yang digunakan untuk mengetahui hasil prediksi terhadap data *test*. Hasil dari prediksi data *test* seperti pada gambar berikut:

```
> table(Predicted = pred, Actual = testy)
      Actual
Predicted 0 1 2
      0 1 0 0
      1 0 1 0
      2 0 0 1
> prob <- model %>% predict_proba(got)
> cbind(prob, Predicted_class = pred, Actual = testy)
      Predicted_class Actual
[1,] 0.81468141 0.1210636 0.06425502      0      0
[2,] 0.02157571 0.9564610 0.02196328      1      1
[3,] 0.33104619 0.1383130 0.53064084      2      2
```

Gambar 5.16 *Hasil Prediksi Data Test*

Pada hasil yang ditunjukkan pada **Gambar 5.15** tabel prediksi setiap klasifikasi citra Andong, Becak, dan Andong memiliki akurasi yang seluruhnya benar dan tepat. Pada hasil kelas prediksi menunjukkan akurasi dari hasil data *test* seluruhnya benar. Hasil tersebut dapat dikatakan bahwa penggunaan metode *Covolutional Neural Network* (CNN) menggunakan *Keras* bisa diimplementasikan terhadap citra transportasi tradisioanl Andong, Becak, dan Pedati.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil dari analisis yang telah dilakukan, maka peneliti dapat menarik beberapa kesimpulan seperti berikut:

1. Metode *Convolutional Neural Network* (CNN) dapat mengimplementasikan data citra transportasi tradisional dengan menggunakan *library keras*. Dalam proses analisis dengan metode CNN menggunakan proses preprocessing citra dan dilanjut pembuatan model hingga didapatkan pemodelan yang baik dalam citra data latih dan data uji.
2. Dengan menggunakan *epoch* 30, *batch size* 32 dan *validation split* 0.2. Pada tabel prediksi dapat dilihat pada citra andong mengalami 7 kesalahan klasifikasi, untuk citra becak mengalami 4 kesalahan klasifikasi dengan 28 citra yang terklasifikasi sebagai becak dan pada klasifikasi citra pedati mengalami 6 kesalahan klasifikasi dengan 26 citra yang terklasifikasi sebagai pedati. Sehingga didapatkan nilai akurasi sebesar 82% dari 100% yang diharapkan. Sedangkan dengan menggunakan *epoch* 30, *batch size* 32 dan *validation split* 0.2. Pada tabel prediksi dapat dilihat pada citra andong mengalami 2 kesalahan klasifikasi dan pada citra becak mengalami 3 kesalahan klasifikasi dengan 5 citra yang diklasifikasi sebagai becak dari 8 data uji. Sedangkan pada klasifikasi citra pedati mengalami 1 kesalahan klasifikasi dengan 7 citra yang terklasifikasi sebagai pedati. Sehingga didapatkan nilai akurasi sebesar 75% dari 100% yang diharapkan.
3. Dengan menggunakan 1 data *test* dari setiap kategori tabel prediksi setiap citra data *test* Andong, Becak, dan Pedati terklasifikasi sesuai dengan prediksi dengan nilai akurasi 100%. Hasil tersebut dapat dikatakan bahwa jumlah dari data *test* yang digunakan dapat mempengaruhi keakurasian metode *Covolutional Neural Network* (CNN).

6.2 Saran

Berdasarkan hasil dari analisis dan kesimpulan, dapat diberikan saran sebagai berikut:

1. Pada penelitian ini dataset yang digunakan memiliki jumlah yang kurang banyak dengan kualitas citra yang masih kurang baik. Sehingga hasil yang didapatkan kurang maksimal, untuk peneliti berikutnya bisa menggunakan jumlah citra yang lebih banyak dan kualitas foto yang lebih mudah terbaca oleh program atau kualitas citra yang tidak *blur* hanya objek yang akan teliti saja yang terlihat jelas.
2. Bisa membuat perbandingan untuk penentuan perhitungan *fit model* dalam menentukan keakurasian *Convolutional Neural Network*.
3. Bagi peneliti berikutnya saat menggunakan data citra yang lebih banyak harus memperhatikan kapasitas perangkat komputer dan laptop agar memperlancar dalam proses penelitian.
4. Membangun arsitektur dari *Convolutional Neural Network* (CNN) dengan memperhatikan jumlah data dan perhitungan *fit model* agar dapat mengklasifikasi citra dengan keakurasian 100%.

DAFTAR PUSTAKA

- Candra, Barnes B. 2014. *Aplikasi Jaringan Syaraf Tiruan untuk Penentuan Kelas Kemasakan Buah Tomat Berdasarkan Warna Citra*. Tugas Akhir. Program Studi Teknik Pertanian Fakultas Teknologi Pertanian Universitas Gajah Mada Yogyakarta.
- Goodfellow. 2016. *Deep Learning*. MIT Press
- Haryoko, Andy, Sholeh H., dan Aziz M. 2016. “Pengenalan Karakter Plat Motor Kendaraan Bermotor Berbasis Citra dengan Menggunakan Metode *Canny* dan Algoritma *Backpropagation*”. Jurnal Ilmiah NERO Vol. 2, No.2. Program Studi Teknik Elektro Fakultas Teknik Universitas Brawijaya.
- Jamizah. 2016. *Perbedaan Alat Transportasi Tradisional dengan Alat Transportasi Modern*. Jurnal. FKIP Universitas Wiralodra Indramayu.
- Nusantara, Ok. 2017. *Alat Transportasi Tradisional*. Diakses pada tanggal 23 Maret 2018 dari <http://oknusantara.com/alat-transportasi-tradisional-detail-43275.html>
- Planologi, Radar. 2015. *Pengertian Transportasi, Manfaat, Fungsi dan Jenis-Jenisnya*. Diakses pada tanggal 23 Maret 2018 dari <http://www.radarplanologi.com/2015/11/pengertian-transportasi-manfaat-fungsi-jenisnya.html>
- Purita, Eva D. 2013. *Pengelolaan Transportasi Umum di Jalan Malioboro Yogyakarta*. Skripsi. Fakultas Ilmu Sosial Universitas Negeri Yogyakarta.
- Pudi, Kefin. 2017. *Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU*. Tugas Akhir. Program Studi Teknologi Industri Universitas Atma Jaya Yogyakarta.
- Sani, Zulfikar. 2012. *Struktur Tata Ruang Kota*. Pustaka Pelajar, Yogyakarta.
- Suartika, Wayan, Arya Y., dan Rully S. 2016. *Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltect 101*. Jurnal Teknik ITS Vol. 5, No. 1, (2016) ISSN: 2337-3539(2301-9271 Print)

- Suprihatin, Hasni. 2017. *Pemertahanan Transportasi Tradisional di Era Modern*. Skripsi. Program Studi Sosial Agama Fakultas Ushuluddin dan Pemikiran Islam Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
- Rismayati. 2016. *Implementasi Convolutional Neural Network untuk Sortasi Mutu Salak Ekspor Berbasis Citra Digital*. Tesis. Program Studi S2 Ilmu Komputer Departemen Ilmu Komputer dan Elektronika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gajah Mada Yogyakarta.

LAMPIRAN

Lampiran I *Script* R Studio CNN *library* Keras

```
# Convolutional Neural Networks
```

```
# Load packages
```

```
library(keras)
```

```
library(EBImage)
```

```
#input data
```

```
setwd("E://datafix/bismillah")
```

```
# Read Images
```

```
images <- list.files()
```

```
images
```

```
summary(images)
```

```
list_of_images = lapply(images, readImage )
```

```
list_of_images
```

```
display(list_of_images[[15]])
```

```
#create train
```

```
train <- list_of_images[c(1:32,41:72,81:112)]
```

```
str(train)
```

```
display(train[[8]])
```

```
#create test
```

```
test <- list_of_images[c(33:40,73:80,113:120)]
```

```
test
```

```
display(test[[9]])
```

```
par(mfrow = c(8,12))
```

```
for (i in 1:96) plot(train[[i]])
```

```

# Resize & combine
for (i in 1:96) {train[[i]] <- resize(train[[i]], 32, 32)}
for (i in 1:24) {test[[i]] <- resize(test[[i]], 32, 32)}

train <- combine(train)
str(train)
x <- tile(train, 8)
display(x, title='Pictures')

test <- combine(test)
y <- tile(test, 3)
display(y, title = 'Pics')

# Reorder dimension
train <- aperm(train, c(4, 1, 2, 3))
test <- aperm(test, c(4, 1, 2, 3))
str(train)

# Response
trainy <- c(rep(0,32),rep(1,32),rep(2,32))
testy <- c(rep(0,8),rep(1,8),rep(2,8))

# One hot encoding
trainLabels <- to_categorical(trainy)

testLabels <- to_categorical(testy)

# Model
model <- keras_model_sequential()

model %>%
  layer_conv_2d(filters = 32,
                kernel_size = c(3,3),
                activation = 'relu',

```

```

        input_shape = c(32, 32, 3)) %>%
layer_conv_2d(filters = 32,
              kernel_size = c(3,3),
              activation = 'relu') %>%
layer_max_pooling_2d(pool_size = c(2,2)) %>%
layer_dropout(rate = 0.25) %>%
layer_conv_2d(filters = 64,
              kernel_size = c(3,3),
              activation = 'relu') %>%
layer_conv_2d(filters = 64,
              kernel_size = c(3,3),
              activation = 'relu') %>%
layer_max_pooling_2d(pool_size = c(2,2)) %>%
layer_dropout(rate = 0.25) %>%
layer_flatten() %>%
layer_dense(units = 256, activation = 'relu') %>%
layer_dropout(rate=0.25) %>%
layer_dense(units = 3, activation = 'softmax') %>%

compile(loss = 'categorical_crossentropy',
        optimizer = optimizer_sgd(lr = 0.01,
                                   decay = 1e-6,
                                   momentum = 0.9,
                                   nesterov = T),
        metrics = c('accuracy'))
summary(model)

# Fit model
history <- model %>%
fit(train,
    trainLabels,
    epochs =30,
    batch_size = 32,
    validation_split = 0.2,

```

```

validation_data = list(test, testLabels))
plot(history)

# Evaluation & Prediction - train data
model %>% evaluate(train, trainLabels)
pred <- model %>% predict_classes(train)
table(Predicted = pred, Actual = trainy)

prob <- model %>% predict_proba(train)
cbind(prob, Predicted_class = pred, Actual = trainy)

# Evaluation & Prediction - test data
model %>% evaluate(test, testLabels)
pred <- model %>% predict_classes(test)
table(Predicted = pred, Actual = testy)

prob <- model %>% predict_proba(test)
cbind(prob, Predicted_class = pred, Actual = testy)

#save model
save_model_hdf5(model,filepath='E://arfian/fix.hdf5')
model=load_model_hdf5(filepath="E://arfian/fix.hdf5")

#####
#####very fixxx
setwd("E://datalatih")
img <- c("x1.jpg", "x2.jpg", "x3.jpg")
image <- list()
for (i in 1:3) {image[[i]] <- readImage(img[i])}

display(image[[2]])
# Get the image as a matrix

for (i in 1:3) {image[[i]] <- resize(image[[i]], 32, 32)}

```

```
for (i in 1:3) {image[[i]]<-toRGB(image[[i]])}
```

```
fixx <- combine(image)
```

```
y <- tile(fixx, 3)
```

```
display(y, title = 'Pics')
```

```
str(fixx)
```

```
got <- aperm(fixx, c(4, 1, 2, 3))
```

```
str(got)
```

```
testy <- c(rep(0,1), rep(1,1),rep(2,1))
```

```
# One hot encoding
```

```
testLabels <- to_categorical(testy)
```

```
pred <- model %>% predict_classes(got)
```

```
table(Predicted = pred, Actual = testy)
```

```
prob <- model %>% predict_proba(got)
```

```
cbind(prob, Predicted_class = pred, Actual = testy)
```