



TUGAS AKHIR - EC 184801

**ESTIMASI JUMLAH DAN KEPADATAN ORANG PADA
CITRA KERUMUNAN TUNGGAL MENGGUNAKAN
METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)**

Deu Aldo Dhavicky
NRP 0721154000016

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

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



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC 184801

**ESTIMASI JUMLAH DAN KEPADATAN ORANG PADA
CITRA KERUMUNAN TUNGGAL MENGGUNAKAN
METODE *CONVOLUTIONAL NEURAL NETWORK (CNN)***

Deu Aldo Dhavicky
NRP 07211540000016

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

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



FINAL PROJECT - EC 184801

**ESTIMATION NUMBER AND DENSITY OF THE CROWD
ON SINGLE IMAGE USING CONVOLUTIONAL NEURAL
NETWORK (CNN) METHOD**

Deu Aldo Dhavicky
NRP 07211540000016

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

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

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Estimasi Jumlah dan Kepadatan Orang Pada Citra Kerumunan Tunggal Menggunakan *Convolutional Neural Network* (CNN)**” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

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

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

Surabaya, Juli 2020



Deu Aldo Dhavicky
NRP. 0721154000016

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

Estimasi Jumlah dan Kepadatan Orang Pada Citra Kerumunan Tunggal Menggunakan Metode *Convolutional Neural Network (CNN)*

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

Oleh: Deu Aldo Dhavicky (NRP: 07211540000016)

Tanggal Ujian : 8 Juli 2020

Periode Wisuda : September 2020

Disetujui oleh:

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

(Pembimbing I)

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

(Pembimbing II)

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

(Penguji I)

Mochamad Hariadi, ST., M.Sc., Ph.D.
NIP: 196912091997031002

(Penguji II)

Susi Juniastuti, ST., M.Eng.
NIP. 196506181999032001

(Penguji III)



Mengetahui
Ket. Departemen Teknik Komputer

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

Halaman ini sengaja dikosongkan

ABSTRAK

- Nama Mahasiswa : Deu Aldo Dhavicky
Judul Tugas Akhir : Estimasi Jumlah dan Kepadatan Orang Pada Citra Kerumunan Tunggal Menggunakan Metode *Convolutional Neural Network (CNN)*
Pembimbing : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D.
2. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

Estimasi jumlah orang pada pusat keramaian saat ini banyak diterapkan, baik menggunakan cara konvensional seperti menghitung secara manual hingga menggunakan bantuan alat atau sensor, hal ini dilakukan dengan tujuan untuk memantau jumlah populasi, karena populasi orang dengan jumlah besar yang berkumpul pada suatu titik di area tertentu akan memunculkan berbagai masalah, salah satunya keamanan. Salah satu metode yang saat ini sedang dikembangkan untuk mendeteksi dan mengenali object pada sebuah citra dengan menggunakan *Convolutional Neural Network (CNN)*. CNN adalah pengembangan dari *Multilayer Perceptron (MLP)* yang didesain untuk mengolah data dua dimensi. Dalam tugas akhir ini dikembangkan estimasi jumlah dan kepadatan orang pada citra kerumunan tunggal menggunakan metode *Convolutional Neural Network* dengan menggunakan Shanghai Tech datasets.

Kata Kunci: *Crowd*, *Crowd-Counting*, Klasifikasi, *Convolutional Neural Network (CNN)*.

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Deu Aldo Dhavicky
Title : *Estimation Number and Density of The Crowd on Single Image Using Convolutional Neural Network (CNN)*
Advisors : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D.
2. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

Estimation of the number of people at the center of the crowd is currently widely applied, both using conventional methods such as counting manually to use the help of tools or sensors, this is done with the aim to monitor the population, because a large population of people who gather at a point in a certain area will bring up various problems, one of which is security. One method that is currently being developed to detect and recognize objects in an image using Convolutional Neural Network (CNN). CNN is the development of Multilayer Perceptron (MLP) which is designed to process two-dimensional data. In this final project, an estimation for the number and density of people in a single crowd is developed using the Convolutional Neural Network method using Shanghai Tech datasets.

Keywords: Crowd, Crowd-Counting, Classification, Convolutional Neural Network (CNN)

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Allah Swt. atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Estimasi Jumlah dan Kepadatan Orang Pada Citra Kerumunan Tunggal Menggunakan Metode *Convolutional Neural Network* (CNN)**.

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

1. Semua keluarga dan kedua orang tua tercinta dan adik saya yang telah memberikan motivasi, dorongan spiritual, dan material dalam penyelesaian buku penelitian ini.
2. Bapak Kepala Departemen Teknik Komputer ITS Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
3. Bapak Reza Fuad Rachmadi, S.T., M.T., Ph.D., dan Bapak Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. yang senantiasa memberikan motivasi, bimbingan, dan arahan selama mengerjakan penelitian.
4. Bapak Ahmad Zaini, ST., M.Sc., selaku dosen wali yang telah memberikan motivasi dan dukungan dalam pelaksanaan penelitian ini.
5. Bapak dan Ibu dosen pengajar Departemen Teknik Komputer ITS, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
6. Seluruh staf Departemen Teknik Komputer ITS yang telah membantu penulis dalam hal administrasi dan perlengkapan penelitian.
7. Seluruh mahasiswa Fakultas Teknologi Elektro angkatan 2013, 2014, 2015, 2016, dan seterusnya yang telah menjadi saudara terbaik selama masa studi saya.
8. Seluruh teman - teman IYD yang sudah menemani keseharian saya dan memberikan motivasi selama masa studi saya (Andhika, Adri, Rivian, Surya, dan yang lainnya).
9. Teman seperjuangan dalam penelitian berbasis CNN (Aufa,

Bagus, Fahmi, Kentani).

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

Surabaya, Juli 2020

Deu Aldo Dhavicky

DAFTAR ISI

Abstrak	i
Abstract	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
NUMENKLATUR	xiii
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan masalah	2
1.5 Sistematika Penulisan	3
2 TINJAUAN PUSTAKA	5
2.1 <i>Related Work</i>	5
2.1.1 <i>Counting Crowd with Fully Convolutional Networks</i>	5
2.1.2 <i>Single-Image Crowd Counting via Multi Column Convolutional Neural Network</i>	5
2.2 <i>Machine Learning</i>	6
2.3 <i>Deep Learning</i>	7
2.4 <i>Convolutional Neural Network</i>	7
2.4.1 <i>Convolutional Layer</i>	8
2.4.2 <i>Fully-Connected Layer</i>	11
2.4.3 <i>Pooling Layer</i>	11
2.4.4 <i>Activation Function</i>	12

3	DESAIN DAN IMPLEMENTASI SISTEM	15
3.1	Desain Sistem	16
3.2	Dataset	17
	3.2.1 Mengolah dataset	17
	3.2.2 <i>Split</i>	18
3.3	<i>Training</i>	19
	3.3.1 <i>Training Models</i>	19
3.4	Evaluasi	21
4	PENGUJIAN DAN ANALISIS	23
4.1	<i>Training Data</i>	23
4.2	Pengujian	24
	4.2.1 Pengujian Model Pertama	24
	4.2.2 Pengujian Model Kedua	26
4.3	Analisa	28
5	PENUTUP	31
5.1	Kesimpulan	31
5.2	Saran	31
	DAFTAR PUSTAKA	33
	Biografi Penulis	35

DAFTAR GAMBAR

2.1	<i>layer</i> yang terdapat pada CNN	8
2.2	<i>Convolutional Layer</i>	10
2.3	<i>Pooling layer</i>	12
3.1	Diagram alir	15
3.2	<i>Processing</i> dataset	18
3.3	Susunan arsitektur pada model pertama	20
3.4	Susunan arsitektur pada model kedua	21
4.1	Hasil estimasi density Part A; dan Hasil estimasi density Part B	23
4.2	Hasil <i>density map</i> beserta gambar asli pada Part A .	24
4.3	Hasil <i>density map</i> beserta gambar asli pada Part B .	24
4.4	Grafik nilai <i>accuracy training</i> model pertama	25
4.5	Hasil <i>density map</i> pada Cascaded-MtL beserta gambar asli	27
4.6	Grafik nilai <i>error</i> pada <i>training</i> untuk kedua model .	28
4.7	Grafik nilai perbandingan <i>error</i> pada kedua model .	29

Halaman ini sengaja dikosongkan

DAFTAR TABEL

3.1	<i>ShanghaiTech</i>	17
4.1	<i>Hasil test pada Sanghaitech dataset</i>	25
4.2	<i>Hasil Estimasi MCNN</i>	26
4.3	<i>Hasil Estimasi CMTL</i>	27

Halaman ini sengaja dikosongkan

NUMENKLATUR

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

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

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

1.1 Latar belakang

Pada tempat – tempat yang terdapat banyak kerumunan orang (seperti stadion, alun-alun kota, pusat pameran dan sebagainya), terutama di pintu masuk atau persimpangan merupakan fenomena yang belakangan cukup membutuhkan perhatian. Terdapat populasi orang dengan jumlah besar yang berkumpul pada suatu titik di area tertentu akan memunculkan berbagai masalah , salah satunya keamanan. Fenomena tersebut memunculkan risiko adanya tindak kejahatan hingga terjadinya kecelakaan yang dapat mengakibatkan cedera serius atau berakibat kematian. Oleh karena itu dibutuhkan manajemen informasi yang baik mengenai fenomena jumlah orang dan kepadatan kerumunan yang ada di suatu tempat[1]. Sistem penghitungan orang (people counting) adalah sebuah sistem yang mampu menghitung orang berdasarkan citra yang ditangkap, dengan tingkat pencahayaan yang cukup [2]. Masalah penghitungan orang ini sebenarnya dapat diselesaikan dengan cara konvensional seperti penghitungan secara manual atau menggunakan sensor, namun hal ini memiliki kelemahan, selain memakai sumber daya manusia, cara tersebut juga membuat tidak nyaman dan membatasi ruang gerak. Sejumlah pendekatan telah diusulkan untuk mengatasi perhitungan kerumunan dan mereka dapat diklasifikasikan menjadi tiga jenis metode, berbasis deteksi objek, berbasis regresi global dan berbasis estimasi kepadatan Metode berbasis deteksi objek [3] selalu menggunakan detektor objek visual, yang akan menangkap objek individu yang terdapat dalam gambar kemudian menjumlahkan objek tersebut. Tetapi, sulit untuk mendapatkan hasil objek individu saat keadaan ramai dengan kondisi hasil tangkap citra kurang baik, ukuran kecil atau adanya blur. Karena metode berbasis deteksi objek kurang dapat diaplikasikan pada keadaan de-

ngan situasi tertentu, sejumlah metode lain mencoba memperkirakan jumlah kerumunan dengan regresi dengan fitur global. Metode regresi termasuk ridge regression, Neural network dan beberapa lainnya. Dibandingkan dengan metode berbasis deteksi, penghitungan secara regresi global dapat lebih di aplikasikan dalam situasi yang lebih sulit. Selanjutnya metode berbasis estimasi kepadatan, yang dapat mempertahankan lebih banyak informasi spasial dari citra yang di dapat[4]. Oleh karena itu, pada tugas akhir ini diusulkan sebuah pengembangan terhadap estimasi jumlah dan kepadatan pada kerumunan menggunakan metode Convolutional Neural Network (CNN).

1.2 Permasalahan

Perumusan masalah dalam penelitian ini adalah diperlukan adanya pendekatan untuk melakukan perhitungan jumlah orang dengan estimasi jumlah dan kepadatan orang berbasis Convolutional Neural Network (CNN).

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk melakukan pengembangan terhadap estimasi jumlah orang dengan memberikan peta informasi kepadatan orang di suatu tempat dengan menggunakan CNN. Harapannya sebagai pengawasan di tempat-tempat berkumpulnya orang dalam jumlah besar.

1.4 Batasan masalah

Untuk memfokuskan permasalahan yang diangkat maka dilakukan pembatasan masalah. Berikut batasan masalah dalam pengerjaan tugas akhir ini, yaitu:

1. Kondisi yang akan di analisa adalah yang terlihat oleh kamera.
2. Data *input* berupa citra tunggal yang digunakan merupakan hasil data dari kamera
3. Citra tunggal yang digunakan adalah yang tersedia untuk publik
4. Proses pendeteksian pada intensitas cahaya yang cukup.
5. Metode yang digunakan adalah Convolutional Neural Network (CNN)

1.5 Sistematika Penulisan

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

1. BAB I Pendahuluan

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

2. BAB II Tinjauan Pustaka

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

3. BAB III Desain dan Implementasi Sistem

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

4. BAB IV Pengujian dan Analisis

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

Halaman ini sengaja dikosongkan

BAB 2

TINJAUAN PUSTAKA

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

2.1 *Related Work*

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

2.1.1 *Counting Crowd with Fully Convolutional Networks*

Penelitian berjudul *Counting Crowd with Fully Convolutional Networks* yang dilakukan oleh Jianyong Wang, Lu Wang, dan Fenglei Yang ini mengusulkan pendekatanyang dipelajari secara menyeluruh dari patch gambar dengan merevisi distribusi kepadatan kerumunan. Model kerumunan FCN ini dapat menampilkan peta kepadatan kerumunan tinggi dan kuantitas kerumunan dapat diintegrasikan oleh peta. Selain itu, untuk menangani masalah distorsi perspektif adegan, kami mengusulkan metode generasi kebenaran kebenaran kepadatan tanah. Hasil percobaan menunjukkan bahwa metode penghitungan kerumunan kami mencapai akurasi terbaik dengan menggunakan dataset WorldExpo 10 dibandingkan dengan metode lainnya.[4].

2.1.2 *Single-Image Crowd Counting via Multi Column Convolutional Neural Network*

Penelitian berjudul *Single-Image Crowd Counting via Multi-Column Convolutional Neural Network*[5] yang dilakukan oleh Yinying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, dan Yi Ma ini merupakan pendekatan yang sederhana namun efektif untuk memetakan gambar ke peta keramaian keramaian. MCNN yang diusulkan memungkinkan gambar input berukuran atau resolusi yang berubah-ubah. Dengan menggunakan filter dengan bidang reseptif dari berbagai ukuran, fitur yang dipelajari oleh setiap kolom CNN adaptif terhadap variasi dalam ukuran orang / kepala karena efek perspektif atau resolusi gambar. Selain itu, peta kerapatan sebe-

narnya dihitung secara akurat berdasarkan kernel geometri-adaptif yang tidak perlu mengetahui peta perspektif gambar input. Karena kumpulan data penghitungan kerumunan yang ada tidak cukup untuk mencakup semua situasi yang menantang, telah mengumpulkan dan memberi label dataset baru yang besar yang mencakup 1.198 gambar dengan sekitar 330.000 kepala . Pada dataset baru ini, beserta semua dataset yang ada, dilakukan percobaan ekstensif untuk memverifikasi efektivitas model dan metode yang diusulkan. Secara khusus, dengan model MCNN sederhana yang diusulkan. Selain itu, percobaan menunjukkan bahwa model yang disebut, setelah dilatih pada satu dataset, dapat dengan mudah ditransfer ke dataset baru.

2.2 *Machine Learning*

Machine Learning (ML) merupakan sebuah cabang dari *Artificial Intelligence* (kecerdasan buatan) yang memungkinkan sistem komputer untuk belajar langsung dari contoh, data, dan pengalaman yang didapatkannya. Dengan membuat komputer mampu menjalankan tugas yang spesifik secara cerdas, ML dapat melakukan algoritma yang kompleks dengan cara mempelajari data yang diberikan tanpa perlu mengikuti program yang dibuat atau diinstruksikan secara manual oleh pengguna[6].

Ciri khas dari ML adalah adanya proses pembelajaran atau *training*. Oleh karena itu, ML membutuhkan data untuk dipelajari yang disebut sebagai data *training*. Setelah berhasil melakukan *training*, maka ML dapat melakukan proses klasifikasi dan prediksi terhadap data baru yang diberikan sesuai dengan hasil *training* yang telah dilakukan. Klasifikasi adalah metode dalam ML yang digunakan oleh mesin untuk memilah atau mengklasifikasikan objek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan. Beberapa contoh implementasi dari ML antara lain adalah *image processing*, *search recommendation*, *text and speech recognition*, dan *health diagnosis*[7].

Di dalam ML, terdapat tiga kategori pembelajaran[8], yaitu:

1. *Supervised Learning*

Supervised learning merupakan metode pembelajaran dimana data yang akan dipelajari sudah diberikan label yang menun-

jukkan klasifikasi atau kelas data tersebut berada. Model yang dihasilkan adalah model prediksi dari data yang telah diberi label.

2. *Unsupervised Learning*

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

3. *Reinforcement Learning*

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

2.3 *Deep Learning*

Deep Learning (DL) adalah salah satu bidang yang muncul dari penelitian ML. DL memungkinkan model komputasi yang tersusun dari beberapa lapisan pemrosesan untuk mempelajari representasi dari data dengan berbagai level abstraksi. DL menemukan struktur sulit yang terdapat di dalam kumpulan data yang besar dengan menggunakan algoritma *backpropagation*. Struktur yang didapatkan menunjukkan parameter internal apa yang harus diubah oleh mesin agar dapat menghitung representasi di setiap *layer* berdasarkan representasi dari *layer* sebelumnya[9].

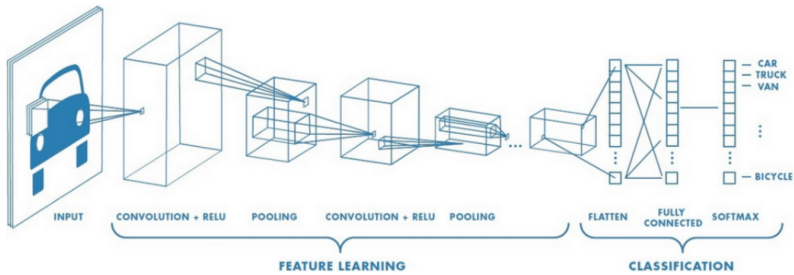
DL melakukan pendekatan dalam penyelesaian masalah dengan menggunakan konsep hierarki. Dengan konsep tersebut, komputer mampu mempelajari sebuah konsep yang kompleks dengan menggabungkan konsep-konsep yang lebih sederhana[10].

2.4 *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah variasi dari *Multilayer Perceptron* yang terinspirasi dari jaringan saraf manusia. Pemberian nama *convolutional neural network* mengindikasikan bahwa jaringan tersebut menggunakan operasi matematika yang dise-

but konvolusi. Konvolusi sendiri adalah sebuah operasi linier. CNN merupakan pengembangan dari *artificial neural network* yang saat ini diklaim sebagai model terbaik untuk memecahkan masalah seputar *object recognition* dan *detection*.

Secara teknis, CNN adalah arsitektur yang bisa di *training* dan terdiri dari beberapa tahap. Input dan output dari masing-masing tahap berupa *array* yang disebut *feature map* atau peta fitur. *Output* dari masing-masing tahap adalah *feature map* hasil pengolahan dari semua lokasi pada input. Struktur CNN dibangun dari tiga jenis layer utama yaitu *convolution layer*, *pooling layer*, dan *activation function*[11]. Susunan *layer* yang terdapat pada CNN dapat dilihat pada gambar 2.1.



Gambar 2.1: *layer* yang terdapat pada CNN[12].

2.4.1 Convolutional Layer

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

untuk mengekstraksi fitur dari citra *input*. Secara umum operasi konvolusi dapat dituliskan dengan persamaan 2.1.

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

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

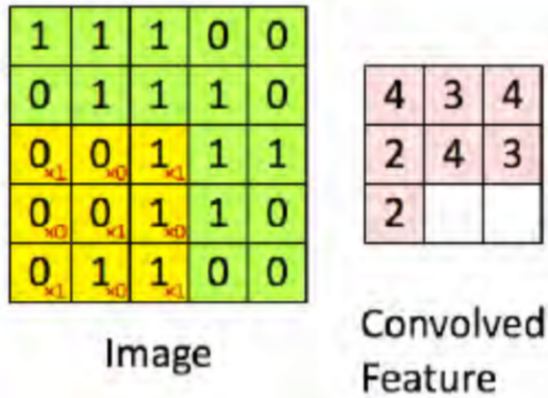
$$S_{(i,j)} = (K \times I)_{(i,j)} = \sum \sum I_{(i-m,j-n)} K_{(m,n)} \quad (2.2)$$

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

Parameter yang menentukan berapa jumlah pergeseran filter disebut "*Stride*". Jika nilai stride adalah 1, maka *convolutioal filter* akan bergeser sebanyak 1 *pixels* secara horizontal lalu vertical. Semakin kecil *stride* maka akan semakin detail informasi yang akan didapatkan dari sebuah *input*, namun membutuhkan proses komputasi yang lebih jika dibandingkan dengan *stride* yang besar. Perlu diperhatikan bahwa dengan menggunakan *stride* bernilai kecil tidak selalu akan mendapatkan performa yang bagus. Persamaan 2.3 digunakan untuk menghitung dimensi dari *feature map*.

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

Padding atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input. Dimensi *output* dari *convolutional layer* selalu lebih kecil dari *input*-nya (kecuali penggunaan 1×1 filter dengan jumlah



Gambar 2.2: Convolutional Layer

stride 1). *Output* ini akan digunakan kembali sebagai input dari convolutional layer selanjutnya, sehingga informasi akan banyak yang terbuang. Dengan menggunakan *padding* dapat mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis, sehingga penggunaan *convolutional layer* yang lebih dalam dapat menghasilkan lebih banyak *features* yang di-*extract*[13]. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan *input*.

2.4.1.1 Stride

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

2.4.1.2 *Padding*

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

2.4.1.3 *Loss Function*

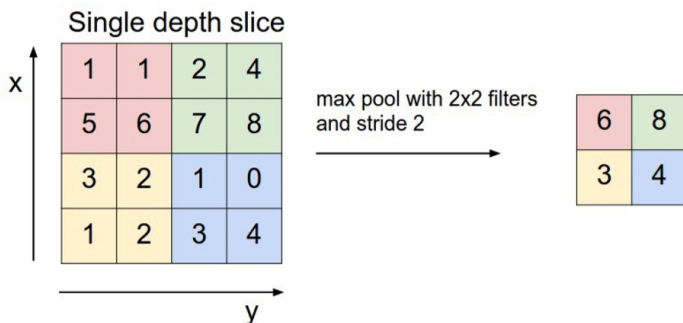
Loss function merupakan fungsi yang menggambarkan kerugian yang terkait dengan semua kemungkinan yang dihasilkan oleh sebuah model. *Loss function* ini bekerja ketika model pembelajaran memberikan kesalahan yang harus diperhatikan. *Loss function* yang baik memberikan nilai *error* yang rendah.

2.4.2 *Fully-Connected Layer*

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

2.4.3 *Pooling Layer*

Fungsi dari *pooling* ini adalah untuk mereduksi input secara *spasial* (mengurangi jumlah parameter) dengan operasi *down-sampling*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max pooling* adalah nilai maksimal. Lapisan *pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume output pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *overfitting*. Ber-



Gambar 2.3: *Pooling layer*[13].

dasarkan gambar 2.3 menunjukkan proses dari *max-pooling*. Output dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan dimensi citra awal. Lapisan *pooling* diatas akan beroperasi pada setiap irisan kedalaman volume input secara bergantian. Jika dilihat dari Gambar 2.3 operasi *max pooling* menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4, dari masing-masing 4 angka pada *input* operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2[14].

2.4.4 *Activation Function*

Activation function atau fungsi aktivasi merupakan sebuah *node* yang ditambahkan di akhir *output* dari setiap *neural network*. Fungsi aktivasi juga dikenal sebagai *transfer function* yang digunakan untuk menentukan *output* dari *neural network*. Fungsi aktivasi dibagi menjadi dua tipe yaitu linier dan *non*-linier. Pada arsitektur CNN, fungsi aktivasi terletak pada perhitungan akhir *output feature map*, sesudah proses perhitungan konvolusi atau *pooling*, untuk menghasilkan suatu pola fitur. Beberapa macam fungsi aktivasi yang sering digunakan dalam penelitian antara lain fungsi *softmax*, sigmoid, tanh, dan *Rectified Linear Unit* (ReLU).[13]

2.4.4.1 *Rectified Linear Unit (ReLU)*

Pada dasarnya fungsi ReLU melakukan *threshold* dari 0 hingga tidak terhingga (*infinity*). Fungsi ini menjadi salah satu fungsi aktivasi yang populer saat ini. Pada fungsi ini input dari neuron - neuron berupa bilangan negatif akan diterjemahkan kedalam nilai 0, dan jika masukan bernilai positif maka output dari neuron adalah nilai aktivasi itu sendiri. ReLU pada intinya hanya membuat pembatas pada bilangan nol.[13]

2.4.4.2 *Softmax*

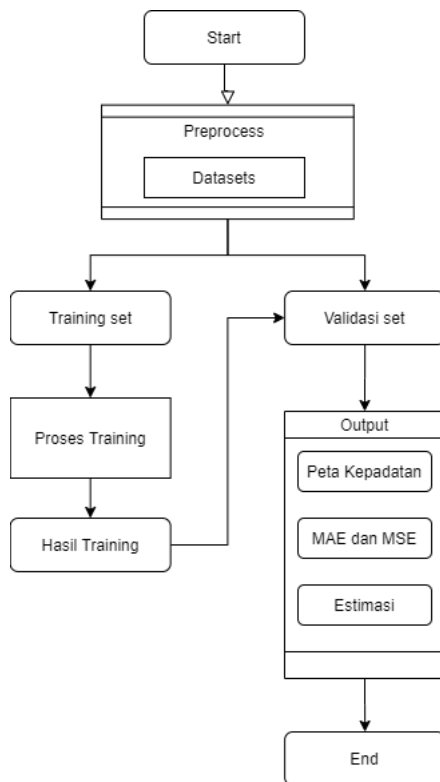
Fungsi aktivasi Softmax adalah contoh lain dari fungsi aktivasi yang digunakan dalam komputasi saraf atau *neural computing*. Fungsi aktivasi ini digunakan untuk menghitung probabilitas distribusi dari vektor bilangan real. Fungsi aktivasi Softmax menghasilkan output yang merupakan kisaran nilai antara 0 dan 1, dengan jumlah probabilitas sama dengan 1. Fungsi Softmax digunakan dalam model *multi-class* dimana ia mengembalikan probabilitas setiap kelas, dengan kelas target memiliki probabilitas tertinggi. Fungsi aktivasi Softmax sebagian besar muncul di hampir semua lapisan output dari arsitektur *deep learning*. Perbedaan utama antara fungsi aktivasi Sigmoid dan Softmax adalah fungsi aktivasi Sigmoid digunakan dalam klasifikasi biner sedangkan fungsi aktivasi Softmax digunakan untuk klasifikasi multivarian[15].

Halaman ini sengaja dikosongkan

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain diagram berikut. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan.



Gambar 3.1: Diagram alir

Tujuan dari penelitian ini adalah untuk melakukan estimasi jumlah dan memberikan peta kepadatan orang di suatu tempat berbasis citra tunggal dengan menggunakan metode *Convolutional Neural Network*. Proses kerja dari penelitian ini ditunjukkan pada gambar 3.1. Berdasarkan pada gambar 3.1, salah satu proses yang dilakukan adalah melakukan sebuah pendekatan pada peta kepadatan. Dataset di berikan koordinat perkiraan posisi tengah setiap kepala manusia di setiap gambar kerumunan yang telah diberi label secara manual. Agar bisa mencapai hal tersebut dataset yang digunakan untuk *training* dalam penelitian ini dilakukan pre-process dataset dengan menggunakan *Kernel Gaussian*. Sehingga didapatkan wilayah yang diubah menjadi probabilitas bahwa wilayah tersebut merupakan kepala manusia. Setelah mendapatkan dataset, langkah selanjutnya adalah pembuatan *split* yang berfungsi untuk memisahkan dataset yang digunakan untuk *training* dan *testing* atau validasi. Metode yang digunakan pada penelitian ini adalah CNN dengan menggunakan arsitektur berdasarkan pada model *Multi-Column Convolutional Neural Network* dan *Cascaded Multi-task Learning*. Hasil dari *training* tersebut adalah *weight* yang berupa model terbaik dalam proses training sebagai validation set. Proses selanjutnya adalah validasi model, yaitu proses dimana validasi model yang telah di proses dari dataset yang digunakan yang akan menghasilkan peta kepadatan.

3.1 Desain Sistem

Secara garis besar, terdapat empat proses yang ada pada bagian ini.

1. Dataset.

Dataset yang digunakan pada tugas akhir ini adalah dataset *ShanghaiTech* yang berisi 1198 gambar yang telah diberikan label, dengan total 330.165 objek orang pada gambar yang telah dilabeli.

2. *Preprocessing* data.

Sebelum bisa digunakan, dataset perlu diproses terlebih dahulu. Dataset yang akan digunakan akan diubah menjadi 9 bagian set gambar, 9 sub-bagian tersebut memiliki masing masing ukuran $1/4$ dari ukuran gambar aslinya. Kemudian

dilakukan *split* untuk memisahkan data yang akan digunakan sebagai input *training* dan validasi

3. *Training*.

Training merupakan sebuah proses pelatihan model CNN menggunakan dataset *ShanghaiTech*. *Training* pada sistem ini akan dilakukan dengan menggunakan model dari Multi-Column Convolutional Neural Network.

4. Evaluasi.

Model yang telah melewati proses *training* kemudian akan dievaluasi untuk menilai performanya. Evaluasi model akan melakukan pengujian terhadap model hasil training.

3.2 Dataset

Tabel 3.1: *ShanghaiTech*

Fitur	Part A	Part B
<i>Resolusi</i>	berbeda	768 x 1024
<i>Jumlah</i>	482	716
<i>Label total</i>	241.667	88.488

Pada penelitian ini dibuat dengan menggunakan dataset Shanghai tech yang berisi 2 jenis bagian, yaitu Part A dan Part B. Pada bagian Part A dataset tersebut 482 gambar yang terdapat didalamnya merupakan kumpulan gambar yang didapat dari *Internet* dan dengan menggunakan *keyword* pencarian yang bervariasi. Sedangkan pada bagian Part B dataset tersebut terdapat sebanyak 716 gambar yang terdapat didalamnya yang didapat dari jalanan metropolitan pada daerah *Shanghai*. Kedua data tersebut baik Part A dan Part B kemudian dibedakan lagi untuk *training* dan *testing*: 300 gambar pada Part A digunakan untuk *training* dan 182 gambar untuk *testing*; serta 400 gambar pada Part B untuk *training* dan 316 gambar untuk *testing*. Selanjutnya, dataset akan diproses sesuai dengan gambar 3.1.

3.2.1 Mengolah dataset

Karena dataset yang digunakan adalah gambar, dataset yang telah didapat diubah dengan menggunakan program yang sudah

dibuat. Dataset diubah menjadi 9 bagian set gambar, 9 sub-bagian tersebut memiliki masing masing ukuran 1/4 dari ukuran gambar aslinya.



Gambar 3.2: *Processing* dataset

3.2.2 *Split*

Split Dataset Shanghai tech yang berisi 2 jenis bagian, yaitu Part A dan Part B. Pada bagian Part A dataset tersebut 482 gambar yang terdapat didalamnya merupakan kumpulan gambar yang didapat dari *Internet* dan dengan menggunakan *keyword* pencarian yang bervariasi. Sedangkan pada bagian Part B dataset tersebut terdapat sebanyak 716 gambar yang terdapat didalamnya yang didapat dari jalanan metropolitan pada daerah *Shanghai*. Kedua data tersebut baik Part A dan Part B kemudian dibedakan lagi untuk *training* dan *testing*: 300 gambar pada Part A digunakan untuk *training* dan 182 gambar untuk *testing*; serta 400 gambar pada Part B untuk *training* dan 316 gambar untuk *testing*.

3.3 Training

Pada proses *training* ini, digunakan data *train* yang sudah diubah menjadi sekumpulan *sub-block* yang kemudian akan di-*resize* menjadi berukuran 1/4 ukuran asli, lalu digunakan untuk proses *training* sehingga membentuk sebuah pola yang hasilnya berbentuk bobot atau *weight*. *Training* dilakukan dengan menggunakan model *Multi-Column Convolutional Neural Networks* atau MCNN dan menggunakan *Cascaded Multi-task Learning of High-level Prior and Density* yang berbasis CNN disebut juga Cascaded-MtL. Pada proses *training* ini perlu dilakukan penentuan ukuran *learning rate* dan *epoch* yang akan dijelaskan sebagai berikut :

1. *Learning Rate*

Learning Rate merupakan sebagai parameter kontrol untuk mengubah model dalam menanggapi perkiraan error setiap kali bobot model yang diperbarui.

2. *Epoch*

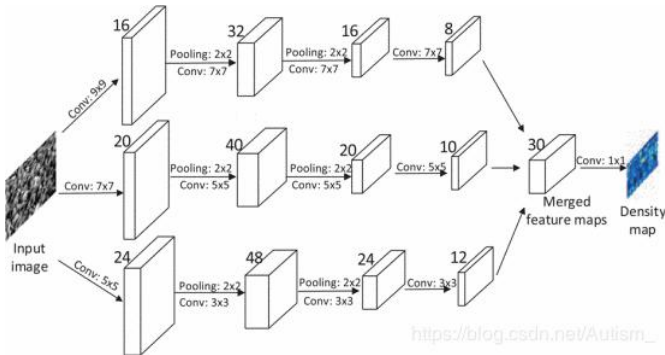
Epoch merupakan satu set putaran *training steps*. Beberapa *epoch* diperlukan untuk pelatihan sebuah neural network sehingga didapatkan kesalahan (*loss*) mendekati nol. Jumlah *epoch* pada proses *training model* ini sejumlah 2000 kali.

3.3.1 Training Models

Pada penelitian ini, digunakan model MCNN dan Cascaded-MtL. MCNN yang diusulkan memungkinkan gambar input berukuran atau resolusi yang berubah-ubah. Dengan menggunakan filter dengan bidang reseptif dari berbagai ukuran, fitur yang dipelajari oleh setiap kolom CNN adaptif terhadap variasi dalam ukuran orang / kepala karena efek perspektif atau resolusi gambar. Selain itu, peta kerapatan sebenarnya dihitung secara akurat berdasarkan kernel geometri-adaptif yang tidak perlu mengetahui peta perspektif gambar input. Sedangkan Cascaded-MtL adalah untuk mempelajari model-model yang memenuhi berbagai tingkat kepadatan yang disajikan dengan menggabungkan *high-level prior* sebelum masuk ke network. *high-level prior* di tujukan untuk mengklasifikasikan penghitungan ke dalam berbagai kelompok yang label kelasnya didasarkan pada jumlah orang yang hadir dalam gambar. Dengan mengeksploitasi jumlah label, metode ini memiliki kemampuan un-

tuk menghitung jumlah orang di dalam peta, terlepas dari berbagai macam skala yang digunakan dalam jaringan.

3.3.1.1 Model Pertama

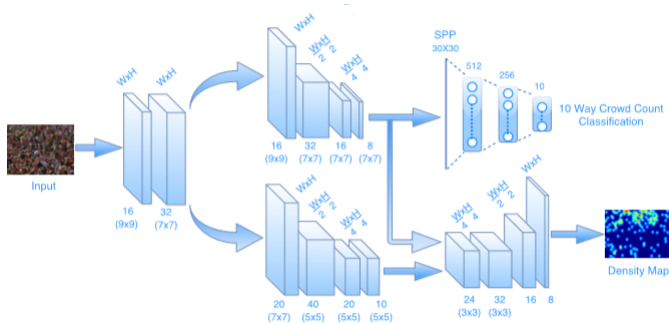


Gambar 3.3: Susunan arsitektur pada model pertama

Model pertama yang digunakan merupakan model MCNN. Sesuai pada Gambar 3.3, model ini terdiri dari 3 kolom *convolutional neural network* yang masing-masing dilambangkan dengan *L-column kernel* berukuran 9×9 , 7×7 , 7×7 , 7×7 Kemudian *M-column kernel* yang berukuran sedang yaitu 7×7 , 5×5 , 5×5 , 5×5 dan *S-column kernel* yang merupakan kernel berskala kecil 5×5 , 3×3 , 3×3 , 7×7 . Masing - masing menggunakan *2D Convolution Layer* yang diikuti dengan *2D max pooling layer* dengan ukuran 2×2 , pada setiap layer 1 dan layer 2 pada masing masing kernel. Masing - masing merupakan *fully-connected layer* dan model ini menggunakan *activation function* ReLU.

3.3.1.2 Model Kedua

Model kedua memiliki dua tahap yang sesuai dengan dua sub-tugas, dengan tahap pertama menggunakan *high-level prior* dan tahap kedua memperkirakan estimasi peta kepadatan. Kedua tahap berbagi seperangkat fitur konvolusional. Cascaded-MtL ini sesuai dengan Gambar 3.4 terdiri dari dari 2 lapisan konvolusional dengan fungsi aktivasi *Parametric Rectified Linear Unit* (PReLU)



Gambar 3.4: Susunan arsitektur pada model kedua

setelah setiap lapisan. Lapisan konvolusional pertama memiliki 16 peta fitur dengan ukuran filter 9×9 dan konvolusi kedua mungkin memiliki fitur tambahan dengan ukuran filter 7×7 . Untuk estimasi kepadatan didapatkan dari lapisan kedua diproses oleh jaringan CNN lain yang terdiri dari 4 lapisan konvolusional dengan fungsi aktivasi PReLU setelah setiap lapisan. Dua lapisan pertama diikuti oleh max pooling layers dengan stride 2. Lapisan konvolusional pertama memiliki 20 peta fitur dengan ukuran filter 7×7 , layer konvolusional kedua memiliki 40 peta fitur dengan ukuran filter 5×5 , layer ketiga memiliki 20 peta fitur dengan ukuran filter 5×5 dan lapisan keempat memiliki 10 peta fitur dengan ukuran filter 5×5 . Output dari jaringan ini dikombinasikan dengan lapisan konvolusional terakhir dari tingkat *high-level prior* menggunakan setelah 2 konvolusi. Dua lapisan konvolusional pertama pada lapisan terakhir ini memiliki ukuran filter 3×3 dengan 24 dan 32 peta fitur konvolusi.

3.4 Evaluasi

Setelah proses *training* selesai, proses selanjutnya adalah melakukan evaluasi terhadap hasil *training* tersebut. Evaluasi ini bertujuan untuk melakukan pengecekan terhadap proses training model yang telah dilakukan. Untuk evaluasi model digunakan metode dengan *Mean Absolute Error* (MAE) dan *Mean Square Error* (MSE).

Halaman ini sengaja dikosongkan

BAB 4

PENGUJIAN DAN ANALISIS

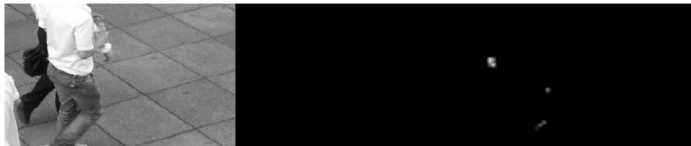
Pada penelitian ini hasil pengujian serta analisis dipaparkan dari desain model dan implementasi. Data yang digunakan dalam pengujian pada penelitian ini sudah disediakan di dalam Shanghai-Tech dataset . Pengujian dilakukan dalam beberapa bagian sebagai berikut :

4.1 *Training Data*

Proses *training* data dilakukan sebanyak 10 kali dengan menggunakan *Convolutional Neural Network* dengan menggunakan model MCNN pada Gambar 3.3 dan Cascaded-MtL dengan model arsitektur seperti pada Gambar 3.4 yang telah dimodifikasi dengan menggunakan python. Training dilakukan dengan menggunakan GPU NVIDIA GEFORCE GTX 1070 6GB. Proses training berlangsung selama kurang lebih 7 hari dengan menggunakan 2000 epoch, dan *learning rate* 0.000001.



(a)



(b)

Gambar 4.1: *a* Hasil estimasi density Part A; *b* Hasil estimasi density Part B

4.2 Pengujian

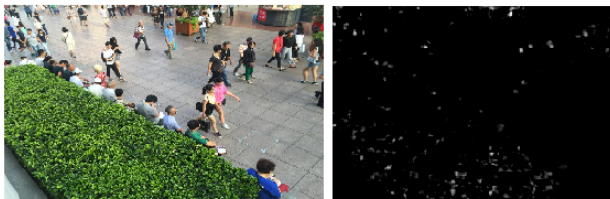
Pada bagian pengujian ini, akan digunakan gambar uji dari *ShanghaiTech* dataset yang memiliki spesifikasi 182 gambar uji coba untuk Part A dengan ukuran gambar yang beragam, untuk uji coba pada Part B digunakan 316 gambar uji coba dengan spesifikasi 768×1024 . Hasil dari proses merupakan *density map*. Disertai hasil evaluasi model yang berbentuk *Mean Absolute Error* (MAE) dan *Mean Square Error* (MSE). Pada pengujian ini juga digunakan 14 gambar yang didapatkan dari Internet, berupa gambar keramaian yang berada pada wilayah sekitar Kota Surabaya.

4.2.1 Pengujian Model Pertama

Pengujian pertama dilakukan dengan menggunakan *weight* yang didapatkan dari hasil *training* menggunakan MCNN. Berikut merupakan *density* dari kedua model uji coba:



Gambar 4.2: Hasil *density map* beserta gambar asli pada Part A



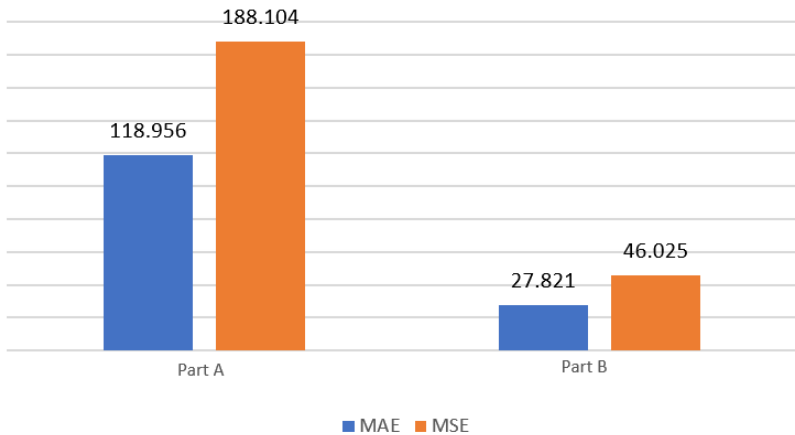
Gambar 4.3: Hasil *density map* beserta gambar asli pada Part B

Berikut merupakan tabel hasil dari *metrics validation* dengan

menggunakan Mean Absolut Error (MAE) dan Mean Square Error (MSE).

Tabel 4.1: Hasil test pada Sanghaitech dataset

Part_A			Part_B		
Test	MAE	MSE	Test	MAE	MSE
1	118.67	187.97	1	28,47	46,79
2	120.42	188.97	2	28,41	46,73
3	118.60	187.82	3	27,31	45,37
4	117.87	187.58	4	27,35	45,48
5	120.35	188.99	5	28,41	46,89
6	118.74	187.72	6	29,66	48,43
7	120.00	188.74	7	25,22	42,61
8	118.72	187.95	8	27,5	45,66
9	118.15	187.62	9	27,95	46,21
10	118.40	187.68	10	27,93	46,08
Average	118,956	188,104	Average	27,821	46,025



Gambar 4.4: Grafik nilai *accuracy training* model pertama.

Dari grafik Gambar 4.4 menunjukkan bahwa nilai akurasi pada proses *training* pertama memiliki 2 hasil berbeda pada Part A dan

Part B. Pada model ini didapatkan bahwa nilai akurasi sesuai dengan keadaan kepadatan yang terjadi pada gambar . Pada model pertama juga dilakukan pengujian dengan menggunakan 14 data gambar yang di peroleh dari internet dengan hasil seperti pada tabel berikut Tabel 4.2 menunjukan hasil estimasi yang menggunakan

Tabel 4.2: Hasil Estimasi MCNN

MCNN			
Image	ground_truth	Estimasi MCNN	MAE
taman bungkul	726	507,79	218,21
cfid_1	600	540,56	59,44
cfid_2	65	264,75	199,75
cfid_3	185	170,49	14,51
cfid_4	265	297,55	32,55
cfid_5	210	176,19	33,81
konser dewa19	106	74,21	31,79
grandcity_7	115	302,45	187,45
grandcity_9	87	626,14	539,14
grandcity_10	140	70,13	69,87
grandcity_11	78	238,11	160,11
kn_12	335	1041,15	706,15
kn_13	362	1115,41	753,41
kn_14	372	1544,62	1217,62

an 14 gambar dengan menyertakan Mean Absolut Error dari setiap gambar yang digunakan sebagai pengujian yang menggunakan hasil proses *training* dari model pertama. Dari data tersebut bisa kita lihat bahwa model ini berhasil melakukan estimasi beberapa gambar pengujian mendekati jumlah asli pada gambar tersebut yang berupa ground truth.

4.2.2 Pengujian Model Kedua

Pengujian model ke dua dilakukan dengan menggunakan metode yang sama dengan model pertama didapatkan dari hasil *training* menggunakan Cascaded-MtL. Berikut merupakan density dari model uji coba:



Gambar 4.5: Hasil *density map* pada Cascaded-MtL beserta gambar asli

Pada model kedua juga dilakukan pengujian dengan menggunakan 14 data gambar yang di peroleh dari internet dengan hasil seperti pada tabel berikut Pada Tabel 4.3 menunjukan hasil esti-

Tabel 4.3: Hasil Estimasi CMTL

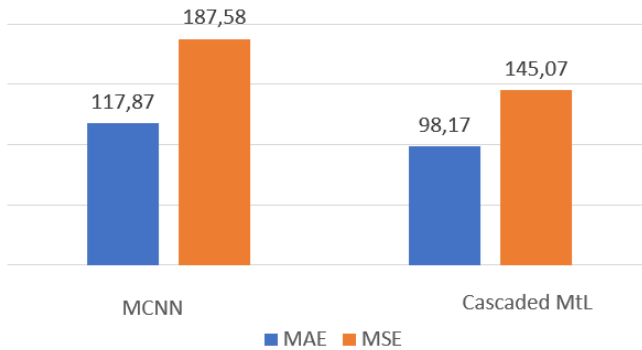
Cascaded-MtL			
Image	ground_truth	Estimasi CMTL	MAE
taman bungkul	726	507,79	247,88
cfid_1	600	540,56	151,13
cfid_2	65	264,75	75,72
cfid_3	185	170,49	8,19
cfid_4	265	297,55	5,7
cfid_5	210	176,19	65,11
konser dewa19	106	74,21	18,16
grandcity_7	115	302,45	138,44
grandcity_9	87	626,14	326,09
grandcity_10	140	70,13	3,02
grandcity_11	78	238,11	109,39
kn_12	335	1041,15	523,94
kn_13	362	1115,41	517,47
kn_14	372	1544,62	926,4

masi dari Cascaded Multi-task Learning yang juga menggunakan 14 gambar dengan menyertakan Mean Absolut Error dari setiap gambar yang digunakan sebagai pengujian, menggunakan hasil proses *training* dari model tersebut. Pada data tersebut bisa kita lihat bahwa model ini juga berhasil melakukan estimasi beberapa gam-

bar pengujian mendekati jumlah asli pada gambar tersebut.

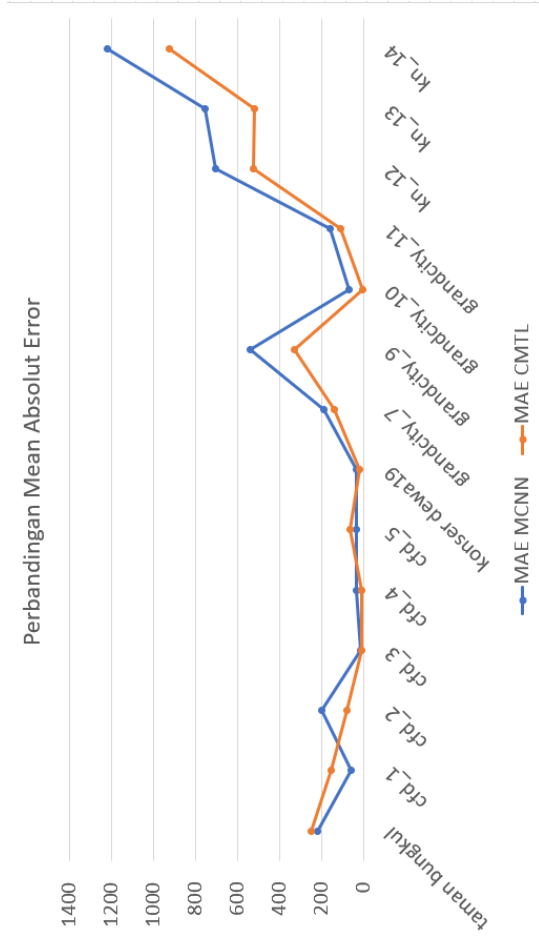
4.3 Analisa

Setelah dilakukan uji coba performa dengan menghadirkan MAE dan MSE pada ke dua model tersebut, didapatkan nilai rata - rata performa dari tiap model.



Gambar 4.6: Grafik nilai *error* pada *training* untuk kedua model.

Dari grafik 4.6 menunjukkan bahwa nilai akurasi pada proses *training* dengan menggunakan Shanghaitech dataset pada model ke 2 memiliki keunggulan hasil dibandingkan dengan model pertama. Berikutnya dilakukan uji coba dengan menggunakan 14 gambar yang didapat dari Internet sesuai dengan tabel 4.2 untuk prediksi dari MCNN dan tabel 4.3 untuk estimasi dari CMTL. Pada gambar 4.7 menunjukkan perbandingan bahwa model ke kedua lebih memiliki keunggulan pada prediksi gambar, yaitu lebih mendekati dari ground truth atau jumlah objek asli pada gambar yang di gunakan untuk melakukan pengujian.



Gambar 4.7: Grafik nilai perbandingan *error* pada kedua model.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

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

1. Berdasarkan hasil *training*, sesuai dengan gambar 4.1 model pertama mendapatkan nilai yang cukup baik pada Part B dikarenakan tingkat kepadatan objek pada gambar tersebut lebih rendah . Tapi, berdasarkan hasil perbandingan performa akurasi dengan model ke dua pada Grafik 4.6 , menunjukkan bahwa model kedua mendapatkan nilai akurasi lebih baik. Hal ini disebabkan pada Cascaded-MtL memiliki fitur layer *high-level prior* untuk mendapatkan hasil yang maksimal.
2. Menurut pengamatan saya, latar tempat pada gambar yang digunakan sebagai dataset juga dapat mempengaruhi hasil estimasi dan juga peta kepadatan. Sebagai contoh, bila pada dataset terjadi pada tempat yang terdapat banyak daun atau objek kepala memiliki fitur lain seperti topi, maka saat pengujian akan menyebabkan kesalahan pada estimasi.

5.2 Saran

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

1. Mencoba meningkatkan *epoch* yang digunakan dalam proses *training*.
2. Melakukan penelitian dengan mengubah model yang sudah ada untuk menemukan model yang memiliki performansi lebih bagus.
3. Melanjutkan penelitian ini ke tahap perhitungan secara *real-time*

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Z. Liu, Y. Chen, and K. Xie, “Research on the impact of crowd flow on crowd risk in large gathering spots,” in 2016 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), pp. 368–371, IEEE, 2016. (Dikutip pada halaman 1).
- [2] H. Celik, A. Hanjalic, and E. A. Hendriks, “Towards a robust solution to people counting,” in 2006 International Conference on Image Processing, pp. 2401–2404, IEEE, 2006. (Dikutip pada halaman 1).
- [3] J. Xing, H. Ai, L. Liu, and S. Lao, “Robust crowd counting using detection flow,” in 2011 18th IEEE International Conference on Image Processing, pp. 2061–2064, IEEE, 2011. (Dikutip pada halaman 1).
- [4] J. Wang, L. Wang, and F. Yang, “Counting crowd with fully convolutional networks,” in 2017 2nd International Conference on Multimedia and Image Processing (ICMIP), pp. 210–214, IEEE, 2017. (Dikutip pada halaman 2, 5).
- [5] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 589–597, 2016. (Dikutip pada halaman 5).
- [6] D. Sharma and N. Kumar, “A review on machine learning algorithms, tasks and applications,” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 10, 2017. (Dikutip pada halaman 6).
- [7] A. Ahmad, “Mengenai artificial intelligence, machine learning, neural network, dan deep learning,” no. October, 2017. (Dikutip pada halaman 6).

- [8] J. DeMuro, “What is a neural network?.” <https://www.techradar.com/news/what-is-a-neural-network>, 2018. Diakses: 2020-04-01. (Dikutip pada halaman 6).
- [9] Y. Bengio et al., “Learning deep architectures for ai,” Foundations and trends® in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009. (Dikutip pada halaman 7).
- [10] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016. (Dikutip pada halaman 7).
- [11] S. R. Dewi et al., “Deep learning object detection pada video menggunakan tensorflow dan convolutional neural network,” 2018. (Dikutip pada halaman 8).
- [12] “Convolutional neural network.” <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>. Accessed: 2020-03-15. (Dikutip pada halaman 8).
- [13] S. Sena, “Pengenalan deep learningpart 7: Convolutional neural network(cnn),” 2017. (Dikutip pada halaman 10, 12, 13).
- [14] T. Nurhikmat et al., “Implementasi deep learning untuk image classification menggunakan algoritma convolutional neural network (cnn) pada citra wayang golek,” 2018. (Dikutip pada halaman 12).
- [15] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” arXiv preprint arXiv:1811.03378, 2018. (Dikutip pada halaman 13).

BIOGRAFI PENULIS



Deu Aldo Dhavicky, lahir pada 24 Mei di Surabaya. Penulis merupakan anak pertama dari dua bersaudara. Saat ini penulis tinggal di Kota Surabaya, Provinsi Jawa Timur. Pada tahun 2009 menyelesaikan pendidikan di SD Kalirungkut 3. Tahun 2012 lulus dari SMP Negeri 35 Surabaya. Dan pada tahun 2015 lulus dari SMA Negeri 17 Surabaya. Penulis sedang menjalani studi di Departemen Teknik Komputer, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember. Penulis memiliki hobi bermain *game* dan membaca buku. Penulis berhasil menyelesaikan tugas akhir dengan judul "**Estimasi Citra Kerumunan Tunggal Menggunakan Metode *Convolutional Neural Network* (CNN)**". Bagi pembaca yang memiliki kritik, saran, atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui surel deudhavicky@gmail.com.

Halaman ini sengaja dikosongkan