



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

RE-IDENTIFIKASI ORANG MENGGUNAKAN LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORK PADA MULTI-MODAL IMAGE

Charles Chang
NRP 0721 17 4000 0027

Dosen Pembimbing
Dr. Reza Fuad Rachmadi, S.T., M.T.
Dr. I Ketut Eddy Purnama, S.T, M.T.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

**RE-IDENTIFIKASI ORANG MENGGUNAKAN
LIGHTWEIGHT CONVOLUTIONAL NEURAL
NETWORK PADA MULTI-MODAL IMAGE**

Charles Chang
NRP 0721 17 4000 0027

Dosen Pembimbing
Dr. Reza Fuad Rachmadi, S.T., M.T.
Dr. I Ketut Eddy Purnama, S.T, M.T.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

[Halaman ini sengaja dikosongkan]



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EC184801

Multi-Modal Person Re-Identification using Lightweight Convolutional Neural Network

Charles Chang
NRP 0721 17 4000 0027

Advisor
Dr. Reza Fuad Rachmadi, S.T., M.T.
Dr. I Ketut Eddy Purnama, S.T, M.T.

DEPARTMENT OF COMPUTER ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2021

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi buku Tugas Akhir saya dengan judul “Re-identifikasi Orang menggunakan Lightweight Convolutional Neural Network pada Multi-Modal Image” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahanbahan 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, Mei 2021

Charles Chang
07211740000027

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Re-identifikasi Orang menggunakan Lightweight Convolutional Neural Network pada Multi-Modal Image

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

Oleh: Charles Chang (NRP. 07211740000027)

Tanggal Ujian : _ 2021
Periode Wisuda : _ 2021

Disetujui Oleh:

Dr. Reza Fuad Rachmadi, S.T., M.T. (Pembimbing I)
NIP: 19850403 201212 1 001

Dr. I Ketut Eddy Purnama, S.T., M.T. (Pembimbing II)
NIP: 19690730 199512 1 001

- (Penguji I)
NIP: _

- (Penguji II)
NIP: _

- (Penguji III)
NIP: _

Mengetahui,
Kepala Departemen Teknik Komputer - ITS

Dr. Supeno Mardi Susiki Nugroho, ST., MT.
NIP. 19700313 199512 1 001

[Halaman ini sengaja dikosongkan]

ABSTRAK

Nama Mahasiswa : Charles Chang
Judul Tugas Akhir : Re-identifikasi Orang menggunakan Lightweight Convolutional Neural Network pada Multi-Modal Image
Pembimbing : 1. Dr. Reza Fuad Rachmadi, S.T., M.T.
2. Dr. I Ketut Eddy Purnama, S.T, M.T.

Sebagai alat pelengkap keamanan, sistem CCTV semakin banyak digunakan di setiap ruang publik untuk memantau dan menganalisa tindakan kriminal pada suatu lokasi. Akan tetapi, pencarian kriminal secara manual masih rentan akan kesalahan manusia. Salah satu solusi membuat pencarian kriminal lebih efektif dan efisien adalah dengan penggunaan Re-identifikasi.

Re-identifikasi merupakan sebuah teknik visi komputer dan deep learning dimana dilakukan pencarian ulang citra atau video milik sebuah identitas. Pada tugas akhir ini, akan dipelajari metode Re-identifikasi orang dengan citra multi-modal, adapun data Input yang akan digunakan berupa sketsa tubuh yang digambar oleh beberapa seniman berbeda.

Teknik yang dipelajari akan diimplementasikan menggunakan Lightweight Convolutional Neural Network dan pre-trained weights dari model yang digunakan untuk mengklasifikasi dataset CIFAR-10. Hasil yang diharapkan melalui Tugas Akhir ini adalah terciptanya sebuah model yang dapat melakukan Re-identifikasi orang riil dari input sketsa menggunakan Lightweight Convolutional Neural Network, sehingga pencarian kriminal di Indonesia dapat dilakukan dengan lebih efisien.

Kata Kunci : Re-identifikasi, Multi-Modal, Kriminal

[Halaman ini sengaja dikosongkan]

ABSTRACT

Name : Charles Chang
Title : *Multi-Modal Person Re-Identification using Lightweight Convolutional Neural Network*
Advisors : 1. Dr. Reza Fuad Rachmadi, S.T., M.T.
2. Dr. I Ketut Eddy Purnama, S.T, M.T

As a complement to security systems, CCTVs are increasingly used to monitor and analyze criminal acts done at a given location. However, the manual search for criminals is still prone to human error. One of the solutions to make the process more effective and efficient is with the use of re-identification.

Re-identification is a computer vision and deep learning technique in which an anonymized identity of an image is matched with its owner. In this paper, we will study the method of re-identifying people with multi-modal images where the query is in the form of a body sketch drawn by several different artists.

Re-identification techniques in this book are implemented using lightweight Convolutional Neural Network, namely the Residual Network used to classify the CIFAR10 dataset. We hope that the result of this research will help the Indonesian police force to find criminals more efficiently.

Keywords: Re-Identification, Multi-Modal, Criminal.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas segala karunia-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Re-identifikasi Orang menggunakan Lightweight Convolutional Neural Network pada Multi-Modal Image**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, Bidang Studi *Telematika*, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa
2. Orang tua saya, atas semangat dan segala dukungan yang telah diberikan
3. Bapak Dr. Reza Fuad Rachmadi, S.T., M.T.
4. Bapak Dr. I Ketut Eddy Purnama, S.T, M.T.
5. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
6. Sesama asisten lab B201 yang menemani pengerjaan Tugas Akhir.
7. Serta teman - teman angkatan 2017 yang telah bersama - sama melalui kehidupan perkuliahan bersama penulis

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

Surabaya, April 2021

Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRACT	iii
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	3
1.3 Tujuan	3
1.4 Batasan masalah	3
1.5 Sistematika Penulisan	4
1.6 Relevansi	5
2 TINJAUAN PUSTAKA	7
2.1 Machine Learning	7
2.2 Convolutional Neural Network	10
2.3 <i>Triplet Loss</i>	11
2.4 CycleGAN	11
2.5 <i>Local Binary Pattern</i>	12
2.6 <i>Fully-Connected Layer</i>	13
2.7 <i>Residual Network</i>	13
2.8 <i>Lightweight Residual Network</i>	14
2.9 Metode Pengujian	17

2.9.1	<i>Precision</i>	17
2.9.2	<i>Recall</i>	18
2.9.3	<i>mean Average Precision (mAP)</i>	18
2.9.4	<i>Precision at n</i>	18
2.9.5	<i>Re-Ranking</i>	18
3	DESAIN DAN IMPLEMENTASI SISTEM	21
3.1	Cakupan Tugas Akhir	21
3.2	Penyesuaian Dataset PKU Sketch Re-ID	22
3.3	Lightweight Convolutional Neural Network	23
3.4	<i>Cross Domain Image-to-Image Translation</i>	35
3.5	<i>Local Binary Pattern</i>	37
3.6	<i>Training dan Testing</i>	37
3.7	Studi Ablasi	38
4	PENGUJIAN DAN ANALISIS	39
4.1	Pengujian menggunakan Lightweight Residual Network	39
4.1.1	<i>Local Binary Pattern (LBP)</i>	40
4.1.2	CycleGAN	41
4.2	Pengujian menggunakan Lightweight Residual Network <i>pre-trained</i> pada Market-1501	42
4.2.1	ResNet 20	42
4.2.2	ResNet56	43
4.2.3	ResNet 110	43
4.3	Pengujian menggunakan Re-Ranking	44
4.3.1	Re-Ranking pada ResNet 56	44
4.3.2	Re-Ranking pada ResNet 110	45
4.3.3	Kesimpulan Re-Ranking untuk Re-Identifikasi Sketsa	46

4.4	Pengujian menggunakan Re-Ranking	46
4.4.1	Re-Ranking pada ResNet 56	46
4.4.2	Re-Ranking pada ResNet 110	47
4.4.3	Kesimpulan Re-Ranking untuk Re-Identifikasi Sketsa	48
4.5	Studi Ablasi	48
4.5.1	Perubahan pada Fully Connected Layer . . .	48
4.5.2	ResNet 56 Fully Connected 128	49
4.5.3	ResNet 56 Fully Connected 256	49
4.5.4	ResNet 56 Fully Connected 768	50
4.5.5	ResNet 56 Fully Connected 1024	51
4.5.6	Kesimpulan FC Layer pada ResNet 56 . . .	51
4.5.7	ResNet 110 Fully Connected 128	52
4.5.8	ResNet 110 Fully Connected 256	52
4.5.9	ResNet 110 Fully Connected 768	53
4.5.10	ResNet 110 Fully Connected 1024	54
4.5.11	Kesimpulan FC Layer pada ResNet 110 . . .	54
4.5.12	Perubahan pada Probabilitas <i>Random Erasing</i>	55
4.5.13	ResNet 56 <i>Random Erasing</i> 0%	55
4.5.14	ResNet 56 <i>Random Erasing</i> 10%	56
4.5.15	ResNet 56 <i>Random Erasing</i> 20%	56
4.5.16	ResNet 56 <i>Random Erasing</i> 30%	57
4.5.17	ResNet 56 <i>Random Erasing</i> 40%	58
4.5.18	ResNet 110 <i>Random Erasing</i> 0%	58
4.5.19	ResNet 110 <i>Random Erasing</i> 10%	59
4.5.20	ResNet 110 <i>Random Erasing</i> 20%	59
4.5.21	ResNet 110 <i>Random Erasing</i> 30%	60
4.5.22	ResNet 110 <i>Random Erasing</i> 40%	61

4.5.23 Hasil Studi Ablasi	62
4.6 Pengujian dengan menggunakan Ensemble	63
4.6.1 Ensemble	63
4.6.2 Ensemble + Spatial Pyramid Pooling	64
4.7 Perbandingan dengan model-model lain	65
5 PENUTUP	67
5.1 Kesimpulan	67
5.2 Saran	68
DAFTAR PUSTAKA	69
BIOGRAFI PENULIS	73

DAFTAR GAMBAR

2.1	Cara Kerja Supervised Learning[1]	7
2.2	Cara kerja Unsupervised Learning [1]	9
2.3	Convolutional Neural Network	10
2.4	Triplet Loss	11
2.5	CycleGAN	11
2.6	Local Binary Pattern	12
2.7	Perbandingan layer ResNet, plain, dan VGG-19 [2] .	13
2.8	Skema struktur ResNet pada CIFAR10	14
2.9	Convolution Layer Residual Network	15
2.10	Layer1 Residual Network	16
2.11	Layer2 Residual Network	16
2.12	Layer3 Residual Network	16
2.13	Residual Network Formula	17
3.1	Desain Sistem	21
3.2	Beberapa contoh gambar pada Dataset PKU Sketch Re-ID	22
3.3	Pembagian Dataset PKU Sketch Re-ID	23
3.4	Contoh hasil CycleGAN generasi ke 200	36
3.5	Contoh dari Random Erasing	37
4.1	Hasil Studi Ablasi pada <i>Fully Connected Layer</i> . . .	62
4.2	Hasil Studi Ablasi pada <i>Random Erasing</i>	63

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

2.1	Parameter ResNet dibandingkan dengan Lightweight ResNet	17
3.1	Jumlah parameter untuk beberapa model populer. .	23
3.2	Susunan Model ResNet 20	24
3.3	Susunan Model ResNet 56	25
3.4	Susunan Model ResNet 110	29
4.1	Spesifikasi Laptop yang digunakan	39
4.2	Rata-rata performa ResNet 20	40
4.3	Rata-rata performa ResNet 20 dengan Local Binary Pattern	41
4.4	Rata-rata performa ResNet 20 dengan CycleGAN . .	41
4.5	Rata-rata performa ResNet 20 <i>pretrained</i> pada Market 1501	42
4.6	Rata-rata performa ResNet 56 <i>pretrained</i> pada Market 1501	43
4.7	Rata-rata performa ResNet 110 <i>pretrained</i> pada Market 1501	44
4.8	Rata-rata performa ResNet 56 dengan re-ranking . .	45
4.9	Rata-rata performa ResNet 110 dengan re-ranking .	45
4.10	Rata-rata performa ResNet 56 dengan re-ranking . .	47
4.11	Rata-rata performa ResNet 110 dengan re-ranking .	47
4.12	Rata-rata performa ResNet 56 Fully Connected 128 <i>pretrained</i> pada Market 1501	49

4.13 Rata-rata performa ResNet 56 Fully Connected 256 <i>pretrained</i> pada Market 1501	50
4.14 Rata-rata performa ResNet 56 Fully Connected 768 <i>pretrained</i> pada Market 1501	50
4.15 Rata-rata performa ResNet 56 Fully Connected 1024 <i>pretrained</i> pada Market 1501	51
4.16 Rata-rata performa ResNet 110 Fully Connected 128 <i>pretrained</i> pada Market 1501	52
4.17 Rata-rata performa ResNet 110 Fully Connected 256 <i>pretrained</i> pada Market 1501	53
4.18 Rata-rata performa ResNet 110 Fully Connected 768 <i>pretrained</i> pada Market 1501	53
4.19 Rata-rata performa ResNet 110 Fully Connected 1024 <i>pretrained</i> pada Market 1501	54
4.20 Rata-rata performa ResNet 56 Random Erasing 0%	55
4.21 Rata-rata performa ResNet 56 Random Erasing 10%	56
4.22 Rata-rata performa ResNet 56 Random Erasing 20%	56
4.23 Rata-rata performa ResNet 56 Random Erasing 30%	57
4.24 Rata-rata performa ResNet 56 Random Erasing 40%	58
4.25 Rata-rata performa ResNet 110 Random Erasing 0%	58
4.26 Rata-rata performa ResNet 110 Random Erasing 20%	59
4.27 Rata-rata performa ResNet 110 Random Erasing 20%	60
4.28 Rata-rata performa ResNet 110 Random Erasing 30%	60
4.29 Rata-rata performa ResNet 110 Random Erasing 40%	61
4.30 Rata-rata performa semua ResNet dengan <i>Fully Con-</i> <i>nected Layer</i> berbeda-beda	62
4.31 Rata-rata performa ensemble	64
4.32 Rata-rata performa ensemble + Spatial Pyramid Po- oling	64
4.33 Perbandingan dengan model-model lain	65

BAB I

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 tahun 2021 ini, Indonesia merupakan negara dengan jumlah penduduk terpadat ke-4 di dunia, dimana pada DKI Jakarta sendiri terdapat 10.4 juta penduduk, yaitu sekitar 333% lebih banyak dibanding Surabaya [3, 4]. Dari data yang diambil dari Badan Pusat Statistik, Polda Metro Jaya mencatat jumlah kejahatan terbanyak yaitu 31.934 kejadian, dan pada tahun 2020 ini terlihat trend adanya peningkatan tindak kriminal di seluruh Indonesia [5, 6, 7, 8].

Saat ini teknologi telah berkembang pesat sehingga penggunaan kamera pengawasan atau *CCTV* sudah umum dipakai. Hasil rekaman dari kamera ini merupakan informasi visual yang sangat vital dan dapat berperan sebagai saksi terjadinya tindakan kriminal. Fakta – fakta inilah yang mendorong riset – riset mengenai pengurangan angka kriminalitas dengan berbagai macam cara, dibutuhkan adanya otomasi yang dapat mengurangi biaya dan beban kerja pihak kepolisian.

Re-identifikasi manusia merupakan sebuah teknik visi komputer dan deep learning dimana pada sebuah lingkungan yang terdapat beberapa kamera pengawas dilakukan adanya pencocokan citra seseorang, kepada citra yang ditangkap pada kamera lain. Masalah utama yang ditangani oleh Re-identifikasi manusia dapat disimpulkan sebagai cara untuk mencari representasi diskriminatif milik indi-

vidu yang ingin dicari. Dengan membuat sistem re-identifikasi manusia, pemeriksaan hasil rekaman yang dilakukan oleh pihak kepolisian dapat dilakukan dengan jauh lebih cepat, dan dapat menekan biaya yang digunakan untuk membayar tenaga kerja. Tidak hanya untuk melakukan pencarian pelaku tindak kriminal, re-identifikasi dapat digunakan juga untuk melihat apakah barang yang tertinggal pada suatu lokasi diambil individu yang sama, dan dapat membantu pihak sekuriti mencari orang yang hilang. Re-identifikasi manusia dapat mempermudah aktivitas - aktivitas yang sebelumnya dilakukan secara manual, maka dari itu pencarian individu sangat dibutuhkan dengan menggunakan teknologi *machine learning* untuk mempercepat proses yang sebelumnya memakan waktu yang sangat lama.

Namun untuk melakukan re-identifikasi manusia dibutuhkan citra dari pelaku, yang tidak selalu siap tersedia dimiliki oleh pihak kepolisian. Maka dari itu diperlukan adanya re-identifikasi yang menggunakan sketsa sebagai input dari model. Meskipun memiliki konsep yang mirip dengan *Face Recognition*, dikarenakan re-identifikasi sketsa menggunakan gambar *full-body* terdapat tambahan kompleksitas yang harus dihadapi oleh model yang dibuat. Selain itu masalah ini sangat menantang dikarenakan sketsa tidak memiliki warna sehingga informasi yang didapatkan oleh model lebih sedikit dibanding pada re-identifikasi manusia. Penelitian ini telah dilakukan sebelumnya oleh Lu Pang et al. dengan menggunakan model yang canggih dan *cross-domain adversarial learning*, dimana model yang dibuat dapat mencapai presisi Rank-1 34%. Namun penelitian lebih lanjut menggunakan model *lightweight* masih belum pernah dilakukan. Penggunaan model *lightweight* sangatlah berguna pada kasus *Edge Computing* dimana terdapat keterbatasan perangkat lunak.

Dengan adanya penelitian ini, diharapkan hasil re-identifikasi sketsa dapat diolah dan dikembangkan untuk membantu pencarian pelaku tindak kriminal oleh pihak kepolisian di Indonesia.

1.2 Permasalahan

Berdasarkan data yang telah dipaparkan di latar belakang, dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Indonesia merupakan negara dengan penduduk ke-empat terbanyak di dunia, sehingga pencarian pelaku tindak kriminal diantara penduduk sipil sulit untuk dilakukan.
2. Data hasil rekaman kamera masih diperiksa secara manual oleh pihak kepolisian, sehingga waktu dan ketepatan pencarian masih tidak optimal.
3. Dataset PKU Sketch Re-ID yang digunakan memiliki jumlah data yang relatif sedikit.
4. Sketsa yang digunakan sebagai input tidak memiliki warna, sehingga informasi yang didapat oleh model lebih sedikit.

1.3 Tujuan

Adapun tujuan dari penelitian Tugas Akhir ini adalah mengembangkan model re-identifikasi orang menggunakan *Lightweight Convolutional Neural Network*, guna mengidentifikasi ulang seseorang yang tertangkap pada beberapa *CCTV* berbeda dengan menggunakan citra sketsa sebagai input.

1.4 Batasan masalah

Batasan masalah yang timbul dari permasalahan Tugas Akhir ini adalah:

1. Data *training* dan *testing* menggunakan data yang diambil dari PKU SketchRe-ID Dataset.
2. Jenis re-identifikasi orang yang akan dilakukan adalah *Closed Set* dan *Short Term*.
3. Training akan dilakukan menggunakan *Lightweight Convolutional Neural Network*.

1.5 Sistematika Penulisan

Laporan penelitian Tugas Akhir ini disusun dalam sistematika yang terstruktur, sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu sebagai berikut:

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 Dasar Teori

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 sistem simulator dan pengambilan data variabel - variabel uji.

3. BAB III Perancangan Sistem dan Implementasi

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah pengolahan data hingga menghasilkan visualisasi. Guna mendukung eksperimen pada penelitian ini, digunakanlah blok diagram atau *work flow* agar penjelasan sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Beberapa teknik visualisasi akan ditunjukkan hasilnya pada bab ini dan dilakukan analisa terhadap hasil visualisasi dan informasi yang didapat dari hasil mengamati visualisasi yang tersaji

5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk mengembangkan lebih lanjut juga dituliskan pada bab ini.

1.6 Relevansi

1. Lightweight Residual Network for Person Re-Identification (Reza Fuad Rachmadi, Supeno Mardi Nugroho, I Ketut Eddy Purnama) [9]

Lightweight Residual Network for Person Re-Identification ini merupakan sebuah implementasi *lightweight CNN* untuk melakukan re-Identifikasi manusia, *lightweight CNN* yang digunakan berbasis *Residual Network* dengan menggunakan pre trained weights yang pernah digunakan untuk memecahkan masalah klasifikasi CIFAR-10. Berdasarkan hasil dari riset yang dilakukan, meskipun *lightweight CNN* tidak mendapatkan akurasi tercanggih dibandingkan model-model lainnya, banyaknya informasi yang didapatkan oleh model ini sangat tinggi, dan dapat dikatakan model ini lebih efisien dari model-model lainnya.

2. Torchreid: A Library for Deep Learning Person Re-Identification in Pytorch. (Kaiyang Zhou, Tao Xiang). [10]

Torchreid merupakan sebuah *library deep learning* yang dibuat oleh Kaiyang Zhou untuk mempercepat implementasi dan percobaan re-identifikasi. Library ini secara umum dibuat dengan menggunakan bahasa Python dengan beberapa kode berbasis Cython untuk optimisasi. Pada library ini dataset sudah di *preprocess* dan di implementasi sesuai dengan protokol evaluasi masing masing dataset sehingga dapat dibandingkan dengan penelitian lain yang terkait.

3. Adaptive L2 Regularization in Person Re-Identification. (Xinyang Ni, Liang Fang, Heikki Huttunen) [11]

Adaptive L2 Regularization in Person Re-Identification merupakan sebuah penelitian untuk menambahkan regularisasi L2, dimana faktor regularisasi dapat berubah ubah secara adaptif pada *baseline* model. Dari hasil penelitian yang dilakukan regularisasi L2 secara adaptif dapat meningkatkan akurasi model sekitar 1 hingga 2% untuk *dataset* Market-1501, Duke-MTMC dan MSMT17.

4. Cross-Domain Adversarial Feature Learning for Sketch Re-identification. (Lu Pang, Yaowei Wang, Yi-Zhe Song, Tiejun Huang, Yonghong Tian) [12]

Cross-Domain Adversarial Feature Learning for Sketch Re-identification merupakan penelitian yang pertama kali menggunakan gambar sketsa sebagai input dari model re-identifikasi manusia, namun dari model yang digunakan sendiri merupakan model yang telah di optimisasi untuk melakukan pengambilan informasi dari sketsa, seperti Triplet SN dan model GN Siamese yang merupakan gabungan dari dua cabang dari model GoogleNet yang dioptimisasi dengan menggunakan pairwise verification loss.

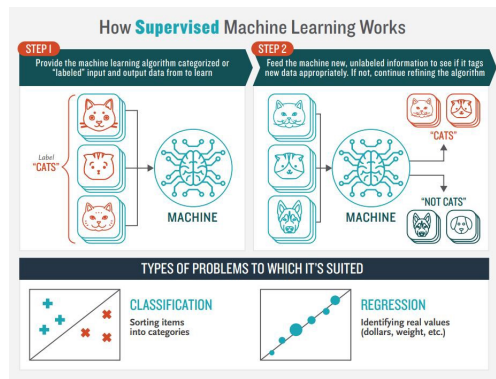
Dapat dilihat dari penelitian-penelitian terkait diatas, selain pada penelitian *Cross-Domain Adversarial Feature Learning for Sketch Re-identification*, fokus dari permasalahan adalah penggunaan metode re-identifikasi pada citra manusia yang tertangkap pada CCTV. Sedangkan pada penelitian yang kami usulkan merupakan implementasi re-identifikasi pada citra orang riil yang tertangkap pada CCTV, menggunakan citra sketsa *full body* sebagai *input*. Selain itu penelitian yang dilakukan menggunakan model *lightweight classical* seperti ResNet, yang bukan merupakan fokus dari penelitian *Cross-Domain Adversarial Feature Learning for Sketch Re-identification*.

BAB II

TINJAUAN PUSTAKA

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

2.1 Machine Learning



Gambar 2.1: Cara Kerja Supervised Learning[1]

Machine learning atau pembelajaran mesin adalah suatu cabang teknologi yang menerapkan penggunaan *artificial intelligence*. *Machine learning* pertama kali diperkenalkan oleh Thomas Bayes, Adrien-Marie Legendre, dan Andrey Markov pada sekitar tahun 1920[13]. Dengan berkembangnya *machine learning*, tugas-tugas yang dilakukan oleh *machine learning* ini pun semakin beragam, dimana secara umum jenis pembelajaran pada Machine Learning

dapat dikelompokkan menjadi dua, yaitu Supervised Learning dan Unsupervised Learning.

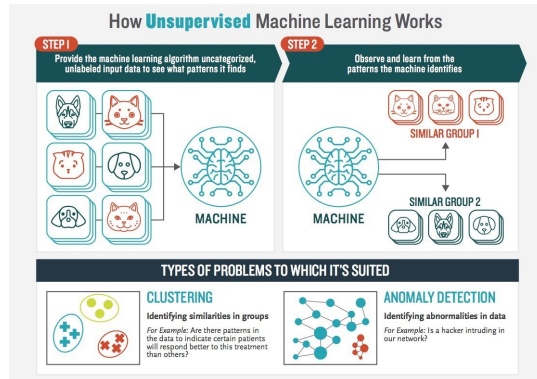
Supervised learning jika diartikan secara harfiah adalah pembelajaran yang ada supervisornya. Disini supervisi dilakukan oleh orang yang melakukan training kepada label di setiap datanya. Sebagai contoh dapat dilihat pada gambar 2.1.

Pada gambar diatas, masing-masing gambar kucing diberi label “CATS” dan yang bukan kucing (“anjing”, “beruang”, “lain-lain”) diberi label “NOT CATS”. Ketika gambar baru dimasukkan setiap label akan dicompare sampai selesai, dan yang memiliki persentase lebih banyak akan diambil sebagai prediksi akhir.

Pada pendekatan *supervised learning*, terdapat input dan output yang dibuat menjadi hubungan matematis. *Supervised learning* cocok untuk digunakan untuk memprediksi dimana sudah ada contoh data yang lengkap, sehingga pola yang terbentuk adalah hasil pembelajaran dari data lengkap tersebut. Beberapa algoritma yang termasuk dalam *supervised learning* adalah sebagai berikut:

1. Regresi Linier Berganda
2. Analisis Deret Waktu
3. *Decision Tree* dan *Random Forest*
4. *Naive Bayes Classifier*
5. *Nearest Neighbor Classifier*
6. *Artificial Neural Network*

Jika dibandingkan dengan *supervised learning*, *unsupervised learning* tidak membutuhkan adanya label sebagai dasar prediksi melainkan menggunakan kesamaan atribut - atribut yang dimiliki oleh data tersebut. Jika atribut - atribut tersebut memiliki kesamaan maka data tersebut akan di *cluster* menjadi satu. Sebagai contoh dapat dilihat pada gambar 2.2 :



Gambar 2.2: Cara kerja Unsupervised Learning [1]

Pada Gambar 2.2 dapat dilihat bahwa disediakan gambar-gambar yang tidak memiliki label ke algoritma *machine learning*. Setelah itu *artificial intelligence* akan memisahkan gambar mana yang memiliki kesamaan di dalam *cluster*. *Cluster* yang ada merupakan hasil akhir klasifikasi yang dilakukan.

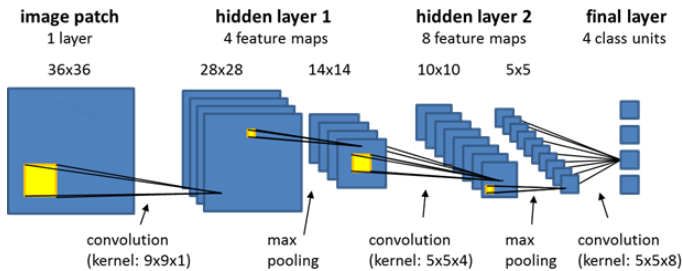
Namun *unsupervised learning* tidak memiliki hasil spesifik layaknya pada *supervised learning*. Hal ini dikarenakan tidak adanya label dasar (ground truth). Beberapa algoritma yang digunakan di *unsupervised learning* :

1. *Clustering*
2. *Anomaly Detection*
3. *Training Model*
4. *Association Discovery*

Deep learning (Pembelajaran Dalam) merupakan bagian yang dalam dari *machine learning* yang terdiri dari pemodelan fungsi yang ditata berlapis dan mendalam dengan menggunakan *Artificial Neural Network* (ANN). ANN merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam

memproses informasi. Jenis pembelajaran dalam deep learning berupa *supervised*, *semi-supervised*, dan *unsupervised*. Deep learning dapat diimplementasikan dalam pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya.

2.2 Convolutional Neural Network



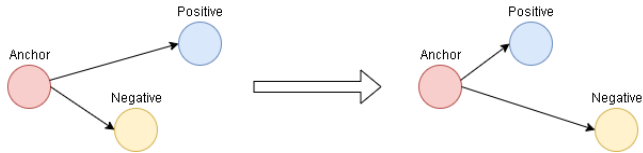
Gambar 2.3: Convolutional Neural Network

Convolutional Neural Network merupakan salah satu algoritma *Deep Learning* yang umum digunakan untuk data berbentuk citra. Convolutional Neural Network memiliki kedalaman yang cukup tinggi sehingga termasuk dalam jenis *Deep Neural Network*. Pada umumnya CNN tidak jauh berbeda dengan neural network pada umumnya, CNN terdiri dari neuron yang memiliki *weight*, *bias*, dan *activation function*. Dimana *weight* dari CNN sendiri didapatkan dari persamaan sebagai berikut

$$neuron_{input} \times neuron_{output} \times tinggi \times lebar$$

Pada Convolutional Neural Network, operasi yang paling utama merupakan Convolutional Layer, dimana terjadi operasi konvolusi pada *output* dari *layer* sebelumnya. Operasi konvolusi merupakan aplikasi kernel pada citra di semua offset sehingga citra secara keseluruhan diubah. Tujuan dari dilakukannya konvolusi itu sendiri adalah untuk melakukan ekstraksi dari fitur-fitur milik citra input.

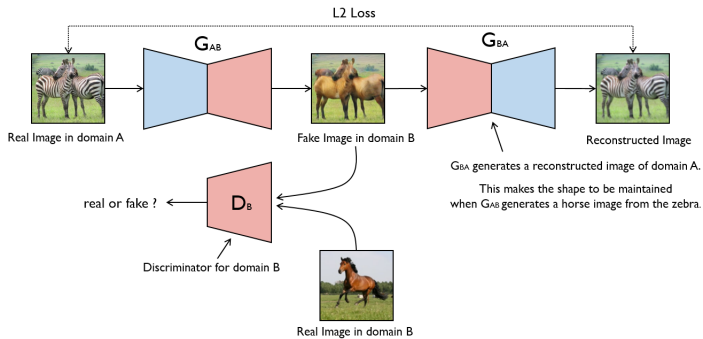
2.3 Triplet Loss



Gambar 2.4: Triplet Loss

Triplet Loss merupakan sebuah *Loss Function* yang umumnya digunakan dalam proses Re-Identifikasi. Pada fungsi ini dilakukan perbandingan jarak antara titik acuan terhadap titik positif yang merupakan gambar dalam kelas sama, dan perbandingan dengan titik negatif yang berasal dari kelas berbeda. Fungsi ini memastikan bahwa jarak ke titik positif akan lebih dekat dibandingkan dengan titik negatif.

2.4 CycleGAN

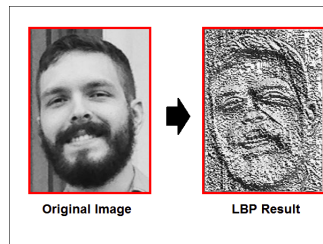


Gambar 2.5: CycleGAN

CycleGAN merupakan sebuah teknik yang menggunakan *Deep Convolutional Neural Network* untuk melakukan sintesis sebuah gambar versi baru dengan modifikasi yang diinginkan, seperti meng-

ubah gambar dari musim panas ke musim salju. Pada umumnya *training* model untuk melakukan hal tersebut membutuhkan data-set dengan contoh berpasangan(*paired example*) yang sangat besar. Cara seperti ini membutuhkan waktu yang sangat lama, dan pada beberapa kasus tertentu tidak dapat dilakukan. Namun dengan menggunakan CycleGAN, model dapat secara otomatis melakukan *training* untuk translasi *Image-to-image*. dilatih secara *unsupervised* dengan menggunakan kumpulan gambar dari sebuah domain X ke sebuah domain Y, tanpa harus memasangkan kedua gambar tersebut.

2.5 Local Binary Pattern



Gambar 2.6: Local Binary Pattern

Local Binary Pattern (LBP) merupakan salah satu deskriptor visual yang digunakan pada visi komputer. Pada umumnya LBP digunakan pada Face Recognition dikarenakan LBP merupakan deskriptor yang sangat kuat untuk melakukan klasifikasi tekstur. Selain itu telah ditemukan bahwa ketika LBP digabungkan dengan deskriptor *Histogram of Oriented Gradients* (HOG), performa yang didapatkan bertambah secara drastis pada beberapa dataset tertentu.

Cara kerja dari LBP sendiri adalah sebagai berikut:

1. Ubah citra menjadi bentuk *grayscale* / hitam putih.
2. Bagi citra menjadi beberapa bagian(*cell*)
3. Untuk setiap *pixel* yang terdapat pada sebuah *cell*, bandingkan dengan pixel milik 8 neighbor yang terdapat di sekeliling-

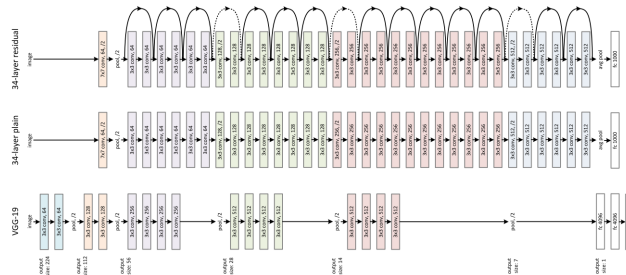
nya

4. Apabila nilai dari pixel yang di tengah lebih besar dari setiap pixel pada neighborsnya maka pixel tersebut diberi nilai 0, selain itu pixel tersebut diberi nilai 1.
5. Hitung Histogram
6. Normalisasi Histogram.

2.6 Fully-Connected Layer

Fully Connected layer merupakan *layer* yang bertujuan untuk melakukan transformasi pada dimensi data agar klasifikasi secara linear dapat dilakukan, dimana setiap neuron pada *convolution layer* ditransformasi terlebih dahulu menjadi satu dimensi. Namun hal tersebut dapat menyebabkan hilangnya data spasial, sehingga pada umumnya *fully connected layer* hanya diimplementasikan pada akhir jaringan. *Convolution layer* dengan ukuran kernel 1x1 dapat melakukan fungsi yang sama dengan *Fully Connected Layer*, namun dapat tetap mempertahankan data spasial.

2.7 Residual Network

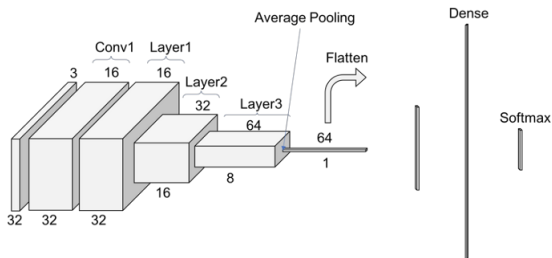


Gambar 2.7: Perbandingan layer ResNet, plain, dan VGG-19 [2]

Residual Neural Network atau *ResNet* merupakan sebuah *Artificial Neural Network* (ANN) yang dibuat berdasarkan bentuk korteks serebral milik manusia. *Residual Network* melakukan hal ini

dengan memperkenalkan *skip connection* atau *shortcut*, dimana model dapat melompat dua atau tiga *layer* jika memang hal tersebut merupakan hasil terbaik. Sebelum adanya model *Residual Neural Network* penambahan layer pada suatu model hanya akan meningkatkan akurasi sampai suatu batas tertentu, sehingga penambahan layer setelah 20 hanya menambahkan kompleksitas model. Namun pada penelitian *Deep Residual Learning for Image Recognition* yang dibuat oleh Kaiming He pada tahun 2015 memproposisikan sebaliknya, apabila layer tambahan yang ada dapat mempelajari matriks identitas, maka akurasi minimal yang didapat pada *layer* akan sama dengan apabila tidak menambahkan *layer*. Untuk membuktikan hal tersebut dibuatlah sistem *skip connection* atau *shortcut* sehingga model dapat mempelajari matriks identitas dengan lebih mudah. Dari penelitian yang dilakukan pada dataset *ImageNet*, *Residual Network* dapat mengurangi *loss* yang didapat ketika menambahkan lebih banyak *layer* pada *Artificial Neural Network* yang dibuat. [2]

2.8 Lightweight Residual Network



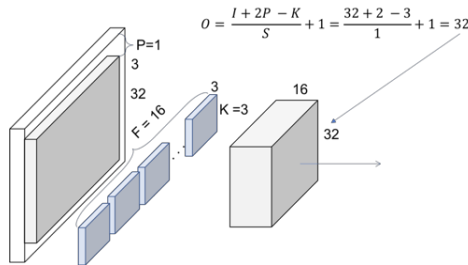
Gambar 2.8: Skema struktur ResNet pada CIFAR10

Lightweight Residual Network merupakan implementasi dari Residual Network pada dataset CIFAR-10, sebuah dataset yang terdiri dari 60000 gambar 32x32, dimana setiap kelas memiliki 6000 gambar. Residual Network yang digunakan untuk memecahkan masalah klasifikasi CIFAR-10 ini berbeda dengan Residual Network pada normalnya yang memiliki jumlah parameter jauh lebih besar, yaitu sekitar 23 juta parameter. Residual Network ini memiliki jum-

lah layer yang relatif dalam apabila dibandingkan dengan Residual Network yang *original*, namun parameter yang terdapat pada model ini jauh lebih sedikit, dengan yang paling banyak sebesar 3 juta parameter.

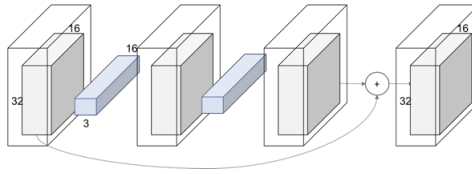
Pada gambar 2.8 dapat dilihat skema yang secara umum digunakan oleh Residual Network pada CIFAR10, dimana terlihat pada skema bahwa tidak terdapat *pooling layer* setelah *convolution layer*. Pada *convolution layer* ini dapat dilihat bahwa layer pertama merupakan *convolution layer* dengan ukuran 3x3 dan batch normalization. Ukuran *stride* dan *padding* yang diberikan adalah 1 untuk menyamakan bentuk output dengan input. Convolution Layer dari ResNet dapat dilihat dari gambar 2.9.

Kemudian setelah *Convolution Layer*, dibuat sebuah layer yang terdiri dari 6 konvolusi dengan ukuran 3x3, banyaknya kumpulan konvolusi yang digunakan akan menentukan ukuran ResNet yang dibuat. Penurunan dimensi dari data dilakukan pada tahap ini dengan melakukan penambahan *stride* menjadi dua untuk setiap konvolusi pertama di setiap layer berikut.



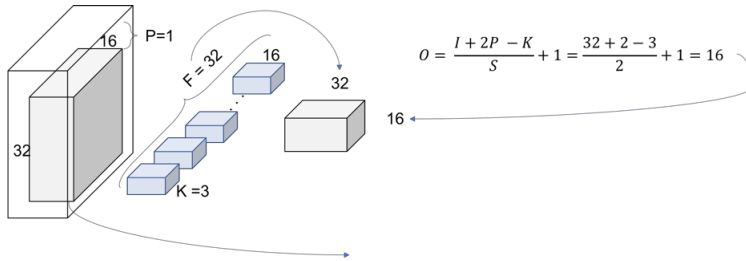
Gambar 2.9: Convolution Layer Residual Network

Pada layer ini diberikan *bypass connection*, dimana apabila terjadi pengurangan performa dari model, dapat dilakukan *bypass connection* dengan cara menambah padding dimensi dengan *zeros* sehingga ukuran output sesuai dengan ukuran sebelumnya.



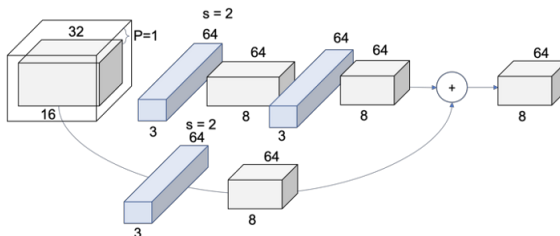
Gambar 2.10: Layer1 Residual Network

Layer kedua memiliki cara kerja yang sama dengan layer pertama, namun dikarenakan ukuran *stride* pada layer pertama dibuat menjadi dua, ukuran output adalah setengah dari input untuk layer kedua, sehingga diberikan *zero padding*.



Gambar 2.11: Layer2 Residual Network

Kemudian Layer 3 akan mengimplementasikan prinsip yang sama dengan Layer 2.



Gambar 2.12: Layer3 Residual Network

Sehingga formula akhir dari struktur Lightweight Residual Network dapat disimpulkan sebagai berikut 2.13

output map size	32×32	16×16	8×8
# layers	$1+2n$	$2n$	$2n$
# filters	16	32	64

Gambar 2.13: Residual Network Formula

Berikut merupakan perbandingan parameter Lightweight Residual Network dengan Residual Network

Name	Parameters
ResNet 20	0.27M
ResNet 32	0.46M
ResNet 44	0.66M
ResNet 56	0.85M
ResNet 110	1.7M
ResNet 50	23M

Tabel 2.1: Parameter ResNet dibandingkan dengan Lightweight ResNet

2.9 Metode Pengujian

2.9.1 *Precision*

Precision merupakan rasio dari prediksi jumlah total contoh positif yang benar diklasifikasikan dibagi dengan jumlah keseluruhan hasil yang diprediksi positif. Precision dapat melihat dari keseluruhan data, berapa persen yang diklasifikasikan secara benar.

$$Recall = \frac{TP}{(TP + FP)}$$

2.9.2 Recall

Recall merupakan rasio dari jumlah total positif yang benar diklasifikasikan, dibagi dengan jumlah total contoh yang benar positif. Recall yang tinggi menunjukkan bahwa kelas yang dikenali dengan benar banyak, atau False Negative yang didapatkan sedikit.

$$Recall = \frac{TP}{(TP + FN)}$$

2.9.3 mean Average Precision (mAP)

mean Average Precision (mAP) merupakan sebuah metrik akurasi yang didapatkan dari rata rata *Average Precision*. Dimana *Average Precision* sendiri didapatkan dari perhitungan nilai presisi dan recall. *mean Average Precision* merupakan sebuah metrik evaluasi yang sangat baik dikarenakan melibatkan presisi dan recall. Dengan menggunakan *mean Average Precision*, sebuah angka dapat diterima untuk mengukur kinerja dari pendeteksian sebuah objek.

$$AP = \Sigma(recall_{n+1}) - recall_n \times precision_{interp} \times (recall_{n+1})$$

2.9.4 Precision at n

Precision at n merupakan sebuah metrik evaluasi yang didapatkan dari pengembalian informasi dalam bentuk daftar dengan ranking. Daftar ranking tersebut kemudian digunakan untuk melakukan penilaian dari akurasi model dengan cara melihat informasi ke- n paling atas yang dikembalikan. Rank-1 accuracy merupakan pengecekan berapa kali prediksi label sesuai dengan yang sebenarnya, sedangkan Rank-5 accuracy merupakan berapa kali prediksi label terdapat pada 5 prediksi paling atas.

2.9.5 Re-Ranking

Re-Ranking merupakan pengambilan *K-Reciprocal Nearest Neighbor* yang mengambil citra serupa dengan citra *query* yang di-

berikan. Hal ini dilakukan dengan adanya pengambilan jarak antara citra *query* dengan citra yang terdapat pada *gallery*.

[Halaman ini sengaja dikosongkan]

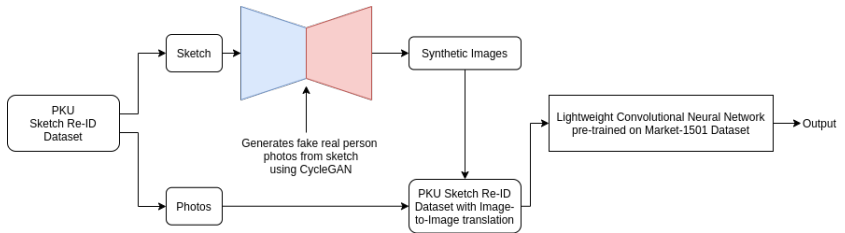
BAB III

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

3.1 Cakupan Tugas Akhir

Tugas akhir ini merupakan salah satu bentuk implementasi Lightweight Convolutional Neural Network untuk melakukan pencarian seorang individu menggunakan gambar multi-modal berupa sketsa sebagai input, berikut pada Gambar 3.1 adalah cakupan Tugas Akhir dari Desain Sistem.



Gambar 3.1: Desain Sistem

3.2 Penyesuaian Dataset PKU Sketch Re-ID

Desain sistem secara umum pada gambar 3.1, yang mencakup beberapa hal, salah satunya ialah penyesuaian dataset. Dataset yang kami akan gunakan adalah dataset PKU Sketch Re-ID yang dibuat oleh Lu Pang [15]. Dataset ini terdiri dari 200 identitas yang unik, dimana setiap identitas telah ditangkap oleh dua kamera berbeda. Selain itu setiap individu memiliki satu gambar sketsa, sehingga seluruh dataset bertotal 600 gambar. Gambar 3.2 menunjukkan beberapa contoh dari dataset PKU Sketch Re-ID.



Gambar 3.2: Beberapa contoh gambar pada Dataset PKU Sketch Re-ID

Dataset PKU Sketch [15] Re-id kemudian akan melalui proses persiapan terlebih dahulu sehingga dapat dilakukan evaluasi performa dari Lightweight Convolutional Neural Network yang akan digunakan. Penyesuaian yang akan dilakukan pada dataset meliputi, penggabungan individu-individu sama yang tertangkap oleh kamera kamera berbeda, kemudian pemasangan individu tersebut ke gambar sketsa sehingga setiap individu unik akan menjadi ke-

las tersendiri. Kemudian dataset tersebut akan dibagi menjadi 150 identitas untuk training dan 50 identitas untuk testing, seperti yang ditunjukkan pada gambar 3.4. Setiap gambar telah di *crop* secara *manual* untuk memastikan setiap gambar dari dataset hanya berisi satu individu yang spesifik. Untuk gambar sketsa yang digunakan, terdapat 5 orang seniman berbeda yang memiliki 5 *art style* yang berbeda pula untuk menggambar masing masing identitas.



Gambar 3.3: Pembagian Dataset PKU Sketch Re-ID

3.3 Lightweight Convolutional Neural Network

Setelah penyesuaian dataset telah dilakukan, akan dilakukan training dengan menggunakan Lightweight Residual Network yang pernah digunakan untuk memecahkan masalah klasifikasi CIFAR-10. Namun pada model yang kami gunakan, *Fully-Connected Layer* terakhir dihilangkan, dan ditambahkan dua layer baru untuk menggantikan layer tersebut sehingga model dapat dipastikan akan mempelajari fitur-fitur yang terdapat pada dataset PKU Sketch Re-ID. Selain itu, berbeda dengan model ResNet aslinya, model yang kami gunakan menggunakan resolusi input sebesar 32x64x3.

Nama	Parameter
ResNet 56	0.85M
ResNet 110	1.7M
GoogleNet	7M
DenseNet121	8.6M
ResNet 50	23M

Tabel 3.1: Jumlah parameter untuk beberapa model populer.

Pada percobaan-percobaan yang kami lakukan, kami menggunakan tiga model Residual Network yang berbeda, yaitu ResNet 20, ResNet 56, dan ResNet 110. Namun kami hanya melanjutkan menggunakan ResNet 56 dan ResNet 110 dikarenakan kedua model tersebut mendapatkan hasil yang jauh lebih baik dibandingkan dengan pada ResNet 20. Meskipun layer dari kedua model yang kami gunakan relatif cukup dalam (56 dan 110 layer) apabila dibandingkan dengan Residual Network normalnya. Parameter yang terdapat pada model ini jauh lebih sedikit, dimana ResNet 110 hanya memiliki 1.7 juta parameter, jauh dari jutaan parameter yang dimiliki model-model populer seperti DenseNet, GoogleNet, dan ResNet 50. Pada tabel 3.2 dapat dilihat bentuk susunan layer dari ResNet 20 sendiri. Untuk susunan model ResNet 56 dapat dilihat pada 3.3 dan untuk ResNet 110 dapat dilihat pada 3.4.

Tabel 3.2: Susunan Model ResNet 20

Type	Channel	Size	Stride
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	32	3x3	2x2
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1

Dilanjutkan ke halaman selanjutnya

Tabel 3.2 – Dilanjutkan dari halaman sebelumnya

Type	Channel	Size	Stride
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	64	3x3	2x2
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
FullyConnected	-	-	-
BatchNorm1D	512	-	-
Dropout	-	-	-
FullyConnected	-	-	-

Tabel 3.3: Susunan Model ResNet 56

Type	Channel	Size	Stride
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
Dilanjutkan ke halaman selanjutnya			

Tabel 3.3 – Dilanjutkan dari halaman sebelumnya

Type	Channel	Size	Stride
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	32	3x3	2x2
BatchNorm2D	32	-	-
Dilanjutkan ke halaman selanjutnya			

Tabel 3.3 – Dilanjutkan dari halaman sebelumnya

Type	Channel	Size	Stride
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	64	3x3	2x2
Dilanjutkan ke halaman selanjutnya			

Tabel 3.4 – Dilanjutkan dari halaman sebelumnya

Type	Channel	Size	Stride
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
FullyConnected	-	-	-
BatchNorm1D	512	-	-
Dropout	-	-	-
FullyConnected	-	-	-

3.4 *Cross Domain Image-to-Image Translation*

Dikarenakan fitur yang dapat dikenali oleh model untuk membantu membedakan satu individu dengan yang lainnya sangat sedikit, maka digunakan *image-to-image translation*, atau lebih spesifik CycleGAN. Penggunaan CycleGAN dapat menjembatani antara modalitas yang dimiliki oleh gambar dari kamera CCTV dan sketsa *full-body*, CycleGAN sendiri dapat mempelajari fitur yang terdapat pada kedua domain permasalahan, dan meniru distribusi dari *gene-*

rator yang diberikan untuk menghasilkan citra CCTV sintesis dari citra sketsa *full-body*. Dikarenakan CycleGAN didesain untuk melakukan translasi antar citra yang tidak dipasangkan, pada penelitian ini kami melakukan pemasangan antara masing-masing citra individu unik yang tertangkap pada CCTV dengan sketsanya, sehingga dapat dipastikan bahwa *style transfer* yang dilakukan merupakan dari masing-masing individu tersebut.



Gambar 3.4: Contoh hasil CycleGAN generasi ke 200

Untuk CycleGAN nya sendiri, kami menggunakan model yang dibuat oleh Jun-Yan Zhu et al. dengan *pre-trained weights* milik *facades_label2photo*. Citra yang digunakan untuk melatih gambar ini merupakan 600 citra yang terdapat pada dataset PKU Sketch Re-ID, dimana model ini di inisialisasi dengan *learning rate* sebesar 0.0002 dan menurun ke 0.0001 setelah 150 epochs. Model CycleGAN ini kami *train* selama 1000 epoch, namun kami memilih generasi ke 200 untuk dataset kami dikarenakan hasil sudah cukup baik.

3.5 *Local Binary Pattern*

Salah satu cara lain yang kami gunakan untuk menjembatani perbedaan modalitas yang dimiliki oleh citra CCTV dan citra sketsa merupakan penggunaan *Local Binary Pattern*. Dengan menggunakan *Local Binary Pattern* dapat dilakukan pengambilan fitur tekstur dari citra CCTV, selain itu *Local Binary Pattern* juga menggunakan citra hitam putih sehingga menyerupai citra sketsa yang ada.

3.6 *Training dan Testing*



Gambar 3.5: Contoh dari Random Erasing

Untuk memastikan bahwa kinerja model yang digunakan dapat diandalkan, kami melakukan proses training dan testing sebanyak 10 kali dan mengambil nilai rata-ratanya sebagai metrik evaluasi *final*. Semua metode dievaluasi menggunakan akurasi Rank-1 milik model masing-masing, mengikuti ketentuan evaluasi yang ditentukan oleh Lu Pang et al.

Pada eksperimen kami, kami mengatasi kurangnya data pada dataset PKU Sketch Re-Identification dengan cara melakukan *pre-training* pada dataset Market-1501. Proses *training* dilakukan selama 100 *epochs*, dimana *training* di inisialisasi dengan *learning rate* sebesar 0,1 . *Learning rate* tersebut akan berkurang dengan faktor 0.1 setiap 40 *epochs*. Selain itu, kami melakukan *random erasing* dan *random crop* untuk menambahkan lebih banyak data *training*. Gambar 3.5 menunjukkan beberapa contoh dari augmentasi data yang dilakukan dengan metode *Random Erasing*.

3.7 Studi Ablasi

Untuk memaksimalkan performa dari model ResNet-CIFAR yang digunakan, dilakukan studi ablası dengan cara mengubah ukuran *Fully Connected Layer* pertama yang digunakan oleh model. Ukuran *Fully Connected Layer* yang digunakan adalah sebagai berikut; FC-128, FC-256, FC-512, FC-768, dan FC-1024. Selain itu dilakukan studi ablası kedua dengan cara merubah probabilitas Random Erasing dari 0% hingga 50%. Pada pengujian ini kami hanya menggunakan model ResNet 56 dan ResNet 110 dikarenakan pada eksperimen sebelumnya kedua model tersebut mendapatkan rank-1 accuracy tertinggi dan kedua tertinggi.

BAB IV

PENGUJIAN DAN ANALISIS

Pada penelitian ini dipaparkan hasil pengujian serta analisis dari desain sistem dan implementasi. Penggunaan dataset dari PKU Sketch Re-ID telah diambil dan telah meminta izin kepada *National Engineering Laboratory for Video Technology*, Universitas Peking. Pengujian ini dilakukan dalam empat bagian, yakni sebagai berikut:

1. Pengujian menggunakan Lightweight Residual Network
2. Pengujian menggunakan Lightweight Residual Network *pre-trained* pada Market-1501
3. Pengujian menggunakan Re-Ranking
4. Studi Ablasi
5. Pengujian dengan menggunakan Ensemble.

Pada pengujian ini, pelatihan masing-masing model dilakukan dengan menggunakan laptop dengan spesifikasi *hardware* sebagai berikut:

Processor	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
RAM	8 GB DIMM DDR4 x 2
Storage	HDD 512 GB
Graphics Card	Nvidia GeForce GTX 1060 6GB MAX-Q
Operating System	Ubuntu 18.04 LTS 64-bit

Tabel 4.1: Spesifikasi Laptop yang digunakan

4.1 Pengujian menggunakan Lightweight Residual Network

Pada pengujian ini, dilakukan *training* dengan menggunakan Lightweight Residual Network, dimana pada model tidak dilakuk-

an pelatihan ke dataset Market-1501 sebelum dilatih untuk melakukan klasifikasi pada dataset PKU Sketch Re-ID. Selain itu pada tahap pengujian ini dilakukan dua metode pengujian, yaitu dengan menggunakan metode Local Binary Pattern dan metode CycleGAN untuk menjembatani modalitas antara citra sketsa dan citra CCTV. Pada tahap pengujian ini hanya dilakukan pengujian pada model ResNet 20, pengujian seperti ini dilakukan karena model ini membutuhkan waktu *training* paling sedikit dibandingkan dengan model *lightweight* ResNet lain yang akan digunakan pada pengujian selanjutnya.

Tabel 4.2 menunjukkan hasil dari training dan testing model sebanyak 10 kali untuk model ResNet 20 tanpa adanya perubahan pada dataset PKU Sketch Re-ID.

Tabel 4.2: Rata-rata performa ResNet 20

No	Rank-1	Rank-5	Rank-10	mAP
1	8	32	48	11.4505
2	10	32	46	11.2543
3	4	32	46	8.426
4	8	24	32	9.842
5	8	36	48	11.1407
6	10	28	42	11.7093
7	6	32	50	9.9849
8	10	28	44	10.8898
9	14	26	44	14.43
10	8	26	42	10.3942
Average	8.6	29.6	44.2	10.95217

4.1.1 *Local Binary Pattern* (LBP)

Tabel 4.3 menunjukkan hasil dari training dan testing model sebanyak 10 kali untuk model ResNet 20 dengan menggunakan *filter* LBP. Namun dikarenakan hasil yang didapatkan lebih rendah dibandingkan tanpa menggunakan LBP pada semua metrik evaluasi, sedangkan waktu yang dibutuhkan untuk *training* menjadi 2 kali lebih banyak pengujian menggunakan LBP diberhentikan.

Tabel 4.3: Rata-rata performa ResNet 20 dengan Local Binary Pattern

No	Rank-1	Rank-5	Rank-10	mAP
1	2	30	40	7.7963
2	4	26	44	8.898
3	6	32	46	10.3007
4	12	30	42	13.5232
5	14	24	40	12.4012
6	8	24	38	10.4669
7	8	32	42	11.4641
8	4	30	44	9.9319
9	8	28	38	10.0783
10	10	14	30	9.7921
Average	7.6	27	40.4	10.46527

4.1.2 CycleGAN

Tabel 4.4 menunjukkan hasil dari training dan testing model sebanyak 10 kali untuk model ResNet 20 dimana citra sketsa pada dataset PKU Sketch Re-Identification telah dilakukan translasi *style*.

Tabel 4.4: Rata-rata performa ResNet 20 dengan CycleGAN

No	Rank-1	Rank-5	Rank-10	mAP
1	8	16	34	8.9563
2	12	28	36	14.2855
3	10	24	40	10.558
4	8	26	36	10.4396
5	8	22	40	10.8883
6	10	26	44	10.9954
7	6	34	50	9.7
8	14	28	42	13.4316
9	8	26	42	10.5551
10	10	26	38	12.737
Average	9.4	25.6	40.2	11.25468

Meskipun Rank-5 dan Rank-10 yang didapatkan lebih rendah, dikarenakan metrik evaluasi yang digunakan disamakan dengan pada paper "Cross-Domain Adversarial Feature Learning for Sketch Re-identification", yaitu menggunakan Rank-1 accuracy. Maka semua pengujian selanjutnya akan dilakukan menggunakan dataset yang telah ditranslasi dengan CycleGAN.

4.2 Pengujian menggunakan Lightweight Residual Network *pre-trained* pada Market-1501

Pada pengujian ini dilakukan *pre-training* terlebih dahulu pada dataset Market 1501 untuk mengatasi keterbatasan data pada dataset PKU Sketch Re-ID. Pada tahap pengujian ini, digunakan tiga jenis *Lightweight Residual Network* yakni; ResNet 20, ResNet 56, dan ResNet 110. Dataset yang digunakan pada tahap pengujian ini semuanya merupakan hasil sintesis dari CycleGAN.

4.2.1 ResNet 20

Tabel 4.5 menunjukkan hasil dari training dan testing model sebanyak 10 kali untuk model ResNet 20 dengan model yang telah di *pre-trained* pada dataset Market 1501 terlebih dahulu.

Tabel 4.5: Rata-rata performa ResNet 20 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	12	28	42	11.6169
2	8	24	40	10.1131
3	10	30	46	12.5311
4	10	24	32	10.3256
5	10	22	40	9.7441
6	10	22	48	10.4837
7	10	34	40	13.0863
8	14	38	48	13.5198
9	8	18	32	9.6554
10	10	24	42	10.9248
Average	10.2	26.4	41	11.30008

Dapat dilihat terdapat peningkatan yang cukup signifikan pada Rank-1 accuracy yaitu dari 9,4% ke 10,2%. Selain itu Rank-5, Rank-10, dan mAP yang didapatkan lebih tinggi dibandingkan pada saat tidak di *pre-trained* pada Market 1501.

4.2.2 ResNet56

Tabel 4.6 menunjukkan hasil dari training dan testing model sebanyak 10 kali untuk model ResNet 56 dengan model yang telah di *pre-trained* pada dataset Market 1501 terlebih dahulu.

Tabel 4.6: Rata-rata performa ResNet 56 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	20	44	54	19.2315
2	16	34	42	20.2117
3	14	34	50	17.2698
4	16	36	40	16.8498
5	10	34	48	14.9505
6	12	26	36	14.3816
7	14	32	46	14.9699
8	12	38	60	16.0066
9	10	38	46	14.5599
10	18	34	46	19.1052
Average	14.2	35	47.8	16.95365

Terlihat terdapat peningkatan yang sangat signifikan pada Rank-1, yaitu sebesar 40% apabila dibandingkan dengan ResNet 20.

4.2.3 ResNet 110

Tabel 4.7 menunjukkan hasil dari training dan testing model sebanyak 10 kali untuk model ResNet 110 dengan model yang telah di *pre-trained* pada dataset Market 1501 terlebih dahulu. Pada pengujian ini terlihat terdapat peningkatan yang sangat signifikan pada Rank-1, yaitu sebesar 70% apabila dibandingkan dengan ketika menggunakan model ResNet 20.

Tabel 4.7: Rata-rata performa ResNet 110 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	18	36	56	17.9811
2	20	46	62	21.6104
3	18	40	56	19.7356
4	16	44	54	18.3177
5	18	46	56	18.1901
6	16	38	42	17.9401
7	18	36	54	20.0548
8	16	38	52	18.2794
9	12	26	38	14.3971
10	22	38	52	22.2614
Average	17.4	38.8	52.2	18.87587

4.3 Pengujian menggunakan Re-Ranking

Pada tahap pengujian ini dilakukan Re-Ranking pada hasil training dan testing 10 kali yang telah dilakukan pada model Residual Network 56 dan 110. Re-Ranking dilakukan dengan harapan dapat meningkatkan performa Rank-1 model, dikarenakan dari penelitian re-identifikasi orang, dapat dilihat bahwa *re-ranking* dapat meningkatkan performa Rank-1 dan mAP model.

4.3.1 Re-Ranking pada ResNet 56

Pada tahap pengujian ini dilakukan re-ranking terhadap 10 metrik evaluasi yang tertera pada tabel 4.6, dimana reranking merupakan pengambilan jarak antara citra query dan citra gallery yang terdekat. Dapat dilihat pada hasil *re-ranking* di tabel 4.10, bahwa semua metrik evaluasi yang digunakan menurun drastis. Yaitu masing-masing sebesar 6.2% , 9.6%, 5%, dan 0.18%. Hal ini menunjukkan bahwa re-identifikasi sketsa merupakan sebuah permasalahan yang sangat berbeda dengan re-identifikasi manusia.

Tabel 4.8: Rata-rata performa ResNet 56 dengan re-ranking

No	Rank-1	Rank-5	Rank-10	mAP
1	8	26	34	10.5473
2	8	28	38	11.8655
3	6	28	38	10.9266
4	8	22	30	10.9797
5	8	20	34	11.7746
6	6	26	34	10.4775
7	6	26	36	10.8615
8	10	26	34	12.5916
9	6	26	38	10.3757
10	14	26	44	14.4551
Average	8	25.4	36	11.48551

4.3.2 Re-Ranking pada ResNet 110

Sama dengan pada Residual Network 56 dapat dilihat bahwa re-ranking menyebabkan terjadinya penurunan pada semua metrik evaluasi.

Tabel 4.9: Rata-rata performa ResNet 110 dengan re-ranking

No	Rank-1	Rank-5	Rank-10	mAP
1	8	24	40	12.0668
2	10	34	36	13.4624
3	6	24	34	10.1525
4	10	26	38	14.866
5	16	32	46	17.1341
6	10	20	26	12.8491
7	14	30	36	16.3645
8	18	30	42	19.7371
9	6	22	36	9.6301
10	8	24	38	10.146
Average	10.6	26.6	37.2	13.640859

4.3.3 Kesimpulan Re-Ranking untuk Re-Identifikasi Sketsa

Dari pengujian yang dilakukan pada kedua model, dapat dilihat bahwa *re-ranking* yang dilakukan menyebabkan penurunan performa yang sangat drastis pada kedua model. Selain itu dapat dilihat bahwa pada semua metrik evaluasi, tidak ada yang meningkat, berbeda dengan pada re-identifikasi manusia dimana Rank-1 dan mean Average Precision meningkat. Oleh karena hasil yang kami dapatkan sebagai berikut maka percobaan dengan *re-ranking* tidak akan dilanjutkan lebih lagi. Tabel dibawah menunjukkan perbedaan performa rata-rata dari kedua model dengan hasil re-rankingnya.

4.4 Pengujian menggunakan Re-Ranking

Pada tahap pengujian ini dilakukan Re-Ranking pada hasil training dan testing 10 kali yang telah dilakukan pada model Residual Network 56 dan 110. Re-Ranking dilakukan dengan harapan dapat meningkatkan performa Rank-1 model, dikarenakan dari penelitian re-identifikasi orang, dapat dilihat bahwa *re-ranking* dapat meningkatkan performa Rank-1 dan mAP model.

4.4.1 Re-Ranking pada ResNet 56

Pada tahap pengujian ini dilakukan re-ranking terhadap 10 metrik evaluasi yang tertera pada tabel 4.6, dimana reranking merupakan pengambilan jarak antara citra query dan citra gallery yang terdekat. Dapat dilihat pada hasil *re-ranking* di tabel 4.10, bahwa semua metrik evaluasi yang digunakan menurun drastis. Yaitu masing-masing sebesar 6.2% , 9.6%, 5%, dan 0.18%. Hal ini menunjukkan bahwa re-identifikasi sketsa merupakan sebuah permasalahan yang sangat berbeda dengan re-identifikasi manusia.

Tabel 4.10: Rata-rata performa ResNet 56 dengan re-ranking

No	Rank-1	Rank-5	Rank-10	mAP
1	8	26	34	10.5473
2	8	28	38	11.8655
3	6	28	38	10.9266
4	8	22	30	10.9797
5	8	20	34	11.7746
6	6	26	34	10.4775
7	6	26	36	10.8615
8	10	26	34	12.5916
9	6	26	38	10.3757
10	14	26	44	14.4551
Average	8	25.4	36	11.48551

4.4.2 Re-Ranking pada ResNet 110

Sama dengan pada Residual Network 56 dapat dilihat bahwa re-ranking menyebabkan terjadinya penurunan pada semua metrik evaluasi.

Tabel 4.11: Rata-rata performa ResNet 110 dengan re-ranking

No	Rank-1	Rank-5	Rank-10	mAP
1	8	24	40	12.0668
2	10	34	36	13.4624
3	6	24	34	10.1525
4	10	26	38	14.866
5	16	32	46	17.1341
6	10	20	26	12.8491
7	14	30	36	16.3645
8	18	30	42	19.7371
9	6	22	36	9.6301
10	8	24	38	10.146
Average	10.6	26.6	37.2	13.640859

4.4.3 Kesimpulan Re-Ranking untuk Re-Identifikasi Sketsa

Dari pengujian yang dilakukan pada kedua model, dapat dilihat bahwa *re-ranking* yang dilakukan menyebabkan penurunan performa yang sangat drastis pada kedua model. Selain itu dapat dilihat bahwa pada semua metrik evaluasi, tidak ada yang meningkat, berbeda dengan pada re-identifikasi manusia dimana Rank-1 dan mean Average Precision meningkat. Oleh karena hasil yang kami dapatkan sebagai berikut maka percobaan dengan *re-ranking* tidak akan dilanjutkan lebih lagi. Tabel dibawah menunjukkan perbedaan performa rata-rata dari kedua model dengan hasil re-rankingnya.

Name	Rank-1	Rank-5	Rank-10	mAP
ResNet 56	14.2%	35%	47.8%	16.95%
ResNet 56 + re-rank	8%	25.4%	36%	11.49%
ResNet 110	17.4%	38.8%	52.2%	18.88%
ResNet 110 + re-rank	10.6%	26.6%	37.2%	13.64%

4.5 Studi Ablasi

4.5.1 Perubahan pada Fully Connected Layer

Pada tahap pengujian ini dilakukan perubahan *Fully Connected layer* pertama yang terdapat pada ResNet 56 dan ResNet 110. Studi Ablasi hanya dilakukan pada kedua model tersebut dikarenakan terjadi peningkatan yang sangat signifikan dibanding dengan menggunakan ResNet 20, yaitu 40% dan 70%. Pengujian ini dilakukan dengan 5 jenis *Fully Connected Layer* yaitu:

1. *Fully Connected* 128
2. *Fully Connected* 256
3. *Fully Connected* 512
4. *Fully Connected* 768
5. *Fully Connected* 1024

Semua pengujian sebelumnya dilakukan dengan *Fully Connected layer* sebesar 512. Oleh karena itu, pada bagian ini dilakukan pengujian pada Fully Connected lainnya yang tertera.

4.5.2 ResNet 56 Fully Connected 128

Tabel 4.12: Rata-rata performa ResNet 56 Fully Connected 128 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	8	42	52	13.4307
2	14	36	48	15.8562
3	12	28	40	13.4272
4	10	34	50	15.0407
5	10	34	46	13.7825
6	10	38	58	15.3561
7	12	36	50	17.1585
8	14	38	50	16.2073
9	10	42	58	16.2594
10	14	36	48	17.6873
Average	11.4	36.4	50	15.42059

Pada penggunaan Fully Connected Layer sebesar 128, terjadi penurunan pada Rank-1 accuracy sebesar 2.8% apabila dibandingkan dengan 512 layer. Selain itu Rank-5 dan mean Average Precision yang didapatkan kedua nya lebih rendah apabila dibandingkan dengan 512 layer, yaitu masing masing mendapatkan penurunan sebesar 2.2% dan 1.53306%.

4.5.3 ResNet 56 Fully Connected 256

Pada penggunaan Fully Connected Layer sebesar 256, terjadi penurunan pada Rank-1 accuracy sebesar 1.4% apabila dibandingkan dengan 512 layer. Selain itu Rank-5, Rank-10 dan mean Average Precision yang didapatkan semuanya mendapatkan hasil yang lebih rendah apabila dibandingkan dengan 512 layer.

Dari dua percobaan ini dapat disimpulkan bahwa menggunakan layer dibawah 512 tidak seefektif menggunakan 512 layer dikarenakan adanya informasi yang hilang yang diakibatkan oleh serialisasi data yang dilakukan oleh adanya Fully Connected Layer.

Tabel 4.13: Rata-rata performa ResNet 56 Fully Connected 256 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	16	32	48	16.7992
2	14	32	48	17.5719
3	14	26	40	15.5429
4	12	26	46	14.9202
5	12	30	50	14.5889
6	10	32	52	13.3994
7	8	28	44	12.6595
8	10	36	48	16.2244
9	16	38	54	18.6509
10	16	38	50	16.84
Average	12.8	31.8	48	15.7197299

4.5.4 ResNet 56 Fully Connected 768

Pada penggunaan FC Layer sebesar 768, terjadi penurunan pada rank1 accuracy sebesar 0.4% apabila dibandingkan dengan 512 layer.

Tabel 4.14: Rata-rata performa ResNet 56 Fully Connected 768 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	8	44	48	13.5317
2	16	36	50	16.791
3	12	38	54	15.9576
4	14	32	42	16.0422
5	10	34	44	13.8197
6	18	26	54	19.7984
7	14	40	48	19.3302
8	14	34	44	15.6428
9	18	38	44	17.7032
10	14	36	40	16.417
Average	13.8	35.8	46.8	16.50338

Namun Rank-5 yang didapatkan lebih tinggi apabila dibandingkan dengan menggunakan 512 layer. Penurunan performa ini dapat terjadi dikarenakan adanya overfitting yang disebabkan oleh penambahan Fully Connected Layer, sehingga performa dari model tidak stabil.

4.5.5 ResNet 56 Fully Connected 1024

Tabel 4.15: Rata-rata performa ResNet 56 Fully Connected 1024 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	14	32	48	15.9998
2	18	46	58	18.8938
3	14	28	52	16.1321
4	24	36	48	21.9224
5	10	30	44	15.1594
6	18	36	56	18.5048
7	14	40	50	18.6785
8	14	40	52	19.3363
9	12	36	52	15.2779
10	12	32	50	15.9013
Average	15	35.6	51	17.58063

Pada penggunaan FC Layer sebesar 1024, terjadi peningkatan pada Rank-1 accuracy sebesar 0.8% apabila dibandingkan dengan 512 layer. Selain itu performa yang didapatkan pada Rank-5, Rank-10, dan mAP semua mendapatkan peningkatan performa, yaitu masing-masing sebanyak 0.6%, 3.2%, dan 0.62%.

4.5.6 Kesimpulan FC Layer pada ResNet 56

Dari semua pengujian yang dilakukan dapat dilihat bahwa pengujian naik perlahan dari Fully Connected layer 128, 256, 512. Namun pada saat mencapai 768 layer, performa dari model menjadi tidak stabil dan naik turun. Hal tersebut dapat dilihat dari rata-rata 10 kali pengujian pada 768 layer dan 1024 layer.

4.5.7 ResNet 110 Fully Connected 128

Tabel 4.16: Rata-rata performa ResNet 110 Fully Connected 128 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	14	36	54	16.5342
2	12	34	50	15.9617
3	16	38	50	17.08
4	12	34	48	15.1201
5	12	32	40	14.5804
6	10	34	50	13.7103
7	12	42	50	15.9939
8	14	38	46	15.9738
9	16	32	44	16.66048
10	12	38	58	16.5114
Average	13	35.8	49	15.812628

Pada penggunaan Fully Connected Layer sebesar 128, terjadi penurunan pada Rank-1 accuracy sebesar 4.4% apabila dibandingkan dengan 512 layer. Selain itu Rank-5, Rank-10 dan mean Average Precision yang didapatkan semuanya lebih rendah apabila dibandingkan dengan 512 layer, yaitu masing masing mendapatkan penurunan sebesar 3%, 3.2%, dan 3.06%.

4.5.8 ResNet 110 Fully Connected 256

Pada penggunaan Fully Connected Layer sebesar 256, terjadi penurunan pada Rank-1 accuracy sebesar 3.6% apabila dibandingkan dengan 512 layer. Selain itu Rank-5 dan mean Average Precision yang didapatkan mendapatkan hasil yang lebih rendah apabila dibandingkan dengan 512 layer.

Dari dua percobaan ini dapat disimpulkan bahwa menggunakan layer dibawah 512 tidak seefektif menggunakan 512 layer dikarenakan adanya informasi yang hilang yang diakibatkan oleh serialisasi data yang dilakukan oleh adanya Fully Connected Layer.

Tabel 4.17: Rata-rata performa ResNet 110 Fully Connected 256 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	14	34	46	15.2228
2	10	38	54	14.5828
3	16	36	50	19.2416
4	20	42	54	21.7159
5	12	32	56	15.9766
6	10	44	56	14.7769
7	12	38	54	15.1684
8	12	38	54	16.4656
9	16	40	50	17.2462
10	16	42	54	18.3998
Average	13.8	38.4	52.8	16.87965999

4.5.9 ResNet 110 Fully Connected 768

Pada penggunaan FC Layer sebesar 768, terjadi penurunan pada Rank-1 accuracy sebesar 3% apabila dibandingkan dengan 512 layer.

Tabel 4.18: Rata-rata performa ResNet 110 Fully Connected 768 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	18	44	60	18.9191
2	16	40	48	17.0235
3	14	36	52	15.8576
4	12	38	54	16.2625
5	16	36	48	19.3128
6	14	34	42	18.631
7	14	36	44	15.2293
8	12	30	38	12.5104
9	16	38	56	17.084
10	12	36	46	14.0698
Average	14.4	36.8	48.8	16.49

Namun Rank-5 yang didapatkan lebih tinggi apabila dibandingkan dengan menggunakan 512 layer. Penurunan performa ini dapat terjadi dikarenakan adanya overfitting yang disebabkan oleh penambahan Fully Connected Layer, sehingga performa dari model tidak stabil, sama seperti pada ResNet 56.

4.5.10 ResNet 110 Fully Connected 1024

Tabel 4.19: Rata-rata performa ResNet 110 Fully Connected 1024 *pretrained* pada Market 1501

No	Rank-1	Rank-5	Rank-10	mAP
1	10	36	50	14.307
2	16	40	52	18.0064
3	20	42	58	21.334
4	16	42	52	16.4967
5	22	48	60	25.3244
6	12	34	50	15.2809
7	20	36	62	21.1675
8	16	34	46	17.2766
9	14	48	60	16.4865
10	12	30	44	14.861
Average	15.8	39	53.4	18.0541

Pada penggunaan FC Layer sebesar 1024, terjadi penurunan pada Rank-1 accuracy sebesar 1.6% apabila dibandingkan dengan 512 layer. Namun performa yang didapatkan pada Rank-5, Rank-10 mendapatkan peningkatan performa, yaitu masing-masing sebanyak 0.2%, 1.2%.

4.5.11 Kesimpulan FC Layer pada ResNet 110

Dari semua pengujian yang dilakukan dapat dilihat bahwa pengujian naik perlahan dari Fully Connected layer 128, 256, 512. Namun pada saat mencapai 768 layer, terjadi overfitting pada model sehingga performa model turun perlahan.

4.5.12 Perubahan pada Probabilitas *Random Erasing*

Pada tahap pengujian ini dilakukan perubahan probabilitas *Random Erasing*, dimana pengujian ini dilakukan dengan 6 jenis *Random Erasing* yaitu:

1. *Random Erasing* 0%
2. *Random Erasing* 10%
3. *Random Erasing* 20%
4. *Random Erasing* 30%
5. *Random Erasing* 40%
6. *Random Erasing* 50%

Namun dikarenakan pengujian sebelumnya telah dilakukan dengan *Random Erasing* 50%, maka hanya pengujian tersisa yang dituliskan.

4.5.13 ResNet 56 *Random Erasing* 0%

Tabel 4.20: Rata-rata performa ResNet 56 *Random Erasing* 0%

No	Rank-1	Rank-5	Rank-10	mAP
1	18	36	56	19.2064
2	14	38	50	17.4292
3	18	38	46	21.0595
4	20	36	62	22.4735
5	18	38	52	20.4456
6	20	36	50	21.4024
7	10	28	42	13.1137
8	14	36	50	19.71
9	14	40	50	18.1666
10	16	38	48	18.4812
Average	16.2	36.4	50.6	19.14881

Performa dari model yang dibuat lebih baik dibandingkan dengan menggunakan *Random Erasing* 50%. Dimana pada Rank-1 yang didapatkan terdapat peningkatan sebesar 2%, selain itu Rank-5, Rank-10, dan mAP yang didapatkan semuanya lebih tinggi.

4.5.14 ResNet 56 *Random Erasing* 10%

Tabel 4.21: Rata-rata performa ResNet 56 Random Erasing 10%

No	Rank-1	Rank-5	Rank-10	mAP
1	14	36	50	18.2589
2	12	48	56	17.2826
3	16	30	44	18.0019
4	22	44	60	24.5899
5	10	34	52	18.5679
6	14	38	48	17.2811
7	10	46	58	16.8315
8	14	30	54	17.3568
9	14	36	48	16.2783
10	20	46	54	22.2397
Average	14.6	38.8	52.4	18.66886

Pada penggunaan Random Erasing sebesar 10%, rank1 accuracy menurun sebesar 2% apabila dibandingkan dengan ketika tidak dilakukan Random Erasing. Selain itu metrik evaluasi yang didapatkan semua nya lebih rendah dari pengujian sebelumnya.

4.5.15 ResNet 56 *Random Erasing* 20%

Tabel 4.22: Rata-rata performa ResNet 56 Random Erasing 20%

No	Rank-1	Rank-5	Rank-10	mAP
1	14	32	50	15.9493
2	16	36	44	17.8214
3	12	28	52	16.0317
4	10	36	48	15.425
5	20	44	52	22.233
6	14	36	48	16.2473
7	14	34	48	18.1056
8	14	48	62	18.3149
9	12	36	58	16.9772
10	16	42	50	21.1495
Average	14.2	37.2	51.2	17.82549

Pada penggunaan Random Erasing sebesar 20%, terjadi penurunan pada Rank-1 accuracy sebesar 2.4% apabila dibandingkan dengan tanpa menggunakan Random Erasing. Selain itu Rank-5, Rank-10 dan mean Average Precision yang didapatkan semuanya mendapatkan hasil yang lebih rendah apabila dibandingkan dengan pengujian sebelumnya.

Dari tiga percobaan ini dapat disimpulkan bahwa penggunaan Random Erasing mengurangi efektifitas model, dimana dapat dilihat bahwa performa model turun sangat signifikan dari tanpa menggunakan Random Erasing ke menggunakan Random Erasing sebesar 10%.

4.5.16 ResNet 56 *Random Erasing* 30%

Pada penggunaan Random Erasing sebesar 30%, terjadi penurunan pada Rank-1 accuracy sebesar 3.4% apabila dibandingkan dengan tanpa menggunakan Random Erasing.

Tabel 4.23: Rata-rata performa ResNet 56 Random Erasing 30%

No	Rank-1	Rank-5	Rank-10	mAP
1	12	34	44	18.1763
2	18	34	42	18.8824
3	14	32	60	16.3065
4	10	22	38	13.5783
5	10	32	46	14.0838
6	22	32	50	22.304
7	4	38	50	11.7184
8	8	28	46	13.8258
9	10	28	38	14.577
10	20	42	50	21.7607
Average	12.8	32.2	46.4	16.5213199

Selain itu terjadi penurunan performa di semua metrik evaluasi lainnya. Penurunan performa ini dapat terjadi dikarenakan adanya underfitting yang disebabkan oleh terlalu banyaknya data yang dihapus oleh Random Erasing.

4.5.17 ResNet 56 *Random Erasing* 40%

Tabel 4.24: Rata-rata performa ResNet 56 Random Erasing 40%

No	Rank-1	Rank-5	Rank-10	mAP
1	10	32	50	14.6807
2	10	40	54	15.7018
3	18	34	46	18.469
4	8	42	56	14.5516
5	14	34	48	17.6434
6	18	32	50	19.003
7	12	32	44	17.2603
8	12	40	50	15.8508
9	18	36	48	19.563
10	14	38	48	17.0508
Average	13.4	36	49.4	16.97744

Terjadi peningkatan performa apabila dibandingkan dengan percobaan sebelumnya, peningkatan performa ini dapat terjadi dikarenakan probabilitas Random Erasing cukup besar sehingga performa model tidak konsisten.

4.5.18 ResNet 110 *Random Erasing* 0%

Tabel 4.25: Rata-rata performa ResNet 110 Random Erasing 0%

No	Rank-1	Rank-5	Rank-10	mAP
1	14	30	42	16.4663
2	24	40	46	22.5784
3	16	34	48	17.7241
4	20	34	44	20.0728
5	22	44	56	23.8577
6	24	38	42	23.8481
7	24	36	52	22.8688
8	20	40	56	22.4153
9	16	38	50	19.2107
10	18	40	42	21.3032
Average	19.8	37.4	47.8	21.03454

Performa dari model yang dibuat lebih baik dibandingkan dengan menggunakan Random Erasing 50%. Dimana pada Rank-1 yang didapatkan terdapat peningkatan sebesar 2%, selain itu Rank-5, Rank-10, dan mAP yang didapatkan semuanya lebih tinggi.

4.5.19 ResNet 110 *Random Erasing* 10%

Tabel 4.26: Rata-rata performa ResNet 110 Random Erasing 20%

No	Rank-1	Rank-5	Rank-10	mAP
1	18	38	54	21.5794
2	18	34	40	18.8688
3	20	40	54	21.5792
4	24	34	42	22.799
5	14	36	48	17.734
6	20	38	50	19.0839
7	16	38	48	17.0347
8	22	42	54	21.1928
9	14	36	54	21.0527
10	20	34	54	20.276
Average	18.6	37	49.8	20.12005

Pada penggunaan Random Erasing sebesar 10%, Rank-1 accuracy menurun sebesar 2.2% apabila dibandingkan dengan ketika tidak dilakukan Random Erasing. Selain itu Rank-1, Rank-5, dan Rank-10 yang didapatkan semuanya lebih rendah dari pengujian sebelumnya.

4.5.20 ResNet 110 *Random Erasing* 20%

Pada penggunaan Random Erasing sebesar 20%, terjadi penurunan pada Rank-1 accuracy sebesar 2.4% apabila dibandingkan dengan tanpa menggunakan Random Erasing. Selain itu Rank-5, Rank-10 dan mean Average Precision yang didapatkan semuanya mendapatkan hasil yang lebih rendah apabila dibandingkan dengan pengujian sebelumnya.

Tabel 4.27: Rata-rata performa ResNet 110 Random Erasing 20%

No	Rank-1	Rank-5	Rank-10	mAP
1	24	36	54	24.1813
2	16	34	58	19.1003
3	16	42	54	17.8024
4	16	36	54	19.0329
5	12	40	46	15.668
6	22	38	58	22.0568
7	26	48	58	23.2754
8	18	40	54	20.0522
9	14	32	46	15.8306
10	22	48	56	22.317
Average	18.6	39.4	53.8	19.93169

Dari tiga percobaan ini dapat disimpulkan bahwa penggunaan Random Erasing mengurangi efektifitas model, dimana dapat dilihat bahwa performa model turun sangat signifikan dari tanpa menggunakan Random Erasing ke menggunakan Random Erasing sebesar 10%. Selain itu performa model juga menjadi tidak konsisten seperti yang dilihat pada Random Erasing 10% dan 20%.

4.5.21 ResNet 110 *Random Erasing* 30%

Tabel 4.28: Rata-rata performa ResNet 110 Random Erasing 30%

No	Rank-1	Rank-5	Rank-10	mAP
1	16	42	52	16.9698
2	16	28	42	17.3695
3	24	34	40	21.3444
4	18	48	50	21.2043
5	16	38	52	18.3336
6	14	40	48	17.6614
7	12	42	56	17.101
8	16	24	38	16.1731
9	18	46	58	20.901
10	16	38	54	18.9849
Average	16.6	38	49	18.60430

Pada penggunaan Random Erasing sebesar 30%, terjadi penurunan pada Rank-1 accuracy sebesar 4.4% apabila dibandingkan dengan tanpa menggunakan Random Erasing.

Selain itu terjadi penurunan performa di semua metrik evaluasi lainnya, yaitu Rank-5 accuracy, Rank-10 accuracy, dan mean Average Precision. Sama seperti pada Residual Network 56, penurunan performa ini dapat terjadi dikarenakan adanya underfitting yang disebabkan oleh terlalu banyaknya data yang dihapus oleh Random Erasing, menyebabkan performa model tidak konsisten dalam melakukan klasifikasi.

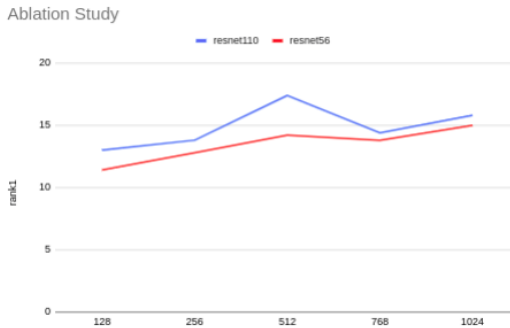
4.5.22 ResNet 110 *Random Erasing* 40%

Terjadi penurunan performa lagi apabila dibandingkan dengan percobaan sebelumnya, yaitu ketika menggunakan Random Erasing sebesar 30%. Penurunan performa ini dapat dipastikan dikarenakan probabilitas Random Erasing cukup besar sehingga performa model tidak konsisten. Dikarenakan kedua model Residual Network yang diuji memiliki performa paling baik ketika tidak menggunakan Random Erasing, maka semua pengujian selanjutnya akan dilakukan dengan tanpa menggunakan Random Erasing.

Tabel 4.29: Rata-rata performa ResNet 110 Random Erasing 40%

No	Rank-1	Rank-5	Rank-10	mAP
1	18	40	54	22.619
2	8	32	50	13.84
3	12	38	52	17.3868
4	14	36	46	17.4407
5	18	36	48	20.2284
6	16	30	42	17.5272
7	20	44	56	24.2339
8	10	44	56	16.0246
9	26	46	64	25.0649
10	16	46	58	22.181
Average	15.8	39.2	52.6	19.65465

4.5.23 Hasil Studi Ablasi

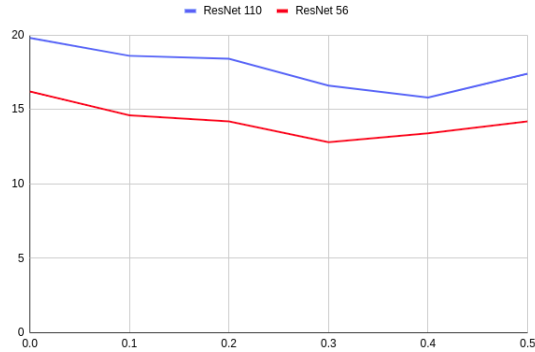


Gambar 4.1: Hasil Studi Ablasi pada *Fully Connected Layer*

Dari hasil percobaan yang dilakukan didapatkan bahwa menggunakan *Fully Connected Layer* sebesar 512 mendapatkan hasil yang paling baik, dimana ketika menggunakan *Fully Connected Layer* lebih dari 512 menyebabkan *overfitting*. Sedangkan ketika menggunakan *Fully Connected Layer* dibawah 512 tidak se-efektif menggunakan 512 layer. Gambar 4.1 menunjukan grafik Rank-1 terhadap *Fully Connected Layer*.

Tabel 4.30: Rata-rata performa semua ResNet dengan *Fully Connected Layer* berbeda-beda

Name	Rank-1	Rank-5	Rank-10	mAP
ResNet 56 FC 128	11.4%	36.4%	50%	15.4205%
ResNet56 FC 256	12.8%	31.8%	48%	15.71973%
ResNet 56 FC 512	14.2%	35%	47.8%	16.95365%
ResNet56 FC 768	13.8%	35.8%	46.8%	16.50338%
ResNet 56 FC 1024	15%	35.6%	51%	17.58063%
ResNet 110 FC 128	13%	35.8%	49%	15.81262%
ResNet 110 FC 256	13.8%	38.4%	52.8%	16.87966%
ResNet 110 FC 512	17.4%	38.8%	52.2%	18.87587%
ResNet 110 FC 768	14.4%	36.8%	48.8%	16.49%
ResNet 110 FC 1024	15.8%	39%	53.4%	18.0541%



Gambar 4.2: Hasil Studi Ablasi pada *Random Erasing*

Sedangkan dari hasil percobaan pada *Random Erasing*, didapatkan bahwa tidak menggunakan Random Erasing merupakan yang paling baik untuk model, dimana ketika menggunakan *Random Erasing* performa model menurun. Sehingga ketika menggunakan Random Erasing 40% keatas performa model menjadi tidak konsisten. Gambar 4.2 menunjukkan grafik Rank-1 terhadap *Random Erasing*.

4.6 Pengujian dengan menggunakan Ensemble

Untuk meningkatkan performa dari model sendiri, dibuat sebuah ensemble dari model ResNet 56 dan ResNet 110, dimana Fully Connected layer yang digunakan sejumlah 512 layer dan tidak dilakukan Random Erasing. Selain itu dilakukan penambahan Spatial Pyramid Pooling pada ensemble dengan harapan terjadi penambahan presisi dan akurasi. Pada semua model telah dilakukan *pre-training* di dataset Market 1501.

4.6.1 Ensemble

Pada model ini terjadi peningkatan pada Rank-1, Rank-5, Rank-10 dan mAP apabila dibandingkan dengan ResNet 110.

Tabel 4.31: Rata-rata performa ensemble

No	Rank-1	Rank-5	Rank-10	mAP
1	24	40	48	23.5601
2	22	40	52	23.8937
3	24	42	60	22.5622
4	14	46	50	18.1273
5	26	48	58	24.3952
6	22	44	48	22.6432
7	24	44	50	24.3334
8	18	42	54	21.0525
9	14	40	58	18.2236
10	22	38	46	21.7579
Average	21	42.4	52.4	22.05491

Namun apabila dilihat dari waktu training yang dibutuhkan ensemble ini membutuhkan lebih lama dari hanya menggunakan ResNet 110.

4.6.2 Ensemble + Spatial Pyramid Pooling

Tabel 4.32: Rata-rata performa ensemble + Spatial Pyramid Pooling

No	Rank-1	Rank-5	Rank-10	mAP
1	22	44	54	21.425
2	18	44	56	19.6589
3	12	34	48	16.9387
4	26	44	50	23.9654
5	16	32	44	17.8318
6	22	48	58	22.1528
7	26	40	52	23.0882
8	22	44	54	21.987
9	20	42	56	21.0866
10	18	48	58	21.3644
Average	20.2	42	53	20.94988

Pada pengujian dengan tambahan Spatial Pyramid Pooling terjadi overfitting sehingga semua metrik evaluasi yang digunakan menurun.

4.7 Perbandingan dengan model-model lain

Tabel 4.33 menunjukkan perbandingan performa model yang dibuat dengan model-model lain. Dari tabel tersebut dapat dilihat bahwa pada model yang dibuat Rank-1 dan Rank-5, dan Rank-10 yang didapatkan bukanlah yang terbaik. Namun pada kolom parameter dapat dilihat bahwa parameter yang dimiliki tidak sebanyak model-model lain yang pernah digunakan untuk memecahkan masalah pada dataset PKU Sketch Re-ID. Bahkan ensemble yang paling berat hanya memiliki sekitar 28.6% parameter dari model Dense-HOG+LBP+Siamese.

Tabel 4.33: Perbandingan dengan model-model lain

Nama	Param	Rank-1	Rank-5	Rank-10
Dense-HOG+LBP+SVM	8.6M	5.1%	16.8%	28.3%
Triplet SN	n/a	9%	26.8%	43.2%
GN Siamese	14M	28.9%	54%	62.4%
Cross-Domain Adversarial	n/a	34%	56.3%	72.5%
Ensemble FC 512	3M	21 %	42.4%	52.4%
Ensemble SPP	3M	20.2%	42%	53%
ResNet 110 FC 512	1.7M	19.8%	37.4%	47.8%

[Halaman ini sengaja dikosongkan]

BAB V

PENUTUP

5.1 Kesimpulan

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

1. CycleGAN dapat meningkatkan Rank-1 *accuracy* dan mAP, namun menurunkan metrik evaluasi sisanya. Dikarenakan model dinilai menurut Rank-1 *accuracy* nya maka CycleGAN disarankan untuk digunakan.
2. *Local Binary Pattern* memiliki *performance cost* yang sangat tinggi, namun apabila dilihat dari metrik evaluasi, performa dari model menurun di semua metrik.
3. *Pre-training* pada dataset Market-1501 sebelum melakukan *training* pada dataset PKU-Sketch Re-ID meningkatkan semua metrik evaluasi.
4. Penggunaan *Fully Connected Layer* sebesar 512 merupakan yang terbaik, apabila menggunakan *Fully Connected Layer* yang lebih besar dari 512 maka akan terjadi overfitting. Namun apabila menggunakan *Fully Connected Layer* dibawah 512, maka tidak se-optimal 512 layer.
5. Penggunaan *Random Erasing* membuat performa model tidak konsisten, dari studi ablasi yang dilakukan lebih baik apabila tidak menggunakan *Random Erasing*.
6. Model yang digunakan memiliki metrik evaluasi yang lebih baik dari DenseNet, sebuah model klasikal yang memiliki parameter lebih banyak 4 kali lipat dari ResNet CIFAR10
7. Model yang digunakan memiliki metrik evaluasi yang lebih baik dari Triplet SN, sebuah model yang dibuat dari tiga model Sketch-a-Nets dan di optimisasikan dengan triplet loss.
8. Meskipun bukan merupakan model yang terbaik, model mampu mendapatkan informasi dengan jauh lebih efisien dibandingkan model-model lain.

5.2 Saran

Untuk pengembangan lebih lanjut pada Tugas Akhir ini, terdapat beberapa saran yang dapat dilakukan, yakni sebagai berikut:

1. Apabila dataset yang digunakan lebih pemasangan sketch dengan dataset membutuhkan kekuatan komputasi yang lebih besar sehingga tidak disarankan untuk menggunakan CycleGAN.
2. Penggunaan Cross Domain Adversarial Learning untuk menjembatani perbedaan modalitas antara citra sketsa dengan citra CCTV.
3. Apabila pada dataset terdapat lebih banyak data maka *pre-training* pada dataset Market 1501 tidak perlu dilakukan.
4. Melakukan studi ablasi lebih lanjut dengan melakukan perubahan pada ukuran citra input, Random Rotation, dan Batch Size.

DAFTAR PUSTAKA

- [1] Booz Allen. How do machines learn?, 2017. <https://www.boozallen.com/s/insight/blog/how-do-machines-learn.html>.
- [2] Pablo Ruiz. Resnet for cifar-10, 2018. <https://towardsdatascience.com/resnets-for-cifar-10-e63e900524e0>, Last accessed on 2020-11-30.
- [3] Badan Pusat Statistik Jakarta. Provinsi dki jakarta dalam angka, 2020. <https://jakarta.bps.go.id/publication/2020/02/28/75c2aa351fc1acb44fcea04f/provinsi-dki-jakarta-dalam-angka-2020--penyediaan-data-untuk-perencanaan-pembangunan.html> Last accessed on 2020-11-27.
- [4] Badan Pusat Statistik Surabaya. Kota surabaya dalam angka, 2020. <https://surabayakota.bps.go.id/publication/2020/05/19/4b5506b7a089a61c75ef6cc9/kota-surabaya-dalam-angka-2020.html>.
- [5] Kasus kriminal meningkat 7,04 persen dalam sepekan, salah satunya perampokan, 2020. <https://nasional.kompas.com/read/2020/05/18/16253371/kasus-kriminal-meningkat-704-persen-dalam-sepekan-salah-satunya-perampokan>.
- [6] Dua pekan terakhir, polri catat peningkatan kejahatan 11,80 persen, 2020. <https://nasional.kompas.com/read/2020/04/20/20542321/dua-pekan-terakhir-polri-catat-peningkatan-kejahatan-1180-persen>.
- [7] Ini alasan angka kriminalitas meningkat pekan lalu menurut polri, 2020. <https://nasional.kompas.com/read/2020/05/18/16253371/kasus-kriminal-meningkat-704-persen-dalam-sepekan-salah-satunya-perampokan>.

- [8] Dalam sepekan, polri catat peningkatan kejahatan jalanan di indonesia, 2020. <https://nasional.kompas.com/read/2020/05/12/17363331/dalam-sepekan-polri-catat-peningkatan-kejahatan-jalanan-di-indonesia>.
- [9] Reza Fuad Rachmadi, Supeno Mardi Susiki Nugroho, and I Ketut Eddy Purnama. Lightweight residual network for person re-identification. In *International Conference on Information Technology and Digital Applications (ICITDA)*, 2020.
- [10] Kaiyang Zhou and Tao Xiang. Torchreid: A library for deep learning person re-identification in pytorch, 2019.
- [11] Xingyang Ni, Liang Fang, and Heikki Huttunen. Adaptive l2 regularization in person re-identification, 2020.
- [12] Lu Pang, Yaowei Wang, Yi-Zhe Song, Tiejun Huang, Yonghong Tian. Cross-domain adversarial feature learning for sketch re-identification. 2018. <https://www.pkuml.org/resources/pkusketchreid-dataset.html>, Last accessed on 2020-11-30.
- [13] Robert Sancrainte. A brief history of artificial intelligence: Thomas bayes, 2017. <https://www.lexpredict.com/2017/02/brief-history-ai-thomas-bayes/>.
- [14] Adam Paszke, Sam Gross ,Soumith Chintala, Gregory Chanan. Pytorch, 2016. <https://pytorch.org/>, Last accessed on 2020-11-30.
- [15] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for image recognition, 2015. <https://arxiv.org/abs/1512.03385>.
- [17] Facundo Bre, Juan Gimenez, and Víctor Fachinotti. Prediction of wind pressure coefficients on building surfaces using artificial neural networks page 4. *Energy and Buildings*, 158, 11 2017. doi: 10.1016/j.enbuild.2017.11.045.

- [18] Mahmoud Harmouch. Local binary pattern algorithm: The math behind it, 2020. <https://medium.com/swlh/local-binary-pattern-algorithm-the-math-behind-it-%EF%B8%8F-edf7b0e1c8b3>, Last accessed on 2021-5-30.
- [19] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. 2017.
- [20] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [21] Zhedong Zheng, Liang Zheng, and Yi Yang. A discriminatively learned cnn embedding for person reidentification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(1):13, 2018.
- [22] Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, and Tao Mei. Vehiclenet: Learning robust visual representation for vehicle re-identification. *IEEE Transaction on Multimedia (TMM)*, 2020.
- [23] Yerlan Idelbayev. Proper ResNet implementation for CIFAR10/CIFAR100 in PyTorch. https://github.com/akamaster/pytorch_resnet_cifar10. Accessed: 2021-06-04.
- [24] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.
- [25] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.
- [26] Ejaz Ahmed, Michael Jones, and Tim K. Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

- [27] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [28] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 2016.
- [29] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [30] Niki Martinel, Gian Luca Foresti, and Christian Micheloni. Aggregating deep pyramidal representations for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

BIOGRAFI PENULIS



Charles Chang atau yang lebih akrab disapa Chang, lahir di Surabaya, Jawa Timur pada 18 Agustus 1999. Merupakan anak pertama dari dua bersaudara. Penulis lulus dari SMP Kristen Petra 5 dan kemudian melanjutkan ke SMA Kristen Petra 1 Surabaya. Penulis merupakan salah satu mahasiswa dari Departemen Teknik Komputer Surabaya, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember. Dalam masa kuliah, penulis sangat tertarik dengan *Machine Learning*, *Image Processing*, dan *Data Science*. Selain itu penulis juga hobi membuat permainan, dan aktif mengikuti lomba membuat permainan di masa kuliah. Bahkan pernah menjadi pemenang juara 3 di MAGE ITS, dan menjadi semi-finalis di Gemastik XII 2019.

[Halaman ini sengaja dikosongkan]