



PROPOSAL TUGAS AKHIR - EF184702

KLASIFIKASI FEW-SHOT CITRA HIPERSPEKTRAL MENGUNAKAN CONTRASTIVE LEARNING UNTUK PEMETAAN TUMPAHAN MINYAK

Fathan Abi Karami

NRP 5025211156

Dosen Pembimbing 1

Wijayanti Nurul Khotimah, S.Kom., M.Sc., Ph. D.

NIP 198603122012122004

Dosen Pembimbing 2

Prof. Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

NIP 197104281994122001

Program Studi S1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024

LEMBAR PENGESAHAN

KLASIFIKASI FEW-SHOT CITRA HIPERSPEKTRAL MENGGUNAKAN CONTRASTIVE LEARNING UNTUK PEMETAAN TUMPAHAN MINYAK

PROPOSAL TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Memperoleh gelar Sarjana Komputer pada
Program Studi S-1 Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh : **Fathan Abi Karami**

NRP. 5025211156

Disetujui oleh Tim Penguji Proposal Tugas Akhir:

- | | | |
|----|---|---------------|
| 1. | Wijayanti Nurul Khotimah, S.Kom., M.Sc., Ph. D. | Pembimbing |
| 2. | Prof. Dr.Eng. Nanik Suciati, S.Kom., M.Kom. | Ko-pembimbing |
| 3. | Nama dan gelar penguji | Penguji |
| 4. | Nama dan gelar penguji | Penguji |
| 5. | Nama dan gelar penguji | Penguji |

SURABAYA
Desember, 2024

APPROVAL SHEET

FEW-SHOT HYPERSPECTRAL IMAGE CLASSIFICATION WITH CONTRASTIVE LEARNING FOR OIL SPILL MAPPING

FINAL PROJECT PROPOSAL

Submitted to fulfill one of the requirements
for obtaining a degree Bachelor of Computer at
Undergraduate Study Program of Teknik Informatika
Department of Teknik Informatika
Faculty of Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

By: Fathan Abi Karami

NRP. 5025211156

Approved by Final Project Proposal Examiner Team:

- | | | |
|----|---|------------|
| 1. | Wijayanti Nurul Khotimah, S.Kom., M.Sc., Ph. D. | Advisor |
| 2. | Prof. Dr.Eng. Nanik Suciati, S.Kom., M.Kom. | Co-Advisor |
| 3. | Name of Examiner and academic title | Examiner |
| 4. | Name of Examiner and academic title | Examiner |
| 5. | Name of Examiner and academic title | Examiner |

**SURABAYA
December, 2024**

KLASIFIKASI FEW-SHOT CITRA HIPERSPEKTRAL MENGGUNAKAN CONTRASTIVE LEARNING UNTUK PEMETAAN TUMPAHAN MINYAK

Nama Mahasiswa / NRP : Fathan Abi Karami / 5025211156
Departemen : Teknik Informatika FTEIC - ITS
Dosen Pembimbing 1 : Wijayanti Nurul Khotimah, S.Kom., M.Sc., Ph. D.
Dosen Pembimbing 2 : Prof. Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

Abstrak

Insiden tumpahan minyak sangat berbahaya bagi lingkungan dan masyarakat di sekitar wilayah terdampak. Oleh karena itu, pemetaan tumpahan minyak penting dilakukan untuk memantau persebaran tumpahan minyak. Citra hiperspektral menyediakan informasi spektral yang kaya sehingga bermanfaat dalam proses pemantauan tumpahan minyak di lautan. Namun, sebagian besar penelitian terdahulu berfokus pada metode klasifikasi *supervised* untuk mendeteksi tumpahan minyak dari citra hiperspektral. Padahal metode *supervised* memerlukan banyak data *training* yang proses pelabelannya memerlukan waktu dan tenaga yang banyak sehingga metode *supervised* kurang cocok diterapkan untuk mengatasi permasalahan tumpahan minyak. Penelitian ini mengimplementasikan metode klasifikasi *Few-Shot* menggunakan strategi *contrastive learning* untuk memetakan tumpahan minyak dari citra hiperspektral, yang membutuhkan data pelatihan jauh lebih sedikit dibandingkan dengan metode *supervised*.

Pengembangan klasifikasi *Few-Shot* dilakukan melalui beberapa tahap, yaitu tahap *pre-processing*, tahap *pre-training*, tahap *fine-tuning*, dan tahap *testing*. Pertama, tahap *pre-processing*, di mana sampel dipisahkan menjadi sampel *training* dan sampel *testing*. Pada tahap ini, sampel *training* juga akan diaugmentasi menggunakan transformasi spektral dan spasial untuk menambah jumlah sampel *training*. Kedua, tahap *pre-training*, di mana *deep learning network* dengan arsitektur VGG-16 dilatih menggunakan strategi *contrastive learning*. Di sini, dua sampel *training* yang dipilih dengan membuat kombinasi pasangan antara sampel *training* akan dilatih agar pasangan *input* yang berasal dari kelas yang sama mempunyai kemiripan yang besar dan pasangan *input* yang berasal dari yang berbeda mempunyai kemiripan yang kecil. Ketiga, tahap *fine-tuning*, di mana *classification layer* ditambahkan ke model yang telah dilatih, lalu *layer* tersebut dilatih menggunakan sampel *training*. Terakhir, tahap pengujian dilakukan untuk mengukur performa model menggunakan sampel *testing*.

Kata kunci: Citra Hiperspektral, Klasifikasi Few-Shot, Contrastive Learning, VGG-16, Pemetaan Tumpahan Minyak.

FEW-SHOT HYPERSPECTRAL IMAGE CLASSIFICATION WITH CONTRASTIVE LEARNING FOR OIL SPILL MAPPING

Student Name / NRP: Fathan Abi Karami / 5025211156

Department : Teknik Informatika FTEIC - ITS

Advisor : Wijayanti Nurul Khotimah, S.Kom., M.Sc., Ph. D.

Co-Advisor : Prof. Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

Abstract

Oil spill incidents are very dangerous for the environment and communities around the affected areas. Therefore, oil spill mapping is important to monitor the distribution of oil spills. Hyperspectral image provides rich spectral information, making it useful in the process of monitoring oil spills in the ocean. However, most previous studies have focused on supervised classification methods to detect oil spills from hyperspectral imagery. In fact, the supervised method requires a lot of training data whose labeling process requires a lot of time and effort so that the supervised method is less applicable to overcome oil spill problems. This study implements the Few-Shot classification method using a contrastive learning strategy to map oil spills from hyperspectral images, which requires much less training data compared to supervised methods.

The development of the Few-Shot classification is carried out through several stages, namely the pre-processing stage, the pre-training stage, the fine-tuning stage, and the testing stage. First, the pre-processing stage, where the samples are split into training samples and testing samples. At this stage, the training samples will also be augmented using spectral and spatial transformations to increase the number of training samples. Second, the pre-training stage, where deep learning network with VGG-16 architecture are trained using a contrastive learning strategy. Here, two training samples selected by creating a pair combination between training samples will be trained so that input pairs from the same class have a large similarity and input pairs from different ones have a small similarity. Third, the fine-tuning stage, where the classification layer is added to the trained model, then the layer is trained using training samples. Finally, the testing stage is carried out to measure the performance of the model using testing samples.

Keywords: Hyperspectral Image, Few-Shot Classification, Contrastive Learning, VGG-16, Oil Spills Mapping.

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
ABSTRAK	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR/GRAFIK/DIAGRAM	viii
DAFTAR TABEL.....	ix
DAFTAR PSEUDOCODE	x
DAFTAR SINGKATAN	xi
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
BAB II TINJAUAN PUSTAKA.....	4
2.1 Penelitian Terkait.....	4
2.1.1 <i>Self-supervised learning-based oil spill detection of hyperspectral Images</i>	4
2.1.2 <i>Hyperspectral Remote Sensing Benchmark Database for Oil Spill Detection With an Isolation Forest-Guided Unsupervised Detector</i>	4
2.1.3 <i>Self-Supervised Spectral–Spatial Transformer Network for Hyperspectral Oil Spill Mapping</i>	5
2.1.4 Perbandingan Penelitian Terkait.....	7
2.2 Dasar Teori	8
2.2.1 Citra Hiperspektral	8
2.2.2 <i>Convolutional Neural Network</i>	9
2.2.3 VGG-16	10
2.2.4 <i>Contrastive Learning</i>	11
2.2.5 Klasifikasi Few-Shot	12
BAB III METODOLOGI.....	14
3.1 Metode yang dirancang	14
3.1.1 <i>Pre-Processing</i>	15
3.1.2 <i>Pre-Training Model</i>	15
3.1.3 <i>Fine-Tuning Model</i>	16

3.1.4	<i>Testing</i>	16
3.2	Peralatan pendukung.....	16
3.2.1	Kebutuhan Perangkat Keras	16
3.2.2	Kebutuhan Perangkat Lunak	16
3.3	Rencana Implementasi dan Uji Coba	17
3.3.1	Implementasi <i>Pre-Processing</i>	17
3.3.2	Implementasi <i>Pre-Training</i>	17
3.3.3	Implementasi <i>Fine-Tuning</i>	18
3.3.4	Implementasi <i>Testing</i>	18
3.3.5	Rencana Uji Coba.....	19
JADWAL KEGIATAN.....		20
DAFTAR PUSTAKA		21
LAMPIRAN-LAMPIRAN ATAU APPENDIKS		25

DAFTAR GAMBAR/GRAFIK/DIAGRAM

Gambar 2.1 Skema Self-supervised learning model yang ditawarkan penelitian 1 (Duan et al., 2022).	4
Gambar 2.2 Skema Unsupervised Oil Spill Detector yang ditawarkan pada penelitian 2 (Duan et al., 2023).	5
Gambar 2.3 Skema Self-Supervised SSTNet untuk oil spill detection yang ditawarkan oleh penelitian 3 (Kang et al., 2023).	6
Gambar 2.4 Struktur 2 modul spektral-spasial (a) SSPM. (b) SSRM source: (Kang et al., 2023).	6
Gambar 2.5 Arsitektur 3 transformer (a) Pooling Transformer. (b) Spektral Transformer. (c) Spasial Transformer (Kang et al., 2023).	7
Gambar 2.6 (a) Citra hiperspektral direpresentasikan sebagai kubus 3D. Suatu titik spektrum pada kubus spektral direpresentasikan pada lokasi spasial. (b) Citra RGB. (c) Citra grayscale yang di-render dari kubus hiperspektral (Khan et al., 2018).	8
Gambar 2.7 Diagram CNN layer dan FC layer (Li et al., 2022).	9
Gambar 2.8 Arsitektur Model VGG-16 (Kamal and EZ-ZAHRAOUI, 2023).	11
Gambar 2.9 Contrastive Learning (Le-Khac et al., 2020).	12
Gambar 3.1 Skema Metode yang Dirancang.	14
Gambar 3.2 Skema Pre-Training Model.	15
Gambar 3.3 Skema Fine-Tuning Model.	16

DAFTAR TABEL

Tabel 2.1 Perbandingan dari ketiga penelitian terkait.	7
Tabel 3.1 Kebutuhan Perangkat Keras.	16
Tabel 3.2 Kebutuhan Perangkat Lunak.	17
Tabel 3.3 Hasil Uji Performa.....	19

DAFTAR PSEUDOCODE

Pseudocode 3.1 Pre-processing.	17
Pseudocode 3.2 Pre-training.	18
Pseudocode 3.3 Fine-tuning.	18
Pseudocode 3.4 Testing.	19

DAFTAR SINGKATAN

VIS	: <i>Visible</i>
IR	: <i>Infrared</i>
MODIS	: <i>Moderate-Resolution Imaging Spectroradiometer</i>
UAV	: <i>Unmanned Aerial Vehicle</i>
HOSD	: <i>Hyperspectral Oil Spill Dataset</i>
SVM	: <i>Support Vector Machine</i>
KCPA	: <i>Kernel Component Principal Analysis</i>
RGB	: <i>Red Green Blue</i>
FC	: <i>Fully Connected</i>
CNN	: <i>Convolutional Neural Network</i>
OA	: <i>Overall Accuracy</i>
AA	: <i>Average Accuracy</i>
AUC	: <i>Area Under the Curve</i>

BAB I PENDAHULUAN

1.1 Latar belakang

Insiden tumpahan minyak sering terjadi di seluruh penjuru dunia. Hal ini tidak lepas dari kegiatan eksplorasi dan transportasi minyak mentah. Akibat dari tumpahan minyak ini adalah rusaknya lingkungan laut dan spesies pesisir laut karena polusi yang parah (Alpers et al., 2017), (Gu et al., 2022), (Kang et al., 2022). Contoh, pada bulan April 2010, ledakan anjungan pengeboran minyak lepas pantai Deepwater Horizon menyebabkan jutaan barel minyak bocor ke area seluas 10.000 kilometer persegi di Teluk Meksiko (Li et al., 2020), (Duan et al., 2022). Akibatnya ikan dan burung laut yang hidup di area sekitar beserta ekosistem laut mengalami kerusakan. Jika tumpahan minyak tidak segera dipantau, minyak yang tumpah akan terbawa ombak ke pesisir pantai. Sumber daya perikanan dan budidaya pesisir serta kesehatan masyarakat sekitar akan terancam oleh insiden tersebut. Dengan demikian, sangat penting untuk mendeteksi tumpahan minyak di permukaan laut secara efektif, guna memantau penyebaran, dampak, dan volume tumpahan minyak (Bhargale et al., 2017).

Pemantauan penyebaran tumpahan minyak di area yang luas dan sulit dijangkau dapat dilakukan dengan menggunakan penginderaan jauh (Cheng et al., 2022). Data multitemporal dan model prediksi lintasan dapat memperkirakan kecepatan dan arah pergerakan tumpahan minyak, yang dimana sangat berperan penting dalam memudahkan proses pembersihan dan sistem peringatan (Karantzalos and Argialas, 2008). Dalam beberapa dekade lalu, para peneliti telah melakukan eksplorasi kemampuan penginderaan jauh untuk mendeteksi wilayah tumpahan minyak. Aplikasi pertama penginderaan jarak jauh untuk deteksi tumpahan minyak adalah didasarkan pada data *visible* (VIS) dan *infrared* (IR) di udara (Fingas and Brown, 2014). Namun, data jenis seperti ini memiliki beberapa kekurangan, seperti rendahnya kemampuan untuk membedakan tumpahan minyak dan objek di sekitarnya karena resolusinya yang rendah (Singha et al., 2013), (Cheng et al., 2022a).

Selain pendekatan berbasis cahaya tampak (*visible*) dan inframerah (*infrared*), banyak penelitian telah memeriksa kemungkinan menggunakan citra multispektral untuk mendeteksi tumpahan minyak. Contohnya adalah instrument *moderate resolution imaging spectroradiometer* (MODIS) telah menunjukkan keefektifannya dalam mendeteksi wilayah tumpahan minyak. Keefektifan ini disebabkan oleh resolusi spektral dan temporal yang tinggi (Lacava et al., 2017), (Satriano et al., 2019), (Liu et al., 2017). Selain data MODIS, data Landsat dengan resolusi spasial yang tinggi juga telah digunakan dalam mendeteksi tumpahan minyak dengan menggunakan informasi spektral dan spasial (Taravat and Del Frate, 2012), (Sun et al., 2015). Kebanyakan metode pendeteksi tumpahan minyak ini sangat mirip dengan proses segmentasi semantik di visi komputer, yang bergantung pada sampel yang telah memiliki label untuk mempelajari *supervised classifier*.

Namun, data MODIS untuk menghasilkan data multispektral dengan resolusi 250 m dan landsat dengan resolusi 30 m tidak mencukupi kebutuhan untuk mendeteksi tumpahan minyak pada wilayah yang lebih kecil dengan baik, karena resolusi spasial dari data citra multispektral lebih besar dari luasan wilayah tumpahan minyak. Di sisi lain, sensor hiperspektral dapat digunakan untuk memperoleh resolusi spektral dan spasial yang lebih kaya dari data multispektral. Sensor ini dapat dipasang di UAV atau pesawat terbang tanpa awak. Sensor tersebut dapat merekam ratusan channel spektral yang sempit dari panjang gelombang cahaya tampak hingga inframerah. Beberapa tahun belakangan, teknik seperti ini telah digunakan secara luas dalam berbagai aspek, seperti deteksi objek (Rao et al., 2022), (Gao et al., 2023), pengambilan komponen air (Duan et al., 2020b), dan pemetaan mineral (Sudharshan et al., 2020), (Duan et al., 2020a). Teknik deteksi tumpahan minyak berbasis *machine learning* dengan tipe data hiperspektral telah dikembangkan dan digunakan untuk menemukan spektral

kelas minyak dari lingkungan laut yang kompleks sehingga dapat sangat mengurangi alarm palsu dari wilayah tumpahan minyak (Kokaly et al., 2013), (Sun and Hu, 2019). Contohnya, pada penelitian berbeda (Liu and Han, 2017), metode pencocokan bentuk spektral telah dikembangkan untuk memperoleh deteksi tumpahan minyak. Pada penelitian lainnya (Zhu et al., 2019), metode berbasis *deep learning* digunakan untuk mendeteksi tumpahan minyak. Metode ini dapat mengurangi pengaruh dari bayangan dan noise pada citra, dan juga meningkatkan perbedaan antara tumpahan minyak dan air laut. Pendekatan - pendekatan seperti ini telah dikonfirmasi efektif dalam mengidentifikasi wilayah tumpahan minyak dengan jumlah training sampel yang banyak.

Namun, untuk membuat training sampel yang banyak dibutuhkan tenaga dan waktu yang banyak. Akibatnya beberapa model *deep learning* kurang cocok untuk diaplikasikan ke permasalahan di dalam dunia nyata dengan sampel yang sedikit. Contohnya pada permasalahan deteksi tumpahan minyak dengan menggunakan citra hiperspektral (Duan et al., 2023). Oleh karena itu, Diperlukan suatu metode *deep learning* yang dapat belajar dari data training dengan jumlah yang terbatas.

Penelitian ini berfokus pada bagaimana memetakan tumpahan minyak dengan menggunakan data *training* yang sangat terbatas (*Few-Shot Classification*) dengan mengimplementasikan metode *contrastive learning*. Diharapkan proses pemetaan tumpahan minyak dapat dilakukan dengan data *training* yang sangat terbatas dan mencapai performa yang baik. Hasil dari penelitian ini dapat menambah opsi dalam mengembangkan model klasifikasi tumpahan minyak dan berperan dalam membantu proses pembersihan tumpahan minyak, terutama dengan sampel yang terbatas.

1.2 Rumusan Permasalahan

Berdasarkan pemaparan latar belakang, rumusan masalah yang ingin diselesaikan adalah:

1. Bagaimana cara melakukan pemrosesan awal *input* sebelum dimasukkan ke proses *training* menggunakan pendekatan *contrastive learning*?
2. Bagaimana membangun model *deep learning* yang cocok untuk melakukan klasifikasi tumpahan minyak?
3. Bagaimana cara mengimplementasikan *contrastive learning* agar model klasifikasi tumpahan minyak dengan *deep learning* tidak membutuhkan data sampel yang banyak?

1.3 Batasan Masalah

Batasan masalah dalam pengembangan model klasifikasi ini adalah:

1. Dataset menggunakan dataset citra hiperspektral hasil penginderaan jauh (*Hyperspectral Oil Spill Dataset*, HOSD) tumpahan minyak, yang diperoleh dari repositori daring di <https://github.com/PuhongDuan/HOSD>.
2. Implementasi *deep learning* menggunakan bahasa pemrograman python.
3. Implementasi *deep learning* menggunakan library PyTorch.

1.4 Tujuan

Tujuan dari penelitian ini adalah:

1. Mengimplementasikan model *deep learning* untuk klasifikasi tumpahan minyak.
2. Mengimplementasikan metode *contrastive learning* agar proses *training* model *deep learning* tidak membutuhkan data sampel yang banyak.

1.5 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah:

1. Mampu menghasilkan metode pemetaan tumpahan minyak yang efektif dengan menggunakan data sampel yang terbatas.
2. Mampu membantu dalam proses pembersihan dan penyelesaian masalah tumpahan minyak.

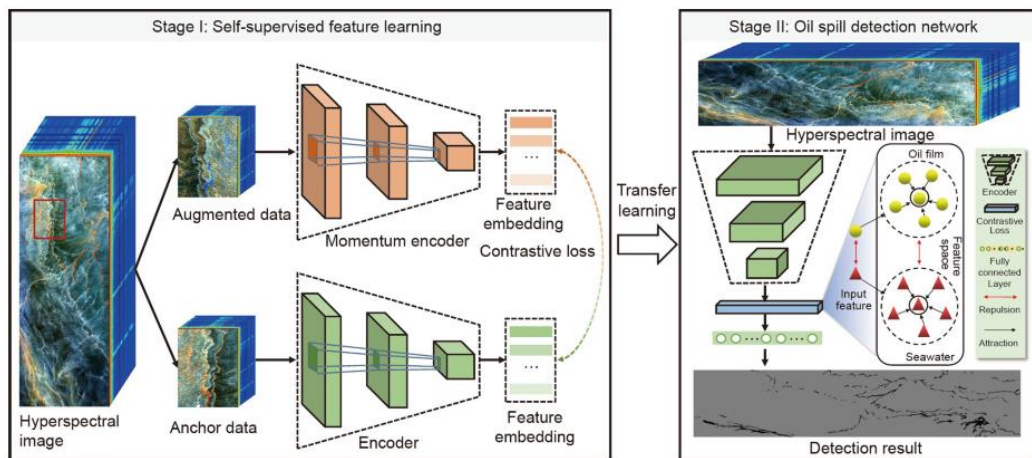
BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada subbab ini akan disebutkan dan dijelaskan secara singkat penelitian yang telah dilakukan oleh para peneliti terdahulu dan berkaitan dengan topik penelitian ini.

2.1.1 *Self-supervised learning-based oil spill detection of hyperspectral Images*

Penelitian yang berjudul “*Self-supervised learning-based oil spill detection of hyperspectral Images*” (Duan et al., 2022) ini mencoba untuk menggunakan metode *self-supervised learning* dalam membangun *deep neural network* dari data *training* berupa data hiperspektral tanpa label. Penelitian ini memiliki tiga langkah utama, yaitu *data augmentation*, *self-supervised feature learning*, dan *oil spill detection network*.



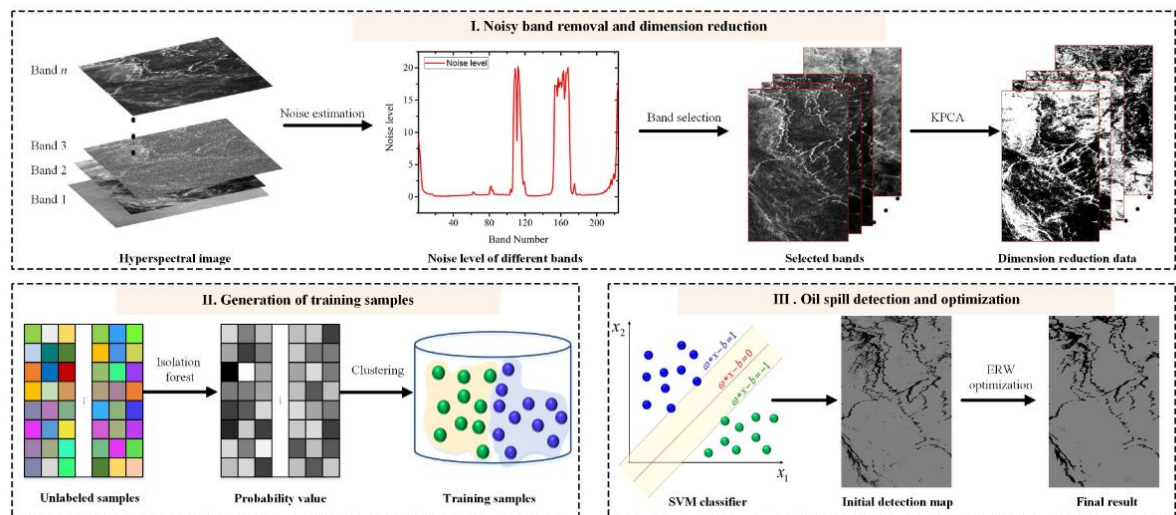
Gambar 2.1 Skema *Self-supervised learning model* yang ditawarkan penelitian 1 (Duan et al., 2022).

Berdasarkan karakteristik data hiperspektral yang berbentuk 3 dimensi, penelitian ini menggunakan augmentasi data berbasis transformasi spasial dan spektral. Teknik augmentasi data ini bertujuan untuk meningkatkan keberagaman data *training*. Beberapa contoh dari transformasinya adalah pencerminan spasial, rotasi spasial, pencerminan spektral, dan pemberian *noise* acak.

Penelitian ini menggunakan ResNet 50 untuk menghasilkan fitur pada input hiperspektral. Penelitian ini juga menggunakan *contrastive learning* untuk melakukan *training* pada *encoder*. Penelitian ini juga menggunakan *momentum bank mechanism* untuk menyelesaikan permasalahan optimasi *mini-batch*. Pada tahap berikutnya, *oil spill detection network* digunakan untuk melakukan *training* pada *CNN classifier* dengan menggunakan *pre-training deep embedding* yang ditransfer dari *self-supervised feature learning*. Hasil dari penelitian adalah metode yang ditawarkan pada penelitian memberikan performa yang baik dalam hal akurasi dengan mean AUC adalah 0,9014.

2.1.2 *Hyperspectral Remote Sensing Benchmark Database for Oil Spill Detection With an Isolation Forest-Guided Unsupervised Detector*

Penelitian terkait lainnya adalah penelitian berjudul “*Hyperspectral Remote Sensing Benchmark Database for Oil Spill Detection With an Isolation Forest-Guided Unsupervised Detector*” (Duan et al., 2023). Penelitian ini menggunakan pendekatan *unsupervised learning* menggunakan metode berbasis *isolation forest* (iForest).



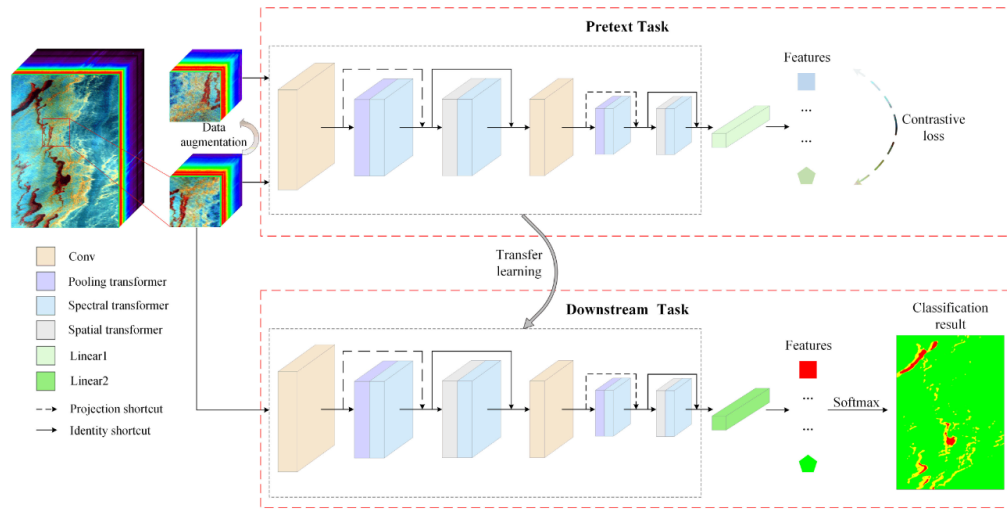
Gambar 2.2 Skema Unsupervised Oil Spill Detector yang ditawarkan pada penelitian 2 (Duan et al., 2023).

Tahap Pertama adalah penghapusan *noisy band* dan reduksi dimensi. Penelitian ini menggunakan metode penyeleksi *band* otomatis berbasis *gaussian statistic model* untuk mengeliminasi *bands* dengan varians *noise* yang tinggi. Setelah menghilangkan *bands* yang memiliki *noise* tinggi. Langkah selanjutnya adalah penggabungan dimensi spektral untuk mengurangi kompleksitas komputasi dan meningkatkan kapabilitas diskriminatif. Proses tersebut dilakukan menggunakan *Kernel Principal Component Analysis* atau KPCA. Tahap berikutnya adalah *generate* sampel *training*. Penelitian ini menggunakan skema otomatis untuk memproduksi sampel *training*. Penelitian ini menggunakan iForest untuk mendapatkan probabilitas tiap piksel. Tahap terakhir adalah deteksi tumpahan minyak dan optimasi. Penelitian menggunakan SVM dan LIBSVM untuk memproduksi peta deteksi awal. Penelitian ini juga memperkenalkan metode ERW untuk menghapus *noisy* piksel dan memastikan peta probabilitas yang telah dihaluskan dapat selaras dengan batas benda nyata.

Hasil dari penelitian ini adalah metode yang ditawarkan memiliki performa yang cukup bagus dibandingkan pendekatan dan metode lainnya dengan mean AUC adalah 0,9006. Selain itu bentuk dari wilayah tumpahan minyak juga selaras dengan peta referensi.

2.1.3 Self-Supervised Spectral–Spatial Transformer Network for Hyperspectral Oil Spill Mapping

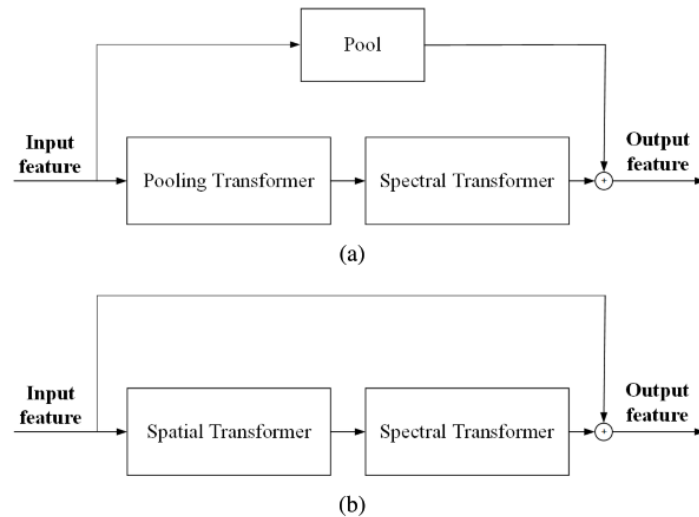
Penelitian lainnya yang terkait adalah penelitian berjudul “*Self-Supervised Spectral–Spatial Transformer Network for Hyperspectral Oil Spill Mapping*” (Kang et al., 2023). Penelitian ini menawarkan pendekatan menggunakan *transformer*, yaitu *Self-Supervised spectral-spatial transformer network* atau SSTNet untuk pemetaan hiperspektral tumpahan minyak.



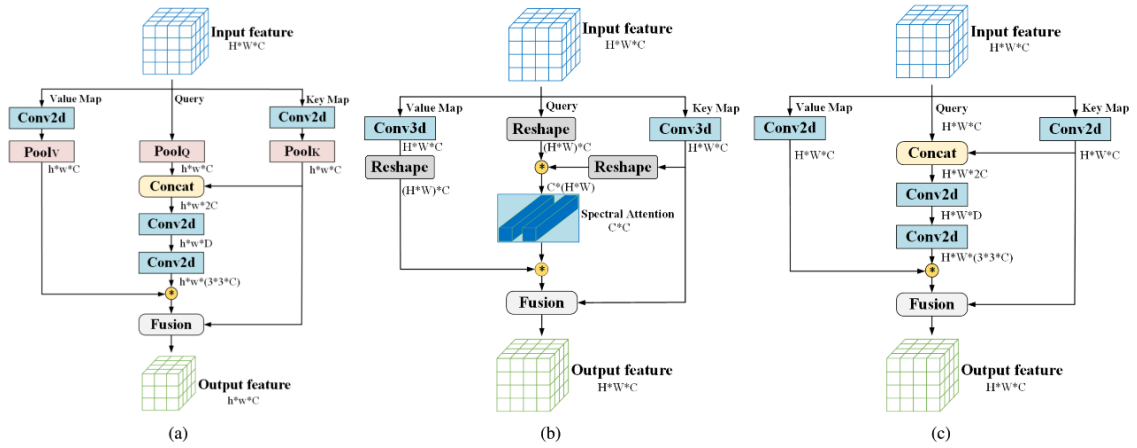
Gambar 2.3 Skema Self-Supervised SSTNet untuk oil spill detection yang ditawarkan oleh penelitian 3 (Kang et al., 2023).

Pertama, data diaugmentasi untuk menambah keberagaman data *training* yang terdiri dari penambahan *noise* pada citra, pencerminan spasial, dan rotasi spektral. Kemudian *pretext task* bertujuan untuk mempelajari *deep embedding features* dari citra asli dan citra yang telah diaugmentasi tanpa menggunakan label. Untuk mendapatkan *deep features* dari sampel tanpa label, strategi pembelajarannya adalah fitur yang mirip akan menghasilkan gap yang lebih kecil, dan fitur yang berbeda akan menghasilkan gap yang lebih jauh. Strategi pembelajaran yang digunakan adalah strategi berbasis *contrastive learning*. Penelitian ini juga menggunakan *momentum bank mechanism* untuk menyelesaikan permasalahan optimasi *mini-batch* yang ditemui.

Setelah itu parameter yang dipelajari dari *pretext task* ditransfer ke *downstream classification network*. Untuk mengekstrak *discriminative features*, sebuah SSTNet dirancang yang utamanya terdiri dari 2 modul kunci, yaitu *Spectral-spatial pooling module* (SSPM) dan *Spectral-Spatial Residual Module* (SSRM)



Gambar 2.4 Struktur 2 modul spektral-spasial (a) SSPM. (b) SSRM
source: (Kang et al., 2023).



Gambar 2.5 Arsitektur 3 transformer (a) Pooling Transformer. (b) Spektral Transformer. (c) Spasial Transformer (Kang et al., 2023).

Hasil dari eksperimen menunjukkan bahwa metode yang ditawarkan oleh penelitian ini lebih superior dibandingkan metode lainnya dalam hasil yang objektif (OA, AA, Kappa, F1) dan subjektif, meskipun dengan training sampel yang terbatas. Metode yang ditawarkan memiliki OA sebesar 95,96, AA 91,89, Kappa 80,48, dan F1 score 90,55.

2.1.4 Perbandingan Penelitian Terkait

Tabel di bawah berisikan perbandingan dari ketiga penelitian terkait yang telah disebutkan sebelumnya.

Tabel 2.1 Perbandingan dari ketiga penelitian terkait.

Penelitian	dataset	setting data training	testing	akurasi
1	HOSD (GM1-GM10)	<ul style="list-style-type: none"> <i>spatial window</i> = 11 <i>batch size</i> = 256 tahap <i>self-supervised feature learning</i> <ul style="list-style-type: none"> <i>maximum training epoch</i> = 200 <i>learning rate</i> = 0,03, dikali 0,1 ketika <i>epoch</i> mencapai 120 dan 160 tahap <i>oil spill detection</i> <ul style="list-style-type: none"> <i>maximum training epoch</i> = 100 <i>learning rate</i> = 0,00001, dikali 0,1 ketika mencapai <i>epoch</i> 60 dan 80 sampel <i>training</i> 10% sampel dengan label dari GM1 dan GM2 	Mengukur AUC	0,9014
2	HOSD (GM1-GM18)	<ul style="list-style-type: none"> <i>Isolation trees</i> = 800 <i>Fused data</i> = 25 Sampel <i>training</i> berjumlah 1% dari <i>automatic generated sample</i> 	Mengukur AUC dan DP	0,9006
3	HOSD (HOSD1-HOSD8)	<ul style="list-style-type: none"> SGD <i>optimizer</i> dengan <i>minibatch</i> = 256 <i>Spatial size</i> tiap cube = 11 Pretext task : <ul style="list-style-type: none"> <i>Learning rate</i> awal = 0,01, dikali 0,1 ketika <i>epoch</i> = 120 dan 200 	Mengukur OA, AA, Kappa, F1 score	Overall Accuracy (OA) adalah 95,96%. Average Accuracy

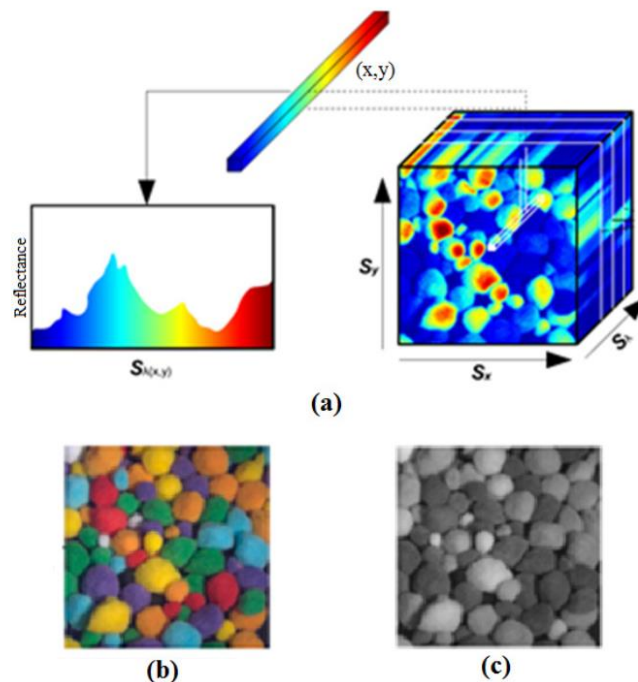
		<ul style="list-style-type: none"> • Total jumlah <i>epoch</i> = 280 <p><i>Classification task :</i></p> <ul style="list-style-type: none"> • <i>Learning rate</i> = 0,03, dikali 0,1 ketika <i>epoch</i> =30 • Jumlah <i>epoch</i> = 50 • Training data = 5% dari sampel <i>training</i> HOSD1 – HOSD2 		(AA) = 91,89%
--	--	---	--	---------------

2.2 Dasar Teori

Pada subbab ini akan dijelaskan teori – teori yang relevan dan berhubungan dengan topik penelitian ini

2.2.1 Citra Hiperspektral

Pencitraan hiperspektral atau pencitraan spektrometri adalah teknik penginderaan spektral di mana suatu objek difoto menggunakan menggunakan beberapa pita optik tertentu dengan rentang spektral yang luas. Pada dua puluh tahun terakhir, pencitraan hiperspektral telah diaplikasikan ke berbagai bidang yang diantaranya adalah pertahanan militer, diagnosis medis (Carrasco et al., 2003), (Afromowitz et al., 1988), arkeologi dan konservasi karya seni (Fischer and Kakoulli, 2006), (Liang, 2012), pencitraan dokumen (Padoan et al., 2008), forensik obat-obatan (Edelman et al., 2012), penjaminan mutu makanan (Feng and Sun, 2012), dan pemetaan mineral di permukaan bumi (Clark and Swayze, 1995).



Gambar 2.6 (a) Citra hiperspektral direpresentasikan sebagai kubus 3D. Suatu titik spektrum pada kubus spektral direpresentasikan pada lokasi spasial. (b) Citra RGB. (c) Citra grayscale yang di-render dari kubus hiperspektral (Khan et al., 2018).

Hasil dari pencitraan hiperspektral adalah citra hiperspektral. Citra hiperspektral memiliki resolusi spasial dan spektral. Resolusi spasial mengukur hubungan geometris antara piksel - piksel pada citra satu sama lain. Sedangkan, resolusi spektral menentukan variasi di dalam piksel – piksel pada citra sebagai suatu fungsi panjang gelombang. Suatu citra hiperspektral memiliki dua dimensi spasial (S_x dan S_y) dan satu dimensi spectral (S_λ). Data hiperspektral

direpresentasikan sebagai kubus data hiperspektral 3D seperti pada gambar. Gambar juga menunjukkan suatu titik spektrum pada kubus pada Lokasi spasial (x, y) dan suatu citra RGB dan *grayscale* di-render dari kubus hiperspektral. Tiap potongan dari kubus sepanjang dimensi spektral disebut *band* atau *channel*.

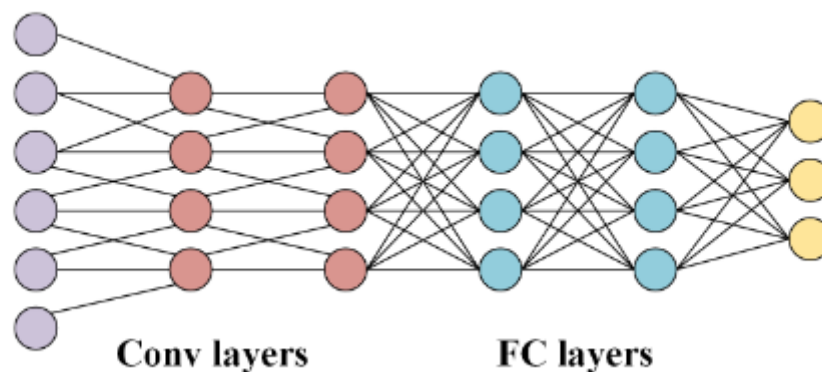
Resolusi spasial adalah detail terkecil yang masih dapat terlihat pada suatu citra. Resolusi ini menggambarkan ukuran objek terkecil yang dapat diamati pada citra tersebut. Objek tersebut juga harus dapat dibedakan sebagai entitas yang terpisah dari objek lain di dalam citra. Dalam situasi praktis, kejernihan citra ditentukan oleh resolusi spasialnya, bukan jumlah piksel pada citra. Karakteristik spasial citra bergantung pada rancangan dari sensor pencitraan dari segi *field of view* dan ketinggian sensor. Sebidang tanah yang dengan luasan tertentu ditangkap oleh tiap detektor di dalam suatu sensor penginderaan jauh. Resolusi spasial berbanding terbalik dengan ukuran luasan bidang tanah yang ditangkap. Semakin kecil ukuran bidang tanah, maka semakin tinggi detail yang bisa diinterpretasikan dari pemandangan yang diobservasi.

Resolusi spektral adalah jumlah *band* spektral dan jangkauan spektrum elektromagnetik yang diukur oleh sensor. Suatu sensor pencitraan dapat merespon ke jangkauan frekuensi yang banyak tetapi memiliki resolusi spektral yang rendah jika sensor tersebut hanya mengambil *band* spektral dalam jumlah kecil. Sebaliknya, sensor yang sensitif terhadap jangkauan frekuensi yang kecil tetapi dapat menangkap spektral *band* dalam jumlah banyak akan memilih resolusi spektral yang tinggi. Sensor hiperspektral mampu mendapatkan citra dengan spektral *band* yang berdekatan dan sangat sempit dalam spektrum *mid infrared*, *near infrared*, dan segmen tampak.

Selain resolusi spasial dan spektral, citra hiperspektral juga memiliki resolusi temporal. Resolusi temporal bergantung pada karakteristik orbit dari sensor. Secara umum resolusi temporal didefinisikan sebagai waktu yang dibutuhkan oleh sensor untuk mengunjungi kembali dan mendapatkan data dari lokasi yang sama (Khan et al., 2018).

2.2.2 Convolutional Neural Network

Convolutional Neural Network atau CNN adalah salah satu jenis dari *feedforward neural network* yang mampu mengekstrak fitur dari data dengan struktur konvolusi. Berbeda dengan metode ekstraksi fitur tradisional lainnya, CNN tidak perlu mengekstrak fitur secara manual. CNN memiliki arsitektur yang terinspirasi oleh persepsi visual. Suatu neuron biologis berhubungan dengan neuron artifisial. *Kernel* dari CNN merepresentasikan reseptor berbeda yang dapat merespons ke beberapa fitur. Fungsi aktivasi mensimulasi fungsi dimana hanya sinyal neural elektrik yang melebihi suatu ambang batas yang dapat ditransmisikan ke neuron berikutnya. Fungsi *loss* dan *optimizer* adalah fungsi yang diciptakan peneliti untuk mengajari seluruh sistem CNN untuk mempelajari apa yang harus diekspektasikan.



Gambar 2.7 Diagram CNN layer dan FC layer (Li et al., 2022).

Dibandingkan dengan *fully controlled* (FC) *network*, CNN memiliki beberapa kelebihan. Salah satu kelebihannya adalah koneksi yang lokal, dimana tiap neuron tidak lagi terhubung dengan semua neuron di *layer* sebelumnya, tetapi hanya beberapa terhubung dengan sejumlah kecil neuron, yang dapat mengurangi parameter dan mempercepat konvergensi. Kelebihan lainnya adalah adanya *weight sharing*, dimana sekelompok koneksi dapat berbagi *weight* yang sama, sehingga dapat mengurangi parameter. Kelebihan terakhir adalah pengurangan dimensi *downsampling*, dimana suatu *pooling layer* memanfaatkan prinsip korelasi lokal untuk melakukan *downsample* pada suatu citra, sehingga dapat mengurangi jumlah dan tetap mempertahankan informasi yang berguna (Li et al., 2022).

Secara umum, arsitektur dari CNN terdiri dari beberapa tahap. Beberapa tahap pertama terdiri dari 2 jenis *layer*, yaitu *convolutional layer* dan *pooling layer*. Unit-unit pada *convolutional layer* tersusun dalam *feature map*. Di dalamnya, tiap unit terhubung ke beberapa bidang lokal di *feature map* dari *layer* sebelumnya melalui sekumpulan *weight* yang disebut *filter bank*. Hasil dari *local weighted sum* kemudian diteruskan melalui suatu *nonlinearity* seperti ReLU. Semua unit di *features map* berbagi *filter bank* yang sama, *feature map* yang berbeda di suatu *layer* menggunakan *filter bank* yang berbeda. Terdapat dua alasan dari arsitektur ini. Pertama, dalam *array* data seperti citra, sekelompok nilai lokal sering memiliki korelasi yang tinggi, membentuk motif lokal yang unik yang dapat dengan mudah dideteksi. Kedua, statistik lokal dari citra dan sinyal lainnya adalah *invariant* ke letaknya. Dengan kata lain, jika suatu motif dapat muncul di suatu bagian dari citra, motif tersebut dapat muncul di mana saja, oleh karena itu terdapat gagasan tentang unit-unit di lokasi berbeda yang berbagi *weight* yang sama dan mendeteksi pola yang sama di bagian *array* yang berbeda. Secara matematis, operasi *filtering* yang dilakukan oleh *feature maps* adalah konvolusi diskrit, oleh karena itu CNN dinamakan demikian.

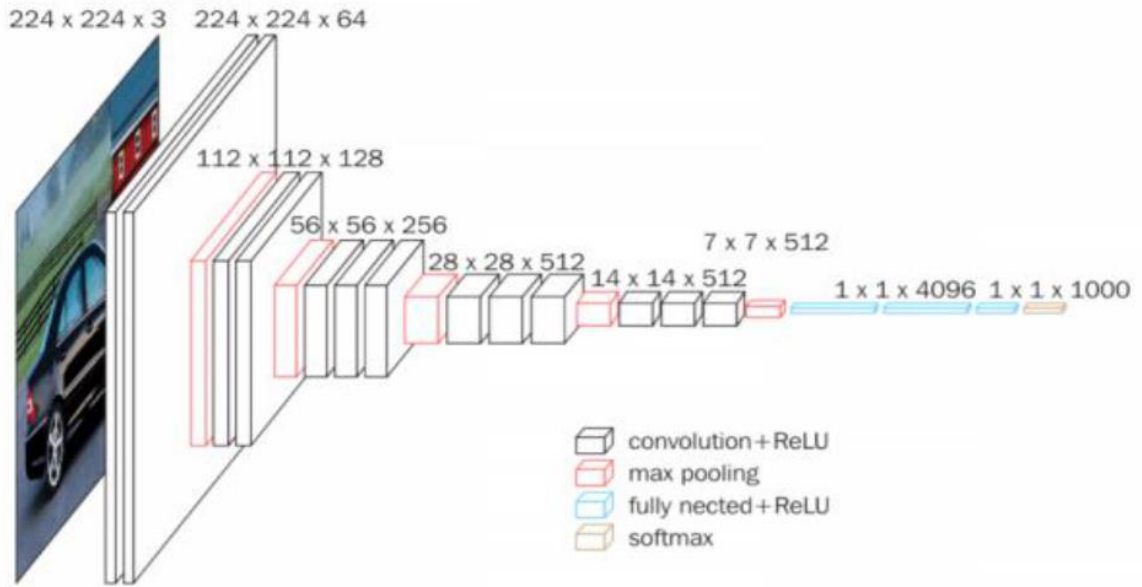
Selain *convolutional layer*, CNN juga memiliki *pooling layer*. Tugas dari *pooling layer* adalah menggabungkan fitur yang mirip menjadi satu secara semantik. Karena posisi relatif dari fitur – fitur yang membentuk motif dapat bervariasi, proses deteksi motif bisa dilakukan dengan *coarse-graining* posisi tiap fitur. Secara umum, unit *pooling* menghitung maksimum dari bidang unit lokal dalam satu *features map*. Unit *pooling* yang bersebelahan mengambil input dari bidang – bidang yang bergeser sebanyak satu baris atau kolom, sehingga dapat mengurangi dimensi dari representasi dan membuat suatu invarian ke pergeseran dan distorsi kecil. Dua atau tiga tahapan dari konvolusi, *non-linearity*, dan *pooling* ditumpuk, diikuti dengan lebih banyak konvolusi dan FC *layer*. *Backpropagation* yang melalui suatu CNN adalah sederhana dengan melalui deep network biasanya, sehingga memungkinkan semua *weights* dalam semua *filter banks* dapat dilatih (Lecun et al., 2015).

2.2.3 VGG-16

VGG-16 adalah sebuah model CNN yang dikembangkan oleh Visual Geometry Group (VGG). VGG-16 pertama kali diperkenalkan oleh K. Simonyan dan A. Zisserman dari University of Oxford pada tahun 2014 (Simonyan and Zisserman, 2014). Tujuan utama dari arsitektur adalah untuk merancang suatu model dengan mempertimbangkan kedalaman lapisan tanpa menambah kompleksitas dari *network*. Arsitektur VGG-16 memiliki kemampuan memproduksi 38.357.544 parameter (Bagaskara and Suryanegara, 2021).

VGG-16 memiliki 16 lapisan konvolusi. VGG-16 menerima *input* berupa citra dengan dimensi 224 x 224 x 3. *Network* ini juga memiliki 16 lapisan konvolusi dengan filter yang memiliki ukuran tetap sebesar 3 x 3. *Network* ini juga memiliki 5 lapisan *Max Grouping* dengan ukuran 2 x 2 pada seluruh *network*. Sebaliknya, pada bagian atas, kedua lapisan tersebut terhubung seluruhnya dengan lapisan *softmax output*. VGG-16 dianggap sebagai *network* berukuran besar dengan parameter berjumlah sekitar 138 juta. Model ini menyusun banyak lapisan konvolusi untuk membuat *deep neural network* yang dapat meningkatkan kemampuan

untuk mempelajari fitur tersembunyi (Kamal and EZ-ZAHRAOUI, 2023).

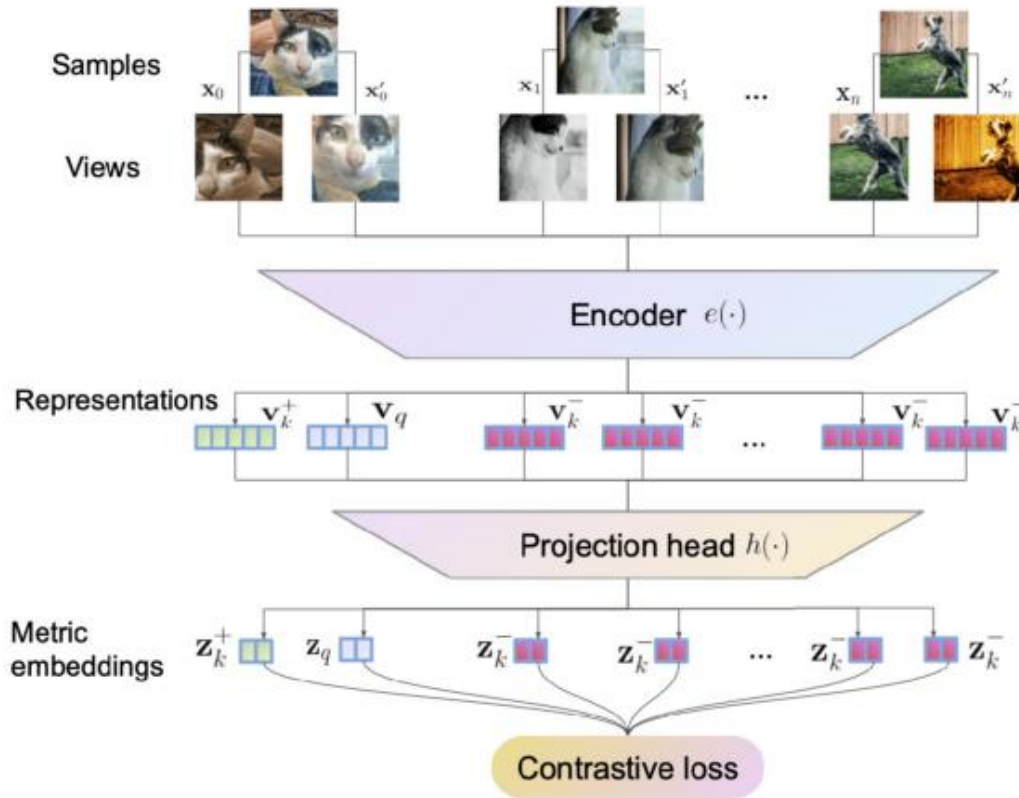


Gambar 2.8 Arsitektur Model VGG-16 (Kamal and EZ-ZAHRAOUI, 2023).

2.2.4 Contrastive Learning

Contrastive learning mempelajari suatu representasi dengan memaksimalkan tingkat kemiripan dan perbedaan antara data sampel yang disusun menjadi pasangan-pasangan yang mirip dan berbeda (Xiao et al., 2020). Ide utama dari *contrastive learning* adalah untuk membedakan pasangan titik data yang mirip (positif) dan berbeda (negatif) secara semantik, yang mendorong representasi f pasangan mirip (x, x^+) menjadi dekat, dan representasi pasangan berbeda (x, x^-) menjadi lebih ortogonal (Chuang et al., 2020).

Alih - alih mendapatkan pseudo-label dari *pretext task*, metode *Contrastive learning* mempelajari model diskriminatif pada beberapa pasangan input, berdasarkan konsep kemiripan tertentu. Mirip dengan *self-supervised pretext task* lainnya, definisi kemiripan ini dapat ditentukan langsung dari data itu sendiri, sehingga dapat mengatasi keterbatasan yang ditemui pada *supervised learning* di mana hanya tersedia beberapa pasang label yang terbatas dari data. Sementara beberapa metode *self-supervised* memerlukan modifikasi arsitektur model selama proses *learning*, metode *contrastive learning* jauh lebih sederhana karena tidak memerlukan modifikasi arsitektur model antara *training* dan *fine-tuning* ke *task* lainnya



Gambar 2.9 Contrastive Learning (Le-Khac et al., 2020).

Gambar diatas adalah ilustrasi dari *Instance discrimination task* sebagai bentuk sederhana dari *contrastive learning*. Suatu pasangan positif dibuat dari dua *augmented views* secara acak dari citra yang sama. Sedangkan pasangan negatif dibuat dari *views* dari citra yang berbeda. Semua *views* telah ter-encode oleh *encoder* dan *projection head* yang sama, sebelum representasi dievaluasi oleh fungsi *contrastive loss*.

$$L_i = -\log \frac{\exp(z^t z' / \tau)}{\sum_{j=0}^K \exp(z_i \cdot z'_i) / \tau} \quad (1)$$

Di atas adalah contoh dari fungsi *contrastive loss*. Di mana $z^t z'$ adalah *dot product* antara dua vektor dan τ adalah *temperature* dari hiperparameter yang mengontrol sensitivitas dari *dot product*. Operator *sum* dihitung pada satu pasangan positif dan K negatif pada *minibatch* yang sama. Secara intuitif, operasi ini dapat dipahami sebagai versi nonparametrik dari (K + 1)-way klasifikasi *softmax* dari z_i ke z'_i yang sesuai (Le-Khac et al., 2020).

2.2.5 Klasifikasi Few-Shot

Klasifikasi *Few-Shot* adalah salah satu metode klasifikasi dengan menggunakan beberapa sampel berlabel. Pada umumnya jumlah sampel yang digunakan pada *Few-Shot Classification* berjumlah 1 sampai 5 sampel. *Few-Shot Classification* bertujuan untuk menyelesaikan permasalahan pada *deep learning*, yaitu banyaknya data yang perlu dikumpulkan dan diberi label. Proses pengumpulan dan pemberian label membutuhkan sumber daya yang tinggi dan terkadang sulit untuk dilakukan.

Tantangan terbesar dalam *Few-Shot classification* adalah risiko tinggi adanya *overfitting*. Beberapa peneliti telah merumuskan beberapa pendekatan untuk menyelesaikan masalah

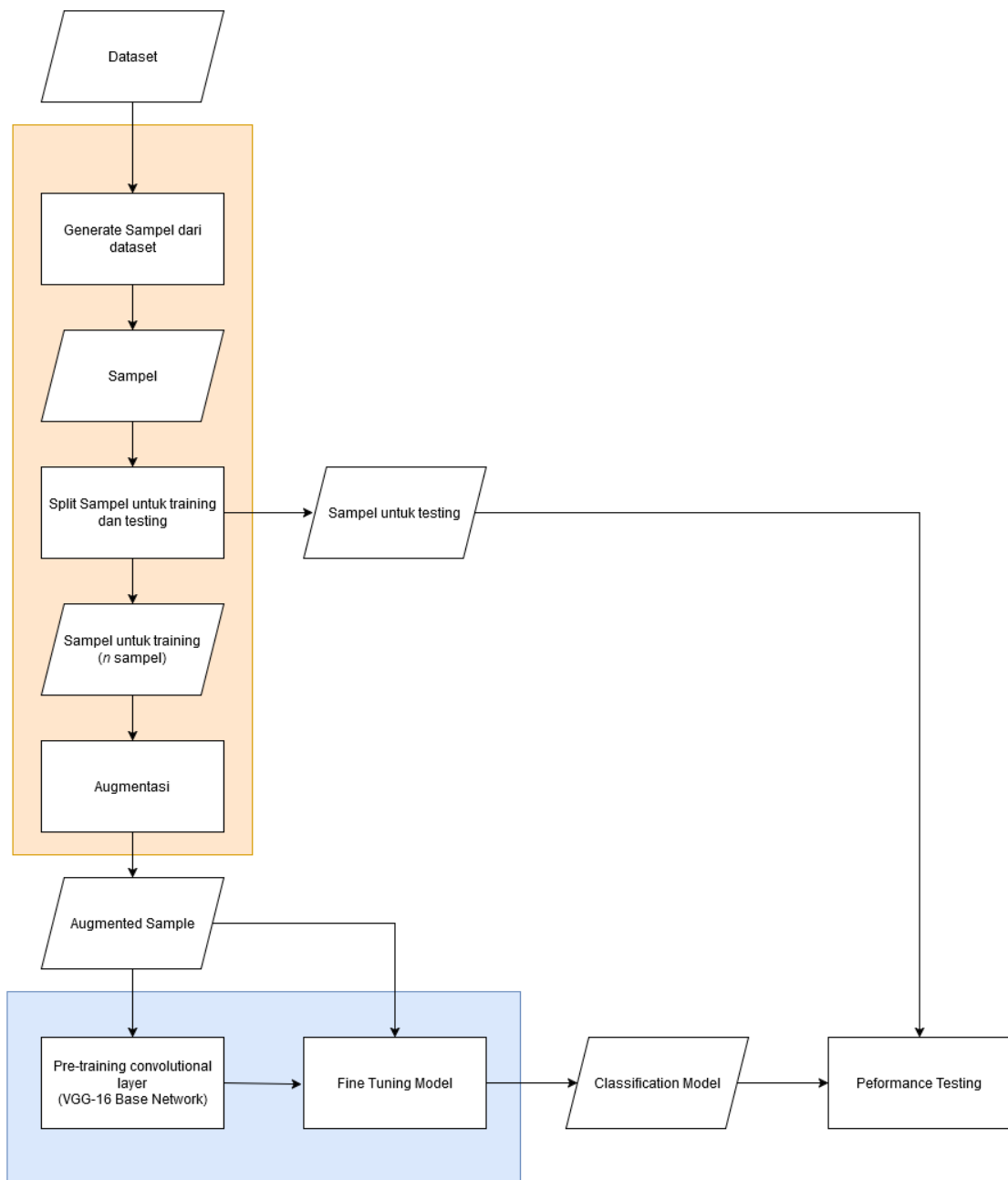
overfitting. Beberapa pendekatan di antaranya adalah metode *meta-learning*, metode *transfer learning*, dan metode *metric learning*. Pada metode *meta-learning*, suatu *meta-learner* akan di-*training* pada banyak *classification task* untuk mendapatkan *generalizable knowledge*, sehingga dapat memungkinkan *training* dengan cepat pada *task* yang berkaitan dengan sampel yang lebih sedikit.

Pada metode *transfer learning*, domain sumber dan target akan dianggap memiliki *shared knowledge*. Pada metode ini, model yang telah dilatih menggunakan sumber data yang berlimpah akan di-*fine-tune* agar sesuai dengan beberapa target sampel berlabel. Selain *meta-learning* dan *transfer learning* peneliti juga telah mengembangkan pendekatan *metric learning*. Pada *metric learning*, *feature embedding* dan atau ukuran jarak akan dipelajari dan sebuah *unseen sample* akan diklasifikasikan berdasarkan jarak antara sampel tersebut dengan *labelled sample* atau representasi dari kelas. Sampel dengan kelas yang sama diharapkan memiliki lokasi yang berdekatan pada *embedding space* dan sampel dengan kelas yang berbeda diharapkan memiliki lokasi yang berjauhan (Li et al., 2023).

BAB III METODOLOGI

Pada bab metodologi, langkah – langkah dan tahapan dari metode penelitian akan dijelaskan secara rinci. Bab ini berisi rancangan dan skema model *deep learning* yang akan dikembangkan beserta algoritma - algoritma pendukung. Bab ini juga berisi peralatan pendukung, seperti *software* dan *hardware*, yang akan digunakan dalam pengerjaan penelitian. Bab ini juga berisi *pseudocode* dari proses pengembangan model *deep learning*. Pada akhir bab juga akan dijelaskan rencana uji coba.

3.1 Metode yang dirancang



Gambar 3.1 Skema Metode yang Dirancang.

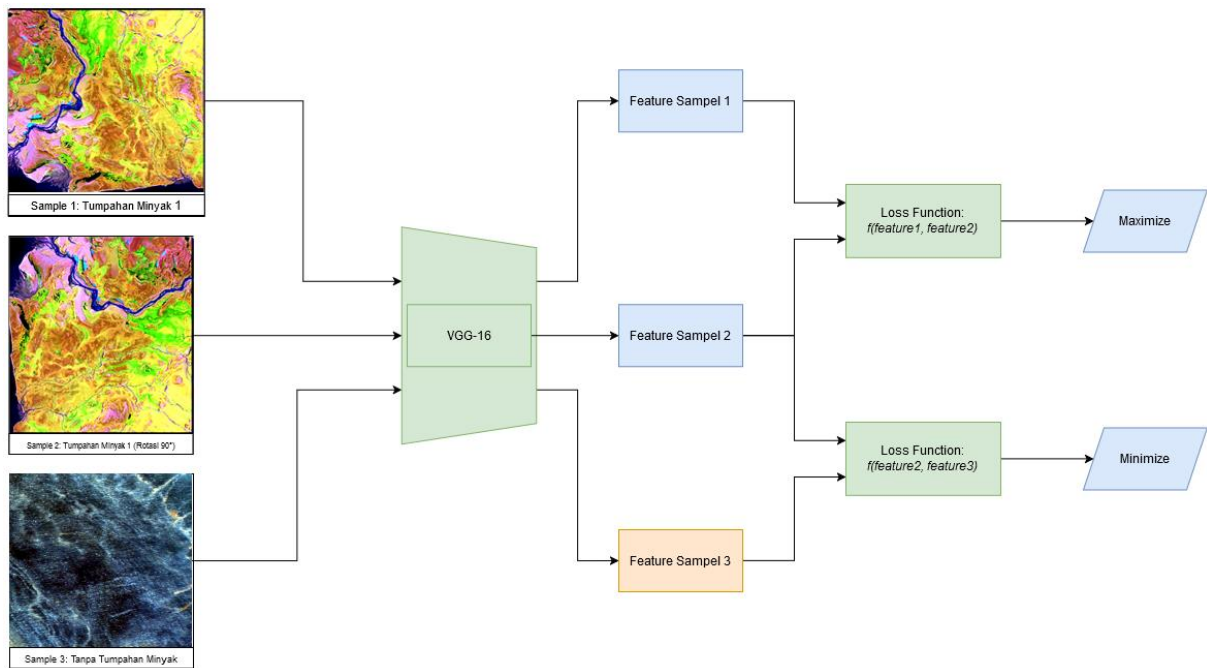
Secara umum, metode penelitian dapat dibagi menjadi beberapa tahap, yaitu *pre-processing*, *pre-training*, *fine-tuning*, dan *testing*. Tahap *pre-processing* bertujuan untuk mendapatkan sampel untuk *training* dan *testing* dari dataset. Tahap *training* berisikan proses pembuatan dan *training* model yang terdiri dari *training* model *deep learning* dan *classification layer*. Tahap *testing* bertujuan untuk menguji performa dari model yang telah dibuat dan di-*training* menggunakan sampel *testing*.

3.1.1 Pre-Processing

Pada tahap *pre-processing*, sampel akan dibuat dari dataset. Sampel merupakan segmen dari citra hiperspektral dengan ukuran $p \times p \times s$, dimana p adalah ukuran *spatial patch* dari sampel dan s adalah ukuran spektral *band* yang mencakup seluruh spektral *band* pada piksel. Sampel yang telah dibuat kemudian dipisah untuk proses *training* dan *testing*.

Sampel untuk *training* yang berjumlah n akan melalui proses augmentasi dimana transformasi spasial dan spektral akan dilakukan pada sampel sehingga akan menambah jumlah sampel. Beberapa transformasi sejumlah t akan dilakukan pada sampel. Transformasi yang akan dilakukan diantaranya adalah pencerminan spasial, pencerminan spektral, rotasi spasial, dan kombinasi dari beberapa transformasi. Dari hasil augmentasi akan dihasilkan sampel sejumlah $(t + 1) \times n$.

3.1.2 Pre-Training Model



Gambar 3.2 Skema Pre-Training Model.

Tahap berikutnya adalah tahap *pre-training*. Pada tahap *pre-training*, model *deep learning* akan di-*training* menggunakan strategi *contrastive learning*. Pada tahap ini, sampel yang telah melalui proses augmentasi akan dipasangkan satu sama lain sehingga akan terdapat jumlah pasangan sampel sebanyak $((t + 1) \times n)C2$. Pada tahap ini juga akan dibuat model *deep learning* dengan menggunakan arsitektur VGG-16.

Setiap pasang sampel kemudian akan dimasukkan ke dalam model *deep learning* untuk mendapatkan *feature* dari masing – masing sampel. Kemudian akan dihitung *loss* dari kedua *feature* menggunakan *loss function*. Berikut adalah persamaan *loss function* yang akan

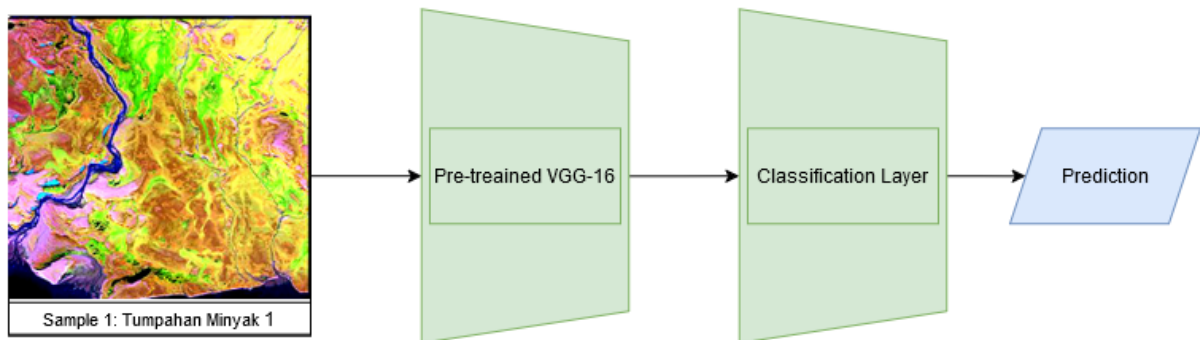
digunakan.

$$L_i = -\log \frac{\exp (\langle f_i^m, f_i^n \rangle / \tau)}{\sum_{j=1}^M \exp (\langle f_i^m, f_j^n \rangle / \tau)} \quad (2)$$

Dimana $\langle f_i^m, f_i^n \rangle$ adalah operasi untuk menghitung kemiripan antara sampel pada pasangan *feature* dan τ adalah parameter *temperature*. Parameter dari *deep network* akan di-*update* berdasarkan nilai *loss* (Duan et al., 2022).

Hasil dari tahap *pre-training* adalah untuk mendapatkan *deep network* yang telah di-*training* dan mampu memberikan *feature* yang sesuai dengan label kelas sampel. *Feature* dari kelas yang sama akan memiliki nilai yang mirip. Sedangkan, *feature* dari kelas yang berbeda akan memiliki nilai yang berjauhan.

3.1.3 Fine-Tuning Model



Gambar 3.3 Skema Fine-Tuning Model.

Pada tahap ini, *classification layer* akan ditambahkan ke dalam model dan akan di-*training* menggunakan sampel *training* yang telah diaugmentasi. Tujuan dari proses *fine-tuning* ini adalah menambahkan dan men-*training classification layer* agar dapat mengklasifikasi sampel dan memberikan prediksi label berupa ada atau tidaknya tumpahan minyak pada sampel.

3.1.4 Testing

Pada tahap *testing*, performa dari model akan diukur dengan memprediksi label dari sampel yang telah disiapkan untuk *testing* dan membandingkannya dengan label yang asli. Metrik yang akan diukur dalam proses *testing* adalah *overall accuracy*, *average accuracy*, *AUC*, dan *F1 score*.

3.2 Peralatan pendukung

Berikut adalah peralatan pendukung meliputi *software* dan *hardware* yang digunakan dalam pengerjaan penelitian ini.

3.2.1 Kebutuhan Perangkat Keras

Berikut adalah tabel kebutuhan perangkat keras yang digunakan selama proses penelitian. Perangkat keras yang digunakan adalah fasilitas komputer yang disediakan di laboratorium GIGA.

Tabel 3.1 Kebutuhan Perangkat Keras.

Perangkat Keras	Spesifikasi
Sistem Operasi	Windows
Processor	Intel Core i9-12900K
GPU	RTX 3080Ti 12GB
Memory (RAM)	64GB DDR5
Storage (SSD)	1 TB

3.2.2 Kebutuhan Perangkat Lunak

Berikut adalah tabel kebutuhan perangkat lunak yang digunakan selama proses penelitian.

Tabel 3.2 Kebutuhan Perangkat Lunak.

Aspek	Spesifikasi
Bahasa	Python
Software	VS Code
Library	PyTorch

3.3 Rencana Implementasi dan Uji Coba

Pada subbab – subbab berikutnya akan dijelaskan rencana implementasi setiap tahapan dalam bentuk *pseudocode*.

3.3.1 Implementasi Pre-Processing

Berikut adalah *Pseudocode* untuk tahap *pre-processing*.

Function PreProcess

Input:

size N of sample

list T of transformation

dataset dataset of hyperspectral image

Steps:

```

1:  function PreProcess( $N$ ,  $T$ , dataset)
2:      samples  $\leftarrow$  generate initial sample from dataset
3:      trainingSamples  $\leftarrow$  pick  $N$  random samples
4:      testingSamples  $\leftarrow$  samples – trainingSamples
5:      augmentedTrainingSamples  $\leftarrow$  []
6:      augmentedTrainingSamples.append(trainingSamples)
7:      for every sample in trainingSamples
8:          augmented_sample  $\leftarrow$  augmentSample(sample,  $T$ )
9:          augmentedTrainingSamples.append(augmented_sample)
10:     end for
11:     return testingSamples, augmentedTrainingSamples

```

Output:

testingSamples, augmentedTrainingSamples

Pseudocode 3.1 Pre-processing.

3.3.2 Implementasi Pre-Training

Berikut adalah *Pseudocode* untuk tahap *pre-training*.

Function preTraining

Input:

list augmentedTrainingSamples of augmented training samples

number epoch of epoch

Steps:

```

1:  function preTraining(augmentedTrainingSamples, epoch)
2:      samplePairs  $\leftarrow$  []
3:      numberOfSample  $\leftarrow$  len(augmentedTrainingSamples)
4:      for  $i \leftarrow 0$  to numberOfSample - 1 do
5:          for  $j \leftarrow i + 1$  to numberOfSample do
6:              samplePairs.append((augmentedTrainingSamples [ $i$ ],
augmentedTrainingSamples [ $j$ ]))
7:          end for
8:      end for
9:      deepNetworkModel  $\leftarrow$  create deep network model with architecture of VGG-16
10:     for  $i \leftarrow 0$  to epoch do

```

```

11:   for every samplePair samplePairs
12:     sample1, sample2  $\leftarrow$  samplePair
13:     feature1  $\leftarrow$  pass sample1 to deepNetworkModel
14:     feature2  $\leftarrow$  pass sample2 to deepNetworkModel
15:     loss  $\leftarrow$  calculate the loss of feature 1 and feature 2 using loss function
16:     deepNetworkModel  $\leftarrow$  update the parameter of deepNetworkModel based on loss
17:   end for
18: end for
19:   return deepNetworkModel

```

Output:

deepNetworkModel

Pseudocode 3.2 Pre-training.

3.3.3 Implementasi Fine-Tuning

Berikut adalah *Pseudocode* untuk tahap *fine-tuning*.

Function finetune

Input:

layers deepNetworkModel of deep network model

number epoch of epoch

list augmentedTrainingSamples of augmented training samples

Steps:

```

1:   function finetune(deepNetworkModel, epoch, augmentedTrainingSamples)
2:     model  $\leftarrow$  deepNetworkModel
3:     model  $\leftarrow$  add classification layer as the final layer
5:     for epoch  $\leftarrow$  0 to epoch do
6:       for every sample in augmentedTrainingSamples
7:         output  $\leftarrow$  pass sample to the model
8:         loss  $\leftarrow$  calculate loss based on the output and the label of the sample
9:         model  $\leftarrow$  update the parameter of the model based on loss
10:      end for
11:    end for
12:    return model

```

Output:

model

Pseudocode 3.3 Fine-tuning.

3.3.4 Implementasi Testing

Berikut adalah *Pseudocode* untuk tahap testing.

Function testing

Input:

model model of trained deep learning model

list testingSamples of testing samples

Steps:

```

1:   function testing(model, testingSamples)
2:     testLabel  $\leftarrow$  get the test label from the testingSamples
3:     prediction  $\leftarrow$  generate prediction for testingSamples using the model
4:     averageAccuracy  $\leftarrow$  calculate the average accuracy of the model using testLabel
5:     overallAccuracy  $\leftarrow$  calculate the overall accuracy of the model using testLabel
6:     f1Score  $\leftarrow$  calculate the f1 score of the model using testLabel
7:     AUC  $\leftarrow$  calculate the AUC score of the model using testLabel
8:     return averageAccuracy, overallAccuracy, f1Score, AUC

```

Output:

return averageAccuracy, overallAccuracy, fIScore, AUC

Pseudocode 3.4 Testing.

3.3.5 Rencana Uji Coba

Uji Coba atau *testing* dilakukan menggunakan sampel *testing* yang diperoleh dari tahap *pre-processing*. Performa yang akan diukur adalah *overall accuracy*, *average accuracy*, *AUC*, dan *F1 score*

Tabel 3.3 Hasil Uji Performa.

Citra	<i>Average Accuracy</i>	<i>Overall Accuracy</i>	<i>F1 Score</i>	AUC
GM01				
GM02				
GM03				
GM04				
GM05				
GM06				
GM07				
GM08				
GM09				
GM10				
GM11				
GM12				
GM13				
GM14				
GM15				
GM16				
GM17				
GM18				

JADWAL KEGIATAN

No	Nama Kegiatan	Bulan																							
		Agustus				September				Oktober				November				Desember				Januari			
1	Studi Literatur																								
2	Penyusunan Proposal																								
3	Perancangan Metode																								
4	Sidang Proposal																								

DAFTAR PUSTAKA

- Afromowitz, M.A., Callis, J.B., Heimbach, D.M., DeSoto, L.A., Norton, M.K., 1988. Multispectral Imaging of Burn Wounds: A New Clinical Instrument for Evaluating Burn Depth. *IEEE Trans Biomed Eng* 35, 842–850. <https://doi.org/10.1109/10.7291>
- Alpers, W., Holt, B., Zeng, K., 2017. Oil spill detection by imaging radars: Challenges and pitfalls. *Remote Sens Environ* 201, 133–147. <https://doi.org/10.1016/J.RSE.2017.09.002>
- Bagaskara, A., Suryanegara, M., 2021. Evaluation of VGG-16 and VGG-19 Deep Learning Architecture for Classifying Dementia People. *Proceedings - 2021 4th International Conference on Computer and Informatics Engineering: IT-Based Digital Industrial Innovation for the Welfare of Society, IC2IE 2021* 1–4. <https://doi.org/10.1109/IC2IE53219.2021.9649132>
- Bhangale, U., Durbha, S.S., King, R.L., Younan, N.H., Vatsavai, R., 2017. High performance GPU computing based approaches for oil spill detection from multi-temporal remote sensing data. *Remote Sens Environ* 202, 28–44. <https://doi.org/10.1016/J.RSE.2017.03.024>
- Carrasco, O., Gomez, R.B., Chainani, A., Roper, W.E., 2003. Hyperspectral imaging applied to medical diagnoses and food safety. <https://doi.org/10.1117/12.502589> 5097, 215–221. <https://doi.org/10.1117/12.502589>
- Cheng, G., Wang, G., Han, J., 2022a. ISNet: Towards Improving Separability for Remote Sensing Image Change Detection. *IEEE Transactions on Geoscience and Remote Sensing* 60. <https://doi.org/10.1109/TGRS.2022.3174276>
- Cheng, G., Xie, X., Chen, W., Feng, X., Yao, X., Han, J., 2022b. Self-Guided Proposal Generation for Weakly Supervised Object Detection. *IEEE Transactions on Geoscience and Remote Sensing* 60. <https://doi.org/10.1109/TGRS.2022.3181466>
- Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., Jegelka, S., 2020. Debiased Contrastive Learning. *Adv Neural Inf Process Syst* 33, 8765–8775.
- Clark, R.N., Swayze, G.A., 1995. Mapping minerals, amorphous materials, environmental materials, vegetation, water, ice and snow, and other materials: The USGS tricorder algorithm. JPL, Summaries of the Fifth Annual JPL Airborne Earth Science Workshop. Volume 1: AVIRIS Workshop.
- Duan, P., Kang, X., Ghamisi, P., Li, S., 2023. Hyperspectral Remote Sensing Benchmark Database for Oil Spill Detection With an Isolation Forest-Guided Unsupervised Detector. *IEEE Transactions on Geoscience and Remote Sensing* 61. <https://doi.org/10.1109/TGRS.2023.3268944>
- Duan, P., Lai, J., Ghamisi, P., Kang, X., Jackisch, R., Kang, J., Gloaguen, R., 2020a. Component Decomposition-Based Hyperspectral Resolution Enhancement for Mineral Mapping. *Remote Sensing* 2020, Vol. 12, Page 2903 12, 2903. <https://doi.org/10.3390/RS12182903>
- Duan, P., Lai, J., Kang, J., Kang, X., Ghamisi, P., Li, S., 2020b. Texture-aware total variation-based removal of sun glint in hyperspectral images. *ISPRS Journal of Photogrammetry and Remote Sensing* 166, 359–372. <https://doi.org/10.1016/J.ISPRSJPRS.2020.06.009>
- Duan, P.H., Xie, Z.J., Kang, X.D., Li, S.T., 2022. Self-supervised learning-based oil spill detection of hyperspectral images. *Sci China Technol Sci* 65, 793–801. <https://doi.org/10.1007/S11431-021-1989-9/METRICS>
- Edelman, G.J., Gaston, E., van Leeuwen, T.G., Cullen, P.J., Aalders, M.C.G., 2012. Hyperspectral imaging for non-contact analysis of forensic traces. *Forensic Sci Int* 223, 28–39. <https://doi.org/10.1016/J.FORSCIINT.2012.09.012>
- Feng, Y.Z., Sun, D.W., 2012. Application of Hyperspectral Imaging in Food Safety

- Inspection and Control: A Review. *Crit Rev Food Sci Nutr* 52, 1039–1058.
<https://doi.org/10.1080/10408398.2011.651542>
- Fingas, M., Brown, C., 2014. Review of oil spill remote sensing. *Mar Pollut Bull* 83, 9–23.
<https://doi.org/10.1016/J.MARPOLBUL.2014.03.059>
- Fischer, C., Kakoulli, I., 2006. Multispectral and hyperspectral imaging technologies in conservation: current research and potential applications. *Studies in Conservation* 51, 3–16. <https://doi.org/10.1179/SIC.2006.51.SUPPLEMENT-1.3>
- Gao, L., Wang, D., Zhuang, L., Sun, X., Huang, M., Plaza, A., 2023. BS3LNet: A New Blind-Spot Self-Supervised Learning Network for Hyperspectral Anomaly Detection. *IEEE Transactions on Geoscience and Remote Sensing* 61.
<https://doi.org/10.1109/TGRS.2023.3246565>
- Gu, Y., Wang, C., Li, X., 2022. An Intensity-Independent Stereo Registration Method of Push-Broom Hyperspectral Scanner and LiDAR on UAV Platforms. *IEEE Transactions on Geoscience and Remote Sensing* 60. <https://doi.org/10.1109/TGRS.2022.3211202>
- Kamal, K., EZ-ZAHRAOUI, H., 2023. A comparison between the VGG16, VGG19 and ResNet50 architecture frameworks for classification of normal and CLAHE processed medical images. <https://doi.org/10.21203/RS.3.RS-2863523/V1>
- Kang, X., Deng, B., Duan, P., Wei, X., Li, S., 2023. Self-Supervised Spectral-Spatial Transformer Network for Hyperspectral Oil Spill Mapping. *IEEE Transactions on Geoscience and Remote Sensing* 61. <https://doi.org/10.1109/TGRS.2023.3260987>
- Kang, X., Wang, Z., Duan, P., Wei, X., 2022. The Potential of Hyperspectral Image Classification for Oil Spill Mapping. *IEEE Transactions on Geoscience and Remote Sensing* 60. <https://doi.org/10.1109/TGRS.2022.3205966>
- Karantzalos, K., Argialas, D., 2008. Automatic detection and tracking of oil spills in SAR imagery with level set segmentation. *Int J Remote Sens* 29, 6281–6296.
<https://doi.org/10.1080/01431160802175488>
- Khan, M.J., Khan, H.S., Yousaf, A., Khurshid, K., Abbas, A., 2018. Modern Trends in Hyperspectral Image Analysis: A Review. *IEEE Access* 6, 14118–14129.
<https://doi.org/10.1109/ACCESS.2018.2812999>
- Kokaly, R.F., Couvillion, B.R., Holloway, J.A.M., Roberts, D.A., Ustin, S.L., Peterson, S.H., Khanna, S., Piazza, S.C., 2013. Spectroscopic remote sensing of the distribution and persistence of oil from the Deepwater Horizon spill in Barataria Bay marshes. *Remote Sens Environ* 129, 210–230. <https://doi.org/10.1016/J.RSE.2012.10.028>
- Lacava, T., Ciancia, E., Coviello, I., Di Polito, C., Grimaldi, C.S.L., Pergola, N., Satriano, V., Temimi, M., Zhao, J., Tramutoli, V., 2017. A MODIS-Based Robust Satellite Technique (RST) for Timely Detection of Oil Spilled Areas. *Remote Sensing* 2017, Vol. 9, Page 128 9, 128. <https://doi.org/10.3390/RS9020128>
- Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521:7553 521, 436–444.
<https://doi.org/10.1038/nature14539>
- Le-Khac, P.H., Healy, G., Smeaton, A.F., 2020. Contrastive Representation Learning: A Framework and Review. *IEEE Access* 8, 193907–193934.
<https://doi.org/10.1109/ACCESS.2020.3031549>
- Li, X., Yang, X., Ma, Z., Xue, J.H., 2023. Deep metric learning for few-shot image classification: A Review of recent developments. *Pattern Recognit* 138, 109381.
<https://doi.org/10.1016/J.PATCOG.2023.109381>
- Li, Y., Lu, H., Zhang, Z., Liu, P., 2020. A novel nonlinear hyperspectral unmixing approach for images of oil spills at sea. *Int J Remote Sens* 41, 4682–4699.
<https://doi.org/10.1080/01431161.2020.1723179>
- Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J., 2022. A Survey of Convolutional Neural

- Networks: Analysis, Applications, and Prospects. *IEEE Trans Neural Netw Learn Syst* 33, 6999–7019. <https://doi.org/10.1109/TNNLS.2021.3084827>
- Liang, H., 2012. Advances in multispectral and hyperspectral imaging for archaeology and art conservation. *Appl Phys A Mater Sci Process* 106, 309–323. <https://doi.org/10.1007/S00339-011-6689-1/FIGURES/14>
- Liu, D., Han, L., 2017. Spectral Curve Shape Matching Using Derivatives in Hyperspectral Images. *IEEE Geoscience and Remote Sensing Letters* 14, 504–508. <https://doi.org/10.1109/LGRS.2017.2651060>
- Liu, S., Chi, M., Zou, Y., Samat, A., Benediktsson, J.A., Plaza, A., 2017. Oil Spill Detection via Multitemporal Optical Remote Sensing Images: A Change Detection Perspective. *IEEE Geoscience and Remote Sensing Letters* 14, 324–328. <https://doi.org/10.1109/LGRS.2016.2639540>
- Padoan, R., Steemers, Th.A.G., Klein, M.E., Aalderink, B.J., De Bruin, G., 2008. Quantitative Hyperspectral Imaging of Historical Documents: Technique and Applications. *e-Journal of Nondestructive Testing* 13.
- Rao, W., Qu, Y., Gao, L., Sun, X., Wu, Y., Zhang, B., 2022. Transferable network with Siamese architecture for anomaly detection in hyperspectral images. *International Journal of Applied Earth Observation and Geoinformation* 106, 102669. <https://doi.org/10.1016/J.JAG.2021.102669>
- Satriano, V., Ciancia, E., Lacava, T., Pergola, N., Tramutoli, V., 2019. Improving the RST-OIL Algorithm for Oil Spill Detection under Severe Sun Glint Conditions. *Remote Sensing* 2019, Vol. 11, Page 2762 11, 2762. <https://doi.org/10.3390/RS11232762>
- Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.
- Singha, S., Bellerby, T.J., Trieschmann, O., 2013. Satellite oil spill detection using artificial neural networks. *IEEE J Sel Top Appl Earth Obs Remote Sens* 6, 2355–2363. <https://doi.org/10.1109/JSTARS.2013.2251864>
- Sudharshan, V., Seidel, P., Ghamisi, P., Lorenz, S., Fuchs, M., Fareedh, J.S., Neubert, P., Schubert, S., Gloaguen, R., 2020. Object Detection Routine for Material Streams Combining RGB and Hyperspectral Reflectance Data Based on Guided Object Localization. *IEEE Sens J* 20, 11490–11498. <https://doi.org/10.1109/JSEN.2020.2996757>
- Sun, S., Hu, C., 2019. The Challenges of Interpreting Oil-Water Spatial and Spectral Contrasts for the Estimation of Oil Thickness: Examples From Satellite and Airborne Measurements of the Deepwater Horizon Oil Spill. *IEEE Transactions on Geoscience and Remote Sensing* 57, 2643–2658. <https://doi.org/10.1109/TGRS.2018.2876091>
- Sun, S., Hu, C., Tunnell, J.W., 2015. Surface oil footprint and trajectory of the Ixtoc-I oil spill determined from Landsat/MSS and CZCS observations. *Mar Pollut Bull* 101, 632–641. <https://doi.org/10.1016/J.MARPOLBUL.2015.10.036>
- Taravat, A., Del Frate, F., 2012. Development of band ratioing algorithms and neural networks to detection of oil spills using Landsat ETM+ data. *EURASIP J Adv Signal Process* 2012, 1–8. <https://doi.org/10.1186/1687-6180-2012-107/FIGURES/7>
- Xiao, T., Wang, X., Efros, A.A., Darrell, T., 2020. What Should Not Be Contrastive in Contrastive Learning. *ICLR 2021 - 9th International Conference on Learning Representations*.
- Zhu, X., Li, Y., Zhang, Q., Liu, B., 2019. Oil Film Classification Using Deep Learning-Based Hyperspectral Remote Sensing Technology. *ISPRS International Journal of Geo-Information* 2019, Vol. 8, Page 181 8, 181. <https://doi.org/10.3390/IJGI8040181>

LAMPIRAN-LAMPIRAN ATAU APPENDIKS