**VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY**

**INTERNATIONAL UNIVERSITY**

**DEPARTMENT OF PHYSICS**

**SPACE ENGINEERING**



**PROJECT’S TOPIC**

Nguyen VAN A

A thesis submitted to

Department of Physics

in partial fulfillment of the Requirements

for the Degree of Engineer IN SPACE ENGINEERING

Ho chi minh city, viet nam  
2020

**PROJECT’S TOPIC**

**Nguyen VAN A (Full name)**

Under the guidance and approval of the committee, and approved by its members, this thesis has been accepted in partial fulfillment of the requirements for the degree.

Approved:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Chairperson

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Committee member

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Committee member

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Committee member

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Committee member

# HONESTY DECLARATION

My name is Nguyen Trong Nghia. I would like to declare that, apart from the acknowledged references, this thesis either does not use language, ideas, or other original material from anyone; or has not been previously submitted to any other educational and research programs or institutions. I fully understand that any writings in this thesis contradicted to the above statement will automatically lead to the rejection from the SE program at the International University – Vietnam National University Ho Chi Minh City.

Date:

Student’s Signature

(Full name)

# TURNITIN DECLARATION

Name of Student: Nguyen Trong Nghia

Date:

Graphical user interface, text, email

Description automatically generated

Advisor Signature Student Signature

# ACKNOWLEDGMENT

It is with deep gratitude and appreciation that I acknowledge the professional guidance of Dr. Nguyen Van B. His constant encouragement and support helped me to achieve my goal.

………………………

# TABLE OF CONTENTS

[HONESTY DECLARATION 3](#_Toc21345696)

[TURNITIN DECLARATION 4](#_Toc21345697)

[ACKNOWLEDGMENT 5](#_Toc21345698)

[TABLE OF CONTENTS 6](#_Toc21345699)

[LIST OF TABLES 7](#_Toc21345700)

[LIST OF FIGURES 8](#_Toc21345701)

[ABBREVIATIONS AND NOTATIONS 9](#_Toc21345702)

[ABSTRACT 9](#_Toc21345703)

[CHAPTER I INTRODUCTION 10](#_Toc21345704)

[1.1. Level-2 Title 10](#_Toc21345705)

[1.1.1. Level-3 title 10](#_Toc21345706)

[CHAPTER II LITERATURE REVIEW 12](#_Toc21345716)

[CHAPTER III METHODOLOGY 12](#_Toc21345717)

[CHAPTER IV RESULTS 12](#_Toc21345718)

[CHAPTER V DISCUSSION AND IMPLEMENTATIONS 12](#_Toc21345719)

[CHAPTER VI CONCLUSION AND RECOMMENDATION 13](#_Toc21345720)

[REFERENCES 13](#_Toc21345721)

[APPENDICES 13](#_Toc21345722)

# LIST OF TABLES

Table 1.1 1

Table 1.2 2

Table 2.1 3

Table 2.2 5

Table 2.3 12

Table 2.4 22

# LIST OF FIGURES

Figure 1.1 1

Figure 1.2 1

Figure 2.1 6

Figure 2.2 13

Figure 2.3 22

Figure 3.1 26

# ABBREVIATIONS AND NOTATIONS

RS: Remote Sensing

DINEOF: Data Interpolating Empirical Orthogonal Functions

TIEOF: Tensor Interpolating Empirical Orthogonal Functions

SVD: Singular Value Decomposition

HOOI: Higher Order Orthogonal Iteration of Tensors

NLP: Natural Language Processing

BERT: Bidirectional Encoder Representations from Transformers

RoBERTa: Robust Optimized BERT Pretraining Approach

HOSVD: Higher Order Singular Value Decomposition

PARAFAC: Parallel Factor Analysis

PCA: Principal Component Analysis

ALS: Alternating Least Squares

# ABSTRACT

Remote sensing data are frequently used in extraction of many interesting physical characteristics such as land use, urbanization, vegetation index, water levels,… One of the most frequent problem in Remote Sensing is the lost of data due to spatial pixels being covered by clouds. We believe that these loss data can be recovered based on the spatial relationship between satellite data taken at different time. While there have been a lot of papers addressing the exact same problem – following the same assumptions as us. These papers usually interpolate continuous numerical data – meaning the interpolated values followed an implicit linear level in their characteristic. In this paper, we try to apply the same method for categorical satellite data and explored the use of a Nature Language Processing neural network in interpolating discrete nominal satellite data.

# CHAPTER I INTRODUCTION

The Introduction presents the background of the thesis work. It reveals the state of the art of the topic by quoting previously published works in the fields and important questions that your work relates to them. It should contain parts such as Rationale, Problem statement, Objectives, Scope, Limitation, Research framework, Structure of research and the summary of the contains of the following parts of the report. It is important that this section be unique and specific to the report. As a common practice, it is conceived before you do your work but is written when you finish writing all other chapters. Most of its contents can be used in your thesis report.

After decades of continuous innovations, satellite-based sensing technologies have achieved remarkable results in their ability to offer high spatial resolution data on a global scale, giving us the necessary tools to make wide and continuous observations of the Earth’s surface. One such outstanding example is the LandSat missions, which have been capturing high quality imagery of the Earth’s surface since July 1972 [1].

While LandSat offer high spatial resolution data of the Earth’s surface every 14 days. Many of these valuable data must be discarded due to atmospheric conditions, mainly clouds, which force many studies to use data with high temporal gap. For example, a study regarding Mangrove Forest change detection in Ca Mau, Viet Nam used LandSat data from 1979 to 2013 [2]. Such limitation limits the amount of data researcher can use to form and validate their hypothesis, creating unnecessary challenges in earth surface analysis.

Several methods for interpolating cloud-covered pixels have been developed. The main assumption of these methods is that covered pixels at a specific point in time can be interpolated using an image in which this pixel is not covered by clouds. Meaning pixels obtained on a cloudy day can be interpolated using information obtained in more favorable conditions. Most of the methods developed for cloud-filling are based on matrix decomposition, most noticeably, DINEOF (Data Interpolation Empirical Orthogonal Functions) [3].

DINEOF assumes that data matrix reconstruction using leading orthogonal functions will capture low-frequency large-scale data, while minimizing the noise and high frequency components resulted from missing values. The aforementioned matrix is a collection of flatten satellites images – meaning it is a 2D matrix, which each row being a satellite image flatten into 1D vector. This representation of satellite data capture both time and spatial information – for which DINEOF will try to find the latent factors represented by high-ranking orthogonal functions. The most recent implementation of DINEOF is called TIEOF (Tensor Interpolation using Empirical Orthogonal Functions) [4], which make use of unflatten satellite images – introducing the distinction between horizontal and vertical into the scheme.

Another scheme based on matrix decomposition were explore by Ruo-Qian Wang (2021) [5]. In which a back propagation implementation of matrix decomposition called Funk-SVD [6] were used. While originally inspired by Recommender System used in Netflix film recommendation, the main ideal behind Funk-SVD is using back propagation to minimize the loss of the reconstructed matrix.

Matrix decomposition-based schemes achieved remarkable results in many cloud-filling tasks [3, 4, 5, 7, 8]. However, due to the inherent underlying assumption of matrix decomposition, these methods only work for continuous numerical data – where there exists a linear relationship in the reconstructed between values in the reconstructed matrix. In another word, a value of in the reconstructed matrix will have an inherent relationship with a value of in the matrix – and the value of is closer to the value of , compared to the value of . Practically speaking, this inherent relationship makes matrix decomposition-based scheme ill-suited for categorical data – where there exists no such relationship between the value of and . This was proven true in our experiment, which will be discussed in Chapter 3.

After such experiment with matrix decomposition-based scheme, Dr. Van suggest the usage of an Attention based NLP model (Natural Language Processing) [9] – whose use of tokenization and query pair eliminate the associated linear relationship found in matrix decomposition. Specifically, BERT (Bidirectional Encoder Representation from Transformer) [10] were suggested.

At its core, BERT is a stack of encoder from the famous Transformer architecture [9]. Developed to solve two main tasks in Nature Language Processing: missing word prediction and next sentence prediction. Of which we took advantage of the missing word prediction workflow – by turning the satellite image into a collection of “sentences”, in which, cloud covered pixels is have the same effects as missing words. Technically, we made use of RoBERTa [11] – which is an optimized implementation of BERT.

In summary, this paper experimented with matrix decomposition schemes in data interpolation for categorical data – with a sharp focus on Remote Sensing imagery. Additionally, another scheme making use of NLP model were explored – which offer arguably better results for categorical data.

## 1.1. Level-2 Title

Your text should begin from here. This is the illustration for the use of the level-2 title style

* 1. 1. Level-3 title

Your text should begin from here. This is the illustration for the use of the level-3 title style

The format of equations, formula:







Table 1.1.Table’s Name

|  |  |  |  |
| --- | --- | --- | --- |
| **Circuit** | **Vref (V)** | **Iref (mA)** | **Iref2 (mA)** |
| Circuit 1 | 0.01 | 1 | 2 |
| Cirucit 2 | 0.1 | 2 | 3 |
| Etc. |  |  |  |

Table 1.2.Table’s Name

|  |  |  |  |
| --- | --- | --- | --- |
| **Circuit** | **Vref (V)** | **Iref (mA)** | **Iref2 (mA)** |
| Circuit 3 | 0.01 | 1 | 2 |
| Cirucit 4 | 0.1 | 2 | 3 |
| Etc. |  |  |  |



Figure 1.1. Figure’s Name



Figure 1.2. Figure’s Name

# CHAPTER II. LITERATURE REVIEW

There have been many approaches for cloud-filling schemes applied specifically for remote sensing data. However, their application is quite narrow, as most of the study were conducted on ocean surface data [3, 4, 7] – where the basis characteristic of the study region remains constant. As such, the underlying high-ranking low frequency orthogonal functions remain relatively consistent over a long period of time. Application of cloud filling scheme over coastal and land regions were also explored [5], but the volume of studies remains quite small.

Additionally, most of the previous cloud-filling schemes were done on numerical continuous data. There exists close to zero study exploring the usage of cloud-filling scheme for discrete nominal satellite data. Under the stated assumption, we hope that our paper would widen the use of cloud filling scheme for two cases: coastal/land region and categorical data.

As stated in the previous chapter, DINEOF is a deeply studied approach [3, 12] – with its most recent implementation being a higher dimension implementation [4]. Traditional DINEOF scheme requires the flattening of 2D satellite images into a 1D vectors – which are then stacked on top of one another to create a 2D matrix (; where are the width and height of the satellite image). Then, will be decomposed using SVD (singular value decomposition), truncating to the first k orthogonal functions, the matrix will be reconstructed and compared to the original matrix. The reconstructed matrix contained the interpolated values. However, calculation of EOF using an SVD requires the matrix to be full (no-missing values), and the EOF is needed to calculate the full matrix. This circular relationship is natural suited for an iterative process described in more details by Becker and Rixen (2003) [12]. Higher dimensional of this method were explored and named TIEOF by Leonid Kulikov et al. (2021) [4]. In which the iterative process described by Becker and Rixen (2003) [12] remains nearly identical, but different decomposition schemes apart from SVD were explored – HOOI (Higher Order Orthogonal Iteration of Tensors) [13] being the best performing scheme.

Aside from matrix reconstruction schemes based on underlying orthogonal functions. Funk-SVD [6] were developed as a back propagation-based matrix reconstruction scheme and was applied successfully in remote sensing workflow by Ruo-Qian Wang (2021) [5]. The main innovative feature of Funk-SVD is: instead of the iterative process described by Becker and Rixen (2003) [12], the reconstruction use gradient descent to update the values of composite matrices.

Matrix decomposition-based method for cloud filling schemes were proven effective due to the sheer volume of supporting literature [3, 12, 8, 4, 7, 5]. Andy Stock et al. (2020) [14] conducted a study specifically verifying the effectiveness of these cloud-filling approaches. However, while there is a high volume of study focused on interpolation of continuous numerical satellite data, interpolation schemes for categorical data remains under-developed.

As such, this study experimented with the aforementioned approaches, but tailored for categorical data. Specifically, Funk-SVD and HOOI TIEOF were used – whose implementations and results are described in more details in Chapter III and Chapter IV. Unfortunately, compared to their continuous numerical counterpart, these methods performed poorly on categorical data.

We hypothesized that this poor performance is due to the very nature of matrix decomposition, which forced a linearity relationship between interpolated value. As such, we decided to experiment with Nature Language Processing model – whose usage of tokenizer and attention mechanisms would eliminate the hidden linear relationship between interpolated value.

The model we choose is belong to the transformer architecture introduced by Ashish Vaswani et al. (2017) [9]. Specifically, BERT [10] were used, as it were designed for missing word interpolation – which is analogous to cloud-covered pixel interpolation. Transformer based NLP model make use of multi-head attention mechanism – which can be shortly described as an information integration scheme of the input query, the current key and the current output query. This attention mechanism allows the model to interpolate a specific word based on the overall context of the input sentence, the current input key and the current (unfinished) output sentence. The differences between the original proposed Transformer model and BERT are:

* BERT consist only of the encoder (while Transformer consist of encoders and decoders),
* The attention mechanism was modified to incorporate information from the output query in both dimensions – not just the unfinished output query (which gave this model the name **B**idirectional **E**ncoder **R**epresentation from **T**ransformer).

The second different gave BERT an edge in context understanding for NLP tasks – which allow BERT to achieve new state of the art performance on the GLUE (Generative Language Understanding Evaluation) benchmark [15].

# CHAPTER III. METHODOLOGY

## 3.1 Training Data

The training data consists of six satellite images taken from LandSat 8 and LandSat 9 from January 10th 2022 to March 7th 2022. These images were chosen due to two characteristics: low cloud-cover and short time span. Short temporal gap is a key requirement as the underlying land surface should remain the same in our data, whose interpolation is the aim of this study. Additionally, low cloud cover ensures that these land surface can be captured by satellite sensors. Specifically, the training data consists of the following images[[1]](#footnote-1):

* LC09\_L2SP\_126054\_20220110\_20220122\_02\_T1
* LC08\_L2SP\_126054\_20220118\_20220123\_02\_T1
* LC09\_L2SP\_126054\_20220126\_20220128\_02\_T1
* LC08\_L2SP\_126054\_20220203\_20220212\_02\_T1
* LC08\_L2SP\_126054\_20220219\_20220302\_02\_T1
* LC08\_L2SP\_126054\_20220307\_20220314\_02\_T1

The data is then preprocessed into classification data and pass into different workflows. A quick summary is:

* Funk-SVD and TIEOF:
  + Train set of only six real satellites image
  + Train set of 87 images, 81 of which is imputed from the set of six real satellite image. This set of 81 imputed images is “skewed” toward days with less cloud.
* RoBERTa: train set of only six satellites image

RoBERTa train set is smaller due to hardware limit of Google Colab – as BERT were not developed to handle image data, creating some complication in feeding such data into BERT.

### 3.1.1 Preprocessing Scheme

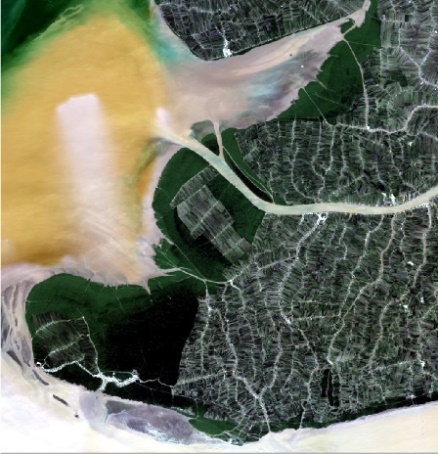
Preprocessing of satellite data were done in QGIS – using SCP (Semi-Automatic Classification Plugin) [16]. The workflow for satellite data preprocessing is quite ordinary, as it was introduced to us during our courses – online documentation describing the process in more detail can also be found [17]. In general, the workflow consists of:

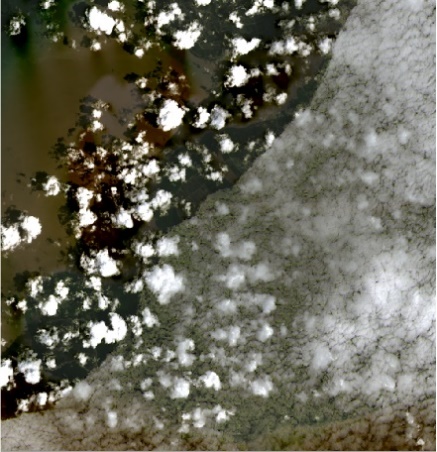
* Cropping satellite image into study region.
* Choose specific band as input into SCP.
* Convert the input bands into false color composite images, which utilized band 2,3,4,5 and 6 of Landsat data.
* :abel these data into 5 classes: Cloud, Water Body, Vegetation, Agriculture and Others.

Firstly, the chosen study region is the “horn” of Ca Mau province, specifically, it is the Man Grove forested region as mentioned by Nguyen-Thanh Son (2014) [2]. This region was chosen specifically due to two main reasons:

* It is a well-study area – and is currently under protection. Which enable us to inherit existing characteristic of land surface from the available literatures.
* Due to various time and logistic constraint. We couldn’t obtain real classification data at each spatial location represented by satellite pixel. Which force us to work entirely on the set of classes mentioned by Nguyen-Thanh Son (2014) [2].

Three real color images for our study region are shown in. These were chosen as demonstration for different clouds condition between different images.

A picture containing outdoor, plant

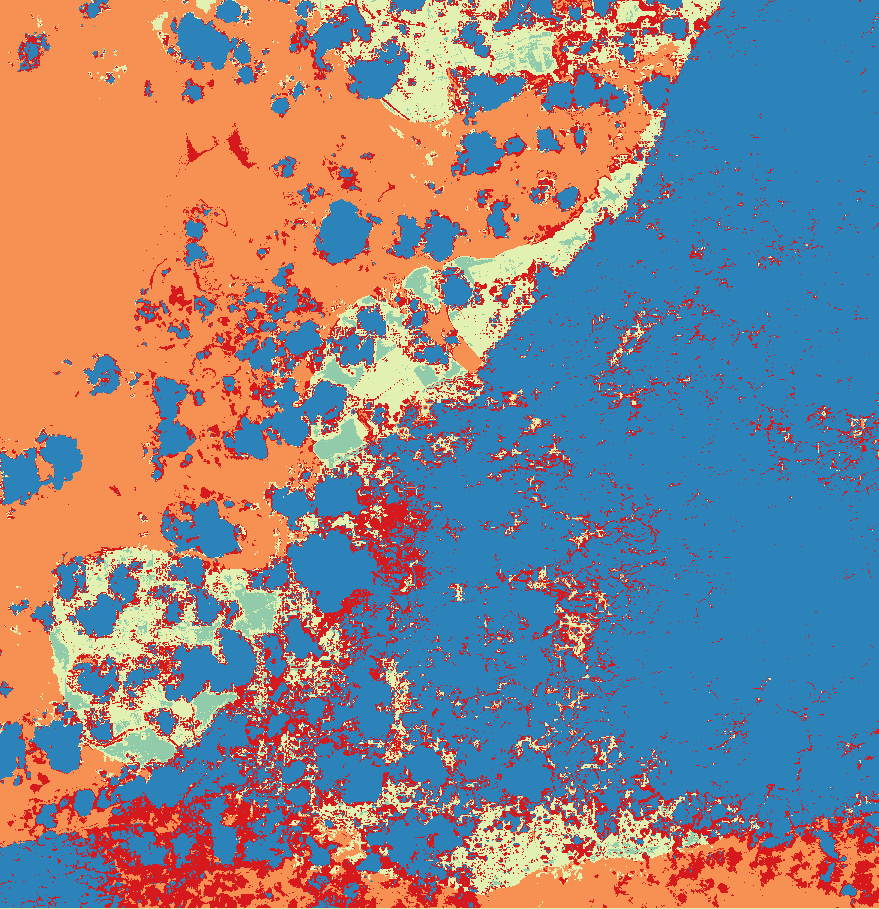
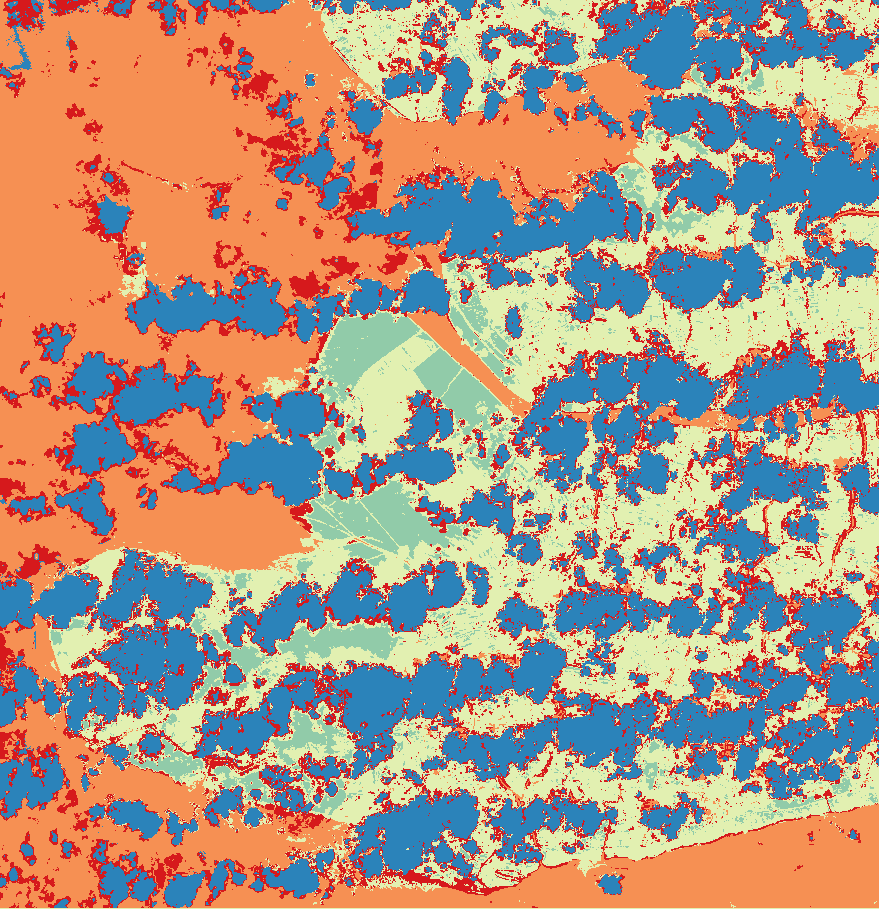
Description automatically generated

Secondly, after the region have been cropped, SCP preprocessing toolbox were used, specifically, by feeding the chosen input band and its respective metadata file into SCP as described in [17]. The chosen bands were Landsat band 1,2,3,4,5,6 and 7 [18].

Thirdly, utilizing SCP virtual band set functionality, multiple color composite images were created, most notably: nature color (4,3,2), color infrared (5,4,3), Short-wave infrared (7,6,4) and Agriculture (6,5,2) [19].

Fourthly and finally, using the aforementioned color composite images and SCP classification toolbox, pixels are labeled as into five classes: Clouds, Water, Vegetation, Agriculture and Others. As shown in

Map

Description automatically generated

It should be noted that the class Other – while initially designed to contained human made building and tents along Agriculture areas – ended up also covered alluvial and darken pixels due to cloud’s shadow. Furthermore, the cloud’s shadows created misclassification in some areas, most noticeable, some vegetation areas are classified as Agriculture. These data imperfections were tolerated, as the focus of this study is not data preprocessing.

### 3.1.2 Synthetic Cloud Cover Generation

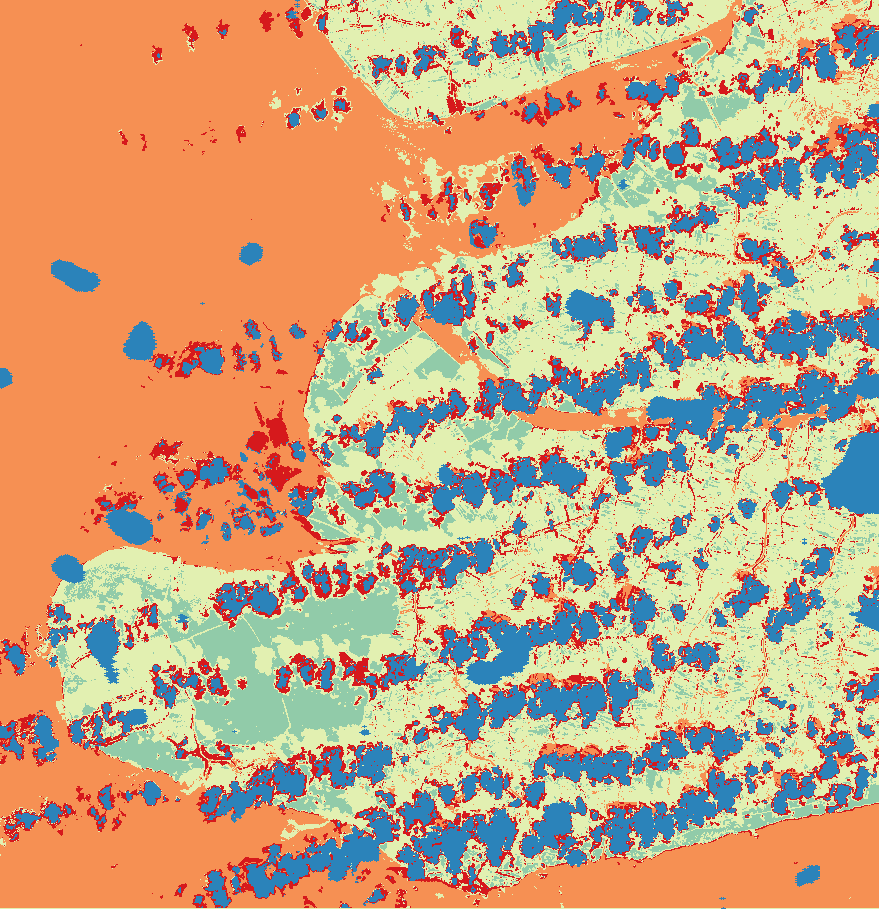
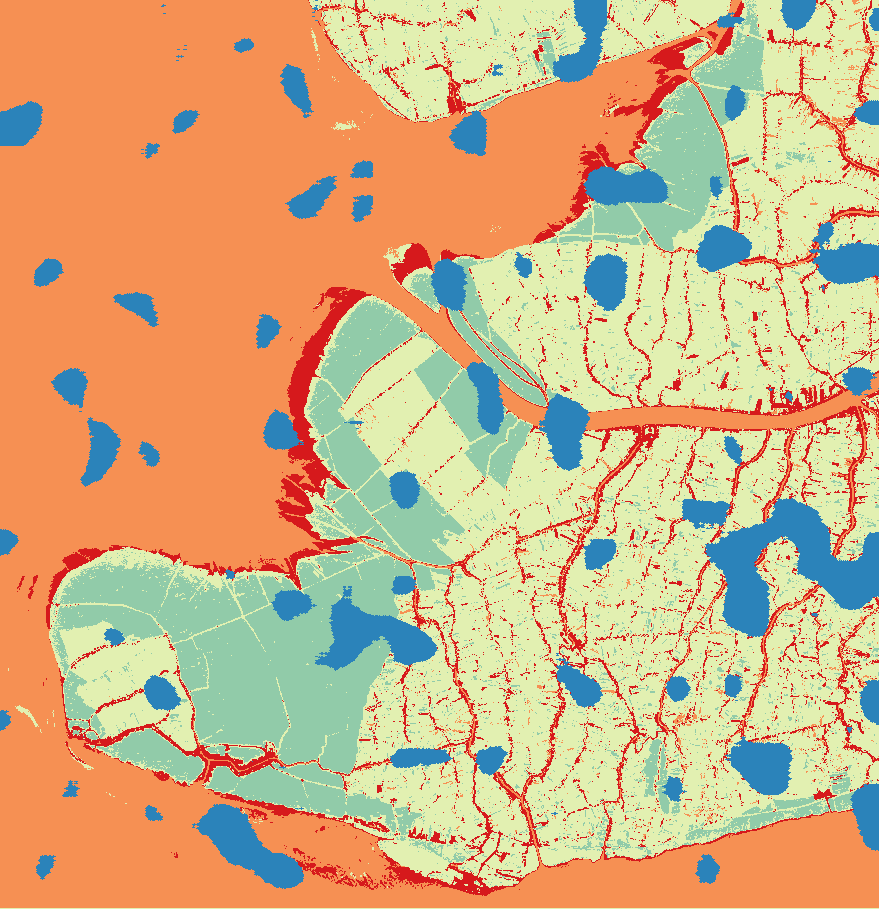
After preprocessing, six Tiff images were generated – each containing a 2D matrix of value in range corresponding to the five classes. Unfortunately, such small amount of training data is not sufficient for matrix decomposition-based scheme. To solve this issue, an adaptive and randomized cloud generator were developed – whose detailed implementation is captured in our **adding\_synthetic\_data\_into\_tensor()** function [20].

The general ideal behind this generator is:

* Take a real image.
* Calculate the percentage of cloud cover pixels.
* If this percentage is lower than , the image is selected as base data for synthetization.
* Create 20 synthetized images, until the cloud cover percentage is higher than

The main driving force behind this implementation is to add synthetized data of less clouded days into our workflow. Which introduced a “bias” for images with favorable atmospheric conditions.

In total, 81 images were synthetized, examples of which are shown in



It should be noted that the synthetized clous does not inherit the surrounding “Other” pixels associated with real clouds - and the misclassification due to cloud shadow were also not captured.

## 3.2 Funk-SVD Matrix Reconstruction

The main ideal behind Funk-SVD is simple: follow the derivative of the loss function between the original matrix and the reconstructed matrix – ignoring missing values in the matrix as describe by Simon Funk (2006) [6].

In this paper, Geoffrey Bolmier’s implementation were used [21]. As Funk-SVD were originally design for a movie recommendation system, our two datasets had to be modified lightly to fit into Funk-SVD workflow. Specifically, Funk-SVD required the data to be in the form of a csv file with three columns: movie\_id, customer\_id and rating. The exact same scheme was used for both of our datasets: “real images” and “real and synthesized images”.

Preprocessing for Funk-SVD started by flattening our array of 2D satellites image, meaning reshaping our data from [ into . Which is analogous to the original Netflix matrix of [22]. Then, the data is preprocessed into a csv file with three needed columns, with value representing class “cloud” ignored. Finally, the correctly formatted data is passed into Funk-SVD, which will learn the two composite matrices, from which our image can be reconstructed. The implementation details can be found here [20].

## 3.3 Tensor Interpolation Empirical Orthogonal Functions

Sitting at the main theoretical foundation of TIEOF are Tensor Decomposition schemes – in which Leonid Kulikov (2021) [4] found the best performing scheme is HOOI (Higher Order Orthogonal Iteration of Tensors) [13]. The main ideas behind HOOI are:

* HOSVD (Higher Order Singular Value Decomposition) [23] is used to find EOFs and a tensor kernel having the same order as the original tensor, whose matrix multiplication with the EOFs create an approximation of the original matrix.
* This approximation matrix is slowly improve using ALS (Alternating Least Squares) Algorithm.

Leonid Kulikov implementation of TIEOF is available on his Github [24]. Unfortunately, the Python implementation consists of some dependency bugs – rendering this version unusable for our purpose. As such, a small bug fix was developed and is currently available on our Github [20].

Nevertheless, the required input for have the format of , in which the second dimension consists of three information . As such, the preprocessing scheme consists of no image flattening – this main idea differs TIEOF from other matrix decomposition schemes, as the information regarding horizontal/vertical direction is kept. For our purpose, is the total number of pixels across all our image, are the row index, column index and image index respectively. Detail implementation of this preprocessing flow is contained in our Github [20]. Similarly to our Funk-SVD implementation, this workflow was applied for both of our datasets: “real images” and “real and synthesized images”.

## 3.4 Masked Language Modelling – RoBERTa Implementation

Natural Langue Processing is an extremely active research field, while the original Transformer architecture was developed by Ashish Vaswani et al. (2017) [9]. Many alterations and ideas have been implemented on this architect – seemingly every year. Of which, the two most well-known implementations are: Google’s BERT [10] and OpenAI’s GPT-3 [25].

BERT’s architecture were chosen for our paper – as BERT offer better documentations and open-source supports. Transformer based models depend on the Multi-Head attention mechanisms – whose inner working is described in-depth in the original “Attention Is All You Need” paper by Ashish Vaswani (2017) [9]. Essentially, Multi-Head Attention can be understood as a concatenation of attentions heads – with each head being an attention vector encoding the importance of a specific words to other words in the sentence. This Multi-Head Attention is applied for both the encoder (block containing input sentence) and the decoder block (block containing input and output sentence). However, on the decoder side, the Multi-Head Attention is called Masked Multi-Head Attention, as the attention mechanism will mask the word behind the current word.

For example, the sentence “He speaks both English and German” will be given an example attention vector on both the Multi-Head Attention and Masked Multi-Head Attention layer. Assuming the current word is “speaks”, using the Multi-Head Attention – the attention vector could be with each value corresponding to a specific word in the sentence. On the other hand, an attention vector using Masked Multi-Head Attention could be - where words behind “speaks” will be masked with . The attention mechanism was designed in this way to “force” the model to learn the underlying context information of sentences – instead of overfitting a specific input sentence to a specific output sentence.

The original transformer architecture encoder and decoder were hypothesized as having the ability to understand linguistic context, where the encoder understand the input sentence, and the decoder understand the relationship between the input and output sentence. As the original goal was language translation – we needed the decoders and the Masked Multi-Head Attention layer.

Under these premises, BERT (Bidirectional Encoders Representation from Transformers) [10] was developed. BERT was originally developed for two tasks: missing word prediction and next sentence classification. These tasks involve zero translating, removing the need for decoders and Masked Multi-Head attention. Essentially, BERT is a model containing language understanding of a single language. Additionally, without Masked Multi-Head Attention, the model “see” the output sentence in both directions – hence the name Bidirectional.

For our purpose, BERT’s workflow for missing word prediction were used, the original BERT implementation can be found on Google Research’s Github [26]. However, a replicate study by Yinhan Liu et al. [11] optimized the original BERT implementation. Following the footsteps of these studies, we implemented a RoBERTa model for Masked Language Modelling using Hugging Face **RobertaForMaskedLM** class [27]. For RoBERTa, we only experimented with “real images” dataset – our implementation is available in our Google Colab Notebook [28].

### 3.4.1 Preprocessing Scheme

As RoBERTa was developed for sentences - not 2D images – formatting satellite imagery into the correct scheme is required. Our preprocessing scheme can be divided into the following steps:

* Write TIFF images into txt files – each file corresponding to an image – where each row represents the actual row of the 2D image.
* Feed these files into a Tokenizer, specifically, ByteLevelBPETokenizer was chosen. This step produces a Tokenizer with a specific set of vocabulary.
* Record the index of the “cloud” class and the “<mask>” class in the tokenizer– which in our case is and respectively.
* Create a dataset, where each sample contains three vectors:
  + input\_ids: tokenized sentence – in our case, is the tokenized row of the original 2D image, with the token replaced by the token.
  + attention\_mask: this is padding mask, conventional sentences has different length, which is not the case for 2D image. As such, this vector has no significance important for our purpose.
  + label: tokenized output sentence, meaning this should have zero “<mask>” token. However, for our case, as the existing of “<mask>” is unavoidable (cloud). As such, this is the same “masked” row, but of a different image.

The last step meant that for training image, the same row will appear in the input times, each time match with the same row of a different satellite image. Specifically, our data set would have the dimension of , where , with and equal to the number of images, the number of images minus one and the number of rows respectively.

This preprocessing scheme was developed with two goals in mind: prevents overfitting and creates a “bias” toward day with favorable atmospheric conditions. As each sentence (row) can have five correct answers, the model is hypothesized to arrive at the “most likely” result for each pixel, instead of overfitting toward one specific “truth” image. Additionally, days with less cloud will have less “<mask>” token, which would bias the model toward such day as it would serve as labels to all remaining day.

### 3.4.2 Training Workflow

After preprocessing, a Dataset class inherited from torch.utils.data.Dataset [29] is created, which will use torch.utlis.data.DataLoader [30] class to load the data into our model. These small technical details are mentioned for the sake of completion, as Hugging Face **RobertaForMaskedLM** required the workflow to utilize Pytorch.

In total, we trained 6 models – with slight variation in hyperparameters of: epoch, number of encoders, number of hidden states and number of attention heads. Specifically, these are:

* 2 epochs – 4 encoders – 512 hidden states – 4 attention heads.
* 2 epochs – 8 encoders – 512 hidden states – 8 attention heads
* 4 epochs – 2 encoders – 1024 hidden states – 2 attention heads
* 4 epochs – 4 encoders – 512 hidden states – 4 attention heads
* 4 epochs – 4 encoders – 1024 hidden states – 4 attention heads
* 8 epochs – 2 encoders – 1024 hidden states – 2 attention heads

It should be noted that training took around 90 minutes for Google Colab’s NVDIA Telsa K80 for each model – the highest time cost between all our schemes. Additionally, deeper model or bigger dataset were not explored due to Google Colab’s GPU RAM limitations.

### 3.4.3 Output Workflow

Output pipeline is implemented using Hugging Face’s transformer.FillMaskPipeline class [31], for which we limit to interference result to the original set of classes of our satellite image.

The pipeline is our reconstruction function, the workflow involves feeding our image into the pipeline row by row. For each row, we replace by “<mask>”, resulting in a “sentence” with multiple “masked” words. Unfortunately, the pipeline was not developed to handle multiple “<mask>” in the same sentence, a more detailed discussion can be found in this open pull request [32].

For our implementation, we wrote a custom output script that ignores other “<mask>” token when interfering a specific “<mask>” token. This implementation might have worse performance in cases where the interpolated value of a “<mask>” token is highly dependent on value at the index of another “<mask>” token. Nevertheless, our implementation achieved acceptable performance with minimal increase in interference time.

The Methodology Chapter shod show the methods and materials used in our work. Explain why you chose them. Describe in such a way that others can replicate exactly your work.

# CHAPTER IV. RESULTS

The experimental results are shown below, specifically these are:

* Funk-SVD on “real images” dataset
* Funk-SVD on “real and synthetic images” dataset
* TIEOF on “real images” dataset
* TIEOF on “real and synthetic images” dataset
* RoBERTa on “real images” dataset

Unfortunately, there exists zero quantitative metric for which the effectiveness of our model can be judged. The main reason is due to the inherent lack of physical data of our study region, while our study assumes that the spatial characteristics of our study region remains constant over the period between January 2022 and March 2022. The legitimacy of this claim is questionable without land surveying and control point data.

As such, our result can only be validated visually over the color-mapped reconstructed class matrix – similarly to many satellite image reconstruction studies [3, 8, 7, 5]. Nevertheless, visual inspections suggest that matrix decomposition method such as Funk-SVD and TIEOF performs poorly on categorical data. Which support our claims regarding the insufficient of matrix decomposition methods for interpolating nominal data. On the other hand, the experiment with NLP model in this paper serve as a primitive proof of concept for the use of NLP in interpolating categorical data – albeit with much more computational complexity.

## Funk-SVD

As mentioned in **METHODOLOGY**, Funk-SVD was trained on two separate datasets – “real images” and “real and synthesized images”, which resulted in two different set of composite matrices. The reconstructions of these two models are shown

Real images dataset

Map

Description automatically generated

# CHAPTER V DISCUSSION AND IMPLEMENTATIONS

The Discussion and Implementations Chapter presents all critical analysis of merits and shortcomings of your results, comparison with reported results in literature, and how you will carry on the incomplete or planned works in your thesis work.

# CHAPTER VI CONCLUSION AND RECOMMENDATION

The Conclusion Chapter summarizes your results, remarks, comments, suggestions and directions to be developed or improved in the future.

# REFERENCES

The References part indicates the sources of your information: books, brochures, catalogs and names and coordinates of people you consulted, if applied. List them in alphabetical order. Use scarcely a website as a reference because the information may not be scientific approved or be erroneous and the sites may disappear in the future. In the chapters refer an article or book by using author's name and date of publication, and put them in parentheses such as (Smith, 1990) or (Smith et al., 1990) if there are more than 2 authors or (Smith et al., 1990a) if there are different cited articles started with the same name. Use appropriate format for the bibliography as advised in the Endnote software.

# References

**There are no sources in the current document.**

|  |  |
| --- | --- |
| [1] | C. N. E. Anagnostopoulos, "License plate recognition: A brief tutorial," *IEEE Intelligent transportation systems magazine,* vol. 6, no. 1, pp. 59–67, Jan. 2014. |
| [2] | W. Jackson, *Learn Android App Development*, Apress, 2013. |
| [3] | K. Aida-zade, E. Mustafaev, and J. Hasanov, "Intelligent reading system based on mobile platform," in *International Conference on Problems of Cybernetics and Informatics*, Baku, Azerbaijan, 2012, pp. 1-4. |
| [4] | A. Mutholib, T. S. Gunawan, and M. Kartiwi, "Design and implementation of automatic number plate recognition on android platform," in *International Conference on Computer and Communication Engineering*, Kuala Lumpur, Malaysia, 2012, pp. 540–543. |
|  |  |

# APPENDICES

The Appendices part contains all related documents, materials and samples you obtained. Use this part wisely to alleviate the containers of the chapters. In other words, give a brief description or results in the chapters to make the reading flows well and use this part to describe in detail the issues for readers who desire to get in depth information. Organize the specific information in different appendices with a title such as Appendix 3: Specifications Sheet of integrated circuit 744. Refer them appropriately in the chapters using the brackets such as [Appendix 1].

1. Landsat naming convention: Landsat\_Sensor\_WRS Path/Row\_Acquisition time\_Processing time\_Collection\_Collection Category [↑](#footnote-ref-1)