CLIP-RS: A Cross-modal Remote Sensing Image Retrieval Based on CLIP, a Northern Virginia Case Study

Larissa Djoufack Basso

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science and Applications

Chang Tien Lu, Chair

Ing Ray Chen

Jin Hee Cho

May 6, 2022

Falls Church, Virginia

Keywords: Remote Sensing Image Retrieval, Textual input, Spatial Database Indexing and Retrieval, Contrastive Learning,Cross-modal

CLIP-RS: A Cross-modal Remote Sensing Image Retrieval Based on CLIP, Northern Virginia Case Study

Larissa Djoufack Basso

Abstract

Satellite imagery research used to be an expensive research topic for companies and organizations due to the limited data and compute resources. As the computing power and storage capacity grows exponentially, a large amount of aerial and satellite images are generated and analyzed everyday for various applications. Current technological advancement and extensive data collection by numerous Internet of Things (IOT) devices and platforms have amplified labeled natural images. Such data availability catalyzed the development and performance of current state-of-the-art image classification and cross-modal models. Despite the abundance of publicly available remote sensing images, very few remote sensing (RS) images are labeled and even fewer are multi-captioned.These scarcities limit the scope of fine tuned state of the art models to at most 38 classes, based on [1],one of the largest publicly available labeled RS data. Recent state-of-the art image-to-image

retrieval and detection models in RS have shown great results. Because the text-to-image retrieval of RS images is still emerging, it still faces some challenges in the retrieval of those images.These challenges are based on the inaccurate retrieval of image categories that were not present in the training dataset and the retrieval of images from descriptive input. Motivated by those shortcomings in current cross-modal remote sensing image retrieval, we proposed CLIP-RS, a cross-modal remote sensing image retrieval platform. Our proposed framework CLIP-RS is a framework that combines a fine-tuned implementation of a recent state of the art cross-modal and text-based image retrieval model, Contrastive Language Image Pre-training (CLIP) and FAISS (Facebook AI similarity search), a library for efficient similarity search. Our implementation is deployed on a Web App for inference task on text-to-image and image-to-image retrieval of RS images collected via the Mapbox GL JS API. We used the free tier option of the Mapbox GL JS API and took advantage of its raster tiles option to locate the retrieved results on a local map, a combination of the downloaded raster tiles. Other options offered on our platform are: image similarity search, locating an image in the map, view images' geocoordinates and addresses.In this work we also proposed two remote sensing fine-tuned models and conducted a comparative analysis of our proposed models with a different fine-tuned model as well as the zeroshot CLIP model on remote sensing data.

CLIP-RS: A Cross-modal Remote Sensing Image Retrieval Based on CLIP, Northern Virginia Case Study

Larissa Djoufack Basso

General Audience Abstract

Satellite imagery research used to be an expensive research topic for companies and organizations due to the limited data and compute resources. As the computing power and storage capacity grows exponentially, a large amount of aerial and satellite images are generated and analyzed everyday for various applications. Current technological advancement and extensive data collection by numerous Internet of Things (IOT) devices and platforms have amplified labeled natural images. Such data availability catalyzed the development and performance of current state-of-the-art image classification and cross-modal models. Despite the abundance of publicly available remote sensing images, very few remote sensing (RS) images are labeled and even fewer are multi-captioned.These scarcities limit the scope of fine tuned state of the art models to at most 38 classes, based on [1],one of the largest publicly available labeled RS data.Recent state-of-the art image-to-image retrieval and

detection models in RS have shown great results. Because the text-to-image retrieval of RS images is still emerging, it still faces some challenges in the retrieval of those images.These challenges are based on the inaccurate retrieval of image categories that were not present in the training dataset and the retrieval of images from descriptive input. Motivated by those shortcomings in current cross-modal remote sensing image retrieval, we proposed CLIP-RS, a cross-modal remote sensing image retrieval platform.Cross-modal retrieval focuses on data retrieval across different modalities and in the context of this work, we focus on textual and imagery modalities.Our proposed framework CLIP-RS is a framework that combines a fine-tuned implementation of a recent state of the art cross-modal and text-based image retrieval model, Contrastive Language Image Pre-training (CLIP) and FAISS (Facebook AI similarity search), a library for efficient similarity search. In deep learning, the concept of fine tuning consists of using weights from a different model or algorithm into a similar model with different domain specific application. Our implementation is deployed on a Web Application for inference task on text-to-image and image-to-image retrieval of RS images collected via the Mapbox GL JS API. We used the free tier option of the Mapbox GL JS API and took advantage of its raster tiles option to locate the retrieved results on a local map, a combination of the downloaded raster tiles. Other options offered on our platform are: image similarity search, locating an image in the map, view images' geocoordinates and addresses.In this work we also proposed two remote sensing fine-tuned models and conducted a comparative

analysis of our proposed models with a different fine-tuned model as well as the zeroshot CLIP model on remote sensing data.

Dedication

Dedicated to my family

Acknowledgments

I want to first thank God for all of his blessings.

I would like to thank the GEM fellowship for aiding in my pursuit of a graduate level education, its resources, guidance, and motivation. I would like to thank Virginia Tech CS department. I would also like to thank Dr. Chang Tien Lu for his support, guidance, and funding through my Master's degree. I would also like to thank the CS department for the various GTA positions, I greatly enjoyed being a GTA, and working with wonderful students.

I would like to thank Virginia Tech's NHGS scholar group, I learned a lot from each and everyone of them. I am grateful to have met fellow scholars as well as the admin team including Renee even if we only met online.

I would also like to thank Dr. Shernita Lee for her advice and guidance in my transition to Graduate School and Virginia

Tech.

This thesis is an expansion of the Spatial Database Management class project. I would like to thank my classmates and team members with whom I work with: Cameron Knight, and Sijia Wang.

Last but not least, I would like to thank my family and my close friends for their constant support. My family is my backbone. I am forever grateful for their support and continuous belief in me and for supporting my dreams. I wouldn't be where I am without them. I feel honored to be part of my family and I thank God everyday for surrounding me with such great support. Words cannot express my gratitude and love for my family.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The rapid development in technology, sensors, digital cameras, and smartphones are creating a vast data on a daily basis. Multimedia data, compared to other types of data, is growing in its share. This also facilitated the accessibility of labelled data that are extensively used in machine learning especially on text-to-image tasks. As the storage capacity of hardware increases, the cross-modal retrieval of related images becomes a challenging research question. An uncommon multimedia data that has increased exponentially in recent years and has spiked interest from various users and organization is remote sensing data. From the first satellite photos of earth taken in August 1959 by the unmanned aircraft Explorer 6 [2] to present time, the amount of satellite images has increased exponentially. Google Earth has also been a catalyst to numerous satellite based application because in June 11th, 2001, it provided the first publicly available earth dataset. On average, commercial satellite images collect 100 terabytes (TB) or more images per day [3]. For instance, Maxar a commercial satellite company **satellite** image library contains 90 petabytes of data over a 15 year span.[4]. The Global Satellite imagery industry is a multi-billion dollar industry. A report by Morgan Stanley predicts that by 2040, the global space industry could generate $ 1 trillion [5] . We are now facing large volumes of satellite images both labelled and unlabelled. Such available data have now raised concerns on efficient cross-modal storage, retrieval, and management of remote sensing images. The need for a cross modal fast, effective, and intelligent image analysis and retrieval that requires lesser manpower and resources is dire. A

solution to this problem is to conduct a text to image and image to image remote sensing image retrieval task. Retrieving satellite images with users' textual input can benefit various sectors looking to acquire the retrieved images based on descriptive query input in a timely manner.For instance, aiding the search and rescue team in geolocating an image based on a descriptive query text. For example, with aerial footage of a large and remote park, park rangers searching for a missing person driving a car with a specific color would enjoy geolocating such information in a timely fashion, a solution that is less resource dependent. Per the National weather service, it costs $178000 to conduct a search on 2,166 square miles using "everything from Coast Guard cutters and a HC-130J long-range surveillance aircraft to an HH-60 rescue helicopter while scouring the seas for survivors"[6]. Satellite images have become crucial resources to solve numerous problems in various fields, including cartography, intelligence and warfare, meteorology, remote sensing, artificial intelligence, etc. In recent years, the high collection and usage of Satellite Image leads to the high demand of high efficient and effective remote sensing image storage and retrieval. Satellite companies are now collecting more data than ever before. In recent years, the demand for applications that could detect objects in satellite images have increased significantly. However, there only exists a handful of labelled satellite image dataset as well as remote sensing captioning dataset. Of the available remote sensing captioned/multilabels dataset, very few are sparsed across multi object detection. Satellite imagery of each area of the planet earth is comprised of diverse and unique categories/objects. However, the few available labelled remote sensing dataset is not representative of the diverse earth categories. For instance, the majority of publicly available labelled dataset are those

of the US or Australia. Therefore, retrieving/detecting objects or categories that are not in the labelled dataset by fine-tuning state of the art classification is nearly impossible. For instance, it will be difficult to detect the Washington monument since those are not present in the publicly available dataset and the performance of image classification of residential areas will decrease in scenarios where the captured remote sensing images are comprised of round hut roof residential areas because the majority of residential areas building roofs are either squared or rectangular shapes. In view of the limitations posed by the lack of more representative and large labelled remote sensing images, our goal in this paper is to propose a cross-modal image retrieval and its geolocation that is not too dependent on the training data.Motivated by the aforementioned problems, we use a fine-tune version of a state of the art cross-modal image retrieval model in remote sensing (RS) images. CLIP, which stands for Contrastive Language Image Pre-training, is a recent state of the art neural network, trained on millions of images provides a robust models for domain specific applications. FAISS (Facebook AI similarity search) is a library for efficient similarity search will be used in our implementation as well.

1.2 Contributions

Our contribution are as follows:

- **We propose a text-to-image and image-to-image Remote sensing retrieval system that combines a fine tuned CLIP model and the FAISS library**: we mined high resolution raster tiles satellite images and, used a fine-tuned clip model with the FAISS library and a PostGIS

database to preprocess, store, and retrieve mined images. We also, provide a geolocation functionality that enables a user to elect to locate results on a map.

- **We conduct an analysis of fine tuned models**: We fine-tuned two clip models with VIT L14 and VIT B16 as base transformers for remote sensing image applications.We also conducted an analysis of our implemented models with a proposed fine tuned CLIP model and a zero shot CLIP [7] model on remote sensing data.

2 Related Work

Literature review on cross-modal image retrieval of remote sensing images can be divided in two sections. The first section will explore previous implementation of image-to-image retrieval and the second section will explore previous work on text-to-image retrieval

2.1 Image-to-image Retrieval

Unlike the text to image retrieval that is dependent on labelled data, the image-to-image retrieval uses distinctive image features from an input and compares those features against other images in the database to find the images that are most similar. Thus the concept of Content-Based Image Retrieval(CBIR) used in numerous image-to-image retrieval work [8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. However, before diving into previously proposed CBIR models, let's first explore some features used in CBIR and how they are extracted.

We divided this section into two main categories: Feature-based retrieval and Content-based image retrieval.

2.1.1 Feature-based retrieval

Color Features

Color features are among the first features that researchers focused on in image recognition problems. It is a low-level visual feature set based on the color distribution in an image. The color features are steady regardless of image transformation like rotation or resize. Histogram-related color features usually need higher computation cost while (DCD) performs better on region-based images. [18] presented a multi-scale distance coherence vector (MDCV) that can help retrieve images more accurately than traditional CBIR methods.

Texture Features

Texture features represent groups of pixels and they are more meaningful semantically than the color features. However, texture features are more sensitive to noises as they extract features based on pixels. The shape of the objects is another important factor on retrieval performance based on texture features.

Shape Features

Shape helps identify objects in images. Region and contour are the key components of shape features. Images with objects that tend to remain the same in size ratio, orientation are better retrieved with methods using shape features. E.g., trademark images retrieval[19].

Spatial Features

Spatial features focus on the location of objects within the image. Bag of Visual Words (BoVW) [20] is one of the most popular methods in removing spatial features and representing the image with a histogram. Further development on spatial features includes Pairs of Identical visual Words (PIW) [21] which represent a global spatial distribution of visual words, and a scale-invariant feature transform-based BoVW) [22].

2.1.2 Content-based Image Retrieval

Previously, papers have proposed solutions on content based image retrieval of remote sensing images. For instance, [23] proposes a content based image retrieval (CBIR) approach that extracts image features with the Speeded Up Robust Feature (SURF) feature extractor and represents them as bag of visual words using clustering and image indexing that are later used to retrieve similar images from the database using cosine similarity. [24] extracts features from image patches and uses an encoder to hash the extracted features and produces indexes that are later stored on a database for easy retrieval. In this hashing approach, it takes 0.4s to hash over 20 millions stored binary codes. The image storage and retrieval blueprint proposed in this paper is very similar to our proposed approach. We index the extracted image features with a similar library and store the indexes in a database for later retrieval.

[25] proposed an image retrieval approach which consists of combining color and shape features where both the cumulative histogram and 7 Hu invariant moments are used to calculate the color features and the shape features respectively. [26] Evaluates

and examines various aerial images feature extraction detection that have been used in the arena of object detection. They include the histogram of oriented images, local binary pattern feature, bag of word feature and sparse coding feature based.

2.2 Text to image retrieval

The text-to-image retrieval of RS images task can be subdivided in two main categories. The first covers the image captioning approach and the second covers the semantic alignment that maps two different modalities (text and images) into the same high dimensional space.

The image caption approach in RS image retrieval consists of creating captions for each of the image and storing them in a database and retrieving the images with captions that are very similar to the query text.[27] proposed a region-driven attention-based Long-short term memory (LSTM) sentence captioning approach.The domain-probabilities used here to caption the image are multiplied with the extracted image features. Those features are extracted with Convolutional Neural Network (CNN) and this approach's performance is evaluated with Bilingual Evaluation Understudy (BLEU) scores. [28] proposed one of the largest remote sensing image captioning dataset that is extensively being used in RS image retrieval work.[29] image captioning task uses a Variational Autoencoder (VAE) and an encoder–decoder architecture. Despite the great performance of image-to-text retrieval of RS via captioning, this approach still faces some issues with overfitting.

The semantic alignment task of RS image retrieval consists of projecting both the RS image and text into the same dimen-

sional space, mapping them, and measuring the cross-modal similarity.

[30] text alignment method consisted of obtaining text semantic representation via a visual semantic embedding that used an LSTM network and the visual feature representation of images were done with Convolution Neural Network. Wang et al[31] proposed a framework that uses semantic embedding as a measure for both the sentence and image representations. A proposed collective representation is used to improve the captioning performance. Qu et al [32] proposed a multimodal neural network model for semantic understanding of remote sensing images. Here, a natural sentence describing a given image is obtained via textual and visual information from RS images. Image features are extracted via a CNN and those features are combined to textual description of the images by RNN or LSTMs.This RNN-based approach wehre CNNs are used for feature extraction that are later used with textual description as input to RNNS has been extensively used in similar task [32, 28, 33].

3 Background

3.1 Models and Libraries used in our proposed solution

In this section, we will provide a brief overview and the motivation behind our usage of the CLIP model and the FAISS library.

3.1.1 The CLIP model

The main model used as an inspiration for this project was CLIP [7]. This is a highly generalizable multi-modal model trained on a large sample of data-mined text and image pairs using contrastive

loss. The model is able to use the zero-shot capabilities of similar GPT-2 and GPT-3 text models to classify images based on their similarities to prompts. The original CLIP model was able to match the performance of ResNet50, the backbone model for an implementation on ImageNet's nearly one and a half million samples without using any of the original labels in it's training data. This means the model is not confined to the original labels of the ImageNet classification task as most models that are trained are but can be given as free-formed text generated from a user or document to describe an image. CLIP was built on a very large dataset, 400 millions image-text pair data scrapped from the internet.

The CLIP model utilizes contrastive loss and while training the model, a method that has yielded high performance results such that the resulting latent-information vectors from both the text and image models are able to correlate when they are describing the same image. Such that, by using a cosine similarity metric on two vectors a similarity between a sentence and an image could be determined. This paper will take advantage of the quality of these vectors in order to find objects in satellite image data. CLIP becomes a natural extension for remote sensing type projects due to it's high generalizability since it's text model has concepts of how to interpret ideas in free text. Some of the ideas that it understands are locality based such as "Near", "Around", "Within", among others. This makes it a proper candidate when looking to vectorize an image containing all of its describable qualities. This means you are not required to conform to classical discrete classification for arial detections in remote sensing data. However, you can search for more descriptive objects. CLIP provides 9 model versions to include: RN50,

RN101, RN50x4, RN50x16,RN50x64, ViT-B/32,ViT-B/16, and ViT-L/14.For this paper we will be using the highest performing version of clip the vision transformer version(ViT) base patch 32.1 shows the robustness of CLIP in comparison to a state of the art CNN model over different dataset. Vision transformers are state-of-the-art computer vision techniques that take advantage of segmenting an image into chunks and using multi-head attention mechanisms to select and mask portions of images in order to select the most relevant pieces to the transformed embedding. This idea will become relevant later in this work when we decide how to pass the satellite image data into our model for the case study.

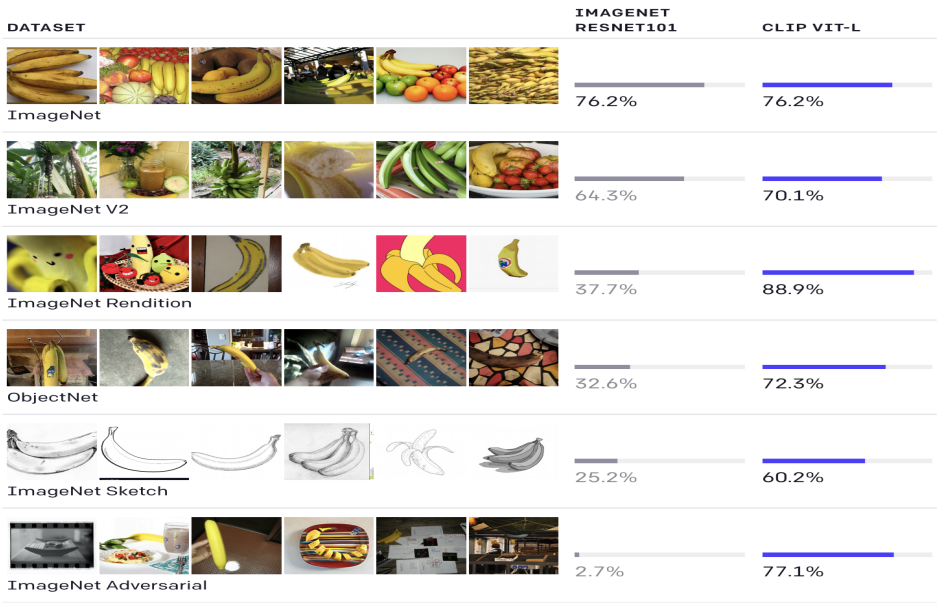


Figure 1: CLIP VIT comparison with Resnet

To fully understand this work, it is important to grasp CLIP's architecture shown in figure 2 with anotations in 3.1.1 The authors of CLIP used a collection of recent state of the art mod-

els. The Contrastive learning approach on the CLIP architecture is very similar to [34]. In this architecture, an image is encoded via an image encoder. The encoded image is then linearly projected to a contrastive embedding. As far as the corresponding image label, the label is first tokenized and encoded via a text encoder. The encoded text is passed to a contrastive embedding as well. The inner product of text to image is the contrastive matrix is taken. The values in blue represent the matching pair and they are closer to 1 while the other values highlighted in white tend to 0. The equations on the right side of the screen are used to compute the loss function. Here, v represents the image vector and u represent the text vector. τ, is the temperature coefficient. N represents the total length of the image batch. λ, the scalar weight, $\lambda \varepsilon [0,1]$

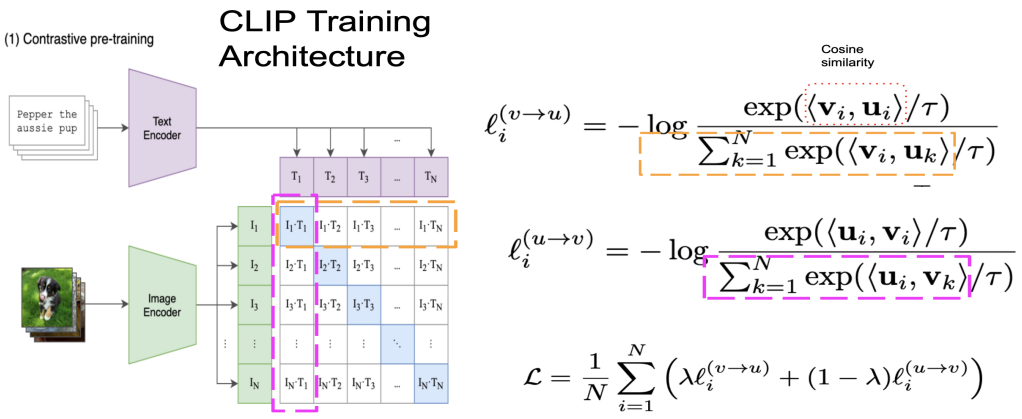


Figure 2: CLIP VIT Architecture

3.1.2 FAISS

The FAISS library is an approximate nearest neighbor (ANN) library that is able to perform a cosine similarity operation at scale across millions or even billions of high dimensional vectors [35]

v – d-dimensional vector representation of image

u – d-dimensional vector representation of text

$l_i^{v->u}$ – image-to-text contrastive loss function for the i-th pair

$l_i^{u->v}$ – text-to-image contrastive loss function for the i-th pair

$\mathcal{L}$ – training loss

$\mathcal{T}$ – temperature coefficient

λ – scalar weight $\lambda \in [0,1]$ scalar weight

such as the ones produced in CLIP. The process was originally defined in the work Billion-scale similarity search with GPUs. This work outlined a methodology for performing this highly scalable similarity search utilizing highly parallel architectures like the GPU. The fully built library which accompanies this paper is completed with a Python API. This can be used to interact with this indexing method from our tool. This tool has the ability to perform precise similarity searches but can also be used to perform approximate nearest neighbor algorithms faster with lower overhead. This model also can store a compressed representation of the high dimensional search vectors which can also be reconstructed with this tool. Consequently , storing computed vectors ends up being unnecessary.Faiss acceleration of exact search with exact distance calculation or approximate one with product quantization(PQ)method is an approach to fight the curse of dimensionality. Faiss also offers a pca option to reduce the number of dimentions in vectors. The different tools proposed by Faiss to mitigate the curse of dimensionality were factors that led us to our choice of the FAISS library in our project. In 2017 when the FAISS library was released, its GPU implementation option was "the fastest exact and approximate (compressed-domain) nearest neighbor search implementation for high-dimensional vectors, fastest Lloyd's k-means, and fastest small k-selection algorithm known" [35]. We chose to use the FAISS library because its provided functionalities could be useful in this project. The algorithmic novelty of FAISS in terms of time complexity is its improvement of an $O(nlog^2n)$ and parallelizes to a time complexity of $O(log^2n)$ [36]

4 Problem Statement and Methodology

4.1 Problem Statement

Cross-modal RS image retrieval system is a necessity for many commercial satellite companies in view of the big satellite data being collected by those companies on a daily basis. Some challenges observed on current remote sensing image retrieval systems are:

- Do not locate the results on a map
- Do not perform well in the text-to-image retrieval of unseen categories
- Do not accurately retrieve images from descriptive input query

It is important to note that the works that we have reviewed do not all face those challenges at once, each of those challenges are often observed on different proposed work. For instance, the retrieved data from the query input can be located on various places on earth. However, in Satellite image applications such as crisis management, a user might be interested and benefits from the geo-location information of the retrieved image. In a crisis management scenario with a search and rescue organization as a potential user, being able to locate the results of a cross-modal search can be a lifesaving, and a less resource dependent solution.

Further, in view of the limited availability of labelled and captioned remote sensing images, the need for a model that can detect objects/categories not present in the training data is crucial.

4.2 Methodology

In this section, we will discuss model fine tuning,image processing, and image retrieval.

5 Implementation

Our proposed implementation follows the architecture depicted in figure 4. The retrieved images via the Mapbox GL free tier library will be extracted using an algorithm, the images will be preprocessed and encoded into a 512 latent vector space from the fine-tuned clip model and will later be indexed with the FAISS library. The indexes as well as each image's metadata are stored in a PostGis library. The text/image retrieval aspect consist of preprocessing and encoding the input with the fine-tuned clip model and and retrieving the indexes of top k RS images similar to the import with the FAISS library. The metadata associated with those indexes are retrieved from the PostGis library and will be passed to the User interface.

5.1 Case-Study Data

We performed a case-study of the Arlington county and national mall region, in which we collected data covering nearly fifty square miles across the Northern Virginia region as shown in figure 5. The data comes from Mapbox [37] API which allows for up to 750,000 raster image retrievals per month. They curate up-to-date data from several sources in order to provide high quality satellite data to its users. We chose Mapbox API for its availability, consistency, and its front-end APIs which make it easy to access locally served, pre-processed data. The data was captured from the highest res-

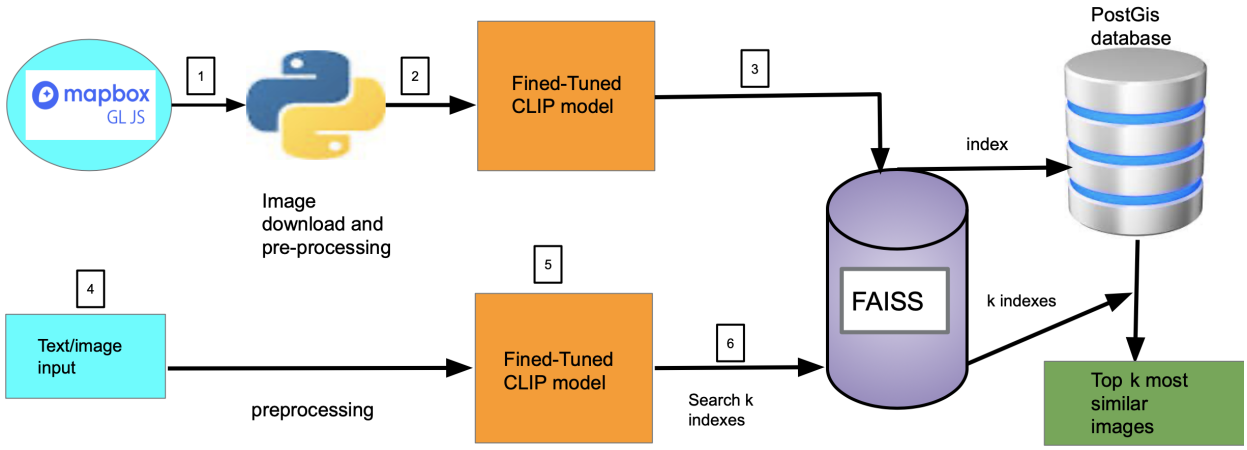


Figure 3: Model Implementation Overview

olution available as well as several other pre-compiled zoom levels to capture larger details in the map data. The data was saved at 10 different zoom levels 8 coming from the Mapbox API at zooms 10-18 and 2 artificially from sub-segmenting the high resolution image to a size acceptable for the CLIP model. The original and fine-tuned CLIP model is only able to embed 224x224 3 channel RGB images. Therefore, in order to embed the raster images from the map, we resized the images and normalized them before passing them into the model. The bounding boxes of the Northern Virginia Region are as follow:

$$bl = (38.81203297268714, -76.9862686605209)$$

$$tr = (38.912582115913104, -77.27389868847722)$$

Bl is a tuple that represents the bottom left latitude and longitude and tr is a tuple for the top right coordinates. To retrieve

those images, we need to first convert the latitude and longitude coordinates at each of the aforementioned bounding boxes from latitude and longitude to numbers. Equations 1 and 2 are input to our image download algorithm.

$$x = \lfloor (\frac{lon + 180}{360} \times 2^z) \rfloor \tag{1}$$

$$y = \lfloor (1 - \frac{\ln(\tan(lat \times \frac{\pi}{180}) + \frac{1}{\cos(lat \times \frac{\pi}{180})})}{\pi} \times 2^{z-1} \rfloor \tag{2}$$

x – tile number x goes from 0 (left edge is 180 °W) to $2^{zoom} - 1$ (right edge is 180 °E)

y – tile number, y goes from 0 (top edge is 85.0511 °N) to $2^{zoom} - 1$ (bottom edge is 85.0511 °S)

z – zoom level

lat – latitude in degrees

lon –longitude in degrees

Figure 4: Model Implementation Overview

z in equations 1 and 2 represents the zoom level.x in 1is the number representation of the longitude and y in 2 is that of the latitude.

5.1.1 Data pre-processing

Due to the nature of the problem in hand, it is a necessity to pass the model over the image and being aware of the sensitive segments of the image. To mitigate any bias with the filter's passage, a

$$\frac{1}{4}$$

Figure 5: An image of the Northern Virginia Bounding boxes used in the project.

step stride was chosen because the CLIP model has demonstrated a bias toward the center of the image due to its attention mechanisms as seen in Figure 7.

The striding mechanism to retrieve the raster tiled data worked as expected. The tiles were iterated over sequentially at each zoom level and in parallel over the different zoom levels. We use multi-threads to implement the parallel concept, due to the number of cores of the hardware being used, we had 8 zoom levels run in parallel at a given time. Within the file-system this data was stored in a folder structure with each zoom level located in its own folder; the latitude rows were stored in the same folder with the sub-folders holding the latitude images in png format. At a larger scale this process would need to run over a Hilbert curve to guarantee locality of data when running the process. Moreover, This will help to get the maximum speed possible and ensure a high fidelity raster data is consistently available locally. For this

Figure 6: A demonstration of how each zoom level works in the context of raster images.

implementation, a straight row / column method was used.

The hardwares used to perform this task are the Nvidia GeForce RTX 3090. The model was loaded into the graphics card, several image tiles were retrieved from the disk at a time and fed into the model for an average throughput of around 200 images per second depending on the size of the starting image. The size transform had the most impact on the performance of processing any given image. The dataset is from nearly 3 million resulting sub-images from striding around 50 square mile area of the studied area. This means for the 8 zoom and 2 finer zoom levels used to process this dataset, it took around an hour on this consumer grade hardware.However, on a less consumer grade hardware such as a computer with an Nvidia RTX 3070 graphic card it took us longer.We achieved speeds of processing about 1 square mile every 3 minutes which could be helpful in rapid instances of smaller areas of search and rescue. The resulting representation for the vector search took up approximately 8gb.

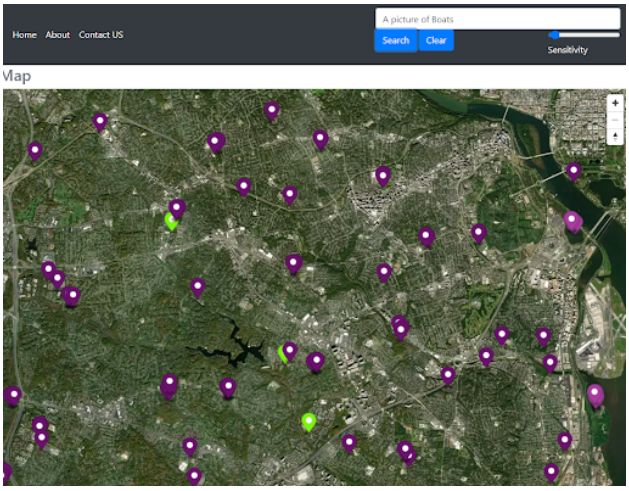


Figure 7: An example take from the user interface from with a search of a picture of boats with mild sensitivity and all points plotted on the map.

Indexing Setup

In the quest to store the resulted indexes into a system that will facilitate their quick retrieval based on the users' requests and with limited processing power and low system latency, we used FAISS. FAISS enabled us to embed representations of high dimensional data where nearest neighbor operations become possible at a very large scales even up to the billions of vectors.

CLIP's functionality of measuring cosine similarity distance between two vectors influenced our election of the inner product indexing method provided by FAISS. This operation was performed on the normalized vectors, that are ultimately congruent to the cosine similarity operation when performing a nearest neighbor operation.

Spatial Database

In the data-processing phase, for each processed image, we stored the following metadata in the database: box geometry object, it encompasses the NorthWest (NW), North East (NE), South West (SW), and South East (SE). Other images metadata stored in the database are the window specific latitude, and longitude coordinates, for the different windows in the map. Therefore, we are able to find images that overlap with a given envelope as well as perform additional indexing based on the image's location and perform spatial data operations. The database platform used was PostGIS. The structure consisted of a single table with the interpolated x,y and zoom for each processed image as well as the converted geo-coordinates and bounding boxes for each of the boxes. To insert the corresponding latitude and longitude to an image, we follow the equations 3 and 4.

$$Lat(y,z) = \frac{ATan(SinH(\pi * (1 - \frac{2y}{2^z})))}{180} \times \pi \tag{3}$$

$$Lon(x,z) = \frac{360 \times x}{2^z} - 180 \tag{4}$$

5.1.2 Text Search

Figure 9, represents the retrieval architecture. The query is transformed into a vector used within the search and retrieval system. To retrieve the data this system takes advantage of the text model from the fine-tuned CLIP. The user's input text from the web-app is embedded into the same latent semantic space as the images in the map. The FAISS library is then used to perform an approx-

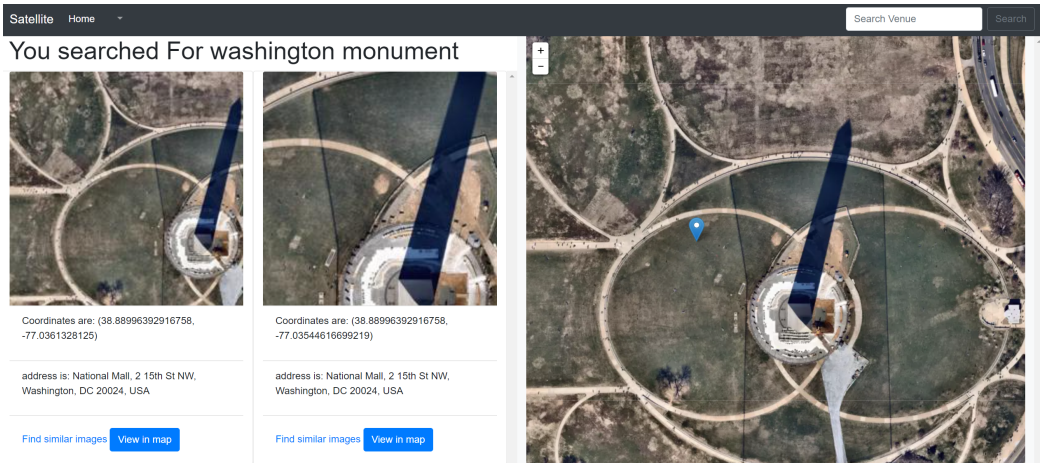


Figure 8: User interface for the Washington monument query. The left side shows the retrieved images from the query text.On each image, labels geo-coordinates and addresses are provided. The right side pinpoints one of the selected images retrieved on the map

imate nearest neighbor operation on the vector to find the most similar k-nearest vectors to the query term.

The output index is used to find relevant rows in the PostGIS table that holds all the vector data for this experiment. The rows hold information on the zoom-level, interpolated x/y coordinates of the image as well as specific latitude and longitude information of the detected image. This information is provided to the front-end. We also used the retrieved geocoordinates to retrieve address by reverse geocoding via the Google map API . The front end then displays the detected images as well as the image's coordinates, address (obtained by reverse geocooding the latitude and longitude), and any other relevant information.

With a garbage or unsanitized raw-text, our system still retrieves some results. If the input text is malformed or irrelevant, the semantic vector will be garbage and thus the FAISS system will

retrieve the nearest neighbors of the malformed input and often return vectors closest to zero with little or no semantic content.

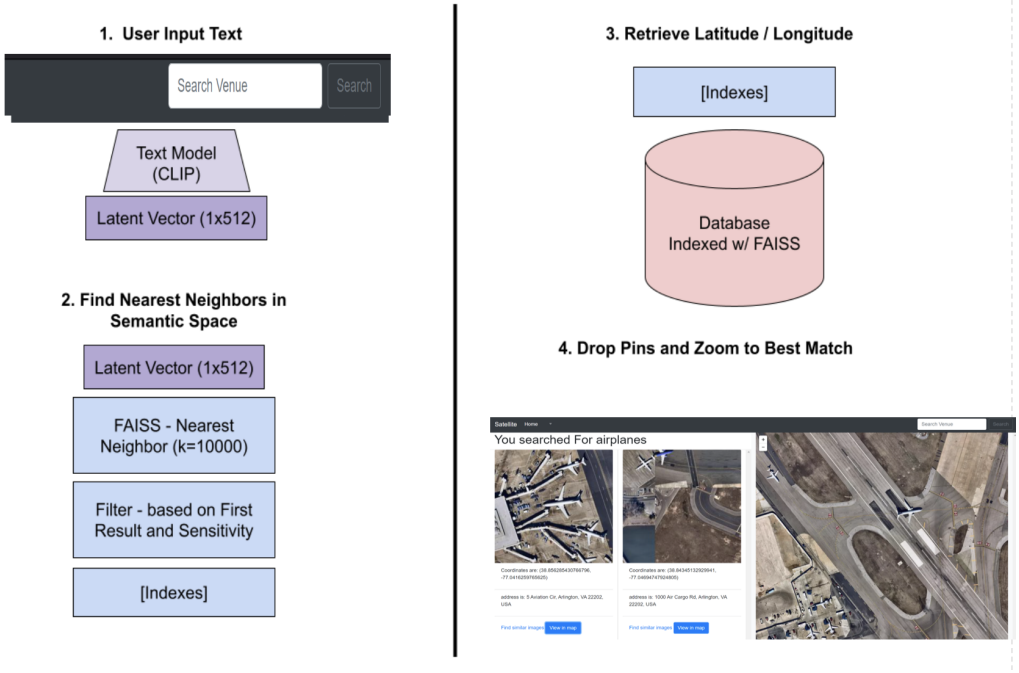


Figure 9: System's retrieval architecture.

5.1.3 Image Search

The image search functionality is implemented in this work via the "find similar iamges" hyperlink in the front-end.To retrieve images similar to the ones displayed in the UI, the retrieval process is the same as the text-to-image retrieval. The image is passed as input to the clip model that encodes the image to a 512 latent vector. The Faiss library is used to retrieve the top k indexes of the images similar to the query image. The metadata of those images are sent back to the front end.

5.1.4 Model Fine-Tuning

Training Dataset

The dataset used in the fine tuned model is a collection of three different labelled datasets.Our attraction to those images was based on the numerous captions available for each image.

The first dataset that we used is the Remote Sensing Image Caption(RSCID) data by [38]. It is a collection of 10000 224x224 RGB images containing upto 5 captions per images. The data is a collection of images retrieve via sources such as Google Earth, Baidu Map, MapABC, and Tianditu.

The second dataset used is the Sydney dataset, a dataset with image resolution of 0.5m of 500x500 RGB images divided into 7 classes with 613 images per classes. These images are from Sydney Australia that were retrieved via the Google Earth API.

The third dataset used is the UCM dataset by [39], a dataset based on the UC Merced Land Use Data. It is subdivided into 21 classes with 100 256x256 RGB images and each image is provided with 5 captions with pixel resolution of 0.3048m. The UCM dataset is a collection of images from the United States Geological Survey (USGS) National Map Urban Area Imagery. The 21 classes can be visualized in figure 13. Figure 10 is a representation of those three datasets [40].

Fine Tuned Model

The fine tuned model that we replicated used Flax/JAX on TPU-v3-8. Even though the Flax/JAX models could be trained on both GPU and CPU, we realized that it was faster to train this model on a TPU. The batch size used on TPU was 1024 with

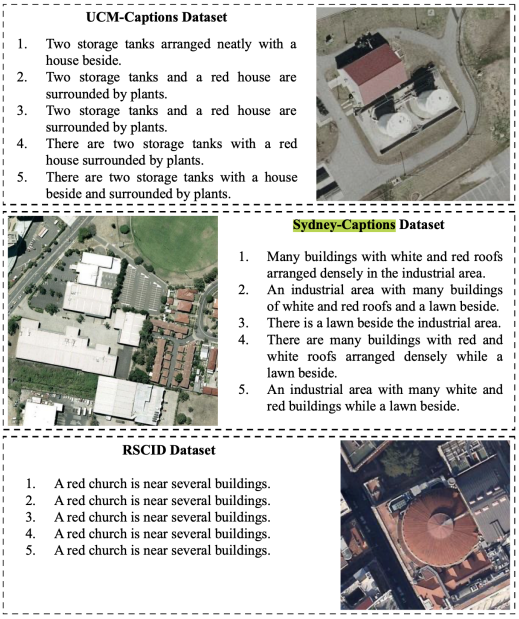


Figure 10: Remote Sensing Captioned Dataset. The top image the image-caption pair from the UCM Dataset. The middle picture is the image-caption pair from the Sydney dataset. The last one is from the RSCID dataset

128 for each TPU device. The model best performed with Best training results were the Adafactor and Adam optimizers and a learning rate of 5e-5 and a linear learning rate schedule. Despite the sizeable amount of data we used as previously described, it was not enough and to avoid overfitting and for data regularization, data augmentation was performed on both the images and their corresponding captions. The model proposed by the authors of clip-rscid [41] was based on VIT B32. However, we did also fine tuned two models with base models VIT L14 and VIT B16 and we realized that the VIT B32 model had a higher performance that our models. We will discuss the results in the experiment section.

The image augmentation consisted of transformations such as color jittering, randomly cropping images, randomly flipping

images vertically and horizontally, and randomly resizing and cropping images. A sample image augmentation from the previously described procedure is shown in 14 The text augmentation was done via backtranslation and in this scenario, the Marian MT family of ROMAIN translation models from Helsinki-NLP were used. Here, each augmentation corresponded to backtranslation through a different pair of ROMAN language models. The abbreviations of the various roman languages covered in the Helsinki-NLP romance model are: fr, fr_BE,
$fr\_CA, fr\_FR, wa, frp, oc, ca, rm, lld, fur, lij, lmo, es, es\_AR$,
$es\_CL, es\_CO, es\_CR, es\_DO, es\_EC, es\_ES, es\_GT, es\_HN$,
$es\_MX, es\_NI, es\_PA, es\_PE, es\_PR, es\_SV, es\_UY, es\_VE$,
$pt, pt\_br, pt\_BR, pt\_PT, gl, lad, an, mwl, it, it\_IT, co, nap, scn$,
vec, sc, ro, la [HugginFace]

A sample text augmentation from the previously described procedure in French is shown in figure 15 The fine tuned clip model we replicated was built by a team that competed on the Flax/JAX community week, a competition organised by hugging face and Google cloud and the goal was to fine-tune the openai CLIP model with RSCID. The link to their Github repo is:https:github.com/arampacha/CLIP−rsicd. In earlier implementation of this projects, we fine-tuned the CLIP model ourselves using both the DOTA dataset and Patternnet dataset. The performance of the numerous models we built/fine-tuned was significantly lesser than that of the ones we replicated. Our own fine tuned models can be provided upon request. The main concept with clip is that in the training portion, the image and text batches are all projected in the same dimensions and the multiplication of those favors images that are more similar and defavor those that are not. The cosine similarity of similar images increases, the im-

ages that map to their respective labels perfectly are those that intersect at the diagonal as shown in figure 11. To evaluate this CLIP implementation, CLIP encodes each image with CLIP's embedded representation of each of thirty caption sentences. Those sentences are of the form "An aerial photograph of {category}" mentioned in the CLIP-rscid repository. The architecture is represented in 12.

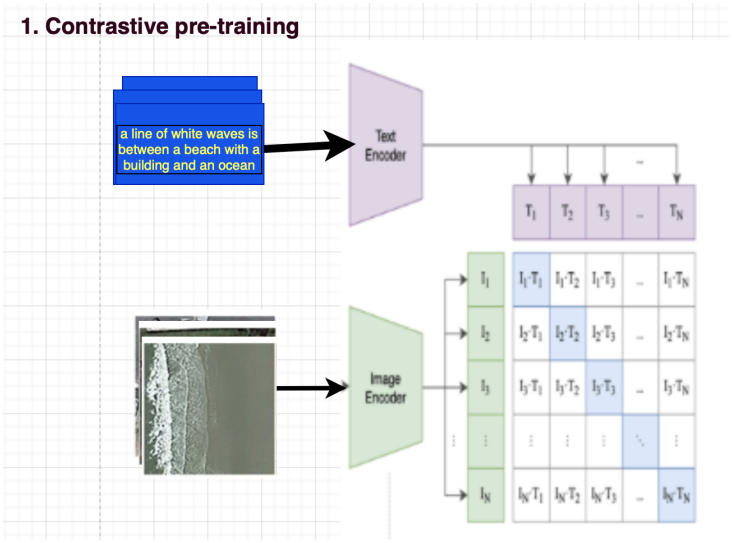


Figure 11: ClIP-RSCID pre-training

5.1.5 User Interface

We built a user-friendly website application to visualize the system.

Mapbox

Mapbox GL JavaScript API provides raster image tiles functionalities and is prominently used in similar projects. In order to perform this case study, we retrieved data between the Latitude

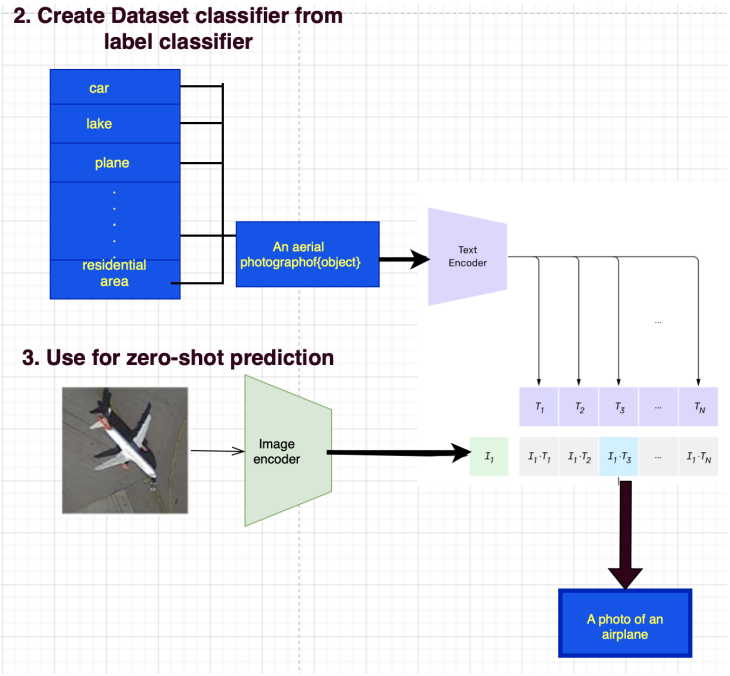


Figure 12: CLIP-RSCID Classification architecture

38.812 and 38.912 and between the Longitude -76.985 and -77.274. The data was captured at the zoom levels as defined by OpenStreetView between 10 and 18. Mapbox's API was used to retrieve all 30,000 images at the different zoom levels at double resolution, being 512 by 512. The Mapbox GL JavaScript API stores the local zoom levels in a quad tree structure. These images were saved in a png format into a filesystem structure defined by "zoom/x/y.png". This structure is used by the Mapbox GL JavaScript API library in order to retrieve images for specific geo-coordinates. The numbers can be transformed into latitude and longitude with the previously defined equations.

These raster images are required for use with the models as defined in the previous section. The images were saved off in an

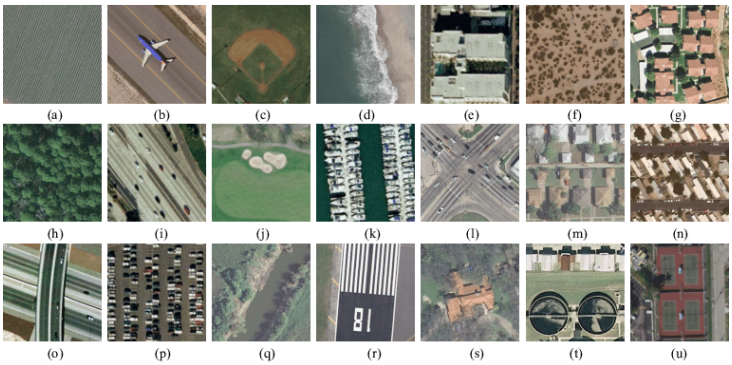


Figure 13: The 21 classes of the UCM dataset: (a)agricultural, (b) airplane, (c) baseballdiamond, (d) beach, (e) buildings, (f) chaparral, (g) denseresidential, (h) forest, (i) freeway, (j) golfcourse, (k) harbor, (l) intersection, (m) mediumresidential, (n) mobilehomepark, (o) overpass, (p) parkinglot, (q) river, (r) runway, (s) sparseresidential, (t) storagetanks and (u) tenniscourt.

appropriate structure where at each zoom level, the tiled raster images would create a seamless image when tiles sequentially in order by their x,y values. These x,y integer values also correspond to the images latitional and longitudinal coordinates and can be calculated for any specific zoom level. This capability will be used when processing and storing the data. Additionally, information can be retrieved from other APIs about specific locations on the map.

Routes Implemented

An example of the interface shown in figure 7. The implementation for this interface takes advantage of two routes. First, there is a route required to get the information required by the Mapbox GL JavaScript API library in the recognizable form "zoom/x/y.png". This route is specialized to also serve the stride based data required to run the inference/ data-loading portion of this project.

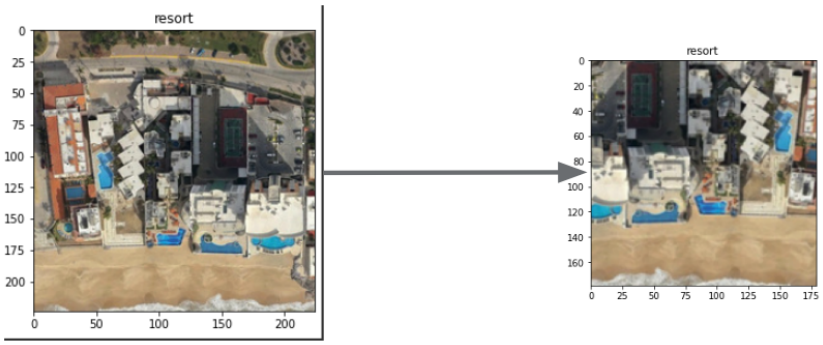


Figure 14: Figure showcases image augmentation. The left image is the original image and the right image is augmented

	image_id	sentence	count	languages_to_back_translate_with	back_translated_sentence
740	928	the sea water is so transparent that it looks ...	2	fr	seawater is so transparent that it looks like ...
741	584	a lot of cars parked on the side of the land .	3	fr	many cars parked on the side of the earth.
742	938	a piece of green ocean is near a yellow beach .	5	fr	a piece of green ocean is near a yellow beach.
743	570	some plants are near a piece of khaki bareland .	2	fr	some plants are near a piece of kaki nueland.
744	599	the bare land has a small patch of water .	3	fr	The bare earth has a small piece of water.

Figure 15: Figure showcases caption augmentation in French. The left side is the original caption and the right caption is backtranslated

This is required for when the user clicks on a pin and needs to be served the source image for the detection. With this method we can use this route to return the exact image from the detection by splicing together images coming from separate raster tiles in real time before serving it to the user. Therefore, this route can be used to retrieve any square coordinate tile from our data-set.

A second route that needed to be implemented was the route to perform the search retrieval. This is implemented as a get and return a JSON packet with information for each of the hits in the system in order of confidence. The system utilizes the

first result as the one to zoom in on. The system could possibly be expanded to use this information to either zoom to other results in order of occurrence, scale the pins dropped based on confidence, or perhaps color code this information to communicate the confidence of the detection to the user thoroughly.

6 Experiment Setup and Results

This section will conduct a thorough analysis of proposed fine-tuned models as well as the performance of domain specific zero shot prediction of the CLIP model.

In an earlier section, we discussed image augmentations methods used in the fine tuned models. Let's first analyze the importance of those augmentations strategies in minimizing overfitting and for regularization. In figure 16, we observe that the model without the image augmentation implementation performs better than the model with image augmentation during training. However, we observe a different scenario during evaluation. The same model that performed better than the model with image augmentation is worst than the model with image augmentation. We observe that its evaluation loss function keeps on going up. This might be due to the presencence of outliers or model overfitting.As such, the model with image augmentation is less prone to overfitting than the model without image augmentation. This conclusion is based on the data that we are currently using. It is however important to note that with a different data, we might have the same or different scenario. As such, our conclusion of this section might vary. Similarly, 17 compares two different models. Model1 is the model with both text and image augmentation implemented and model 2 is the model with image augmentation

only. Model 2 performs a little better in training than model 1 but model 1 performs better than model 2. In the aim of reducing model overfitting, model1 is the better option. As a result, the model with both text and image augmentation is less prone to overfitting than the others, thus the importance of image and text augmentation in this text-image pair task.

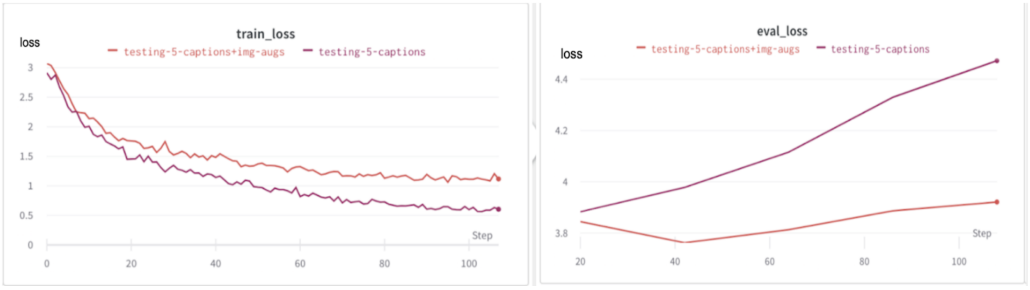


Figure 16: Loss function of models with image augmentation and without image augmentation

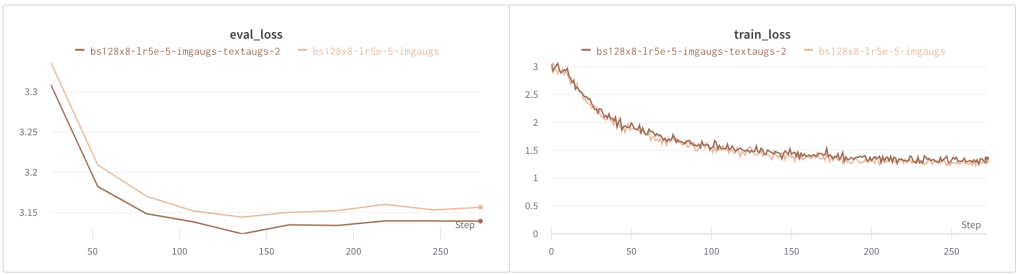


Figure 17: Loss function of models with image text and augmentationa and with image augmentation only

To setup this experiment, a portion of the RSCID data was used as evaluation set. The encoding of each image and a version of CLIP model was compared against a CLIP version encoding of 30 artificial categories that are passed through the root sentence "An aerial photograph of category". This setup can be visualized in 12.In this comparison, the image and text encoding that

are most similar will have a higher score than those that do not. In other words, each image in the test dataset will be compared against the 30 synthetic phrases. The 30 categories used are: "stadium,railwaystation,mediumresidential,sparseresidential,viaduct, center,airport,playground,church,square,beach,school, mountain ,bareland,resort,baseballfield,denseresidential,river,forest,desert, bridge,storagetanks,port,commercial,parking, industrial,meadow, park,pond,farmland". An image can have many captions. Here, k scores levels were defined. The different values of k are 1, 3, 5, and 10. The test set contains labels/captions associated with each images. To evaluate the performance of either CLIP version, the top k categories that match a specific image was evaluated against the actual image label. A score of 1 signifies that the top-k score categories were all a subset of the image's original labels.A score of 0 will mean the opposite. Because, there isn't always perfect score, the scores are averaged.

Evaluation metrics To evaluate the performance of our models with the aforementioned approach, we used various multilabel evaluation metrics. In the following metric definitions and equations, let D represent the evaluation dataset , L represent the total number of labels, Z_i the set of predicted labels and Y_i represent the actual labels.

- **Accuracy** determine the potion of correctly predicted labels. $Accuracy(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \bigcap Z_i|}{|Y_i \bigcup Z_i|}$ [42]
- **Precision** refers to the portion of predicted labels that are truly positive.
  $Precision = \frac{TruePositive}{Truepositive+FalsePositive}$
  The paper [42] uses the following equation to calculate the precision in a multilabel classification task

$Precision(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \bigcap Z_i|}{|Z_i|}$

- **Recall** calculates the number of the actual positives that have been predicted as positive by the model. An ideal recall score also should be near one for a good classifier. In fact, Recall is a metric that indicates how well a model can classify relevant data. It's also known as True Positive Rate or Sensitivity.
  $Recall = \frac{TruePositive+TrueNegatives}{Truepositive+FalsePositive+TrueNegative+FalseNegative}$

  An extension of the previous recall function in multilabel classification class is $Recall(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \bigcap Z_i|}{|Y_i|}$ [42]

- **F1-score** is a harmonic mean of Precision and Recall, and it provides a more accurate picture of cases that were wrongly classified than the Accuracy Metric. F1-score might be a better measure to use if for an uneven class distribution.
  $F1 = \frac{2*(Precision*Recall)}{Precision+Recall}$

- **Hamming Loss** helps determine the portion of incorrectly predicted labels.
  $HammingLoss(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \Delta Z_i|}{|L|}$ [42]

For the precision, recall, and f1 score, we used the macro average.

Figure 1 shows the performance of the different clip versions and their averaged score. The fine-tuned clip model, clip-rscid-v2 significantly outperforms the zeroshot CLIP model at all the different k scores. Further, while looking at all the other metrics at k=1, we observe that CLIP-rscid-v2 outperforms the zero-shot CLIP model.

Model Name	Metrics(%)	k=1	k=3	k=5	k=10
zero-shot CLIP	Accuracy	57.546251	74.294060	82.181110	93.086660
	Precision	54.355713	28.850909	22.256283	14.469290
	Recall	56.470518	74.090959	82.561763	93.325388
	F1 score	50.659561	39.564359	33.171769	23.744307
	Hamming Loss	2.830250	8.380396	14.521259	30.460889
CLIP-rscid-v2	Accuracy	**88.3**	**96.8**	**98.2**	**99.8**
	Precision	**86.625**	**45.53**	**31.731**	**16.39**
	Recall	**84.137**	**95.366**	**98.252**	**99.417**
	F1 score	**83.306**	**57.737**	**44.277**	**26.191**
	Hamming Loss	**1.045**	**6.959**	**13.444**	**30.039**

Table 1: Model accuracy performance of ClIP-RSICD and zeroshot CLIP in RS application

The original CLIP paper demonstrated the robustness of CLIP and its adaptability to domain specific applications. This can be seen here, by looking at the top 3,5,and 10 scores. The model's performance is very significant and for the top 10 categories, it shows a near 94% on a domain specific application.

In this experiment, we see that the fine tune clip model, clip-rscid-v2 a near above 97 % accuracy on 3 or more categories. This is a very important result in this work because, our goal was to be able to retrieve image based on descriptive input. Descriptive input are often comprised of multi-labels.

In this paper, we also fine tuned two different clip models to the remote sensing image domain. The finetuned models that we implemented were based on both VITL14 and VITB16 transformers.Table 2 evaluates the performance of the three fine tuned clip models under the lens of various evaluation metrics. The fine tuned model with vit-base-patch 32 outperforms the other models in view of the evaluation scores. For all the models, we observe

that the hamming loss score increases with higher values of k which means that with higher values of k the number of wrongly predicted labels increases. However, at k=1 we observe that the precision, recall, and f1 scores are high and the hamming loss is very low. Meaning that the models perform best with lower values of k especially for k=1 and has higher precision and recall values especially the precision values. The high models performance with k=1 for evaluation metrics other than the accuracy score is normal because, as shown in Table 3, for each test case, we only have one actual label. Therefore, with k values greater than 1, the number of incorrectly predicted labels increases thus affecting the precision, F1, and hamming loss scores. Furthermore, we calculated the accuracy for each of the different models by counting the number of times an actual label appeared in the first k labels with the highest predictions and we divided that number by the total number of samples.This explains the reason why the accuracy score increases with the increase of k. We can thus conclude that the high precision,recall, F1 scores, and the low Hamming loss score for k=1 of the models especially the model with based transformers vit-base patch 32 is indicative of a good model and classifier.

Figure 18 helps visualize the result and showcases several examples of retrieved images as well as their corresponding text query. In the similar way, figure 19 also showcases the retrieved images and their corresponding text query. However, the focus was more towards an image retrieval from very descriptive input. This is one of the reasons behind our motivation to use a fine tuned clip model in this work. In Figure 18, one of the examples is a picture of the washington monument, a categorie or an image that is not part of the training dataset used in this work. At the

time this work is written, of the publicly available remote sensing captioning dataset, the washington monument is not part of those data. In this work, our aim was to detect unseen objects or objects that were not part of the training data. We also wanted to be able to retrieved images based on descriptive query as shown in both 18 and 19.

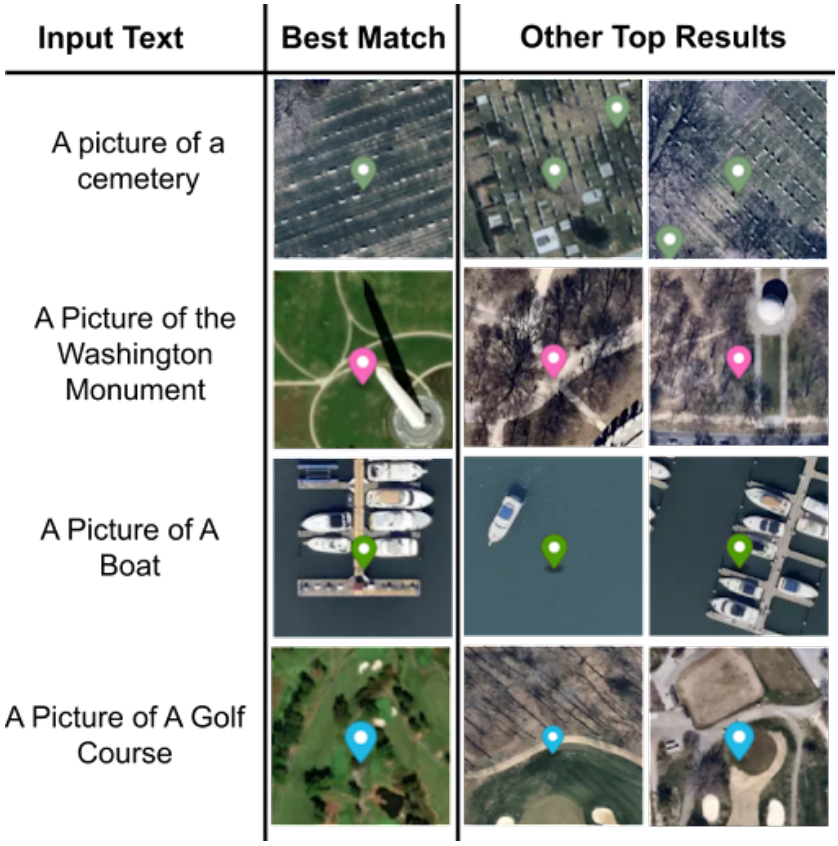


Figure 18: Retrieved image examples with their text query

Base Transformers	Metrics(%)	k=1	k=3	k=5	k=10
VIT-L-14	Accuracy	58.423	81.889	87.244	96.982
	Precision	59.914	37.016	27.322	16.69
	Recall	57.404	81.463	87.212	97.071
	F1 score	54.055	47.662	38.525	26.398
	Hamming loss	2.772	7.874	14.183	30.201
VIT-Base-patch 16	Accuracy	62.317	81.110	87.829	96.397
	Precision	61.656	32.891	24.776	14.279
	Recall	60.751	80.301	87.27	96.102
	F1 score	57.457	43.923	36.245	23.690
	Hamming loss	2.512	7.926	14.145	30.240
VIT-Base-patch 32	Accuracy	**88.3**	**96.8**	**98.2**	**99.8**
	Precision	**86.625**	**45.53**	**31.731**	**16.39**
	Recall	**84.137**	**95.366**	**98.252**	**99.417**
	F1 score	**83.306**	**57.737**	**44.277**	**26.191**
	Hamming loss	**1.045**	**6.959**	**13.444**	**30.039**

Table 2: Performance metrics of three fine tuned CLIP models with different based transformers.

Actual Label	**Predicted labels**
airport	'airport', 'parking', 'bareland', 'industrial', 'desert', 'railwaystation', 'meadow', '
bareland	'bareland', 'desert', 'mountain', 'storagetanks', 'sparseresidential', 'denseresident
airport	'airport', 'meadow', 'parking', 'resort', 'bareland', 'railwaystation', 'commercial'
bridge	'bridge', 'pond', 'viaduct', 'river', 'bareland', 'port', 'parking', 'park', 'desert', 'b
industrial	'industrial', 'commercial', 'denseresidential', 'mediumresidential', 'parking', 'spa
playground	'stadium', 'bareland', 'playground', 'meadow', 'parking', 'park', 'pond', 'center',

Table 3: A sample of the actual and predicted labels obtained from the evaluation dataset and a model's prediction

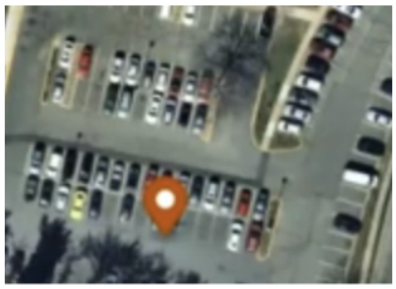
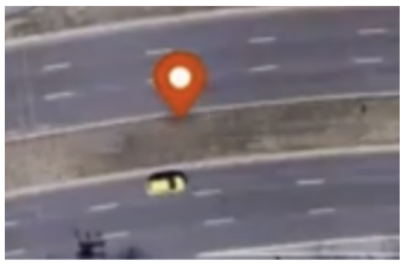
Input Text	Matches
A picture of cars	
A picture of a red car	
A picture of a yellow car	
A picture of an airplane on a runway	
A picture of a residential areas with trees	

Figure 19: Retrieved image examples from descriptive text querries

7 Future Work

There is still room for improvement of this work and the following are some suggested improvements.

1. This model could be used as a root-model for more complex detection schemes used to perform land and damage surveys.

2. This model performs well in detecting solar panels on roofs and could possibly be extended to query other domain specific tasks with zero-shot queries or by building a classifier with a relatively limited set of training data. We showed promising results for zero-shot queries, extending it to a zero-shot or few-shot datasets to evaluate our model.

3. Fine-tuning the CLIP model on more descriptive RS data with quantitative captions of the various objects in the image will be beneficial and will help overcome some of the current limitations of both the original and the fine-tuned CLIP model.

4. This work can also extend to Real-time object detection on satellite images for rescue purposes.

5. For the ethical concerns with regards to the potential and capability of this software on satellite images, we need to add more information to urge the user to follow related regulations. Before making this project available to the public, we can either wipe out sensitive information that might be collected here or we could limit access to the user we approve by enabling account login.

6. Use GAN-based models for cloud detection and removal of

mined images. Even though we did not observe clouds from our visual inspection of a portion of our mined data, cloud presence in satellite image data is common. In future work, we plan to use GAN-based models for cloud detection and removal on satellite images.

8 Conclusion

In this work, we proposed a work that facilitated a cross-modal RS image retrieval. Some of the problems we aimed to solve were to build a text-to-image and image-to-image retrieval system that focused on the retrieval of images especially those based on descriptive query text and query inputs whose classes and categories are beyond those of the training data used during the model's fine-tuning process.Figures 18 and 19 showcases retrieved images from both descriptive inputs such as images of red, yellow cars,and residential areas with trees and categories not present in the training dataset such as the image retrieval of the Washington monument we dropped pins in the locations where those images are located on a map. We also used the FAISS library for its suitability for this work as far as image retrieval speed and curse of dimensionality reduction and the ability to handle large high dimensional vectors.Furthermore, in the related work, we covered various image to image and text to image retrieval models. The text-to image retrieval approaches only detected the objects that were from the training dataset.We chose to use CLIP because it was trained on 400 million diverse captioned data and in view of its robustness in domain specific applications, we believed that using a fine-tuned clip model will help us detect images from descriptive inputs and inputs that have not been seen by the model during training.

Besides displaying the retrieved images from the user's search on the user interface, we also provided options where the user could locate their image of interest from the ones display on the web application on the map and find similar images. On our web application, we also provided other detailed information such as the geo-coordinates of each images and we gathered each image location by fully taking advantage of the reverse geocoordinate API from google. To build up the proposed system, we broke the implementation into 3 different parts. We first downloaded satellite images using the Mapbox GL JS API and we limited our image download to the Northern Virginia area. The downloaded images, were downloaded in a raster tile structure file in a zoom/x/y/image.jpeg format. The downloaded images were preprocessed and embedded to a 512 latent vector space per image with the fine-tuned CLIP model that we replicated. We then used the FAISS library to generate indexes for those images and the images as long as metadata relevant to the images were stored on a PostGis database.

Upon a retrieval request either via the search bar or the image similarity search, the input(image/text) are then encoded by the fine-tuned CLIP model and the FAISS library searches for the top k(in this scenario we retrieved 5) nearest neighbor vectors and provides the indexes of those vectors. We then used those indexes to retrieve the image metadata attached to each of them. The image metadata are then sent to the front end.We also fine tuned two CLIP models with base transformers VIT B32 and VIT L14. The fine-tuned clip model proposed by the authors of CLIP-rsicd outperformed our models and showed an accuracy of nearly 97% on the top-3 categories and greater.This shows that it performs well on multi-class input queries.Nonetheless our implemented model

showed great performance and outperformed the zeroshot CLIP model.

References

[1] Weixun Zhou, Shawn D. Newsam, Congmin Li, and Zhenfeng Shao. Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *CoRR*, abs/1706.03424, 2017.

[2] Explorer 6: 60 years since first earth photo from space. *BBC News*, August 2019.

[3] Doug Mohney. Terabytes from space: Satellite imaging fills data centers. *Data Center Frontier*, Apr 2020.

[4] Fun facts about digitalglobe satellites. *Maxar Technologies*, September 2016.

[5] Space: Investing in the final frontier. *Morgan Stanley*, July 2020.

[6] NOAA US Department of Commerce. cost for mayday hoax.

[7] Federico Bianchi, Giuseppe Attanasio, Raphael Pisoni, Silvia Terragni, Gabriele Sarti, and Sri Lakshmi. Contrastive language-image pre-training for the italian language. *CoRR*, abs/2108.08688, 2021.

[8] Yelisetty Srivarsha and V. M. Manikandan. A novel content-based image retrieval scheme based on textual information. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–4, 2021.

[9] Mathivanan. P, Kokilambal S, Snehashri. V, and Swetha. A. Intelligent content based image retrieval model using adadelta

optimized residual network. In *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5, 2021.

[10] Nehal M. Varma and Arshi Riyaz. Content retrieval using hybrid feature extraction from query image. In *2018 International Conference on Information , Communication, Engineering and Technology (ICICET)*, pages 1–4, 2018.

[11] S. Sahaya Sujithra Mary, Sasithradevi A, S. Mohamed Mansoor Roomi, and J. Jebas Immanuvel. A random vector functional link network based content based image retrieval. In *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, volume 1, pages 486–492, 2019.

[12] Baljit Kaur Saini and S. D. Sawarkar. Visual entity identification using content based image retrieval technique. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–4, 2018.

[13] Nehal M. Varma and Akanksha Mathur. A survey on evaluation of similarity measures for content-based image retrieval using hybrid features. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pages 557–562, 2020.

[14] Khawaja Tehseen Ahmed, Syed Ali Haider Naqvi, Amjad Rehman, and Tanzila Saba. Convolution, approximation and spatial information based object and color signatures for content based image retrieval. In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pages 1–6, 2019.

[15] K Karthik and S Sowmya Kamath. A hybrid feature modeling approach for content-based medical image retrieval. In *2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS)*, pages 7–12, 2018.

[16] Ahmad Raza, Hassan Dawood, Hussain Dawood, Sidra Shabbir, Rubab Mehboob, and Ameen Banjar. Correlated primary visual texton histogram features for content base image retrieval. *IEEE Access*, 6:46595–46616, 2018.

[17] Ayesha Khan, Ali Javed, Muhammad Tariq Mahmood, Muhammad Hamza Arif Khan, and Ik Hyun Lee. Directional magnitude local hexadecimal patterns: A novel texture feature descriptor for content-based image retrieval. *IEEE Access*, 9:135608–135629, 2021.

[18] Xiupeng L. Jiexian, Z. and F. Yu. Multiscale distance coherence vector algorithm for content-based image retrieval. 2014.

[19] Zhiling Hong and Qingshan Jiang. Hybrid content-based trademark retrieval using region and contour features. In *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, pages 1163–1168, 2008.

[20] Bajwa K.B. Sablatnig R. Ali, N. A novel image retrieval based on visual words integration of sift and surf. 2016.

[21] Barat C. Muselet D. Khan, R. and C. Ducottet. Spatial orientations of visual word pairs to improve bag-of-visual-words model. in proceedings of the british machine vision conference (pp. 89-1). September 2012.

[22] Zambanini S. Anwar, H. and M. Kampel. A rotation-invariant bag of visual words model for symbols based ancient coin classification.in 2014 ieee international conference on image processing (icip). October 2014.

[23] Amruta Rudrawar. Content based remote-sensing image retrieval with bag of visual words representation. In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, pages 162–167, 2018.

[24] A.-C. Grivei, C. Văduva, and M. Datcu. Improved earth observation data retrieval through hashing algorithms. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5909–5912, 2019.

[25] Zhiwen T. Xiaowen C. Xue-min Z. Kaibin Z. Zenggang, X. and Y. Conghuan. Research on image retrieval algorithm based on combination of color and shape features. 2019.

[26] Hu J. Hu F. Shi-B. Bai X. Zhong Y. Zhang L. Xia, G.S. and X. Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. 2017.

[27] S Chandeesh Kumar, M Hemalatha, S Badri Narayan, and P Nandhini. Region driven remote sensing image captioning. *Procedia Computer Science*, 165:32–40, 2019. 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019.

[28] Fengpeng Li, Ruyi Feng, Wei Han, and Lizhe Wang. En-

semble model with cascade attention mechanism for high-resolution remote sensing image scene classification. *Opt. Express*, 28(15):22358–22387, Jul 2020.

[29] Xiangqing Shen, Bing Liu, Yong Zhou, Jiaqi Zhao, and Mingming Liu. Remote sensing image captioning via variational autoencoder and reinforcement learning. *Knowledge-Based Systems*, 203:105920, 2020.

[30] Qimin Cheng, Yuzhuo Zhou, Peng Fu, Yuan Xu, and Liang Zhang. A deep semantic alignment network for the cross-modal image-text retrieval in remote sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:4284–4297, 2021.

[31] Binqiang Wang, Xiaoqiang Lu, Xiangtao Zheng, and Xuelong Li. Semantic descriptions of high-resolution remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 16(8):1274–1278, 2019.

[32] Bo Qu, Xuelong Li, Dacheng Tao, and Xiaoqiang Lu. Deep semantic understanding of high resolution remote sensing image. In *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–5, 2016.

[33] Xiangrong Zhang, Xiang Li, Jinliang An, Li Gao, Biao Hou, and Chen Li. Natural language description of remote sensing images based on deep learning. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4798–4801, 2017.

[34] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D.

Manning, and Curtis P. Langlotz. Contrastive learning of medical visual representations from paired images and text, 2020.

[35] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

[36] Vlad Feinberg. Facebook ai similarity search(faiss), part2. https://vladfeinberg.com/2019/07/18/faiss-pt-2.html, Jul,2019.

[37] Mapbox. Mapbox gl javascript api.

[38] Xiaoqiang Lu, Binqiang Wang, Xiangtao Zheng, and Xuelong Li. Exploring models and data for remote sensing image caption generation. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4):2183–2195.

[39] Jiayuan Fan, Tao Chen, and Shijian Lu. Unsupervised feature learning for land-use scene recognition. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–12, 01 2017.

[40] Gencer Sumbul, Sonali Nayak, and Begüm Demir. SD-RSIC: summarization driven deep remote sensing image captioning. *CoRR*, abs/2006.08432, 2020.

[41] Arutiunian Artashes, Vidhani Dev, Venkatesh Goutham, Bhaskar Mayank, Ghosh Ritobrata, and Pal Sujit. Clip-rsicd. https://github.com/arampacha/CLIP-rsicd, 2022.

[42] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3:1–13, 09 2009.