

# **Technische Universität Berlin**

Faculty of Electrical Engineering and Computer Science  
Dept. of Computer Engineering and Microelectronics  
**Remote Sensing Image Analysis Group**



---

## **Content Based Image Retrieval by Deep Multi Modal Hashing for Sentinel Images**

---

Master of Science in Computer Science

November, 2020

**Hasan Bank**

Matriculation Number: 396582

**Supervisor:** Prof. Dr. Begüm Demir

## **Acknowledgements**

This study has been prepared in the scope of a master thesis of computer science in the Technische Universität Berlin.

Firstly, I would like to thank my family. They always support and helped me both materially and emotionally, even they are in the distance.

Secondly, I express my sincere appreciation to Prof. Begüm Demir, who introduced me to remote sensing and guided me with her knowledge during this long journey. Also, I am grateful to all other people of the Remote Sensing Image Analysis Group.

Finally, I would like to thank my girlfriend, Merve Kortel, who understood and supported me in this tiring period.

## Abstract

As improving Earth Observation technologies, many remote sensing data from different sources are produced every day. Creating an image retrieval system to retrieve relevant images from these big data sources by an image query would benefit different fields like urban area study, climate change analysis, and forestry study. Hence, a content-based multi-modal image retrieval system has been proposed in this thesis to increase the efficiency of using the big remote sensing data. Instead of using single modality, multi-modality, Sentinel-1, and Sentinel-2 images have been used to utilize various data sources' varied characteristics.

Our CBIR system has been created hashing based because of its advantages in searching time and storage units. When big images are hashed to small compact binary codes, the retrieving mechanism is getting faster, and the cost of storage decreases. Deep neural networks, specifically Convolutional Neural Networks, have been created to obtain data sources' hash codes. These networks have been trained in two different ways: mean square error loss and triplet loss. Results have been evaluated with mean average precision and weighted mean average precision.

**Keywords:** Content-Based Image Retrieval, Remote Sensing, Deep learning, Convolutional Neural Networks, Multi-modality, Cross-modality

## Zusammenfassung

Im Zuge der Verbesserung der Erdbeobachtung Technologien werden täglich viele Fernerkundungsdaten aus verschiedenen Quellen produziert. Die Schaffung eines Bildabfragesystems zum Abrufen relevanter Bilder aus diesen großen Datenquellen durch eine Bildabfrage würde verschiedenen Bereichen wie der Untersuchung städtischer Gebiete, der Analyse des Klimawandels und der Forstwirtschaft zugute kommen. Daher wurde in dieser Arbeit ein inhaltsbasiertes multimodales Bildabfragesystem vorgeschlagen, um die Effizienz der Nutzung der großen Fernerkundungsdaten zu erhöhen. Anstatt eine einzelne Modalität zu verwenden, wurden multimodale, Sentinel-1- und Sentinel-2-Bilder verwendet, um die unterschiedlichen Eigenschaften der verschiedenen Datenquellen zu nutzen.

Unser CBIR-System wurde aufgrund seiner Vorteile bei der Suche nach Zeit- und Speichereinheiten auf Hashing-Basis geschaffen. Wenn große Bilder in kleine kompakte Binärcodes gehasht werden, wird der Abrufmechanismus immer schneller, und die Speicherkosten sinken. Tiefe neuronale Netzwerke, insbesondere Convolutional Neural Networks, wurden geschaffen, um die Hash-Codes der Datenquellen zu erhalten. Diese Netzwerke wurden auf zwei verschiedenen Arten trainiert: Verlust durch mittlere quadratische Fehler und Verlust durch Triplets. Die Ergebnisse wurden mit mittlerer Durchschnittspräzision und gewichteter Durchschnittspräzision ausgewertet.

**Schlüsselwörter:** Bildgewinnung durch Fernerkundung, Tiefes Lernen, Neuronale Faltungsnetze, Multimodalität, Crossmedialität, Inhaltsbasierte Bildgewinnung

# Contents

<b>List of Acronyms</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation	1
1.2 Objective	2
1.3 Outline	2
<b>2 Foundation and Related Work</b>	<b>3</b>
2.1 Deep Learning	3
2.1.1 Convolutional Neural Networks	3
2.1.2 Loss Functions	6
2.1.3 CNN Architectures	7
2.2 Hashing Based Image Retrieval Methods	7
2.2.1 Single Modality Hashing Methods	9
2.2.2 Multi-Modality Hashing Methods	10
2.3 Basics on Remote Sensing	11
2.4 Data Set Description	12
2.5 Evaluation Metrics	14
2.5.1 Average Cumulative Gains	15
2.5.2 Normalized Discounted Cumulative Gains	15
2.5.3 Mean Average Precision	15
2.5.4 Weighted Mean Average Precision	16
<b>3 Proposed Multi-Modality Hashing Methods</b>	<b>18</b>
3.1 Multi-Modality Hashing with Mean Square Error Loss	19
3.2 Multi-Modality Hashing with Triplet Loss	21
<b>4 Experimental Results</b>	<b>23</b>
4.1 Results of Mean Square Error Multi-Modality Hashing	23
4.2 Results of Triplet Loss Approach	25
4.3 Comparison Between Proposed Methods	27
4.4 Visual Representations of Retrieved Images	28

<b>5 Conclusion and Future</b>	<b>39</b>
5.1 Conclusion	39
5.2 Future Work	40
<b>Bibliography</b>	<b>41</b>
<b>Appendix</b>	<b>44</b>

# List of Acronyms

ACG	Average Cumulative Gains
CBIR	Content-Based Image Retrieval
CNN	Convolutional Neural Network
DCG	Discounted Cumulative Gains
EO	Earth Observation
EC	European Commission
ESA	European Space Agency
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LIDAR	Laser Imaging Detection and Ranging
mAP	Mean Average Precision
MSE	Mean Square Error
NDCG	Normalized Discounted Cumulative Gains
WAP	Weighted Mean Average Precision

## List of Figures

2.1	The Structure of one example neuron which applies the activation function . . .	4
2.2	Different activation functions . . . . .	5
2.3	ILSVRC winner methods with top-5 error on the classification task [22] . . . . .	7
2.4	ResNet architectures for ImageNet [7] . . . . .	8
2.5	Representation of the content-based image retrieval system . . . . .	8
2.6	Passive and active remote sensing [10] . . . . .	11
3.1	Representation of mini-batch used in MSE loss based approach . . . . .	19
3.2	Euclidean distance and cosine distance . . . . .	20
3.3	Workflow of the proposed triplet based method to retrieve images as using Sentinel-1 query . . . . .	22
4.1	Errors during the training of MSE based approach . . . . .	24
4.2	Elapsed time during training and validation, number of retrieved images per query = 20, 50 . . . . .	24
4.3	Errors during the training of the triplet loss-based approach . . . . .	25
4.4	Elapsed time during training and validation of triplet loss-based method, number of retrieved images per query = 20, 50 . . . . .	26
4.5	Average mAP and WAP results comparisons among different hash bits for 20 and 50 retrieved images per query . . . . .	29
4.6	Consumed time to train and validate both models for 20 and 50 retrieved images per query . . . . .	30



# List of Tables

2.1	Example hashed images	9
2.2	Sentinel-2 image bands [21]	12
2.3	Unrepresented classes in Serbia patches	14
2.4	Distribution of classes in Serbia patches	17
4.1	Results of Mean Average Precision (mAP) and Weighted Mean Average Precision(WAP) of MSE based approach for different numbers of bits on 20 retrieved images per query	25
4.2	Results of Mean Average Precision (mAP) and Weighted Mean Average Precision(WAP) of MSE based approach for different numbers of bits on 50 retrieved images per query	26
4.3	Results of Mean Average Precision (mAP) and Weighted Mean Average Precision(WAP) of triplet loss based approach for different numbers of bits on 20 retrieved images per query	27
4.4	Results of Mean Average Precision (mAP) and Weighted Mean Average Precision(WAP) of triplet loss based approach for different numbers of bits on 50 retrieved images per query	28
4.5	Visual results on retrieved 20 Sentinel-1 images by Sentinel-1 query under various hash length bits in MSE based approach	31
4.6	Visual results on retrieved 20 Sentinel-2 images by Sentinel-1 query under various hash length bits in MSE based approach	32
4.7	Visual results on retrieved 20 Sentinel-1 images by Sentinel-2 query under various hash length bits in MSE based approach	33
4.8	Visual results on retrieved 20 Sentinel-2 images by Sentinel-2 query under various hash length bits in MSE based approach	34
4.9	Visual results on retrieved 20 Sentinel-1 images by a Sentinel-1 query as employed a neural network trained with triplet loss under various hash length bits	35
4.10	Visual results on retrieved 20 Sentinel-2 images by a Sentinel-1 query as employed a neural network trained with triplet loss under various hash length bits	36
4.11	Visual results on retrieved 20 Sentinel-1 images by a Sentinel-2 query as employed a neural network trained with triplet loss under various hash length bits	37

*List of Tables*

4.12	Visual results on retrieved 20 Sentinel-2 images by a Sentinel-2 query as employed a neural network trained with triplet loss under various hash length bits	
	.....	38

# 1 Introduction

## 1.1 Motivation

Remote sensing can be defined as a process that detects and monitors emitted or reflected signals from the objects on the earth and use these radiations to obtain physical characteristics of the objects. These signals are gathered by satellites or aircraft. Shortly, it can be described as sensing from a great distance. Thanks to improvements in Earth Observation technologies, the volume of remote sensing images is getting higher with higher spatial and higher spectral resolutions. Using these big data archives effectively would be beneficial in several different domains, such as urban area study, forestry research, risk management. That is why implementing an efficient and correct remote sensing image retrieval systems is a need.

Early remote sensing image retrieval methods have used text-based manual annotations like geographical locations, visual descriptions, sensor types. However, these annotations are not always available, and the quality of these annotations directly affects the retrieval systems' performance. Therefore, the recent and popular approach of image retrieval in remote sensing is content-based.

In content-based image retrieval systems, the feature representation of the query image is computed. This representation is compared with the feature representations of all images used as a database. Feature representations can be obtained by handcrafted feature descriptors or data-driven feature descriptors.

CBIR systems relying on handcrafted features have shallow learning architectures. These simple architectures can not afford a feature learning which is optimized simultaneously. Therefore, it has a limited capability to represent the high-level semantic content of remote sensing images. On the other hand, deep learning-based CBIR systems simultaneously optimize feature learning during the image retrieval process. It removes the need for descriptors which are created by human [34].

In order to have a quick search and find related images in big data archives, hashing has been used. It is getting more critical to retrieve data in terms of time and storage efficiency. Hashing methods convert original high dimensional images into small binary hash codes. Therefore, original data storage requirements have been reduced significantly, and similarities of the images are measured by the calculated Hamming distance between the binary hash codes. Decreased cost of storage and less searching time is the fundamental purpose of using hashing in image retrieval. Using the hashing techniques with deep neural networks(DHNNs) has achieved good results on large-scale remote sensing image retrieval, as proposed by Li et al. (2017) [17].

## 1.2 Objective

Thanks to advances in sensor technologies, remote sensing images' spectral and spatial resolution are getting higher, and the volume of remote sensing images increases. According to Kempeneers and Soille, Sentinel satellites operated by the Copernicus program of the European Commission (EC) produce around 10 TB of Earth Observation (EO) data per day [12]. Finding semantically similar images or classifying similar images together in this big data source by human-based approaches takes a considerable amount of time. In order to utilize this data source effectively, deep hashing techniques can be used. Therefore, the retrieval process's speed can be increased, and storage requirements can be decreased because of hashing advantages.

CBIR systems do not have to be in one modality. Some datasets can have multi modalities such as image, audio, text, so on so forth. There are several examples of multi modalities. For example, images or audio can have text descriptors as labels or tags to keep some text information related to the data's class or category. In this study, two different remote sensing datasets have been used. Sentinel-1 and Sentinel-2 are missions of the European Space Agency. Creating hashed codes of Sentinel-1 and Sentinel-2 images by using deep hashing techniques was the first step of this study, and then, these hashed codes have been used to find similar images in the same and different datasets.

## 1.3 Outline

This thesis is separated into 5 chapters. A short introduction of each chapter is given below.

**Chapter 2** is the Foundation and Related Work. Deep learning techniques have been used in this study to hash the images. In this chapter, the basics of deep learning have been discussed. A literature review about CBIR and hashing in Section 2.2 and remote sensing basics in Section 2.3 has been shared. Details of the used dataset have been written in Section 2.4, and the most commonly used evaluation metrics in multi-labeled image retrieval problems have been detailed in the last part of this chapter.

**Chapter 3** is the Proposed Multi-Modality Hashing Methods. Two different loss functions have been implemented to fulfill the purpose of the study. These approaches have been shared in this chapter.

**Chapter 4** is the Experimental Results. Train, validation, and test results have been shared for both of the proposed methods. Time and evaluation metrics based comparisons are also in this chapter. Finally, the visual representations of retrieved images by an example query have been presented.

**Chapter 5** is the Conclusion and Discussion. A summary of the thesis has been shared. It has been concluded by discussing future works.

## 2 Foundation and Related Work

Using deep learning techniques in CBIR systems is attractive and influential in computer vision and remote sensing fields. This chapter explains the fundamentals of Deep Learning, hashing mechanisms, remote sensing basics, used dataset details, and evaluation metrics.

### 2.1 Deep Learning

Deep learning is one of the machine learning techniques, and most of the deep learning methods are based on neural networks. Deep neural networks try to mimic how the human brain operates and recreate it artificially because of the human brain's massive capability for learning, adapting skills and then applying them. If computers could copy that, computer algorithms can utilize this robust learning structure. Hence, an artificial structure is created, which has nodes or neurons. There are several layers with connected neurons. Information is propagated and processed from one layer to another in artificial neural networks. This structure allows for learning directly from data. Therefore, it has shown impressive performance in several fields like computer vision, speech recognition, machine translation, and bioinformatics.

There are two big categories in machine learning or deep learning. These are supervised and unsupervised learning. Although labels as extra information are needed in supervised learning, unsupervised learning realizes the learning process without label information.

#### 2.1.1 Convolutional Neural Networks

In the computer vision field, the convolutional neural networks is a popular deep learning technique because of its robust ability to find images pattern. Finding patterns is done in a hierarchical way in CNN(Convolutional Neural Network), which means low-level patterns are detected in the first layers, and high-level patterns are recognized towards to last layers.

Steps are followed in CNN:

1. Convolution
2. Pooling
3. Flattening
4. Full Connection

Filters are applied to input images to generate feature maps in the convolutional layer. However, input images are converted n-dimensional arrays or tensors in the pre-processing step. For example, a Sentinel-1 image with double polarization 60x60 resolution is represented 60x60x2 tensor to use in the convolutional layer.

## 2 Foundation and Related Work

Filters are slid over the image spatially and computing dot products. Therefore, a feature map or activation map is obtained. Multiple filters are created in order to have different feature maps. Thus, different features are kept.

To define how many pixels are shifted over in the input matrix while filter slides, the stride is defined. When the stride is 1, filters move 1 pixel at a time, or if it is 2, filters move 2 pixels at a time. The output size is calculated with Formula [2.1](#) where  $N$  is for input size,  $F$  is for filter size.

$$\text{Output Size After Convolution} = (N - F) / \text{stride} + 1 \quad (2.1)$$

The convolution step can shrink images very quickly, and it is not always the requested case. In order to adjust the output size, new dummy pixels are added to the input. This action is called by padding, which increases the input data's width and height. If zeros fill these added pixels, it is named zero padding, and this is a common padding approach. As a result of employing padding, information losses are lessened. Pixels on the edges are less processed than pixels on the center when no padding is applied. Consequently, padding can overcome this problem.

The convolution step aims to make the image smaller while detecting certain features of the images. Thanks to this reduction of the size, processing will be faster and easier.

After every convolution operation, the activation function is applied. Each activation function operates a mathematical function between input and output neurons using the input nodes' values and weights, as seen in Figure [2.1](#)

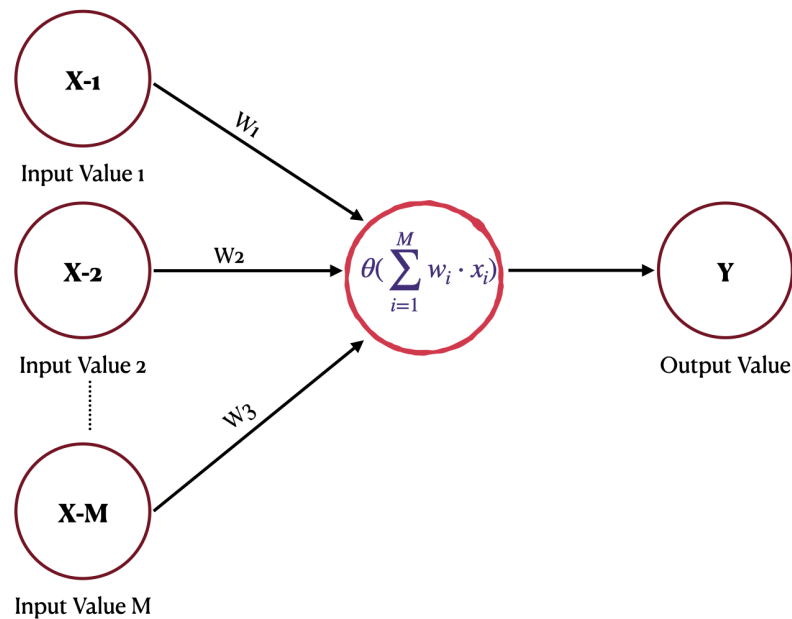


Figure 2.1: The Structure of one example neuron which applies the activation function

Some popular activation functions are presented in Figure [2.2](#)

Pooling is applied to each activation map, which has been created in the convolution layer. It makes the representations smaller and easier to manage. Max pooling, average pooling, L2-

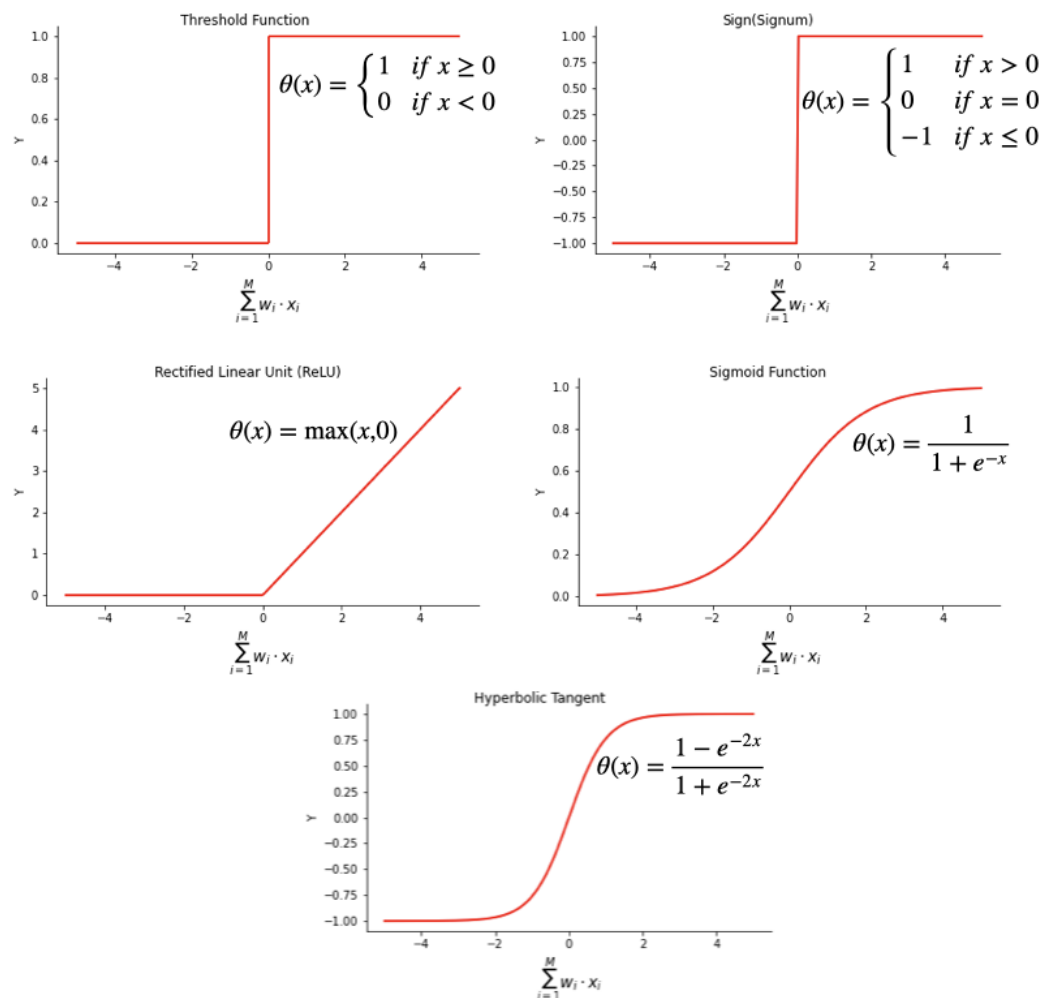


Figure 2.2: Different activation functions

norm pooling are different types of pooling. As in the convolution layer, filter size and stride is also used in pooling. Max pooling is the most popular pooling technique with stride 2 and filters size 2 or 3.

Pooled feature maps are flattened into one-dimensional vectors. These vectors are used as input for further processing.

Fully connected layers are the last layers of the CNN. The flattened tensor is processed in the fully connected layer, and it is made smaller until the desired number of neurons is obtained. Therefore, these layers build the connections between flattened tensors and all output neurons.

All these processes are repeated to minimize the error of the model. Epoch is a hyperparameter, and it defines how many times the learning algorithm runs through the entire dataset. It is like a for loop, which means each loop proceeds over the all training dataset to update weights and biases of the model effectively.

### 2.1.2 Loss Functions

Loss functions indicate how well the convolutional neural network is performing. In order to optimize the network, the value resulted from the loss function should be minimized. After calculating the error using the loss function, the network is backpropagated, and weights are adjusted to reduce the error.

Mean Square Error is the sum of squared distances between the target and predicted values. It is shown in Formula 2.2. MSE (Mean Square Error) and Mean Absolute Error is commonly used regression losses.

$$L_{MSE} = \frac{\sum_{i=1}^n (y_i - y_i^p)^2}{n} \quad (2.2)$$

Cross-Entropy Loss is used in classification, image retrieval problems like in [18], [40]. The calculation is shown in Formula 2.3, where  $y_i$  is the real class value and  $\hat{y}_i$  is the prediction.

$$L_{CrossEntropy} = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \quad (2.3)$$

Instead of feeding the network with pointwise or pairwise training examples, comparisons between three data points are fundamental of the triplet loss functions [25]. The Anchor, Positive and Negative are named these data points. Anchor and positive images are similar images because they belong to the same class. However, anchor and negative images are dissimilar images because they are in different classes. The purpose of the triplet loss function is to minimize the distance between the anchor and positive sample while maintaining a distance between anchor and negative sample more than a defined margin [34]. Thus, Formula 2.4 has been obtained for the triplet loss function where  $f()$  refers to the trained model, A is for Anchor, P is for positive, and N is for negative samples,  $\alpha$  is the margin.

$$L_{Triplet} = \sum_{i=1}^M \max(\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha, 0) \quad (2.4)$$

There are some other loss functions to achieve different purposes. To have better binarization from the last layer of the neural network to hashed value, push loss is used, as shown in Formula 2.5 where K refers to the desired hash length and  $\mathbf{1}$  is the K-dimensional vector with all elements 1 [26].

$$L_{Push} = -\frac{1}{K} \sum_{i=1}^P \|f(g_i) - 0.5 \cdot \mathbf{1}\|^2 \quad (2.5)$$

In order to increase the utilization of every bit, balancing loss is defined. It tries balancing the number of 0s and 1s in each binary code. The calculation is shown in Formula 2.6.

$$L_{Balancing} = \sum_{i=1}^P (\text{mean}(f(g_i)) - 0.5)^2 \quad (2.6)$$



### 2.1.3 CNN Architectures

Different numbers and types of layers are in a convolutional network, and various CNN architectures are obtained by permuting these layers' order. Researchers proposed the architectures like AlexNet [36], GoogleNet [30], VGGNet [31], ResNet[7] to provide better results to solve computer vision problems. The networks are getting larger and deeper with each new architecture.

ImageNet Large Scale Visual Recognition Challenge(ILSVRC) [27] gives a chance to compare these CNN architectures to traditional methods. Figure 2.3 shows ILSVRC winner methods and their corresponding top-5 error on the classification task. The challenge winner of 2015 is ResNet, which is the lowest error and largest network compared to previous years. Because of that, ResNet has been chosen for the CNN configuration of this study. Figure 2.4 shows details of ResNet architectures.

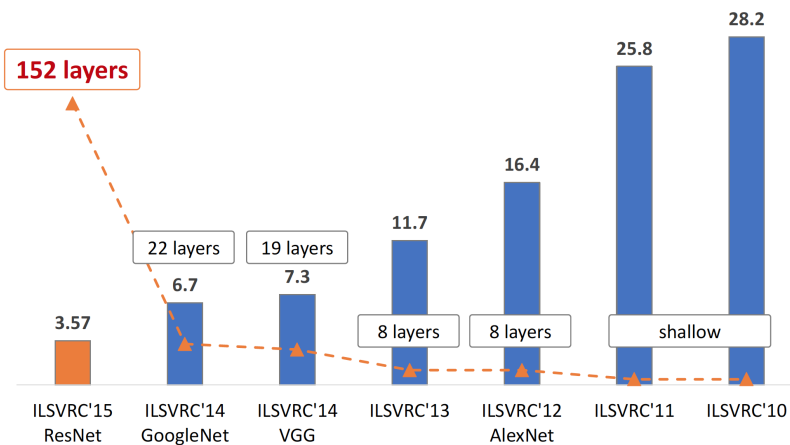


Figure 2.3: ILSVRC winner methods with top-5 error on the classification task [22]

## 2.2 Hashing Based Image Retrieval Methods

Image retrieval systems try finding similar images when a query image is given. Conventional methods of image retrieval use metadata of images such as labels, tags, text descriptions. However, content-based image retrieval systems have provided better results without metadata or human-made annotations. Instead of using manually added text information, CBIR systems use the visual contents acquired from the images. These contents are named features that represent the attributes of the images like color, texture, shape.

There are two main steps of the CBIR systems. Firstly, the image description step is done. Characteristics of the images are defined in the image description step. The image retrieval step is executed to retrieve images similar to a query by comparing the similarity between image descriptors of the dataset and the query [34]. Figure 2.5 shows the general flow of the CBIR system.

## 2 Foundation and Related Work

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

Figure 2.4: ResNet architectures for ImageNet [7]

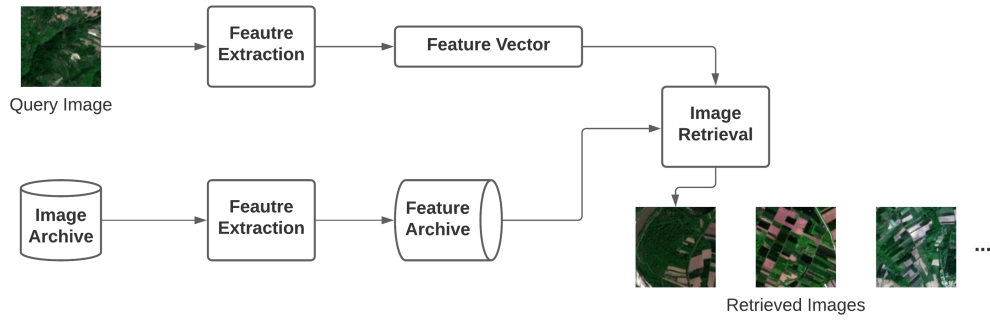


Figure 2.5: Representation of the content-based image retrieval system

Among all CBIR methods, hashing methods take significant attention because of its retrieval accuracy and time efficiency. Hashing mechanisms convert images kept in big storage units into binary codes, which are small and compact representations [34]. During this conversion, the similarity between the raw images is kept in the Hamming space [11]. Hence, image retrieval is done by using the Hamming distance between hashed images. The Hamming distance is the number of positions which are different between two equal-length input. For example, a query image has been hashed to binary 8 bits, [0,0,0,0,1,1,0,0]. Hashed database images are shown in Table 2.1 with their hamming distances to the query. The image with index 3 has minimum differences with the query when compared with others. Therefore, the image with index 3 is called the closest image to the query. Thanks to using hamming distance, finding and retrieving similar images of a query in a big hashed database are quicker and more effective than traditional human-based approaches.

Conventional hashing-based CBIR systems use hand-crafted features to obtain binary hashed

Table 2.1: Example hashed images

Index	Binary Hash	Hamming Distance to Query
0	[1,1,1,1,0,0,0,0]	6
1	[1,1,0,1,0,0,0,0]	5
2	[0,0,0,0,0,0,0,1]	3
3	[1,0,0,0,0,1,0,0]	2

codes. However, it does not provide a satisfactory result to represent the big volume of remote sensing images because this feature extraction process is not directly connected to the hash-code learning phase [11]. That is why hand-crafted features can not be fit to use in hash code learning. However, deep neural networks hashing-based CBIR systems provide end-to-end learning architecture with more accurate results due to their capability to convert images into binary codes as preserving the semantics of the images more adequately. These deep hashing techniques are categorized into two headings. There is one type of data source in single-modality, and this data source is used for both the query and retrieval set. In multi-modalities, there are different types of data sources.

### 2.2.1 Single Modality Hashing Methods

Query and database for retrieval images are from the same source in hashing based single modality CBIR methods [5]. Using an image for a query to retrieve similar images from the same dataset using hashing techniques is a typical example of single modality hashing usage. Many studies have used a single modality hashing approach in CBIR systems of different domains like computer vision, remote sensing.

Lin et al. (2016) proposed an unsupervised deep hashing approach for image retrieval in a single modality [19]. It did not require labeled training data and had three goals during the creation of binary hash codes. These were minimal quantization loss, evenly distributed codes, and uncorrelated bits. It has two deep learning networks. The first network has been initialized with pre-trained weights from 16 layers VGGNet [31] which is trained on the ImageNet dataset. Therefore, it benefits from transfer learning. However, transfer learning does not help to provide satisfactory results always. The parameters from a network trained in computer vision datasets can not be efficiently used in a network trained in remote sensing datasets because computer vision and remote sensing datasets have different characteristics like spatial and spectral resolution [35] [18]. Because of that reason, transfer learning from a network that was trained on a computer vision dataset into the remote sensing domain can not produce good results.

A supervised deep hashing approach was proposed for fast and accurate image search in remote sensing by Roy et al. (2018) [26]. It utilizes a pre-trained network(Inception Net [37]) trained on ImageNet besides, the second stage is trained with diverse losses such as triplet loss, representation penalty, and a balancing loss.

Instance similarity deep hashing(ISDH) by proposed Zhang et al. (2018) [40] and Deep Multi-Similarity Hashing(DMSH) by proposed Li et al. (2017) [16] have used multi-label computer vision datasets instead of using single-label datasets. In single-label datasets, images can be

classified as similar when they have the same label. However, the similarity between two images in multi-label datasets is based on how many labels are shared. While the number of common labels is increasing, the similarity between images is getting higher. Hence, binary codes should be more closer to preserve this similarity. This primary purpose has been tried to achieve by defined new different pairwise loss functions [40], [16].

### 2.2.2 Multi-Modality Hashing Methods

Query and retrieval images are located in the same dataset, while single-modality hashing methods are applied. However, heterogeneous multi-modal data sources have been converted compact binary hash codes by multi-modality hashing methods. It allows searching semantically similar images in a modality as using a query from a different modality.

Although data can represent better with multiple modalities, most of the hashing studies still base on a single modality. Different modalities can complete each other to reflect the features of the object or scene. Though, the number of multi modalities hashing studies are much less than single modality hashing studies. Therefore, multi-modalities hashing is needed more attention [2].

Plenty of multi-modality hashing studies [[2], [38], [11], [1], [4], [33]] are based on computer vision datasets like MIRFlickr[9], NUS-WIDE[3], Wiki[23], IAPR TC-12[6], Microsoft Coco[20] rather than remote sensing datasets. However, Sentinel-1 radar images and Sentinel-2 multispectral optical images are used in this thesis.

Creating a correlation between modalities based on handcrafted features is not easy because the modalities have notably different statistical properties. Deep neural network techniques have remarkable abilities to capture a correlation between these heterogeneous data sources [1].

Supervised hashing methods utilize supervised information to obtain semantic information of the images, and it creates better correlation and decreases the semantic gap between modalities; ergo, more accurate results are achieved with supervised learning instead of unsupervised in multi modalities [1].

Deng et al. (2018) presented a supervised triplet-based deep hashing network for cross-modal retrieval in a computer vision dataset [4]. This triplet based loss function works on inter-modal and intra-modal. Anchor, positive and negative examples have been chosen from the same source like text or image in intra-modal. However, inter-modal works to create a cross-modality correlation. Anchor and positive, negative examples are from different sources. For example, while anchors have been collected from the text dataset, positives and negatives have been collected from the image dataset to calculate inter-modal loss value from text to image.

Li et al. (2018) presented a cross-source hashing approach for image retrieval based on remote sensing datasets by using panchromatic and multispectral single label images [18]. The panchromatic and multispectral image is a pair that were taken from the same location, but they show different properties of the land. Generally, their approach can be summarized into two main categories. One of them is uni-source LSRSIR(Large Scale Remote Sensing Image Retrieval). Intrasource pairwise similarity constraint(IASC) has been defined for uni-source image retrieval, which means query and retrieval are in the same source. This loss function keeps binary values of the similar images closer while keeping the distance between binary values of the dissimilar images in each source. The other category is the cross-source LSRSIR. The intersource pairwise

similarity constraint(IRSC) has been defined to use in this category. IRSC operates like IASC, but IRSC is operated between two data sources. The query and retrieval images are from different sources in cross-source LRSIR. To minimize the information loss during binarization from the final feature representation to binary hashed codes, binary quantization constraint(BQC) has been defined. Feature distribution constraint(FDC) has been employed to have balanced bit values across the hashed dataset.

## 2.3 Basics on Remote Sensing

Remote sensing is a technology to observe the objects on the earth using sensors on the aircraft or satellite. According to the type of energy resources during data acquisition, remote sensing systems are categorized as passive and active systems. As illustrated in Figure 2.6, passive remote sensing systems exploit solar radiation emitted or reflected by objects. However, active remote sensing systems produce signals towards target objects and then register reflected radiation from the target. Active or passive sensors can get different types of information like Multispectral, Hyperspectral, Synthetic Aperture Radar, Laser Imaging Detection and Ranging(LIDAR), Scatterometer, Radiometer. Multi and hyperspectral imaging sensors are some examples of passive remote sensing systems. Radar systems or LIDAR are examples of active remote sensing systems. Remote sensing image analysis is getting more popular because of its wide range of applied fields like climate analysis, urban area study, forestry research, risk and damage management, water quality evaluation, and monitoring.

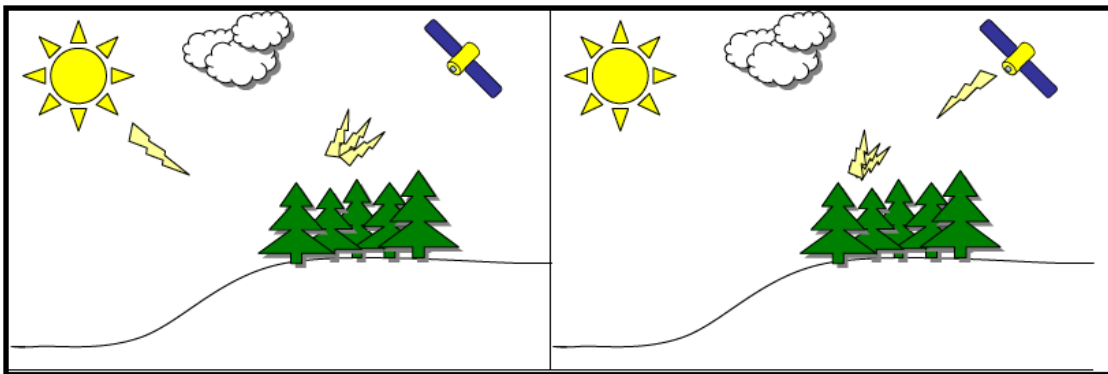


Figure 2.6: Passive and active remote sensing [10]

Sentinel-1 is the first mission of the Copernicus program developed by European Space Agency(ESA). The mission comprises a constellation of two-polar orbiting satellites, performing C-band synthetic aperture radar instrument which provides acquiring images in all weather conditions, day or night. It offers images in single or double polarization [28].

Electromagnetic radiation varies in a spectrum range from the shorter wavelengths like gamma and x-rays to longer wavelengths like microwaves and broadcast radio waves. Different wavelengths are used in remote sensing to observe different types of materials. The absorption

## 2 Foundation and Related Work

and emission of radiation are affected by the molecular form and shape of the observed object. Therefore, a multispectral image has 2-15 bands, and each band detects varied information obtained by a particular wavelength signal.

Sentinel-2 is another earth observation mission that ESA is developing for the Copernicus initiative. It comprises a constellation of two identical satellites, which are Sentinel-2A and Sentinel-2B, in the same orbit that acquires multispectral optical images at high spatial resolution (10 m to 60 m) over land and coastal areas [29].

Pixel is the smallest element of information in a digital image. A digital image comprises pixels in two-dimensional, which are columns and rows. Spatial resolution is named for the size of an area, which is represented by a pixel. The ability of the sensor affects spatial resolution, which means detected details on the observed area.

Sentinel-2 multispectral images have 13 bands with different spatial and spectral resolutions. A Sentinel-2 image has 4 bands with 120x120 pixels 10 meter spatial resolution, 6 bands with 60x60 pixels 20 meter spatial resolution, and 2 bands with 20x20 pixels 60 meter spatial resolution, which are presented in Table 2.2

Table 2.2: Sentinel-2 image bands [21]

Sentinel-2 Bands	Spatial Resolution (m)	Description	Sentinel-2A		Sentinel-2B	
			Central Wavelength (nm)	Bandwidth (nm)	Central Wavelength (nm)	Bandwidth (nm)
B1	60	Aerosols	442.7	21	442.2	21
B2	10	Blue	492.4	66	492.1	66
B3	10	Green	559.8	36	559.0	36
B4	10	Red	664.6	31	664.9	31
B5	20	Red Edge 1	704.1	15	703.8	16
B6	20	Red Edge 2	740.5	15	739.1	15
B7	20	Red Edge 3	782.8	20	779.7	20
B8	10	Near infrared	832.8	106	832.9	106
B8A	20	Red Edge 4	864.7	21	864.0	22
B9	60	Water vapor	945.1	20	943.2	21
B10	60	Cirrus	1373.5	31	1376.9	30
B11	20	Short-Wavelength Infrared 1	1613.7	91	1610.4	94
B12	20	Short-Wavelength Infrared 2	2202.4	175	2185.7	185

## 2.4 Data Set Description

The deep learning models aim to generalize patterns in training data to predict new data that the model has never processed. In order to have a high generalization ability, the neural networks

should be supplied by large data sources. That is why using a limited number of annotated remote sensing images to feed deep learning networks causes unsatisfied results. A trained neural network with a small data set can suffer from the overfitting problem. Overfitting is adjusting the model too closely to a particular set of data, extracting some inexistent patterns, and it may cause failure to predict future observations reliably. When there is insufficient data for a domain such as remote sensing, transfer learning is used to benefit a trained network over big data set into the new task. Using and applying well-trained on large data sets and well-constructed networks to models with smaller data set to increase the performance is called transfer learning. Transfer learning is using what was learned for a particular task to solve a different task. In transfer learning, all layers can be removed after a particular layer. A new fully-connected layer with a different number of neurons and random weights can be added to these transferred layers. Transfer learning from the computer vision domain to the remote sensing domain is a way to overcome the lack of big annotated remote sensing images, but the source and destination tasks should be similar to have good results by transfer learning. This precondition can not be satisfied between computer vision and remote sensing because the properties of the computer vision and remote sensing images are not similar. Hence, using transfer learning to overcome small data set problem is not an effective way.

BigEarthNet is a new comprehensive multi-label Sentinel-2 archive presented by Sumbul et al. (2019) [35]. It is a multi-label archive, which means each image can have several land-cover class labels, and it is larger than existing archives in remote sensing. Our study has been trained and tested with Sentinel-1 and BigEarthNet Sentinel-2 images. Thanks to these adequate data sources, transfer learning has not been needed, and overfitting has not occurred.

BigEarthNet Sentinel-2 archive is used with Sentinel-1 images, which have double polarization, VV, and VH. The focused area is the country of Serbia. A subset of BigEarthNet has been created. That has only images related to the area of Serbia. Although BigEarthNet has 43 classes [35], some of these classes have not been represented in Serbia patches because of the land's nature, and totally there are 31 classes in Serbia patches. These not represented classes are written in Table 2.3. Classes in all Serbia patches and the number of images associated with each land-cover class have been presented in Table 2.4

Sentinel-1 and Sentinel-2 image pairs have been created. Paired Sentinel-1 and Sentinel-2 images mean that they have the same coordinates and same land-cover classes. These pairs have been used to feed CNN models.

Sentinel-2 multispectral images have 13 bands. However, the 10th band, which does not keep the information about the land-cover class, had been excluded from BigEarthNet [35]. Therefore, the datasets have 12 bands for Sentinel-2 images and 2 polarizations for Sentinel-1 images.

Totally 71,855 paired images have been processed to feed both of the CNNs. It means 71,855 Sentinel-1 and 71,855 Sentinel-2 images have been used. These sets have been divided as train, validation, and test sets by applying ratios respectively 50%, 25%, and 25% of the whole sets. Therefore, the train set with 35928 image pairs, the validation set with 17963 image pairs, and the test set with 17964 image pairs have been obtained. The train set is a sample of data to train the model. The model learns weights and biases as processing this data. In order to have an unbiased evaluation while tuning the model's hyperparameters, validation sets are used. When

Table 2.3: Unrepresented classes in Serbia patches

<b>Land-Cover Classes</b>
Agro-forestry areas
Annual crops associated with permanent crops
Coastal lagoons
Estuaries
Intertidal flats
Olive groves
Peatbogs
Permanently irrigated land
Rice fields
Salines
Salt marshes
Sea and ocean

training loss decreases but error increases in the validation set, this is a sign of overfitting to the training set. That is why the validation set is used to detect overfitting problems. A model is trained many times, defined by the number of epochs. A validation set is used to find the best trained model by comparing the accuracy values between different epochs. The test set is used to determine the general performance of the system. During the testing phase, query images are chosen from the test set, and images are retrieved from the validation set. In other words, the validation set is defined as a database.

Land-cover classes are kept as texts in a JSON file per image. They are categorical data, not numeric. Categorical data has values from the fixed categories, but these categories can not order. There are 31 land-cover classes, so images have some of them, and these class names do not have an ordering. Most deep learning algorithms can not operate categorical data directly. Numeric data like coordinates can be compared, and intervals between these values can be calculated. This study's proposed algorithms require numeric land-cover classes to calculate label similarities between images using Hamming distance. Therefore, one-hot encoding has been applied to the text-based categorical land-cover classes to convert them into numeric values. The length of the encoded label's array is 31 bits because of the total number of classes in the patches. Every class has been represented by a bit. When the bit is 1, the image has the related class. When the bit is 0, the related class is nonexistent for that image.

### 2.5 Evaluation Metrics

Evaluation of a CBIR system's performance is based on measuring the ranking quality of response for a given query. The highly relevant images are more useful than moderately relevant images, and these moderately relevant images are more useful than irrelevant images during retrieving similar images of a query. Therefore, this prioritizing should be taken into account in evaluation metrics.



Metrics most commonly used to evaluate a CBIR system's performance have been shared in this section. These metrics are mostly used in multi-labeled image retrieval methods [25].

### 2.5.1 Average Cumulative Gains

Average Cumulative Gains (ACG) is for the average number of shared labels between the query image and the top n retrieved images, as shown in the formula [2.7] where  $C(q,i)$  refers to the number of shared class labels between query q and retrieved i image [39].

$$ACG@n = \frac{1}{n} \sum_i^n C(q,i) \quad (2.7)$$

### 2.5.2 Normalized Discounted Cumulative Gains

There is a drawback with Average Cumulative Gains. ACG calculation does not include the ranking of retrieved images. However, the images with the most number of shared labels with the query should be retrieved earlier than the images with fewer shared labels. Discounted Cumulative Gains(DCG) calculation involves the retrieved images' position beside the relevance score and is calculated as in Formula [2.8]

$$DCG = \sum_{i=1}^n \frac{2^{C(q,i)} - 1}{\log(1+i)} \quad (2.8)$$

Discounted Cumulative Gains are normalized to keep it in the range [0,1]. DCG is divided by the ideal ranking order of DCG to have Normalized Discount Cumulative Gains(NDCG) as in Formula [2.9]. Ideal ranking order means that retrieved images should be in decreasing order of the number of shared labels with the query.

$$NDCG = \frac{DCG}{iDCG} \quad (2.9)$$

### 2.5.3 Mean Average Precision

The mAP metric is a well-known evaluation metrics used in information retrieval [25]. Many studies[[26], [39], [15], [16], [41], [14]] have used this metric to show the accuracy of their methods. It is the mean of average precision for each query, calculated as in Formula [2.10] where Q refers to the query set's size. The calculation of average precision per query has been shown in Formula [2.11] where  $Tr(q,i) \in 0,1$  is an indicator function. If q and i share at least 1 class, it is 1. If they do not have any common labels, it equals 0.  $N_{Tr}(q)@i$  returns the number of relevant images, which means the query and retrieved image has at least 1 shared class, from the query q within the top i images [39].

$$mAP = \frac{1}{Q} \sum_q AP(q) \quad (2.10)$$

## 2 Foundation and Related Work

$$AP(q) = \frac{1}{N_{Tr}(q)@n} \sum_i^n (Tr(q,i) \frac{N_{Tr}(q)@i}{i}) \quad (2.11)$$

### 2.5.4 Weighted Mean Average Precision

Weighted Mean Average Precision(WAP) is similar to MAP. However, WAP uses Average Cumulative Gains(ACG). WAP calculates the average ACG at each top n retrieved images instead of average precision. The formula [2.12](#) shows the calculation of the WAP.

$$WAP = \frac{1}{Q} \sum_q^Q \left( \frac{1}{N_{Tr}(q)@n} \sum_i^n (Tr(q,i) * ACG@i) \right) \quad (2.12)$$

Table 2.4: Distribution of classes in Serbia patches

<b>Land-Cover Classes</b>	<b>Number of Images</b>
Broad-leaved forest	43079
Non-irrigated arable land	34064
Land principally occupied by agriculture, with significant areas of natural vegetation	31955
Complex cultivation patterns	27832
Transitional woodland/shrub	25904
Discontinuous urban fabric	11331
Natural grassland	7763
Pastures	7239
Mixed forest	3916
Water courses	3596
Coniferous forest	2089
Industrial or commercial units	1764
Inland marshes	1440
Water bodies	1191
Fruit trees and berry plantations	1052
Vineyards	623
Mineral extraction sites	610
Sparsely vegetated areas	548
Sport and leisure facilities	245
Road and rail networks and associated land	138
Construction sites	111
Green urban areas	100
Burnt areas	73
Dump sites	71
Bare rock	62
Airports	42
Beaches, dunes, sands	35
Sclerophyllous vegetation	24
Continuous urban fabric	20
Moors and heathland	12
Port areas	2

### 3 Proposed Multi-Modality Hashing Methods

Two separate deep learning architectures have been implemented to create hash codes of Sentinel-1 and Sentinel-2 images. Both architectures are based on ResNet50, whereas the only difference is the number of input channels. Used Sentinel-1 images have two polarizations, and used Sentinel-2 images have 12 bands. Therefore, CNN Sentinel-1 number of input channels is 2, and CNN Sentinel-2 number of input channels is 12.

After learning the features of the images by ResNet50 architectures, a linear transformation has been applied in the fully connected layer. Thus, feature vectors have been shrunk to a hash length vector. After a fully connected layer, a sigmoid function has been applied to have values between 0 and 1. The sigmoid function keeps the values in the range of 0, 1. However, the binarization loss function penalizes the system when the codes returned from the neural network has 0.5. Therefore, it pushes the system to produce codes close to 0 or 1. In the validation phase, these decimals should have been converted to binary codes to save and use them as a database for the testing phase. Thus, a faster searching and retrieving mechanism has been provided thanks to benefiting the Hamming distance between binary codes. That is why a sign function is applied per modality after results obtained from the neural networks in the validation. Hence, saved hash codes have only 0s and 1s. The calculation of the sign function has been shown in Formula [3.1](#)

$$Binary = (sign(Logits - 0.5) + 1)/2 \quad (3.1)$$

These networks have been operated in two different ways, which are detailed in the section [3.1](#) Multi-Modality Hashing with Mean Square Error Loss and [3.2](#) Multi-Modality Hashing with Triplet Loss. However, both methods are empowered by using push (binarization) and balancing losses, as shown in the formula [3.2](#). The coefficients have been determined as following the study of Roy et al. (2018) [\[26\]](#) as  $\lambda_1 = 0.001$ , and  $\lambda_2 = 1$ .

$$L_{Combined} = L_{Main} + \lambda_1 * L_{Push} + \lambda_2 * L_{Balancing} \quad (3.2)$$

Mini batch gradient descent has been used during the training of both methods. It splits the whole data into a lot of small batches. The size of small batches has been defined as 200 in this study. In each iteration, this small batch of data has been used to train the neural networks.

Adam Optimizer [\[13\]](#) has been used to optimize loss functions with learning rate  $10^{-3}$  and weight decay  $10^{-4}$ . It is an algorithm for the first-order gradient-based optimization of stochastic objective functions.

### 3.1 Multi-Modality Hashing with Mean Square Error Loss

As defined in Section 2.1.2, MSE is the sum of squared distances between the target and predicted values. The predicted values of the loss function are cosine similarities of images' hash codes, and the target value is the cosine similarities of labels. It means cosine similarities of two images' hash codes should close to their labels' similarities. Therefore, binary codes of the images have preserved the label's similarity.

The mini-batch has been read like in Figure 3.1. Each pair has two rows. Every row has one paired Sentinel-1 and Sentinel-2 images that share the same land-cover classes.

Mini-Batch			
		Sentinel 1	Sentinel 2
Pair 1	Label1	S1_Image1	S2_Image1
	Label2	S1_Image2	S2_Image2
Pair 2		...	
Pair 200		...	

Figure 3.1: Representation of mini-batch used in MSE loss based approach

Cosine similarity is a metric to measure how similar two data are regardless of their size. Cosine similarity has been used to measure how similar labels and binary codes of images, and it is calculated as in the formula 3.3. The cosine similarity is useful because even if two data points are far apart by the Euclidean distance due to the magnitude of the points, they may still be closer in terms of cosine distance. As seen in the figure 3.2, the Euclidean distance of A2-B2 is bigger than A1-B1, but the cosine distance is the same. The length of the labels and hashed binary codes can be different. That is why using Euclidean distance to find similarity does not always produce realistic results [1]. However, cosine similarity looks at the angle between vectors, not magnitude.

$$\text{Cos}(a, b) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_i^n a_i b_i}{\sqrt{\sum_i^n a_i^2} \sqrt{\sum_i^n b_i^2}} \quad (3.3)$$

MSE Loss has been implemented with intramodality and intermodality loss calculations. These calculations have been done in pair based. Thus, they have been operated in every pair of the batch. Intra refers to the calculation inside of modality, and inter refers to the calculation between the modalities. There are two intramodality loss calculations. One is for Sentinel-1, and another is for Sentinel-2. In Sentinel-1 intramodality, cosine similarity of Sentinel-1 image 1's and Sentinel-1 image 2's hash codes have been used to input MSE. Cosine similarity of label 1 and label 2 is the target of the MSE. MSE loss function for Sentinel-1 intramodality has been calculated as in the formula 3.4. Replacing hash codes of Sentinel-1 images with hash codes of Sentinel-2 images in the calculation, Sentinel-2 intramodality loss calculation has been obtained, as shown in the formula 3.5

### 3 Proposed Multi-Modality Hashing Methods

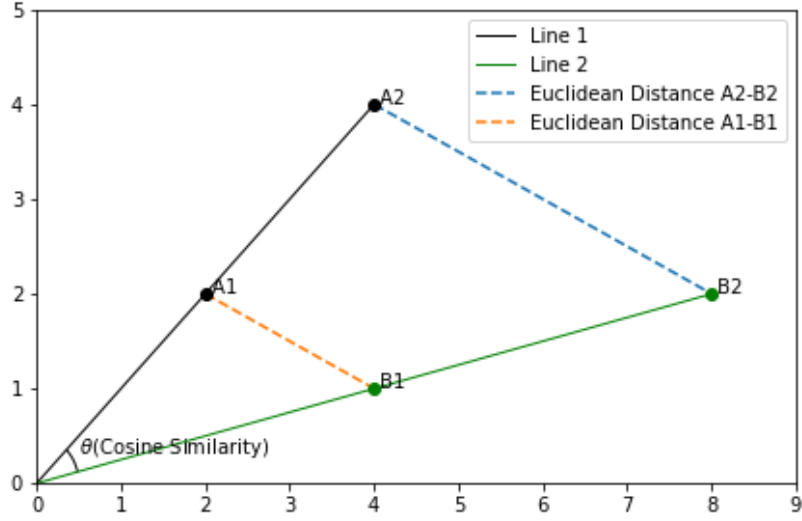


Figure 3.2: Euclidean distance and cosine distance

$$MSE_{IntraS1} = MSE(Cos(S1Image1, S1Image2), Cos(label1, label2)) \quad (3.4)$$

$$MSE_{IntraS2} = MSE(Cos(S2Image1, S2Image2), Cos(label1, label2)) \quad (3.5)$$

Besides the intramodality loss calculation, there are four intermodality loss calculations. Two of them have been calculated in the same row. The labels in the same row are the same. That is why 1.0, which is the cosine similarity of the same labels, is written to target values of the MSE loss. Paired hash codes of Sentinel-1 and hash codes of Sentinel-2 images should be similar because they share the same land cover classes. The formula 3.6 and the formula 3.7 has been operated to calculate respectively for row 1 and row 2 intermodality MSE Loss.

$$MSE_{InterSameLabel_1} = MSE(Cos(S1Image1, S2Image1), 1.0) \quad (3.6)$$

$$MSE_{InterSameLabel_2} = MSE(Cos(S1Image2, S2Image2), 1.0) \quad (3.7)$$

The cosine similarity of the hash codes is the input of the MSE Loss. If these hash codes have been chosen diagonally in the pair, MSE loss for intramodality in different labels have been obtained. Two equations have been operated, as shown in the formula 3.8 and 3.9, in order to calculate the loss value of intermodality MSE for different labels.

$$MSE_{IntrerDifferentLabel_1} = MSE(Cos(S1Image1, S2Image2), cos(label1, labe2)) \quad (3.8)$$

$$MSE_{IntrerDifferentLabel_2} = MSE(Cos(S1Image2, S2Image1), cos(label1, labe2)) \quad (3.9)$$

### 3.2 Multi-Modality Hashing with Triplet Loss

The general view of the approach like this:

1. Sentinel-1 intramodality as shown in the formula [3.4](#)
2. Sentinel-2 intramodality as shown in the formula [3.5](#)
3. Intermodality
  - a) Same labels(same row)
    - i. The formula [3.6](#)
    - ii. The formula [3.7](#)
  - b) Different labels
    - i. The formula [3.8](#)
    - ii. The formula [3.9](#)

It has different equations to fulfill all purposes. These equations are gathered in the formulas [\(3.10, 3.11, 3.12, 3.13\)](#).

$$L_{MSE} = 0.33 * MSE_{IntraS1} + 0.33 * MSE_{IntraS2} + 0.33 * MSE_{Inter} \quad (3.10)$$

$$MSE_{Inter} = 0.5 * MSE_{SameLabel} + 0.5 * MSE_{DifferentLabel} \quad (3.11)$$

$$MSE_{SameLabel} = 0.5 * MSE_{InterSameLabel_1} + 0.5 * MSE_{InterSameLabel_2} \quad (3.12)$$

$$MSE_{DifferentLabel} = 0.5 * MSE_{InterDifferentLabel_1} + 0.5 * MSE_{InterDifferentLabel_2} \quad (3.13)$$

## 3.2 Multi-Modality Hashing with Triplet Loss

Our MSE based approach or any pairwise loss functions check the similarities between image pairs, not among multiple images. This issue can cause a decrease in the accuracy of hashing based CBIR systems. In order to cope with this problem, triplet loss can be applied [\[34\]](#).

Hamming distances between one-hot encoded labels have been evaluated to find positive (closest) and negative (farthest) per anchor in every batch. Triplet loss keeps close binary hash codes of anchor and positive images to preserve labels' similarity after hashing. Moreover, it pushes away the binary codes of negative images from the anchor to maintain labels' differences in the hashed codes.

Anchor, positive and negative samples have been chosen in the same modality while intramodality losses are calculating.

Triplet-Based Deep Hashing Network for Cross-Modal Retrieval by Deng et al. (2018) has been followed for adapting triplet loss to the cross-modality [\[4\]](#). Although it is in the computer vision domain, the fundamental is the same. A query is chosen from a source, while the positive

### 3 Proposed Multi-Modality Hashing Methods

and negative instances are from another source. Anchor behaviors like the query of the retrieving mechanism. The anchor has been chosen from Sentinel-1, and positive, negative samples have been chosen from Sentinel-2 to simulate retrieving Sentinel-2 images by using the Sentinel-1 query. That is  $InterLoss_1$  of the formula 3.16. The anchor is from Sentinel-2, and positive, negative samples are from Sentinel-1. That is the  $InterLoss_2$  of the formula 3.16.

$$L_{Triplet} = 0.5 * L_{TripletIntra} + 0.5 * L_{TripletInter} \quad (3.14)$$

$$L_{TripletIntra} = 0.5 * L_{TripletS1Intra} + 0.5 * L_{TripletS2Intra} \quad (3.15)$$

$$L_{TripletInter} = 0.5 * L_{TripletInterLoss_1} + 0.5 * L_{TripletInterLoss_2} \quad (3.16)$$

As shown in Figure 3.3, Sentinel-1 and Sentinel-2's neural networks have been trained with different loss functions. These trained networks have been utilized to generate hash binary codes of the query in the testing phase. In the figure, only a Sentinel-1 query has been presented in testing for the sake of simplicity. Sentinel-2 images have been hashed using trained Sentinel-2's neural network in testing, and these hashed codes have been compared with hashed Sentinel-1 and Sentinel-2 archives to retrieve semantically similar images.

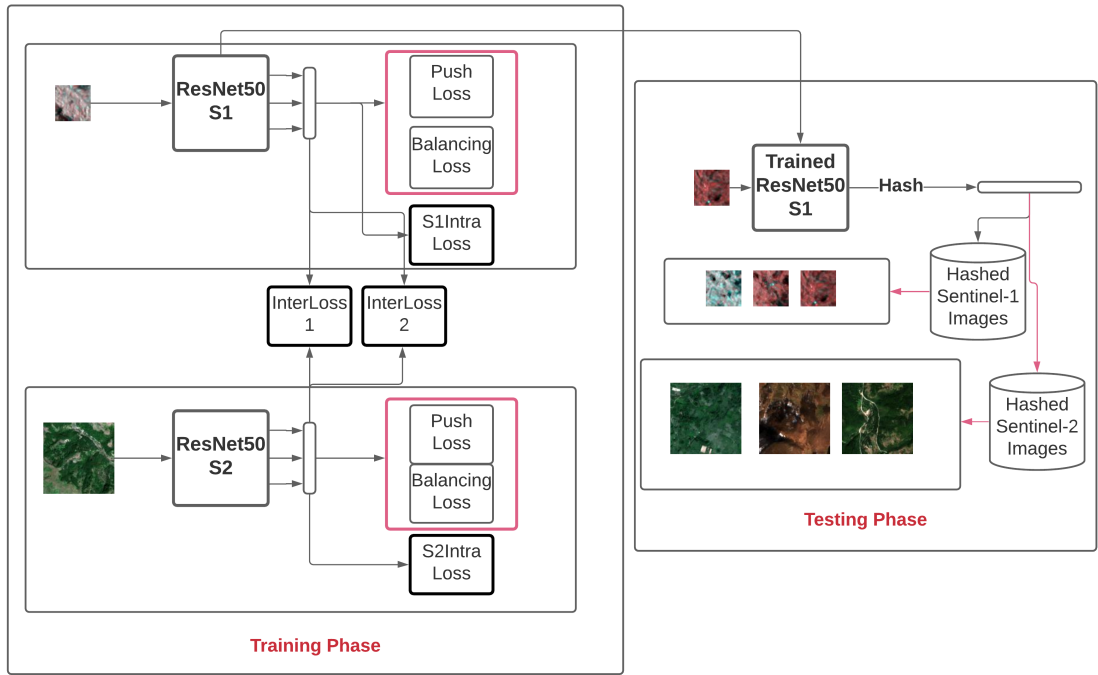


Figure 3.3: Workflow of the proposed triplet based method to retrieve images as using Sentinel-1 query



## 4 Experimental Results

MSE loss and triplet loss have been implemented to train the neural networks. The networks have been trained, validated, and tested with different parameters, which are the length of hash bits, number of retrieved images per query, and fixed parameters, which are the mini-batch size and number of epochs. The length of hash bits has been varied in a wide range of 8, 16, 32, 64, and 128 bits. The number of retrieved images per query has been defined as 20 and 50. The mini-batch size is 200.

The number of epochs is usually large to allow the learning algorithm to run until the model's error has been sufficiently minimized. However, as detailed in the next chapters, no vital decrease in loss function errors has been observed while the number of epochs increases and 10 has been determined as the number of epochs.

A model trained for the number of epochs. The model with the best results in a particular epoch should be saved to be used in the testing. mAP values have been compared in the validation phase to find the best epoch. The epoch that produced the highest mAP value in the validation has been saved as the best-trained epoch to use in testing.

All models have been trained and tested in TU Berlin High-Performance Cluster [8]. 2 GPU nodes have been reserved from the cluster to run the processes. This configuration has 1 NVIDIA Tesla P100 16GB HBM2 per node.

The results of all cases in both approaches have been evaluated by mAP and WAP metrics. In this chapter, these results have been shared.

### 4.1 Results of Mean Square Error Multi-Modality Hashing

In this section, the method's results, which is the mean square error-based approach, have been given. A line plot that shows epochs along the x-axis as time and the model's loss function error occurred during several training experiments on the y-axis has been presented, as shown in the figure 4.1. As the desired result of a deep learning algorithm, loss values have been decreased in every epoch. From 1st epoch to 5th epoch, a well-nigh 29% decrease of the loss error has been observed. However, the ratio has lessened after epoch 5. From the 5th epoch to the 10th epoch, the loss error reduction is about 9%. To be sure about this movement of the error among epochs, the model has been trained 5 times with the same parameters. Although the large epoch numbers will need a long time to finish the training, there would not be a drastic change in the result. Therefore, the train of the model has been done with 10 epochs number.

Figure 4.2 shows the time spent in training and validation of the models for different length of hash bits with 20 and 50 retrieved images per query.

Tables 4.1 and 4.2 are a statistical summary of our mean square error multi-modality hashing. The cells which have yellow background represent the best value among all length of hash bits.

## 4 Experimental Results

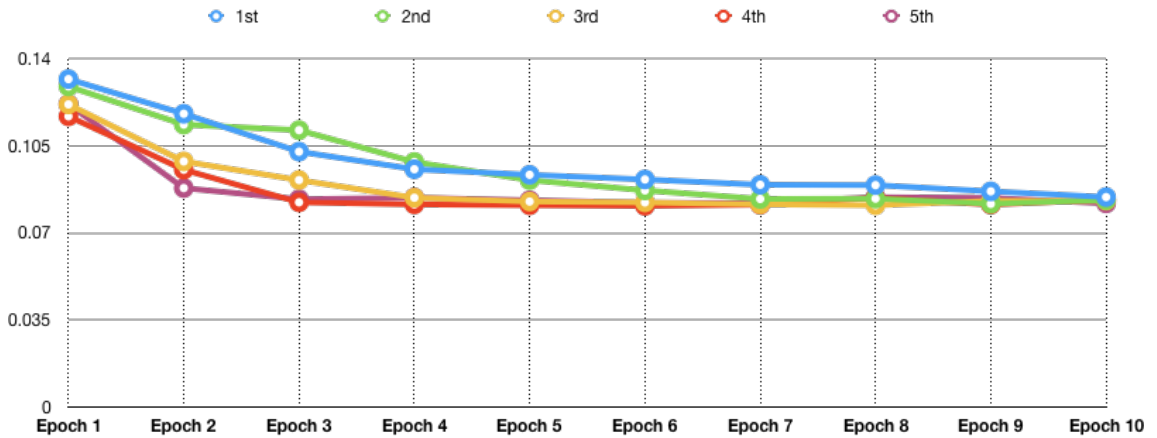


Figure 4.1: Errors during the training of MSE based approach

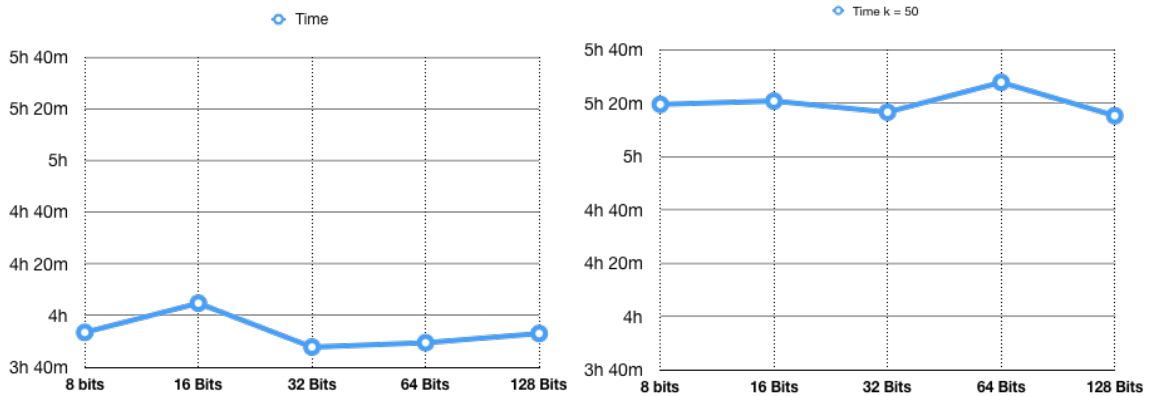


Figure 4.2: Elapsed time during training and validation, number of retrieved images per query = 20, 50

The mAP and WAP results for different numbers of bits have been shown in Table 4.1. The number of retrieved images for these results is 20. It can be observed that 32-bit and 64-bit hash lengths have outperformed to retrieve 20 images per query in the Mean Square Error based method.

Table 4.2 also has been shown the mAP and WAP results for different numbers of bits, but 50 is the number of retrieved images for these results. Although the 8-bit hash length has the highest average mAP and WAP, other hash lengths also have the best scores in different scenarios, like the method which produces the 32-bit length of hashes outperforms to retrieve 50 Sentinel-1 images on intramodality.

Table 4.1: Results of Mean Average Precision (mAP) and Weighted Mean Average Precision(WAP) of MSE based approach for different numbers of bits on 20 retrieved images per query

mAP					
	8-bit	16-bit	32-bit	64-bit	128-bit
<b>S1→S1</b>	0.911	0.886	0.905	0.918	0.883
<b>S1→S2</b>	0.545	0.602	0.760	0.590	0.451
<b>S2→S1</b>	0.457	0.531	0.502	0.588	0.528
<b>S2→S2</b>	0.860	0.909	0.796	0.960	0.906
<b>Average mAP</b>	0.693	0.732	0.741	0.764	0.692
WAP					
<b>S1→S1</b>	1.533	1.425	1.536	1.546	1.380
<b>S1→S2</b>	0.812	0.791	1.238	0.824	0.661
<b>S2→S1</b>	0.633	0.725	0.664	0.837	0.804
<b>S2→S2</b>	1.410	1.455	1.477	1.603	1.319
<b>Average WAP</b>	1.097	1.099	1.229	1.202	1.041

## 4.2 Results of Triplet Loss Approach

In this section, the method's results, which is the triplet loss-based approach, have been given. A line plot shown in the Figure 4.3 has also been created to illustrate how training loss errors appeared on several training experiments among the epochs. The rate of loss error's change from 1st epoch to 5th epoch is more notable in the triplet Loss-based approach than MSE based approach. Approximately 58% decrease in the loss errors has been observed from 1st to 5th epoch. This ratio is decreasing after the 5th epoch. From the 5th epoch to the 10th epoch, reducing the training loss errors is only about 8%.

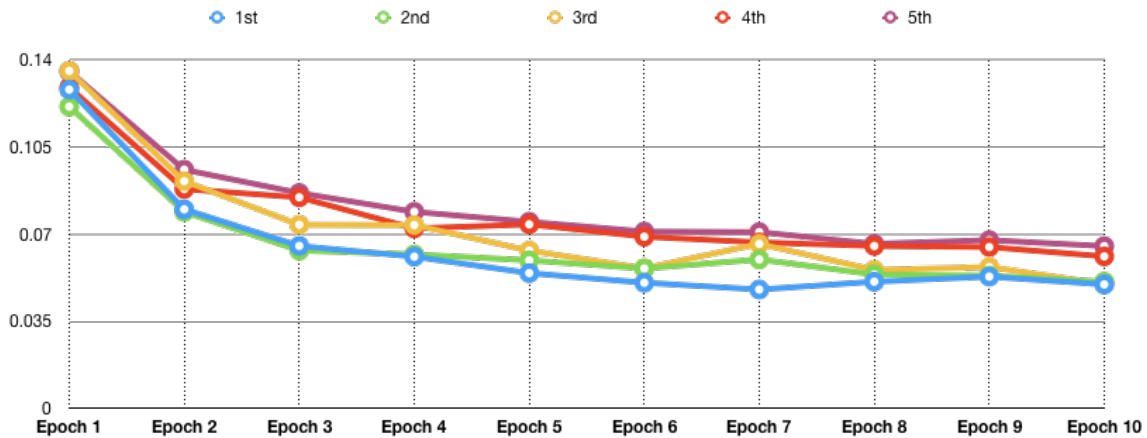


Figure 4.3: Errors during the training of the triplet loss-based approach

Figure 4.4 shows the time spent in training with triplet loss function and validating the models

## 4 Experimental Results

Table 4.2: Results of Mean Average Precision (mAP) and Weighted Mean Average Precision(WAP) of MSE based approach for different numbers of bits on 50 retrieved images per query

mAP					
	8-bit	16-bit	32-bit	64-bit	128-bit
<b>S1→S1</b>	0.600	0.876	0.919	0.864	0.777
<b>S1→S2</b>	0.679	0.552	0.476	0.370	0.477
<b>S2→S1</b>	0.891	0.500	0.491	0.641	0.600
<b>S2→S2</b>	0.666	0.885	0.909	0.938	0.905
<b>Average mAP</b>	0.709	0.703	0.699	0.703	0.690
WAP					
	8-bit	16-bit	32-bit	64-bit	128-bit
<b>S1→S1</b>	0.910	1.386	1.484	1.341	1.127
<b>S1→S2</b>	1.066	0.841	0.618	0.481	0.657
<b>S2→S1</b>	1.463	0.703	0.672	1.028	0.849
<b>S2→S2</b>	1.029	1.313	1.424	1.393	1.511
<b>Average WAP</b>	1.117	1.061	1.049	1.061	1.036

for different length of hash bits with 20 and 50 retrieved images per query.

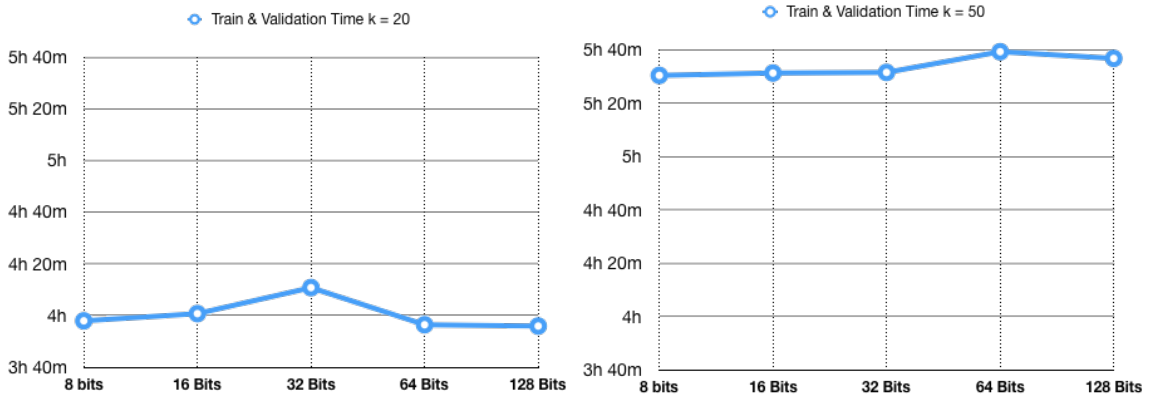


Figure 4.4: Elapsed time during training and validation of triplet loss-based method, number of retrieved images per query = 20, 50

The mAP and WAP results of the triplet loss-based method for different numbers of bits have been shown in Tables 4.3 and 4.4. The number of retrieved images per query is 20 for Table 4.3 and 50 for Table 4.4. A specific length of hash bits outperforms others in all cases can not be said in our triplet loss multi-modality hashing. Evaluation results of a length of hash bits depend on the number of retrieved images per query and retrieving type. Although that length has lower metrics in a case, it can have a higher outcome in another case. Even though 32 and 64-bit hashes produce better outcomes than others, results are so similar among the number of bits, as can be observed.

### 4.3 Comparison Between Proposed Methods

Table 4.3: Results of Mean Average Precision (mAP) and Weighted Mean Average Precision(WAP) of triplet loss based approach for different numbers of bits on 20 retrieved images per query

mAP					
	8-bit	16-bit	32-bit	64-bit	128-bit
<b>S1-&gt;S1</b>	0.933	0.921	0.897	0.952	0.836
<b>S1-&gt;S2</b>	0.917	0.950	0.943	0.921	0.772
<b>S2-&gt;S1</b>	0.933	0.847	0.924	0.939	0.927
<b>S2-&gt;S2</b>	0.964	0.924	0.961	0.960	0.901
<b>Average mAP</b>	0.936	0.910	0.931	0.943	0.859
WAP					
<b>S1-&gt;S1</b>	1.471	1.399	1.540	1.589	1.383
<b>S1-&gt;S2</b>	1.265	1.563	1.676	1.369	1.322
<b>S2-&gt;S1</b>	1.486	1.234	1.503	1.353	1.488
<b>S2-&gt;S2</b>	1.640	1.423	1.688	1.644	1.409
<b>Average WAP</b>	1.465	1.405	1.602	1.489	1.400

### 4.3 Comparison Between Proposed Methods

Mean square error loss and triplet loss do not have the capability of bit balancing and binarization. The same loss functions have been added to the MSE and triplet-based approach to have balanced bits in hashing and better binarization from the neural network's output to hashed codes.

Mean Square Error loss focuses on similarities among only image pairs, and similarities among multiple images are ignored. It causes poor performance to learn similarities of the images for CBIR problems. However, triplet loss works on image triplets, and it provides more effective image descriptors [34]. This performance difference also can be observed clearly in our study. As shown in Figure 4.5, triplet loss has produced better results than MSE loss in average mAP and WAP among all length of hash bits.

In terms of consumed time during training and validation, triplet loss needs more time, as presented in Figure 4.6. Although the training phase has not used the parameter, which is the number of images per query, the validation performance is directly connected to the number of retrieved images per query. As the number of retrieved images per query increases, the difference between the models' time spent is grown. For example, 3 hours and 56 minutes have been spent to train and validate the triplet loss-based model to create 128-bit hashes to retrieve 20 images, and the consumed time has been raised approximately 43% to retrieve 50 images in the same length of hash bits. This increment ratio is about 35% for the 128-bit length of hashes in the MSE-based approach. Therefore, the increment rate for consumed time in the validation of triplet loss is more prominent than MSE loss.

## 4 Experimental Results

Table 4.4: Results of Mean Average Precision (mAP) and Weighted Mean Average Precision(WAP) of triplet loss based approach for different numbers of bits on 50 retrieved images per query

mAP					
	8-bit	16-bit	32-bit	64-bit	128-bit
<b>S1→S1</b>	0.909	0.819	0.890	0.878	0.852
<b>S1→S2</b>	0.909	0.867	0.919	0.902	0.892
<b>S2→S1</b>	0.927	0.910	0.921	0.883	0.931
<b>S2→S2</b>	0.895	0.930	0.959	0.929	0.951
<b>Average mAP</b>	0.910	0.882	0.922	0.898	0.906
WAP					
<b>S1→S1</b>	1.352	1.267	1.412	1.425	1.267
<b>S1→S2</b>	1.316	1.372	1.415	1.308	1.404
<b>S2→S1</b>	1.396	1.418	1.455	1.455	1.520
<b>S2→S2</b>	1.461	1.500	1.616	1.453	1.455
<b>Average WAP</b>	1.381	1.389	1.474	1.410	1.412

### 4.4 Visual Representations of Retrieved Images

Visual representations of Sentinel-1 images have been created as building an RGB color image. The composite RGB image has been built using the VV channel for red, VH channel for green, and ratio  $|VV|/|VH|$  for blue [24].

Image visualizations of Sentinel-2 have been done by following the natural color representations. The composite RGB image has been built using band 4 for red, band 3 for green, and band 2 for blue [32]. Visualizations of the RGB format of Sentinel-2 images are presented with a scale of 0.5 to fit them into the page.

The same example image pair (Sentinel-1 and Sentinel-2 images) has been chosen as a query to visualize retrieved images of all cases in both methods. These cases are:

- Query: Sentinel-1, retrieved images: Sentinel-1
- Query: Sentinel-2, retrieved images: Sentinel-2
- Query: Sentinel-2, retrieved images: Sentinel-1
- Query: Sentinel-2, retrieved images: Sentinel-2

The number of retrieved images is 20 for these visualizations, and these results have been presented in different hash length bits.

Tables 4.5, 4.6, 4.7, and 4.8 have been created to visualize the results of retrieved images by mean square error multi-modality hashing. Sentinel-1 and Sentinel-2 intra-modalities have been shown in Table 4.5 and 4.8, respectively. From Sentinel-1 to Sentinel-2 is in Table 4.6, and from Sentinel-2 to Sentinel-1 is in Table 4.7.

#### 4.4 Visual Representations of Retrieved Images

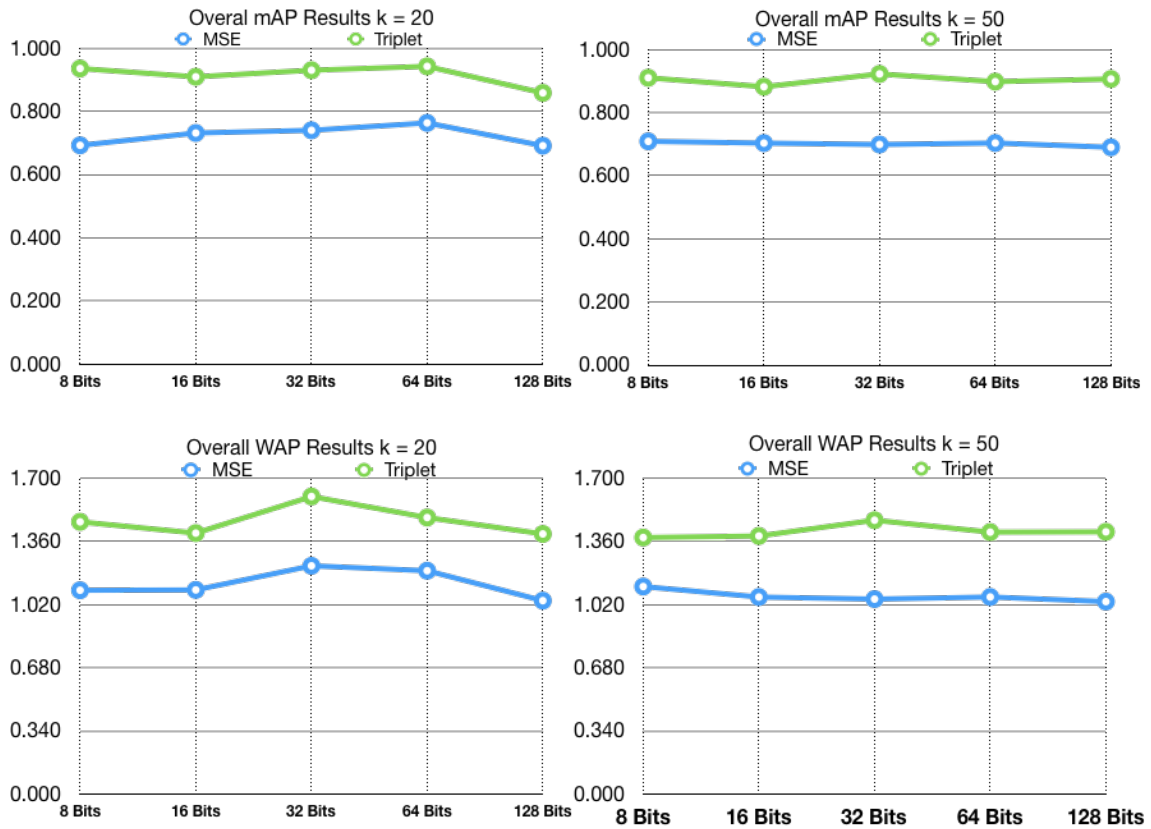


Figure 4.5: Average mAP and WAP results comparisons among different hash bits for 20 and 50 retrieved images per query

Tables [4.9](#), [4.10](#), [4.11](#), and [4.12](#) have been created to visualize the results of retrieved images by triplet loss multi-modality hashing. Tables [4.9](#) and [4.12](#) have been used to present intra-modalities' results of Sentinel-1 and Sentinel-2, respectively. Table [4.10](#) is for cross-modality from Sentinel-1 to Sentinel-2, and Table [4.11](#) is for cross-modality from Sentinel-2 to Sentinel-1 in the triplet-based approach.

## 4 Experimental Results

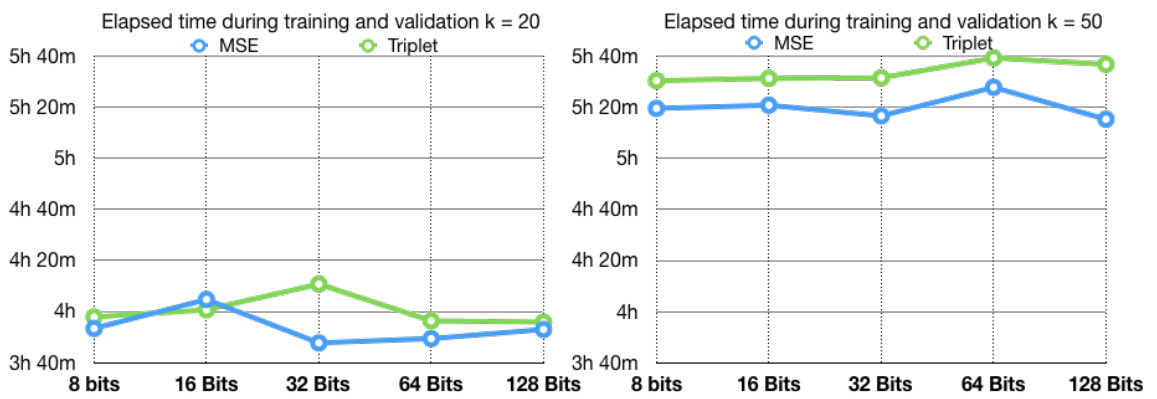
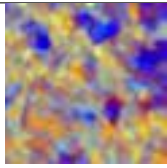
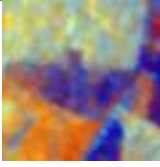
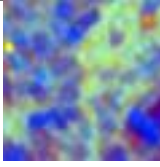
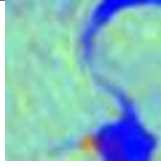
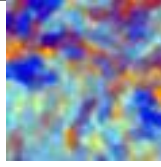
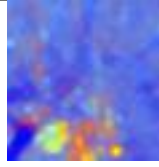
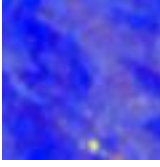
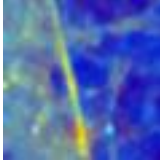
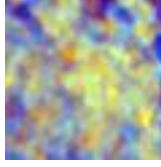
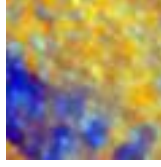
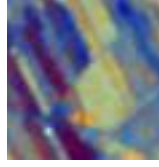
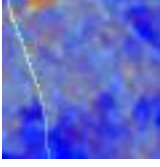
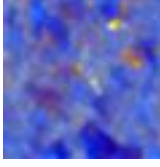
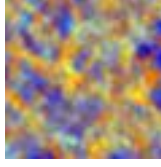
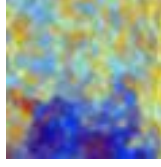
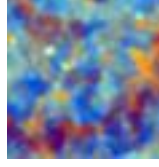
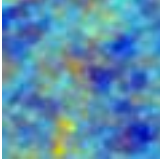
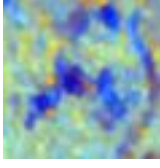
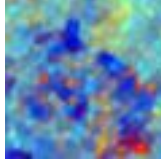
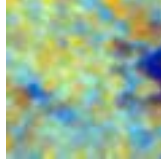
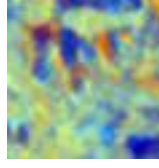
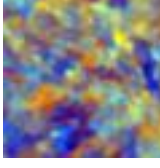
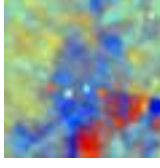
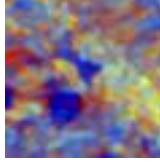
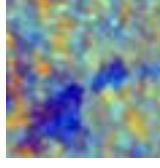
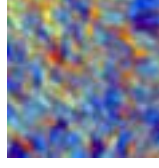


Figure 4.6: Consumed time to train and validate both models for 20 and 50 retrieved images per query



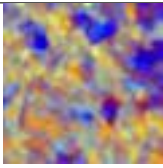


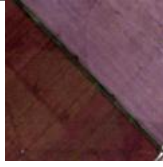



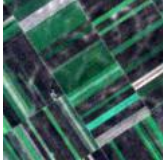












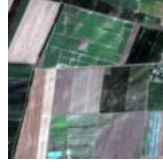



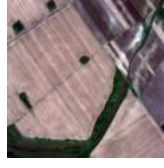

#### 4.4 Visual Representations of Retrieved Images

Table 4.5: Visual results on retrieved 20 Sentinel-1 images by Sentinel-1 query under various hash length bits in MSE based approach

Query					
	1st	5th	10th	15th	20th
<b>8 Bits</b>					
<b>16 Bits</b>					
<b>32 Bits</b>					
<b>64 Bits</b>					
<b>128 Bits</b>					


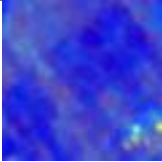
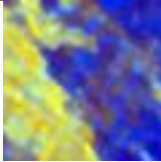
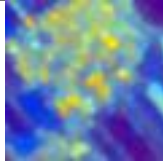
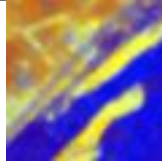
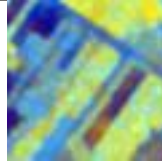
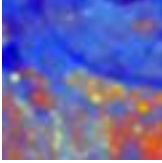
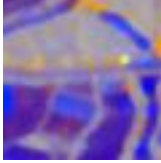
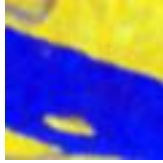
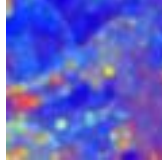
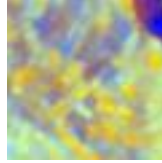
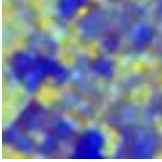
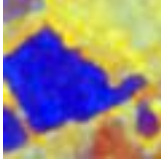
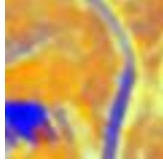
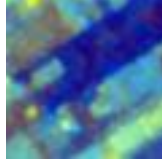
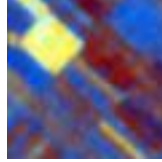
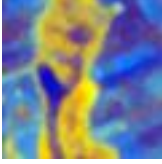
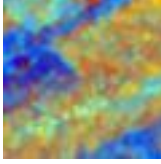
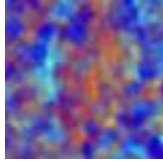
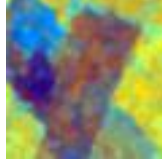
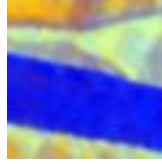
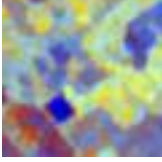
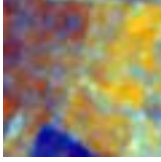
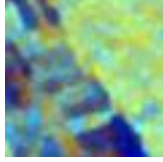
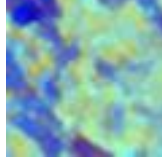

#### 4 Experimental Results

Table 4.6: Visual results on retrieved 20 Sentinel-2 images by Sentinel-1 query under various hash length bits in MSE based approach

Query					
	1st	5th	10th	15th	20th
<b>8 Bits</b>					
<b>16 Bits</b>					
<b>32 Bits</b>					
<b>64 Bits</b>					
<b>128 Bits</b>					








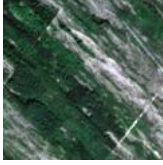














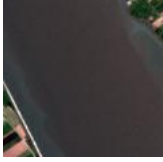

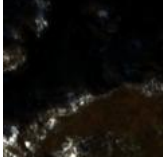

#### 4.4 Visual Representations of Retrieved Images

Table 4.7: Visual results on retrieved 20 Sentinel-1 images by Sentinel-2 query under various hash length bits in MSE based approach

Query					
	1st	5th	10th	15th	20th
<b>8 Bits</b>					
<b>16 Bits</b>					
<b>32 Bits</b>					
<b>64 Bits</b>					
<b>128 Bits</b>					

#### 4 Experimental Results

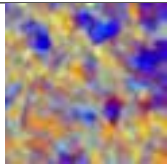
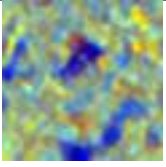
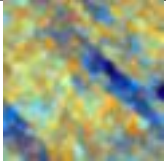
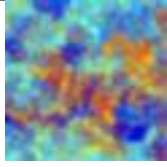
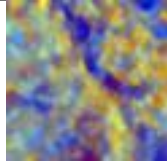
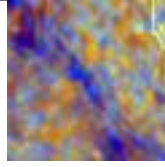
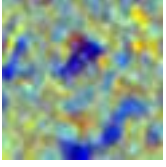
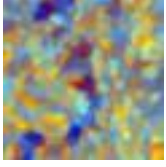
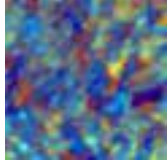
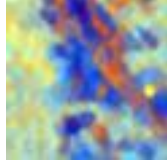
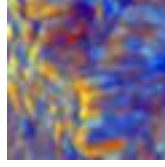
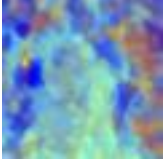
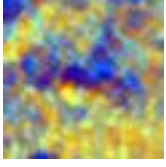
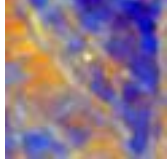
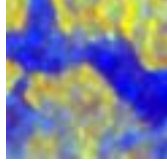
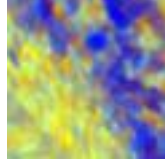
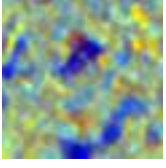
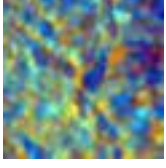
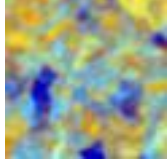
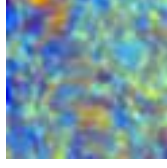
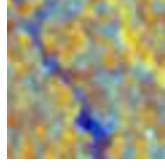
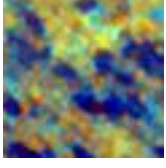
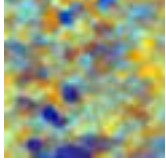
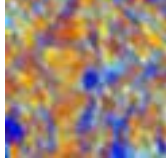
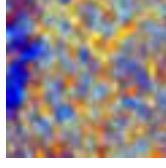
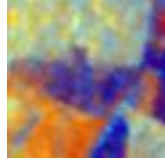
Table 4.8: Visual results on retrieved 20 Sentinel-2 images by Sentinel-2 query under various hash length bits in MSE based approach

Query					
	1st	5th	10th	15th	20th
<b>8 Bits</b>					
<b>16 Bits</b>					
<b>32 Bits</b>					
<b>64 Bits</b>					
<b>128 Bits</b>					



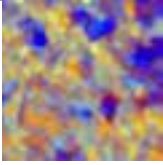


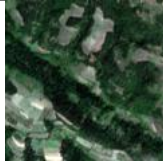







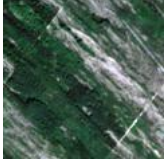







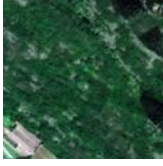






#### 4.4 Visual Representations of Retrieved Images

Table 4.9: Visual results on retrieved 20 Sentinel-1 images by a Sentinel-1 query as employed a neural network trained with triplet loss under various hash length bits

Query					
	1st	5th	10th	15th	20th
<b>8 Bits</b>					
<b>16 Bits</b>					
<b>32 Bits</b>					
<b>64 Bits</b>					
<b>128 Bits</b>					


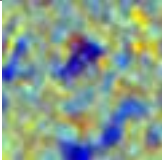
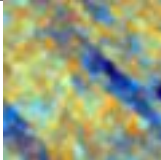
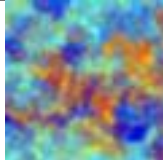
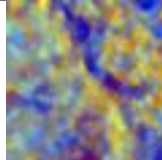
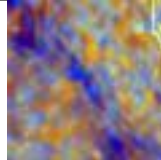
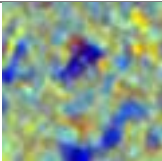
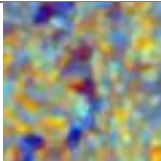
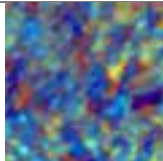
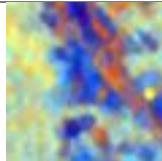
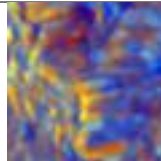
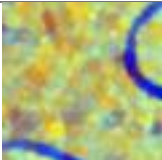
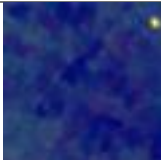
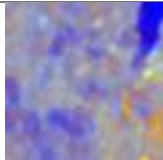
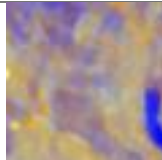
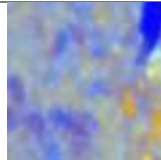
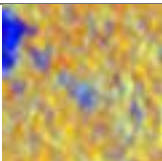
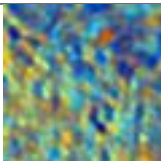
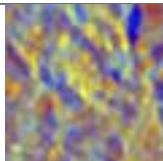
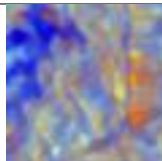
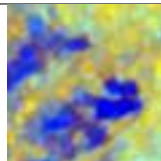
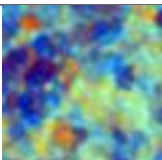
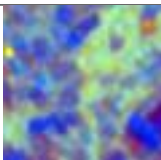
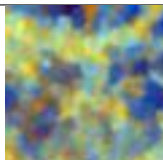
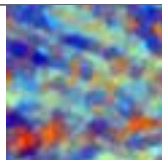
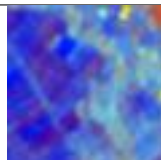
#### 4 Experimental Results

Table 4.10: Visual results on retrieved 20 Sentinel-2 images by a Sentinel-1 query as employed a neural network trained with triplet loss under various hash length bits

Query					
	1st	5th	10th	15th	20th
<b>8 Bits</b>					
<b>16 Bits</b>					
<b>32 Bits</b>					
<b>64 Bits</b>					
<b>128 Bits</b>					

#### 4.4 Visual Representations of Retrieved Images









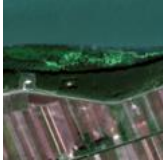








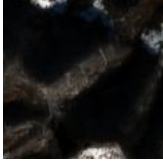








Table 4.11: Visual results on retrieved 20 Sentinel-1 images by a Sentinel-2 query as employed a neural network trained with triplet loss under various hash length bits

Query					
	1st	5th	10th	15th	20th
8 Bits					
16 Bits					
32 Bits					
64 Bits					
128 Bits					



#### 4 Experimental Results

Table 4.12: Visual results on retrieved 20 Sentinel-2 images by a Sentinel-2 query as employed a neural network trained with triplet loss under various hash length bits

Query					
	1st	5th	10th	15th	20th
<b>8 Bits</b>					
<b>16 Bits</b>					
<b>32 Bits</b>					
<b>64 Bits</b>					
<b>128 Bits</b>					



# 5 Conclusion and Future

## 5.1 Conclusion

A content-based image retrieval system has been implemented with a deep hashing technique to get semantically similar images in a multi-modality remote sensing data source in this thesis. This multi-modality structure has two subheadings. One of them is intramodality, which means query and retrieved images are in the same modality. The other one is intermodality, which means query and retrieved images are in different modalities.

If images are hashed to binary codes while preserving label similarities, a low storage required and fast CBIR system can be obtained because the distance between hashed binary codes is calculated by Hamming distance quickly, and hashed binary codes need much less storage than original images. Consequently, our CBIR system has been created in a hashing based way.

In order to hash images to binary codes, feature descriptors are required. Feature descriptors of the images have been computed by using convolutional neural networks. It has provided a more optimized learning mechanism than hand-crafted feature descriptors. ResNet50 architecture has been followed to build neural networks per modality. A network has been created for Sentinel-1 images, and a network has been created for Sentinel-2 images. These deep neural networks have been trained with two different loss functions, and these two approaches have been implemented to present an effective multi-modality CBIR system in remote sensing. These approaches are:

- Multi-modality hashing with mean square error loss
- Multi-modality hashing with triplet loss

The same neural network architectures have been employed in both approaches. Binarization and balancing loss functions have been added to the next of the primary loss function.

Mean square error loss has been adapted to the problem of this study. In this approach, neural networks have been feed by the pairs of data. Cosine similarities of the binary codes of the images should have been near to cosine similarities of the class labels of the images. Mean square error has calculated the distance between these two cosine similarities.

Although the Mean square error-based approach has produced favorable results in some particular cases, using MSE to train CBIR systems' neural networks is not the best way. Triplet loss calculates similarities among triple images, not a pair of images like in MSE, and it makes better performance than MSE loss. In this approach, images have been categorized as the anchor, positive and negative. A positive image means the image has the most number of shared labels with the anchor in the batch, and a negative image means the image has the least number of shared labels with the anchor. The purpose of the triplet loss keeps the anchor's and positive's binary codes closer while maintaining a distance between the anchor and the negative binary codes.

## 5 Conclusion and Future

Both approaches have been trained, validated, and tested with various length of hash bits and 20, 50 retrieved images per query. The length of hash bits is 8, 16, 32, 64, and 128. Evaluation results have been shared, and as estimated, the triplet loss-based approach outperforms than the MSE-based approach. Besides evaluation metrics, visual representations of all cases in both methods have been shared with 20 retrieved images per query.

### 5.2 Future Work

Although proposed methods, especially triplet loss-based, succeed in CBIR of multi-modality remote sensing dataset, some directions have been shared to improve the study in future research.

The results could not be compared with the state of the art methods because most previous multi-modality hashing studies used computer vision datasets. Using our data source in these studies is not possible because of the differences in neural network structure. While multi-modality hashing in computer vision uses image-text datasets, Sentinel-1 and Sentinel-2 images have been used in this study. This difference also affects the architecture of the neural networks. Significant modifications are needed to use our data source in a computer vision study, but it means changing almost all parts of the related study. That is why it has not been done. There are a few papers for multi-modality hashing in remote sensing, but they did not share the source codes. Therefore, using their solutions with our data source is not possible. Getting the results of other studies with our data source and comparing them with our methods' results would be a valuable contribution.

ResNet50 architecture has been followed to build neural networks for Sentinel-1 and Sentinel-2 images. ResNet also has deeper architectures like ResNet101 and ResNet152-layer. These deeper architectures or completely new architectures can be implemented to see other architectures' performance in the future.

In addition to mAP and WAP, Normalized Discounted Cumulative Gains (NDCG) and Average Cumulative Gains(ACG) can also be used as evaluation metrics to increase the variety in the results of the study.

# Bibliography

- [1] Yue Cao et al. “Deep visual-semantic hashing for cross-modal retrieval”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 1445–1454.
- [2] Zhen-Duo Chen et al. “Dual deep neural networks cross-modal hashing”. In: *AAAI*. 2018.
- [3] Tat-Seng Chua et al. “NUS-WIDE: a real-world web image database from National University of Singapore”. In: *Proceedings of the ACM international conference on image and video retrieval*. 2009, pp. 1–9.
- [4] Cheng Deng et al. “Triplet-based deep hashing network for cross-modal retrieval”. In: *IEEE Transactions on Image Processing* 27.8 (2018), pp. 3893–3903.
- [5] Venice Erin Liong et al. “Cross-modal deep variational hashing”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4077–4085.
- [6] Hugo Jair Escalante et al. “The segmented and annotated IAPR TC-12 benchmark”. In: *Computer vision and image understanding* 114.4 (2010), pp. 419–428.
- [7] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: [1512.03385 \[cs.CV\]](https://arxiv.org/abs/1512.03385).
- [8] HPC. TU Berlin HPC. URL: <https://hpc.tu-berlin.de/> (visited on 11/15/2020).
- [9] Mark J Huiskes and Michael S Lew. “The MIR flickr retrieval evaluation”. In: *Proceedings of the 1st ACM international conference on Multimedia information retrieval*. 2008, pp. 39–43.
- [10] George-Alexandru Ilie et al. “Spaceborne SAR tomography: application in urban environment”. In: (2011).
- [11] Qing-Yuan Jiang and Wu-Jun Li. “Deep cross-modal hashing”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3232–3240.
- [12] Pieter Kempeneers and Pierre Soille. “Optimizing Sentinel-2 image selection in a Big Data context”. In: *Big Earth Data* 1.1-2 (2017), pp. 145–158.
- [13] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [14] Hanjiang Lai et al. “Instance-aware hashing for multi-label image retrieval”. In: *IEEE Transactions on Image Processing* 25.6 (2016), pp. 2469–2479.
- [15] Chao Li et al. “Self-supervised adversarial hashing networks for cross-modal retrieval”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4242–4251.

## Bibliography

- [16] Tong Li, Sheng Gao, and Yajing Xu. “Deep multi-similarity hashing for multi-label image retrieval”. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 2159–2162.
- [17] Yansheng Li et al. “Large-scale remote sensing image retrieval by deep hashing neural networks”. In: *IEEE Transactions on Geoscience and Remote Sensing* 56.2 (2017), pp. 950–965.
- [18] Yansheng Li et al. “Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval”. In: *IEEE Transactions on Geoscience and Remote Sensing* 56.11 (2018), pp. 6521–6536.
- [19] Kevin Lin et al. “Learning compact binary descriptors with unsupervised deep neural networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1183–1192.
- [20] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.
- [21] *MultiSpectral Instrument (MSI) Overview*. ESA. URL: <https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument> (visited on 11/15/2020).
- [22] Kien Nguyen et al. “Iris recognition with off-the-shelf CNN features: A deep learning perspective”. In: *IEEE Access* 6 (2017), pp. 18848–18855.
- [23] Jose Costa Pereira et al. “On the role of correlation and abstraction in cross-modal multimedia retrieval”. In: *IEEE transactions on pattern analysis and machine intelligence* 36.3 (2013), pp. 521–535.
- [24] *Polarimetry*. ESA. URL: <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/product-overview/polarimetry> (visited on 11/15/2020).
- [25] Josiane Rodrigues, Marco Cristo, and Juan G Colonna. “Deep hashing for multi-label image retrieval: a survey”. In: *Artificial Intelligence Review* (2020), pp. 1–47.
- [26] Subhankar Roy et al. “Deep metric and hash-code learning for content-based retrieval of remote sensing images”. In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2018, pp. 4539–4542.
- [27] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: [10.1007/s11263-015-0816-y](https://doi.org/10.1007/s11263-015-0816-y).
- [28] *Sentinel-1*. ESA. URL: <https://sentinel.esa.int/web/sentinel/missions/sentinel-1> (visited on 11/15/2020).
- [29] *Sentinel-2*. ESA. URL: <https://sentinel.esa.int/web/sentinel/missions/sentinel-2> (visited on 11/15/2020).
- [30] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

- [31] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).
- [32] Blaž Sovdat et al. “Natural color representation of Sentinel-2 data”. In: *Remote Sensing of Environment* 225 (2019), pp. 392–402.
- [33] Shupeng Su, Zhisheng Zhong, and Chao Zhang. “Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3027–3035.
- [34] Gencer Sumbul, Jian Kang, and Begüm Demir. “Deep Learning for Image Search and Retrieval in Large Remote Sensing Archives”. In: *arXiv preprint arXiv:2004.01613* (2020).
- [35] Gencer Sumbul et al. “Bigearthnet: A large-scale benchmark archive for remote sensing image understanding”. In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 5901–5904.
- [36] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.
- [37] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.
- [38] Wei Wang et al. “Effective deep learning-based multi-modal retrieval”. In: *The VLDB Journal* 25.1 (2016), pp. 79–101.
- [39] Zheng Zhang et al. “Improved deep hashing with soft pairwise similarity for multi-label image retrieval”. In: *IEEE Transactions on Multimedia* 22.2 (2019), pp. 540–553.
- [40] Zheng Zhang et al. “Instance similarity deep hashing for multi-label image retrieval”. In: *arXiv preprint arXiv:1803.02987* (2018).
- [41] Fang Zhao et al. “Deep semantic ranking based hashing for multi-label image retrieval”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1556–1564.

# Appendix