UNSUPERVISED SEGMENTATION OF MULTISPECTRAL HIGH-RESOLUTION SATELLITE IMAGES

A Dissertation Submitted in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology in Artificial Intelligence

by

KAUSTUV RAY

04-03-06-10-51-21-1-19308

under the guidance of

PROF. RAHUL SINGH

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF SCIENCE

BENGALURU - 560012

June 2023

DECLARATION

I, **Kaustuv Ray (04-03-06-10-51-21-1-19308)**, hereby declare that the work presented in this Mtech Thesis report entitled **"Unsupervised Segmentation of Multispectral High-Resolution Satellite Images"** is the result of the work performed by me in the **Electrical Communication Engineering, Indian Institute of Science, Bengaluru**, under the supervision of **Prof. Rahul Singh** and that it has not been submitted elsewhere for any degree or diploma. In keeping with the general practice, due acknowledgments are made wherever the work is based on findings of other investigations.

Bengaluru - 560012 **Kaustuv Ray**

June 2023

CERTIFICATE

This is to certify that the work contained in this project report entitled **"Unsupervised Segmentation of Multispectral High-Resolution Satellite Images"** submitted by **Kaustuv Ray,** towards the partial requirement of **Master of Technology** in **Artificial Intelligence** has been carried out by him under my supervision at the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, and no part of it has not been previously submitted for a degree, diploma or any other qualification at this university or any other institution

Bengaluru- 560012 Prof.

June 2023 Rahul Singh

ACKNOWLEDGEMENT

I am grateful to my supervisor, Rahul Singh, for his guidance and support and to the faculty members for sharing their knowledge. I thank my family and friends for their constant support and encouragement. I dedicate this thesis to my parents, whose love and support have guided me throughout my academic journey.

Bengaluru - 560012 **Kaustuv Ray**

June 2023

ABSTRACT

Name of the student: **Kaustuv Ray**

Roll No: **04-03-06-10-51-21-1-19308**

Department: **Department of Electrical Engineering**

Degree for which submitted: **Master of Technology in Artificial Intelligence**

Thesis title: **Unsupervised Segmentation of Multispectral High-Resolution Satellite Images**

Thesis supervisor: **Rahul Singh**

Date of thesis submission: **June 2023**

Many supervised semantic segmentation methods rely heavily on a large-scale pixelwise annotated dataset, but it is time-consuming and laborious to provide manual annotation. The main aim of the project is to develop deep learning algorithms to perform semantic segmentation of high-resolution multispectral (HRMS) satellite images in an unsupervised manner.The segmentation results will be invariant to the domain-shift problem for the scenes belonging to different imaging conditions and satellite sensors. This will enable us to generate segmentation masks for a large number of cities situated in different geographical conditions.

Keywords: Segmentation, Unsupervised Segmentation, Satellite Images, Unsupervised Learning, Deep Learning

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A remarkable development in earth observation instruments has led to the generation of satellite images with improved spatial, spectral, and temporal resolutions. Consequently, there has been a high demand for "smart" identification and classification of land use and land cover maps from these images. Specifically, semantic segmentation is used to partition and classify the image into meaningful parts to later classify each part at the pixel level into one of the pre-defined classes such as buildings, roads, vegetation and other categories. Accurate identification of different types of objects in images helps significantly to keep the maps up to date; these maps can be further used for urban planning, environment monitoring, and disaster relief[1]. Traditional unsupervised segmentation

techniques rely heavily on domain expertise and consequently require human input. Moreover, substantial human intervention is also needed to generate training datasets for many current state-of-the-art supervised semantic segmentation techniques. This has created a wide gap between the vast archives of globally collected remotely sensed data and their corresponding semantically labeled image interpretation. The traditional unsupervised image segmentation techniques suit well for lower-resolution satellite data where the different "classes" of interest are well-separated spatially, for example, distinguishing features between water bodies and land masses. They are usually not robust for spatially mixed classes often encountered in higher-resolution satellite data. The biggest challenge in unsupervised segmentation is to simultaneously satisfy the following criteria (a) the same label should be assigned to pixels of similar features, and (b) the same label should be assigned to spatially continuous pixels.

Thus, researchers more often seek to find a plausible approach to assign labels, one that balances the aforementioned criteria well [2]. Although state-of-the-art segmentation methods are based on supervised deep learning techniques, they heavily rely upon clean train data in the form of image and pixel label pairs, and more often, the training gets limited to a particular geographical city location. Thus, these supervised models suffer from

domain generalization issues, as well as scaling up for the prediction of a new pixel class label. In this work, we experiment with unsupervised image segmentation techniques that extract pixel class context-based information using spatial and spectral textures generated by adjacent pixels. Recently, unsupervised and self-supervised learning have gained significant attention in machine learning. Such approaches have been devised for different problems, e.g., image clustering, video analysis, and change detection in Earth observation images [3]. While most deep-learning-based semantic segmentation methods are supervised, unsupervised semantic segmentation methods have been proposed in the literature exploiting deep clustering [4].

Chapter 2

Related Approaches

Different image segmentation methods proposed in the literature so far are based on features that are generated by using one of these techniques: (i) handcrafted features, (ii) features generated by unsupervised learning techniques, and (iii) features generated by deep neural networks. Generating good handcrafted features requires domain expertise and is often limited to classes that are well separated spatially. Hence, these are hard to scale. Deep learning-based methods are known to generate superior features [5], but they often require a "huge amount of" supervised training data. Since good training data is hard to generate and is often limited to moderate geographical coverage, these methods are, again, hard to scale.

The practicality of supervised methods is limited due to the difficulty in

acquiring labeled data. Unsupervised learning alleviates these limitations by learning semantic representations from unlabeled images without relying on predefined annotations. Sudipana et al. [6] use a self-supervised approach that exploits the idea of reducing the gap among feature representations of multiple views of the same image in an iterative manner without using any labeled data. They take two views of an image along with a third dissimilar image and get the respective feature vectors. Pseudo-labels are assigned to pixels by finding the channel with the highest value. The loss function takes into account the pseudo labels, the distance between the two representations, and also the disparity from the dissimilar image.

There are several attempts to carry out semantic segmentation without labels. IIC [7] extends mutual information-based clustering to pixel-level representation by generating a probability map over image pixels. IIC uses paired samples of an image and its randomly perturbed version using geometric and photometric transforms. Then the objective is to maximize the mutual information between the two images. This ensures that the model learns a representation that preserves what is common between the two images while discarding instance-specific details.

InfoSeg [8] leverage image-level representation learning for pixel-level segmentation. They use multiple high-level features, each capturing se-

mantically similar image areas. They use a two-step learning procedure. At each iteration, they perform a Segmentation and Mutual Information Maximization step. In the first step, images are segmented using the current features. In the second step, the features are updated based on the segmentation from the first step. Due to this procedure, InfoSeg does not require a pretrained backbone or labeled images.

STEGO [9] uses a transformer-based architecture for unsupervised semantic segmentation. They use features from an unsupervised pre-trained network and try to distill the features into a compact form.

They use a (Visual Transformer)ViT backbone and train a shallow network on top of it by using contrastive learning. The loss has three contrastive terms that distill connections between an image and itself, similar images, and random other images respectively.

Chapter 3

Methodology

3.1 Overview

We can consider segmentation as a function F that maps an image $I \in \mathbb{R}^{h \times w \times c}$ to its segmentation map$y \in \mathbb{R}^{h \times w \times l}$. h, w denote the height and width of the image, respectively and c, l denote the number of input channels and classes, respectively. In an unsupervised setting, we need to find this function F without using any ground truth.

3.2 Network Architecture

We consider two different architectures. Both architectures have two stream of network, the difference is present in the last layer. For the first architecture, the last layers are separate. This is the Disjoint network. In the second architecture, the last layers are shared, and this is the Joint network.

3.2.1 Disjoint Network

We follow a two-stream architecture. Each stream consists of a Fully Convolutional Network (FCN). We follow a similar architecture to [10]. We add a prediction layer to one of the networks. The larger network is called the online network, denoted as g', and the smaller network is called as target network, denoted as g. The model is shown in Fig 3.1. The original image I is given to g, and the augmented version I' is given to g'.

3.2.2 Joint Network

We use a two-stream FCN. This network is similar to [6]. Both streams of the network share the last layer. We will denote the two streams as g

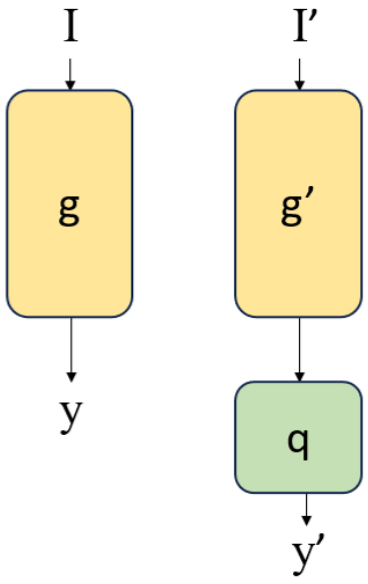

Figure 3.1: Disjoint Network Architecture

and g' and the last layer as q. g and g' consist of 5 convolutional layers, each followed by an activation layer and a batch normalization layer. q has a single convolutional layer followed by a batch normalization layer. The original image I is given to g, and the augmented image I' is given to g'. The model is shown in Fig 3.2.

3.3 Training Objective

The networks take I and I' as input, producing y and y' as output, where $y, y' \in R^{h \times w \times l}$. In the output, each pixel i has a l-dimensional vector

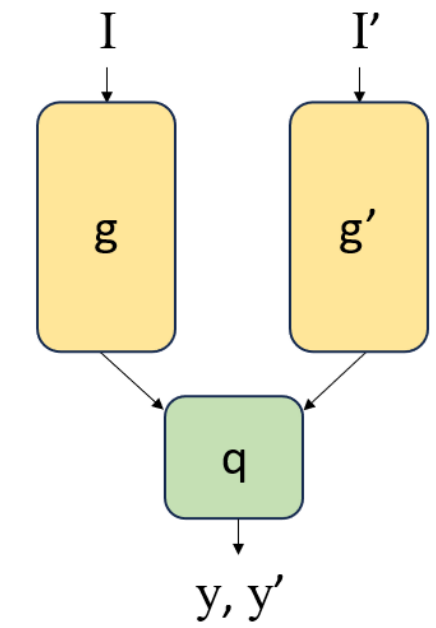

Figure 3.2: Joint Network Architecture

which is denoted as y_i. The class label for pixel i is denoted as c_i and is obtained by $c_i = \arg\max y_i$.

3.3.1 Geometric Equivariance

We want the network to be equivariant to geometric transforms. We denote a geometric transform as G and its inverse as G^{-1}. Let y be the output of the network for an image x, i.e., $F(x) = y$. Now if we give $G(x)$ as input, the network should give $G(y)$ as output, i.e., $F(G(x)) = G(y)$.

In our experiments, we have used random rotations of ninety degrees for geometric transforms. We used geometric equivariance on the augmented image only. The features for the augmented image I' are given below.

$$y' = G^{-1}F(G(I')) \tag{3.1}$$

3.3.2 Pseudo Labels

Since this is an unsupervised setting, we cannot use the ground truth. Instead, we consider the outputs of the network as pseudo labels and use them for training. We take the cross entropy loss between the l dimensional vector and its pseudo label. We find the mean loss over all pixels in the image, and over all images in the batch. We calculate this mean cross-entropy loss for both streams of the network.

$$\mathcal{L}_{P1}(y, c) = \frac{1}{N \times h \times w} \sum_{i,p} \mathcal{L}_{CE}(y_p[i], c_p[i]) \tag{3.2}$$

$$\mathcal{L}_{P2}(y', c') = \frac{1}{N \times h \times w} \sum_{i,p} \mathcal{L}_{CE}(y'_p[i], c'_p[i]) \tag{3.3}$$

3.3.3 Feature Similarity

The two images, I and I', correspond to the same location, so their features should be identical. We take the mean squared error between the features for each pixel in an image. Then we take the mean loss over all pixels in an image and all images in a batch.

$$\mathcal{L}_S(y, y') = \sum_{i,p} \|y_p[i] - y'_p[i]\|^2 \tag{3.4}$$

The total loss is gievn by the average of $\mathcal{L}_{P1}, \mathcal{L}_{P2}$ and $\mathcal{L}_S$.

3.4 Post processing

3.4.1 Mapping segments to Ground Truth

In a unsupervised setting the network cannot find the semantic class of a pixel, i.e., it cannot give the class name. It can only differentiate between different classes. So we need a method(Fig 4.5) to map the output segments to class labels. [6] use a method that maps segments to a label with maximum overlap. [7] use linear assignment for mapping. We have used a different method that considers the intersection over union(IoU)

between two segments. The method is given in Algorithm 1.

While mapping function uses ground truth labels, the mappings obtained are not used to train the network. Therefore our approach is unsupervised.

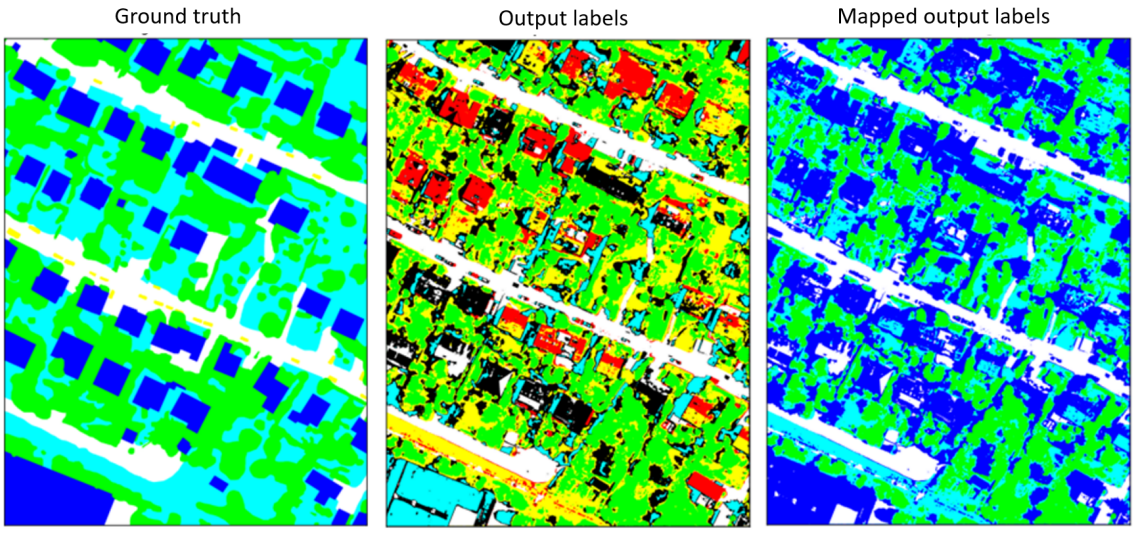

Figure 3.3: Mapping output labels to ground truth

3.4.2 Superpixel Correction

We want neighboring pixels to have the same class label. Towards this end, we use superpixels to improve the output segments [4]. Superpixels partition the image into multiple segments that have similar color values. The boundaries of superpixels also align better with the edges and corners present in an image.

Algorithm 1 Max mIoU matching

Input Ground truth l with M classes
Input Model output s with N classes
for $i \leftarrow 1$ to N **do**
  for $j \leftarrow 1$ to M **do**
    $a[j] \leftarrow$ number of pixels in intersection of i and j
    $b[j] \leftarrow$ number of pixels in union of i and j
    $c[j] \leftarrow \frac{a}{b}$
  end for
  $t \leftarrow \arg\max_{j \in M} c$
  `Map prediction class` i `to label` t
end for

For a given image, we extract K superpixels, here K is a large number. Then we constrain all the image pixels within a superpixel to have the same class label. The superpixel is assigned the label that occurs the most inside it. The method is given in Algorithm 2.

We have used SLIC[11] to extract superpixels, with $K = 10,000$ segments in our experiments. Fig 3.4 shows an example of superpixel correction.

Algorithm 2 Superpixel Correction

Input: Predicted Segmentation I containing M classes
Input: Number of Segments K
$S \leftarrow SLIC(I, K)$
for $k \leftarrow 1$ to K **do**
  $P_k \leftarrow$ indices of pixels present in k^{th} superpixel
  $t \leftarrow \arg\max_{i \in M} |c_i|$
  `Assign class` t `to` k^{th} `superpixel`
end for

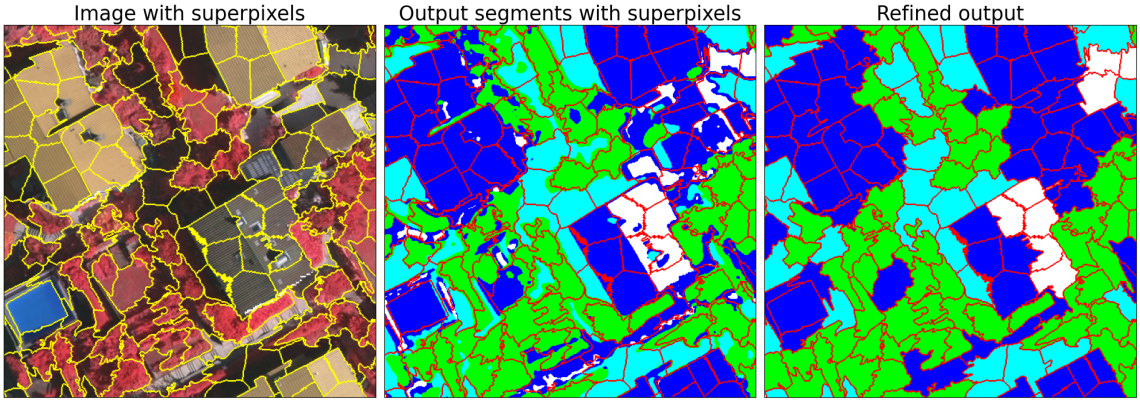

Figure 3.4: Superpixel correction

Chapter 4

Experiments

4.1 Datasets

We used the data from ISPRS Test Project on Urban Classification and Semantic Labelling [12]. We used images from two cities – Vaihingen and Potsdam, both areas covering urban scenes. The ground truth has six classes- Impervious surfaces, buildings, low vegetation, trees, cars, and background.

The Vaihingen dataset has 33 scenes with an average size of 2500×2000 pixels, with a ground sampling distance of 9cm. Vaihingen images have 3 channels NIR-R-G(near-infrared, red, green). Potsdam has 38 scenes of size 6000×6000 pixels, with a ground sampling distance of 5cm. The Pots-

dam images are available in R-G-B(red, green, blue) and R-G-B-IR(red, green, blue, infrared).

The class distributions are given in Fig 4.1, 4.2.

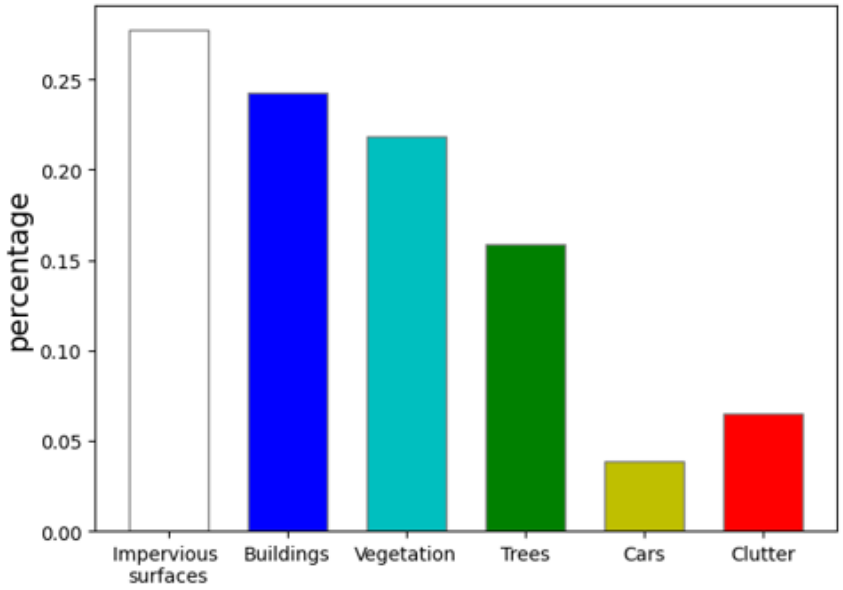

Figure 4.1: Potsdam class distribution

We chose a patch size of 448×448. We note that the Potsdam dataset has a higher resolution than Vaihingen. To cover the same area in each patch, we divide Potsdam scenes into patches of size 1000×1000 and resize them to 448×448.

We also merge the six classes into buildings, vegetation, and impervious surfaces to obtain Potsdam RGB-3 and Potsdam RGBIR-3, which are three-class variants of the Potsdam datasets.

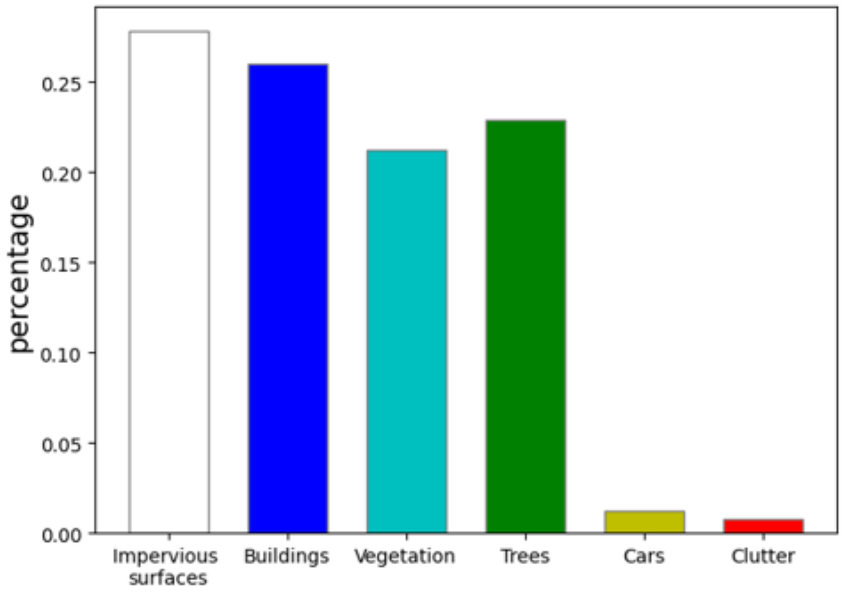

Figure 4.2: Vahingen class distribution

4.2 Implementation Details

4.2.1 Training

We have used He initialization[13] and SGD optimizer with a momentum of 0.9 for both networks. For the joint network, the learning rate is 0.001. The disjoint network uses two different learning rates for the online and target networks. The target network g uses a higher learning rate of 0.002, and the online network g' uses a learning rate of 0.001. We have used L2 regularizer with a penalty of 0.001. We run the experiments for four epochs.

We used the PyTorch library[14] for implementation. All experiments were carried out on an NVIDIA RTX A4000 GPU.

4.2.2 Augmentations

As we use datasets with multi-spectral images, we require augmentations that are independent of the number of channels. To get the augmented image I' we used channel-wise random jittering in brightness and contrast. We also blackout small portions of the image so that the model can learn the structures present in the image. We remove N square patches of side $p \times p$. We have taken $N = 70$ and $p = 20$.

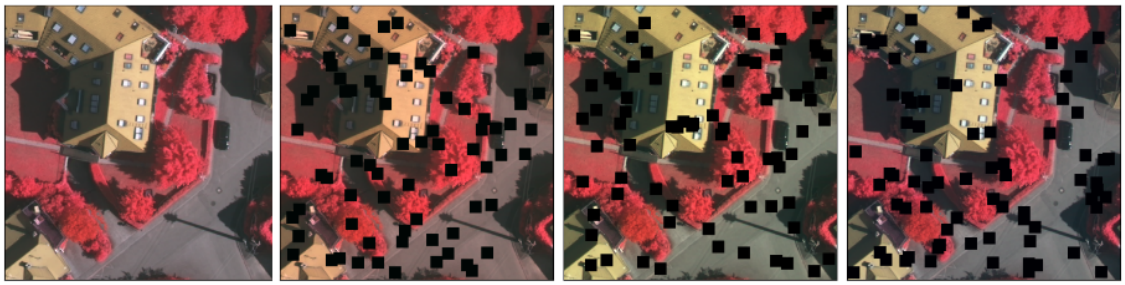

Figure 4.3: Examples of augmentations

4.3 Evaluation

We evaluate our models based on accuracy and mean intersection over union(mIoU). We note that accuracy is not a good performance measure because the dataset has a class imbalance, and the network can get a good accuracy by simply learning a single class.

4.4 Results

We compare our model with [6], IIC[7], InfoSeg[8] and STEGO[9]. The results are given in Tables 4.1 and 4.2.

[6] is the only paper that gives the results in mIoU. Both our models perform better than [6] in both mIoU and accuracy.

On the Potsdam RGBIR dataset, the models perform better than IIC in terms of accuracy. STEGO outperforms our model on the Potsdam RGBIR dataset.

From the tables, we can observe that the Joint model performs better than the Disjoint model in all cases. Having more channels in training improves performance.

	Vaihingen	Pot-RGB-3	Pot-RGB	Pot-RGBIR-3	Pot-RGBIR
Disjoint Network	37.5	41.5	26.9	45.5	28.6
Joint Network	39.2	46.0	30.5	49.1	31.1
Sudipan et. al.	33.4	-	-	-	-

Table 4.1: Comparision on mIoU

	Vaihingen	Pot-RGB-3	Pot-RGB	Pot-RGBIR-3	Pot-RGBIR
Disjoint Network	58.2	59.1	44.9	63.4	48.4
Joint Network	63.0	63.2	52.7	66.1	54.9
Sudipan et. al.	48.5	-	-	-	-
IIC	-	-	-	65.1	45.4
InfoSeg	-	-	-	71.6	57.3
STEGO	-	-	-	-	77.0

Table 4.2: Comparison on accuracy

The outputs of the Joint network are shown in Fig 4.4.

4.4.1 Oversegmentation

We experiment with different numbers of output classes. The results are given in Table 4.3 We observe that having more segments is beneficial. This is because the model cannot exactly identify the semantic classes. By oversegmenting some of the mistakes made by the model are rectified. We observe that having eight output segments is the best. Increasing the output labels beyond 8 affects the performance negatively.

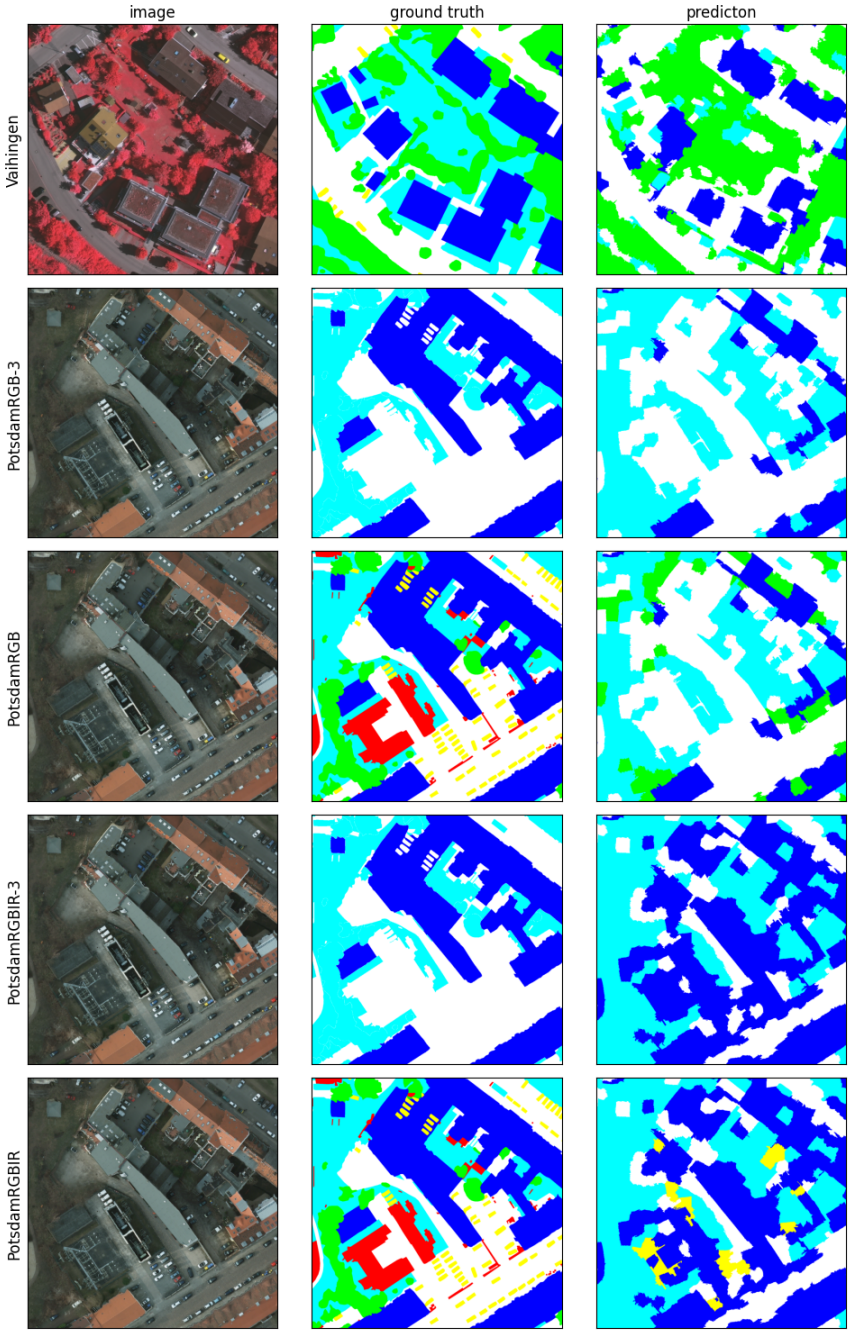

Figure 4.4: Joint Network Outputs on different datasets

number of output labels	mIoU	accuracy
6	25.0	49.2
8	39.2	63.0
10	35.1	58.8

Table 4.3: Comparing different number of output labels

	Max mIoU	Max overlap	Linear Assignment
Joint Network	39.2	37.6	36.6
Disjoint Network	36.0	34.9	35.4

Table 4.4: Comparing different mapping functions on Vaihingen dataset

4.4.2 Mapping functions

We evaluated different mapping functions on the Vaihingen dataset using the Joint and Disjoint models. Table 4.4 shows the mIoU for combinations of mapping functions and networks. Fig 4.5 compares the outputs of different mapping functions using Joint model on the Vaihingen dataset. From the figures and tables we can conclude that our mapping function is the best.

4.4.3 Ablation Studies

We perform the following ablation studies on the Joint model to understand the contribution of individual components. We want the contributions of the individual terms of the loss function. We also want to see the

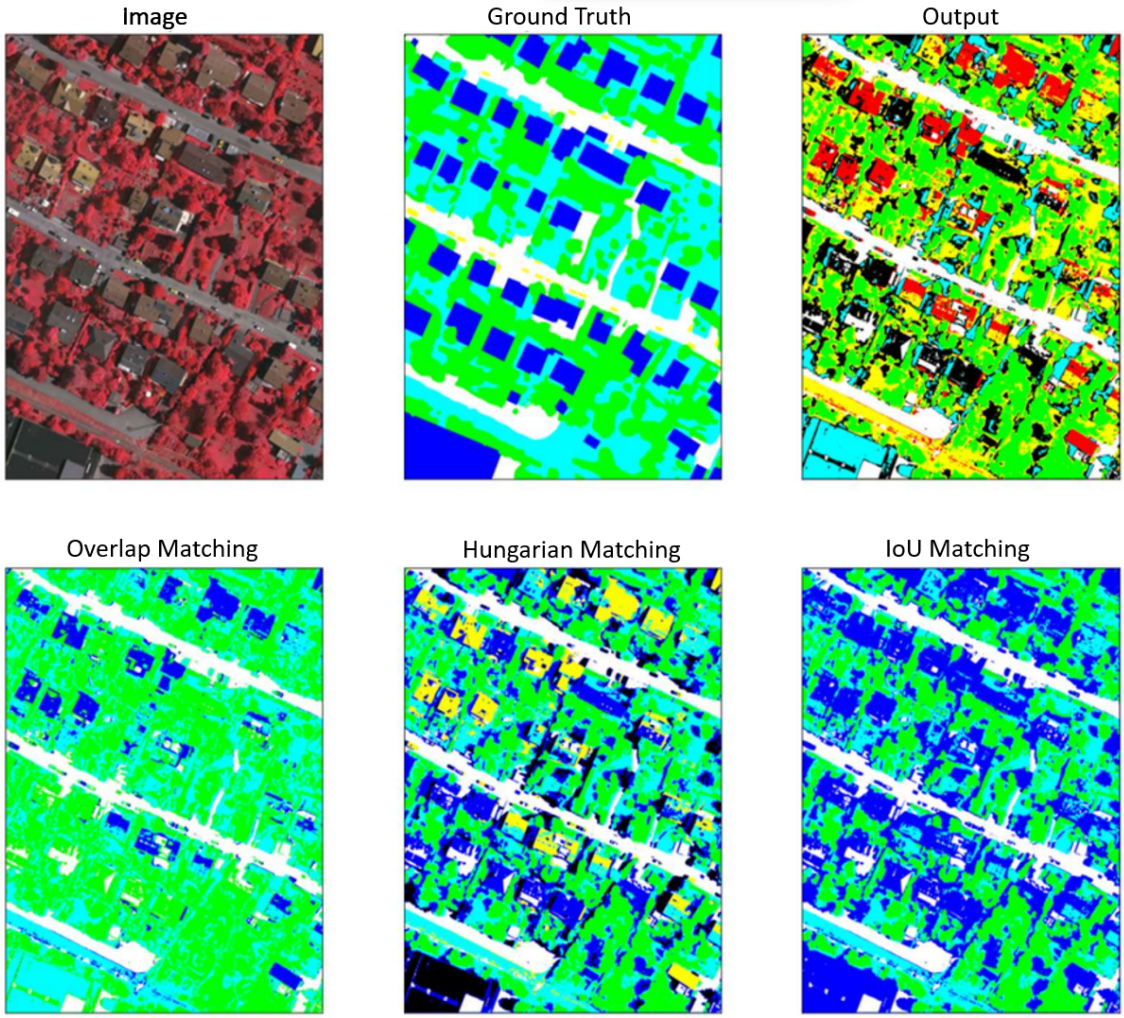

Figure 4.5: Joint Model using different Mapping functions on Vaihingen dataset

effect of superpixel correction and He initialization. The results are given in Table 4.5.

We see that pseudo-label loss has the highest contribution to the performance. Superpixel also adds to the mIoU. All other factors have small contributions.

	Joint Network
original	39.2
feature similarity loss	38.1
invariance loss	38.5
pseudo loss	29.9
superpixel	36.6
He Initialization	38.2

Table 4.5: Ablation studies

Chapter 5

Conclusion and Future work

Our work aimed to create a model that performed the performance in unsupervised image segmentation. We experimented with two different architectures. We found that Joint architecture performs better than the disjoint architecture. We also found that model performance increases when trained on images with more channels. u

We can always do hyperparameter tuning to find optimize our models. We will use Linear probing [15] to evaluate the quality of representations learned by our model. Linear probing is used for evaluating Unsupervised methods. Using different self-supervised approaches like [10] [16] [17], may help in learning better representations, and also increase the unsupervised performance. Vision Transformers(ViT)[18] have emerged as a

new approach in representation learning for computer vision tasks. They have achieved competitive or even state-of-the-art results on benchmark datasets. We can use ViT as a backbone and follow an unsupervised approach to learn dense representations.

Bibliography

[1] Vladimir Iglovikov, Sergey Mushinskiy, and Vladimir Osin. *Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition*. 2017. arXiv: 1706.06169 [cs.CV].

[2] Wonjik Kim, Asako Kanezaki, and Masayuki Tanaka. "Unsupervised Learning of Image Segmentation Based on Differentiable Feature Clustering". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 8055–8068. DOI: 10.1109/TIP.2020.3011269.

[3] Sudipan Saha, Patrick Ebel, and Xiao Xiang Zhu. "Self-Supervised Multisensor Change Detection". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–10. DOI: 10.1109/TGRS.2021.3109957.

[4] Asako Kanezaki. "Unsupervised Image Segmentation by Backpropagation". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 1543–1547. DOI: 10.1109/ICASSP.2018.8462533.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[6] Sudipan Saha et al. “Unsupervised Single-Scene Semantic Segmentation for Earth Observation”. In: *IEEE Transactions on Geoscience and Remote Sensing* (2022).

[7] Xu Ji, João F. Henriques, and Andrea Vedaldi. *Invariant Information Clustering for Unsupervised Image Classification and Segmentation*. 2019. arXiv: 1807.06653 [cs.CV].

[8] Robert Harb and Patrick Knöbelreiter. *InfoSeg: Unsupervised Semantic Image Segmentation with Mutual Information Maximization*. 2021. arXiv: 2110.03477 [cs.CV].

[9] Mark Hamilton et al. *Unsupervised Semantic Segmentation by Distilling Feature Correspondences*. 2022. arXiv: 2203.08414 [cs.CV].

[10] Jean-Bastien Grill et al. *Bootstrap your own latent: A new approach to self-supervised Learning*. 2020. arXiv: 2006.07733 [cs.LG].

[11] Radhakrishna Achanta et al. “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282. DOI: 10.1109/TPAMI.2012.120.

[12] F. Rottensteiner et al. *The ISPRS benchmark on urban object classification and 3D building reconstruction*. 2012.

[13] Kaiming He et al. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015. arXiv: 1502.01852 [cs.CV].

[14] Adam Paszke et al. “Automatic differentiation in PyTorch”. In: (2017).

[15] Mark Hamilton et al. *Unsupervised Semantic Segmentation by Distilling Feature Correspondences*. 2022. arXiv: 2203.08414 [cs.CV].

[16] Mathilde Caron et al. *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*. 2021. arXiv: 2006.09882 [cs.CV].

[17] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG].

[18] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

[19] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. 2016. arXiv: 1511.00561 [cs.CV].

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[21] Liang-Chieh Chen et al. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. 2017. arXiv: 1606.00915 [cs.CV].

[22] Hengshuang Zhao et al. *Pyramid Scene Parsing Network*. 2017. arXiv: 1612.01105 [cs.CV].

[23] Hengshuang Zhao et al. *Pyramid Scene Parsing Network*. 2017. arXiv: 1612.01105 [cs.CV].