Technische Universität München
TUM School of Engineering and Design
Data Science in Earth Observation

Masked Image Modeling for Representation Learning in Earth Observation

Author: Hugo Hernández Hernández

Master Thesis

Earth Oriented Space Science and Technology - ESPACE

Supervisors:
Conrad M. Albrecht
Yi Wang
Xiaoxiang Zhu

February 6, 2023

Technische Universität München
TUM School of Engineering and Design
Data Science in Earth Observation

Masked Image Modeling for Representation Learning in Earth Observation

Author: Hugo Hernández Hernández

Master Thesis

Earth Oriented Space Science and Technology - ESPACE

Supervisors:
Conrad M. Albrecht
Yi Wang
Xiaoxiang Zhu

February 6, 2023

I confirm that this master thesis is my own work and I have documented all sources and material used.

Munich, February 6, 2023 Hugo Hernández Hernández

Acknowledgments

First of all, I am very grateful to my supervisor Yi Wang who has been my main guidance and support through the last year of my research career, his guidance has allowed me to get a better picture of my goals, the research panorama, and how to to be clearer in my ideas. I would also express my gratitude with my supervisor Conrad Albrecht who even though all the difficulties he has provided clear advises and comments over how to present and organize scientific ideas. I would like to thank Prof. Dr. Xiaoxiang Zhu, whom with her innovative ideas has been a guidance for my research.

I would like to express my thanks to all people from DLR Earth Observation Centre, which whom I have hold interesting discussions about the ongoing projects and researches concerning data science in Earth observation, and I have realized the potential for future works and new ideas that could be the preamble of a new era in remote sensing.

I would like to thank Large-scale Data Mining in EO group from Helholtz AI Cooperation Unit for giving me the opportunity to develop a thesis work with them, meanwhile, I also extend my thanks to the Helmholtz Association's Initiative and Networking Fund on the HAICORE@FZJ partition for supporting my work with their computational resources. I would like to thank to the scholarship program DAAD-CONACyT for its financial and integrative support during 2 years, hoping they will continue supporting people like me to achieve their goals.

I extend special thanks to my family whom gave me their ground values and support to overcome any difficulty no matter if I am on the other side of the world, meanwhile, I thanks to my friends whom make me realize it takes long time to ensure who will be there in bad times and is willing to support me.

Finally, thanks to TUM ESPACE master's program, now I have a better picture of the world and it has to be changed. I can state that bubbles shall be broken, inequality shall be reduced, elitism shall be eradicated.

This is just the beginning and our side has to win.

Abstract

Deep learning has been widely used in the field of Earth observation *(EO)* and brought impressive outcomes. However, one big challenge in EO is the contradiction between rapidly increasing data volume and limited annotation resources. To tackle this issue and make use of large-scale unlabeled data, self-supervised representation learning *(SSL)* has been developed within the years. This methodology is focused on learning useful representations by itself with few or without human intervention from immense and unlabeled datasets. Various types of self-supervision have been studied, among which recent Masked Image Modelling *(MIM)* has shown great potential. The principle of MIM is masking out a defined ratio of an input image, and training a model to predict the masked patches from visible ones. The learned encoders can then be transferred to downstream tasks to extract good data representations.

In this thesis, we explore a new approach of MIM in EO with the combination of two architectures presented in the state of the art: Masked Autoencoder *(MAE)*, which masks and reconstructs the raw input image with an asymmetrical structure to increase the efficiency; and Masked Feature Prediction *(MFP)*, where image feature descriptors are seen as reconstructing targets. The proposed approach is performed on an EO dataset for pretraining, and evaluated on a classification downstream task. Experimental results show an optimal reconstruction of the images in multispectral domain and promising downstream performance in scene classification.

Contents

In memory of our ancestors, for the generations to come

"Herman Melville, in Moby Dick, spoke for wanderers in all epochs and meridians:"I'm tormented with an everlasting itch for things remote. I love to sail forbidden seas..."

"Maybe it's a little early. Maybe the time is not quite yet. But those other worlds-promising untold opportunities-beckon.

Silently, they orbit the Sun, waiting."

Carl Sagan, Pale Blue Dot: A Vision of the Human Future in Space

1 Introduction

1.1 Motivation

Recently, the main option to deal with image-oriented databases (Deng et al., 2009) was using human-annotated datasets with methods like supervised learning, however, this is costly and time-consuming. To make use of unlabeled data, self-supervised learning has become within the years a useful tool to achieve the reduction of human supervision in computer vision.

Earth observation presents the challenge of dealing with massive-scale datasets and limited annotation resources, thus, self-supervised representation learning *SSL* has been introduced to address this issue (Wang et al., 2022b) (Zhu et al., 2017). The idea is characterized by the concept of the model can learn by itself without human intervention (LeCun et al., 2015).

1.2 Background: Self-Supervised Learning

Self-supervised learning comes from the concept of representation learning, this representation of data make a task easier to extract useful information when building predictors (Bengio et al., 2012) (Ericsson et al., 2022).

The idea of self-supervised learning *(SSL)* in computer vision comes from the necessity of learning increasingly rich semantic features of images. This is performed by relying on large-scale annotated datasets which require manual annotation, thus, expensive and time-consuming tasks to be performed (Wang, 2022). The main idea behind SSL is the capability of the model to learn by itself how to label the massive amount of data which it is feed with. It is not necessary to perform labeling manually when the method provide already this skill. In the computer vision process, it is necessary to extract image features used in different output tasks. Similarly, is the case with remote sensing where, after the features acquisition step a sort of tasks are performed such as

classification, segmentation, object identification, etc.

In both cases SSL has demonstrated a good performance when addressing the drawbacks of supervised learning: poor feature generalization, vulnerability to attacks, manually labeled data, etc. (Wang, 2022), mainly because it works with unlabeled images. While large databases leads to noisy labels which affect the model generating bias, small ones with good quality tend to lead to overfitting (Wang et al., 2022b). Therefore, SSL could lead to successfully address this challenges in remote sensing.

The general pipeline of SSL is divided into two phases: the pre-training task and downstream task as it is illustrated in Figure 1.1.

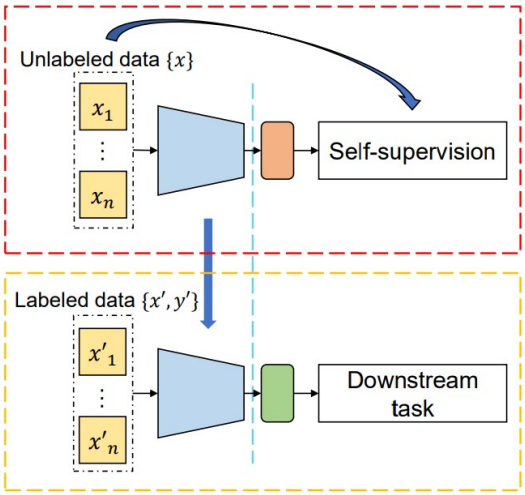


Figure 1.1 Self-Supervised Learning Pipeline (Wang et al., 2022b)

1.2.1 Pre-training Task

During pre-training task, the model is trained using a dataset with similar characteristics of the one used for the downstream task, a land-usage oriented dataset, this with the goal to learn similar characteristics for the labels during the classification task. This stage uses a specific model and architecture which their main task is to extract image features. As it can be seen in Figure 1.1, a model is feed with a set of unlabeled data x, then it is processed by the model which generates as an output a visual representation learned through self-supervision later to be used in the next stage.

In the current work, an Earth oriented dataset is used as a pretext task for pre-training: self-supervised learning for Earth Observation *SSL4EO-s12* (Wang et al., 2022c), which

relies on spaceborne imagery acquired by Sentinel 1/2 missions.

1.2.2 Downstream task

During the downstream task, the process transfers what does the model has learned to a smaller dataset with similar characteristics with the one it was pre-trained, usually, specialized architectures are used to extract image features to be transferred. These architectures could be convolutional neural networks (Lecun et al., 1998), (LeCun et al., 1989) or vision transformers (Dosovitskiy et al., 2020), then an evaluation of the transfer learning task is performed. In remote sensing the tasks are presented in a wide variety of options such as classification or segmentation, in the present work we perform a classification task.

1.2.3 Categorization of Self-Supervised Learning

Self-supervised learning is divided into three different methods: generative, contrastive and predictive. Generative methodologies are oriented to reconstruct or generate from input data by teaching the model how to perform the task, contrastive methods perform a maximization of the of the similarity between the augmented inputs, and predictive methods allow the model to predict self-generated labels (Wang et al., 2022b).

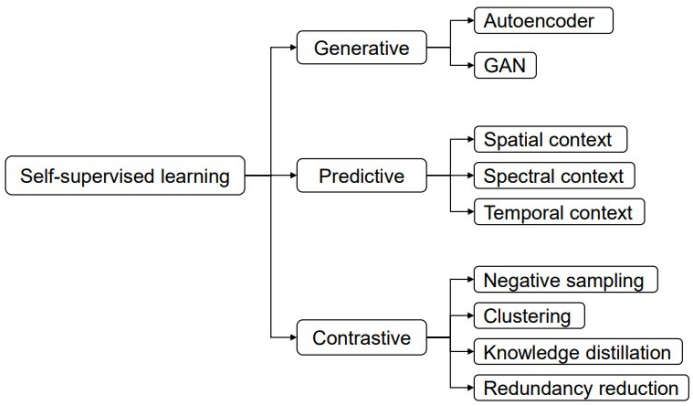


Figure 1.2 Categorization of self-supervised learning (Wang et al., 2022b)

Generative Methods

In the present work we focus on generative methods, which are focused on learning representations by reconstruction from the input data. They train an encoder to encode input x into an explicit vector z, and a decoder to reconstruct x from z (Liu et al., 2020), they are divided into two categories: autoencoders *(AE)* and generative adversarial networks *(GAN)*. For AE models, the objective is to reconstruct inputs from corrupted inputs, while for GAN, the training process is divided into two parts, where one generates fake samples while the other one tries to distinguish them from real ones.

One of the backbones of this thesis are the autoencoders which were first defined in (Ballard, 1987) as part of the work on modular learning networks. They are defined as a group of hierarchies that have a compact encoding of input-output pairs to take advantage of the backpropagation algorithm. Autoencoders are oriented to dimensionality reduction and their traditional architecture is set as a feed-forward neural network which has as a goal to produce its input as the output layer (Liu et al., 2020). They train a defined encoder E to map input x to a latent vector $z = E(x)$, and a decoder D to reconstruct $x = D(z)$ from z. In this context a joint function $D \circ E$ is specified to contribute to a self-supervised loss:

$$\|x - D(E(x))\| \tag{1.1}$$

As the encoder-decoder structure may lead into trivial solutions such as $E = D = 1$, it may be necessary to apply strategies to prevent trivialities. In the case of the Autoencoders, the dimensionality reduction is usually used (Wang et al., 2022b) regarding its application, constraining a small dimension z is not a must to prevent identity mapping. It is also an option to construct $D \circ E$ such the dimension of z is greater that the input's x dimension with additional sparsity constrains, this gives its name as sparse autoencoder.

A basic structure of an autoencoder can be visualized in Fig. 1.3, it is a densely connected neural network with two parts, one working as an encoder and the second towards the output as a decoder. The AE architecture used in the present work involves a bigger complexity but shares the same principles.

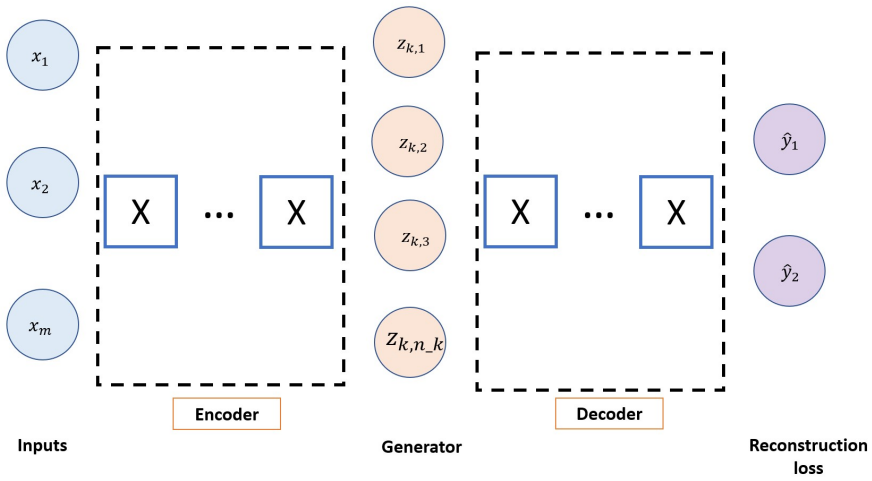


Figure 1.3 Basic Autoencoder Architecture

1.2.4 Self-Supervised Learning and Supervised Learning

Self-supervised learning and supervised learning share characteristics in common as well as differences, in Table 1.1 we present their characteristics. It is worth to mention that, the main difference is that SSL works with unlabeled data while supervised learning with labeled-data.

Table 1.1 Comparison SSL & SL.

Feature	Supervised Learning	Self-supervised Learning
Methodology	• Learn from an already labeled dataset • Initializes from the transfer task	• Uses an unlabeled dataset, then learn the labels by itself • Divided in two phases: pre-training task and downstream task • Initialize the weights from the pre-training phase, e.g. using fine-tuning
Usage of labels	• Requires hand-crafted labels to work	• Generates its own labels during pre-training phase
Complexity	• Straightforward, assumes labels of each given image	• Requires two phases: pre-training and downstream task • Requires two datasets • Large-scale dataset

Continued on next page

Table 1.1 – continued from previous page

Feature	Supervised Learning	Self-supervised Learning
Versatility	• Eases up the process of learning, but limited to annotated datasets	• Although complex, versatile for any kind of data • Different taxonomies make it adaptable for different kinds of outputs • Necessary to pre-train with similar dataset
Performance	• Poor performance for small datasets • Costly and regular performance for large-scale datasets	• Very good performance for large-scale datasets

Table 1.1 explores the similarities and differences with respect concepts that may help to understand better the behavior and performance of the methodologies used.

1.3 Background: Masked Image Modelling

1.3.1 Masked Image Modelling in SSL

For the general structure of the algorithm we use Masked Image Modelling *MIM* (Xie et al., 2021) which belongs to generative SSL. MIM concept is simple: a task that learns how to create, within this structure the models are trained to predict further information similar to the one it was trained, so it provides a good opportunity to be applied in the field of remote sensing. Masked Autoencoders *MAE* (He et al., 2021) as well as Masked Feature Predictors *MFP* (Wei et al., 2021) are part of MIM. One of the most important parts of this architecture is the encoder which maps the observed signal to a latent representation to later be decoded (He et al., 2021), however, this

encoder requires a structure to work. Usually, while working with images CNNs are used but recent studies have demonstrated the performance of different structures that provides satisfactory results. Vision transformers (Dosovitskiy et al., 2020) have been recently included for image processing tasks, where basically the image is split into patches treated as tokens then it is processed by the structure. This architecture had already provided very good results for image recognition in classification, thus, its usage in remote sensing oriented to Earth observation provides a good opportunity to be explored.

1.3.2 Masked Image Modelling in EO

While many of this techniques and architectures have been already explored, they have been oriented to ordinary tasks or just intended to demonstrate its performance for explanatory purposes. However, a complete study in remote sensing oriented to Earth observation is novel and has potential to provide promising results. The present study takes masked image modelling architectures *(MAE and MFP)* with vision transformers as their backbone to perform a study oriented to multispectral images by creating a hybrid capable to provide a better framework for image classification in remote sensing with the help of self-supervised learning (Wang et al., 2022b). As a MIM study includes the performance of pre-training on a Earth observation oriented unlabeled dataset (Wang et al., 2022c) using self-supervised pre-training, therefore, performs a transfer learning classification (Zhuang et al., 2019) on the Eurosat dataset (Helber et al., 2019), (Helber et al., 2018) which consists of ten different classes describing the land usage.

Thus, the current study modifies the traditional view on pixelwise images in the RGB spectrum to a multispectral spectrum which provides a better approach about the characteristics of the land. Moreover, with the help of feature descriptors such as HOG, it improves the performance of the classification output.

1.4 Thesis overview

The present work proposes a hybrid architecture for the pre-training of multispectral images from Earth observation data provided by Sentinel-2 mission by means of SSL, the architecture is based on MAE with HOG as feature prediction targets. The work can be addressed as follows:

- We use two datasets for the self-supervised learning analysis, SSL4EO-s12 (Wang et al., 2022c) during the *self-supervised pre-training* phase and EuroSat (Helber et al., 2019), (Helber et al., 2018) for the *downstream task*.
- The data used is *multi-spectral imagery* obtained by the mission Sentinel-2, we use the 13 bands for the classification task.
- We implement a hybrid architecture based on *masked autoencoders* (He et al., 2021) and *masked feature predictors* (Wei et al., 2021) which work at their encoder with *vision transformers* (Dosovitskiy et al., 2020) and follow the *masked image modelling* strategy for image prediction.
- We perform a classification task over 10 different classes on the EuroSat imagery and we evaluate it in terms of accuracy and a confusion matrix.

The thesis is structured as follows:

- In Chapter 1 we address the motivation and background concepts for our masked image modelling architecture oriented to Earth observation.
- In Chapter 2 we structure a literature review about similar and baseline works for the ideas presented in this work, structured in terms of the main concepts and architectures used for the proposed model.
- In Chapter 3 we describe the proposed hybrid MAE+MFP methodology for self-supervised representation learning in Earth observation.
- In Chapter 4 we present the experimental setup and possible ablation studies.
- In Chapter 5 we show the results and their posterior analysis.
- In Chapter 6 we set our conclusions and outlooks.

2 Related work

2.1 Self-supervised Learning

2.1.1 Self-supervised Learning in computer vision

Self-supervised learning is the backbone of the present thesis work, by means of this strategy we are capable to generate a reconstruction of images and perform a classification of a dataset. To get with the current developments, some studies have invest their efforts in understanding how does SSL works in computer vision.

As a preamble, ImageNet (Russakovsky et al., 2014) starts a new chapter in the research field by providing a large scale visual recognition challenge. The database is linked with several studies to deal with the classification and recognition of the objects, therefore, it serves as a tool for the developed of strategies and models in computer vision. From these baselines, some researches come with new ideas to improve the tasks and focus on new strategies that have helped to computer vision works. Doersch (Doersch et al., 2015), explores the use of spatial context for image context prediction task, a relevant feature for our land-use classification task. Additionally, they use an unsupervised visual representation learning strategy which can be compared to self-supervised learning.

After some years of research in deep learning, new studies have come with a clearer panorama of self-supervised learning applied to images. Goyal (Goyal et al., 2019) aims to learn representations independently and using a massive-scale dataset for its training, a relevant characteristic addressed in our work. Additionally, the research provide an extensive benchmark using different datasets and tasks providing an additional quality necessary for this thesis: the variety in its performance. Caron (Caron et al., 2019) study also aims to a massive-scale dataset, their approach focuses in raw data with unsupervised methods using convolutional neural networks. By training on *96 M* of images the study validates the potential of unsupervised learning, a similar idea is developed in our work. Another novel research in representation learning that achieved

notable results using self-supervised learning is (Misra and Maaten, 2019). Its main contribution is the invariance of images under semantic representations improving their semantic quality and outperforming traditional supervised learning methodologies. Momentum Contrastive learning *MoCo* (He et al., 2019) enhances the use of encoders for self-supervised learning in computer vision. Since our work uses autoencoders this study takes relevance and represents a milestone in the present research field of computer vision.

More recently, new approaches have came to light, such is the case of *SimCLR* (Chen et al., 2020b). In this study, it is refined the idea of contrastive self-supervised learning, therefore, new findings and improvements over previous methods are achieved. This research points that self-supervised learning remains still undervalued, so it encourages new works, such as the present thesis, to explore new strategies and make relevant SSL. In a context oriented to the downstream tasks, *Bootstrap Your Own Latent* (Grill et al., 2020) presents an interesting approach to self-supervised image representation learning where the algorithm utilizes the output of a network as a target for an enhanced representation. It generates a more robust model in comparison with the contrastive methods available in the literature. Finally, Goyal (Goyal et al., 2021) presents the premise of self-supervised learning that it can learn from any random image and from any unbounted dataset. The main contribution and its relation with our work is that self-supervised learning works in a real world setting, this is an important characteristic to the present work which deals with spaceborne imagery.

2.1.2 Self-supervised learning for remote sensing

One of the main baselines of this thesis work is remote sensing. Since self-supervised learning is a tool to achieve the goals and address the motivations of the current thesis, is also important to mention what is the role of remote sensing using tools from deep learning that helps to understand better the behavior of the Earth surface in the field of Earth observation studies.

An extensive study relating deep learning and remote sensing in various ways is presented in (Zhu et al., 2017). The paper questions if it is suitable to use these tools as presented in deep learning and more oriented to this work self-supervised learning, as a key to resolve all the tasks and challenges presented in remote sensing. Zhu et, al. also provides an extensive analysis about how deep learning with all its variations is oriented to achieve mainly Earth oriented remote sensing analysis. They perform the

analysis by presenting lists of resources for its application an easy usage to get with relevant results. Moreover, studies address the importance of remote sensing outputs which are relevant for the current study, such as image classification. One of these studies perform an image classification relying in ImageNET with Deep Convolutional Neural Networks (Krizhevsky et al., 2017), where a relatively big dataset with several classes is trained to achieve relevant results in the field of deep learning oriented to remote sensing applications. Although, the study performs with "normal" images not related with the current study, it performs a classification task with prominent results which are relevant for later studies oriented to a more remote sensing oriented researches.

In this concern, some efforts have been made to create a milestone in the analysis of remote sensing with deep learning with relevant results for a wide field of applications specially oriented to satellite acquired imagery. A scene classification task in (Zhao et al., 2020) uses Convolutional Neural Networks to perform scene identification and classification tasks with images obtained by satellite, being capable to identify scenery like a golf course, an airfield, sea ice or a river in the RGB domain. The study achieves high accuracies for the classification task, while other studies focuses their efforts in other aspects of the remote sensing sciences e.g. using SAR for the acquisition of data but processing it by means of deep learning. One SAR study is performed in (Zhang et al., 2019) where a variation in how do the images should be treated to perform a better training and demonstrate its effectiveness, adding one more study to the literature about feature learning with satellite images, relevant for the current thesis. Another effort in remote sensing is an study oriented to semantic segmentation as a final output and task using SSL techniques (Singh et al., 2018). In this research the authors achieve an improvement over training from scratch by proposing architectural changes in the traditional structure. Finally, in a novel study (Tao et al., 2020) a new architecture for self-supervised learning is proposed to deal with the scene classification for multispectral imagery. One of the authors' motivations is to have a better approach to multispectral and hyperspectral data because they usually have a different imaging mechanism from the widely used RGB images, providing a new opportunity to develop more studies with these type of data, widely common in remote sensing.

2.2 Masked Image Modelling

Masked Image Modelling forms part of self-supervised learning and it is the focus of the present thesis. It has been a recently explored strategy to predict and generate images from a given input and various works have focused their efforts to get relevant results by studying its performance in diverse environments and frameworks. However, in remote sensing they are still several aspects to be explored. The main idea is masking a given input and predict the masked patches of the images. Hence, the researches goes from the percentage and the form of the patches to the integration with learning methodologies and architectures such as self-supervised learning.

2.2.1 Framework studies on Masked Image Modelling

A clear, simple yet extensive approach regarding MIM was performed by Xie et. al. (Xie et al., 2021). In this study is presented a framework with several variations in masking strategies to achieve one common goal: how to learn good representations. The presented architecture also deals with the existence of en encoder composed by architectures like ViT or Swin transformers (Liu et al., 2021), but the main contribution relies in how the masking strategies are varied in different ways while keeping the masking ratio as a constant. Other frameworks take as a baseline the success of masking strategies on other fields e.g. masked language modelling, and replicate the methodology in the visual spectrum by the usage of an *online tokenizer* (Zhou et al., 2021). This study perform a self-distillation on masked patch tokens and take the teacher network as the online tokenizer to acquire visual semantics. The tokenizer addresses a variation in the main idea on how masked modelling works and achieves reasonable accuracies with a more robust model. Moreover, the study also uses vision transformers in its main architecture, an essential architecture used in our thesis.

One of the baselines for MIM is a previously research in context-based pixel prediction study (Pathak et al., 2016), the authors present by the use of Context Encoders a feature learning tool. Although, they used a CNN for the training of the model, the results demonstrate a good prediction and reconstruction of the masked patch of the original image. Moreover, the contributions also point classification, detection, and segmentation tasks. Additionally, studies indirectly contributing to MIM have been performed showing good results (Chen et al., 2020a), the generative pretraining from pixels relies on a sequence transformer to auto-regressively predict pixels without incorporating knowledge of the 2D input image. Hence, it serves as a strategy to be

considered by the masking strategies presented in our work.

Another approaches goes more than predicting missing patches, the strategy follows other direction yet relevant in masked modelling. Henaff et. al. (Hénaff et al., 2019) predict patches using a verification task by means of Contrastive Predictive Coding (CPC), where the predicted images are evaluated using a contrastive loss and enhances the model to correctly classify future representations among non-"correct" representations. At the same time *Selfie* (Trinh et al., 2019) (self-supervised image embedding) takes CPC to teach a model how to learn to select the correct patch among other "distractor" patches in a similar way to Henaff's model, relying on ImageNet and CIFAR-10 datasets to achieve their results. The motivation and objectives are relevant to our pipeline and provide several ideas to be analyzed when performing MIM in the studies. Nevertheless, most of the modelling strategies lacks of a remote sensing context missing important issues to be addressed in this field.

2.2.2 Masked Image Modelling for remote sensing

In the field of remote sensing, masked image modelling is still an approach to be explored. There are many open issues that have not been addressed and few researched have been done orientating their efforts to a narrow spectrum of the field. However, these investigations provides interesting ideas and results on how these modelling strategies perform when more complex images are masked. Moreover, they represent important reference points for later studies including the present one in this thesis work.

A hyperspectral image classification performed by Guan and Lam (Guan and Lam, 2022) involves the MIM strategy within a remote sensing analysis by a cross-domain contrastive learning framework oriented to learn image representations with these characteristics across the spectral and spatial domains. Their main output is the signal representation in these two domains to achieve high accuracies at the classification performance. Nevertheless, one challenge this study presents relies in the time consuming task solver due to the abundance of unlabeled samples. Most recently, Wang (Wang et al., 2022a) proposed a new approach for remote sensing using masked image modelling by proposing a global semantic integrated self-distilled complementary MIM where their main goal is to address the information loss by generating two complementary masked views for the same image. Thus, with the help of an auxiliary network pipeline they extract global semantic information from the images and transfer to MIM by

self-distillation. Their approach provides a new baseline to be taken into consideration for future developments.

There are few studies relating MIM and remote sensing, hence, it is necessary to deep into more studies considering the benefits of masked images for SSL in the field of remote sensing. Month by month new studies come with new approaches and results contributing to the state of the art.

Finally, a complete setup of models, strategies and methodologies have been addressed independently and within the years to tackle the challenges in deep learning and remote sensing. Step by step, researchers combine different ideas to solve more specific problems, get better results and improve the application on complex tasks such as remote sensing. In the present study, we address one of the challenges in remote sensing which is the usage of features prediction with a framework of several strategies which had worked efficiently in past approaches. Thus, we expect new knowledge of the architectures and models' behavior with a remote sensing baseline to be added to the state of the art.

3 Methodology

3.1 Masked Image Modelling pre-training

The concept of masked image modelling comes from masked signal modeling, their task is to learn how to mask a portion of input signals and trying to predict these masked signals (Xie et al., 2021). Recently this idea has been explored going from language processing to image processing in computer vision, using the same methodology with SSL, although not without difficulties to get relevant results.

The concept of masked image modelling *(MIM)* is to learn representations by masking a portion of the input image signals and perform a prediction of the original input at the masked area. One generic architecture is described by the model SimMIM (Xie et al., 2021), which provides a clear structure oriented to image reconstruction. Their framework consists of 4 major components:

1. **Masking strategy.** It is developed on the input image, its task is to get with the area to mask in the most efficient way (also depends on the architecture used in the model). The input image after the masking will be used as the input to the encoders.

2. **Encoder architecture.** At this stage it depends on the structure or architecture used by the model, as a baseline, the encoder is relevant because it extracts a latent feature representation for the masked image, later to be used to predict the masked patches from the image. The output of the encoder will depend on the general architecture of the model.

3. **Prediction head.** It deals with the encoder output, its goal is to accomplish the prediction target, its structure and complexity will depend on how is defined the structure in the model it is used. In some architectures a decoder is used as the prediction head but the complexity is bigger.

4. **Prediction target.** At this final component it is decided the form of the original

signals to predict. Depending on the application could be the raw input values or a transformation of this values, in the case of SimMIM raw pixel values are used as an example, but feature representations can be used in the framework. At this stage it is also defined the output loss type.

Masked Image Modelling is open to several modifications that can improve its performance or may be oriented for specific tasks, there are many developed architectures in the literature but for the present project two of them result relevant for the study and its goals: *masked autoencoders (MAE)* (He et al., 2021) and *masked feature prediction (MFP)* (Wei et al., 2021).

3.1.1 Masked Autoencoders

Oriented to computer vision tasks and designed as self-supervised learners, masked autoencoders are one of the core architectures for the present project. They take the idea of MIM by randomly masking out patches from an input image and reconstruct the missing pixels. Its structure is based on the encoder-decoder flowchart but with a novel modification: an asymmetrical architecture, where the encoder operates on the visible patches and the lightweight decoder reconstructs the original image from the latent representation and mask tokens. Since they use *vision transformers (ViT)* (Dosovitskiy et al., 2020) architecture in the encoder as well as implementing its idea (Fig. 3.1) to divide into patches while masking some of them out.

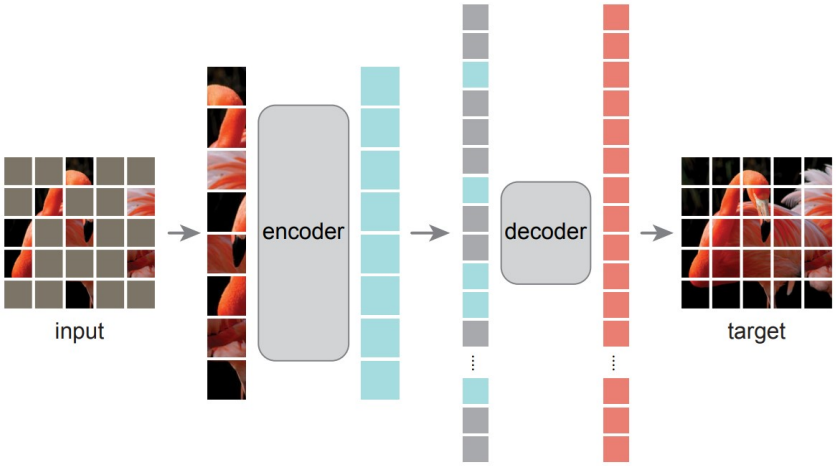


Figure 3.1 Masked Autoencoders architecture (He et al., 2021)

MAE architecture is an encoding approach that reconstructs the original signal given

its partial observation. This approach has an asymmetric encoder-decoder structure where the encoder maps the observed signal to a latent representation, and the decoder reconstructs the original signal from that latent representation. Due to its asymmetric nature, allows the model to improve their performance by setting the encoder to operate partially on the observed signal (He et al., 2021). The flowchart is divided into 5 steps to provide a better picture about the approach:

1. **Masking.** As it is presented in the masking approach of MIM as well as the patching division for ViT, the architecture takes these two ideas to generate a new masking approach. The image is divided into regular non-overlapping patches which are then sampled in a subset, then a mask removes the remaining ones. The idea goes as follows: the approach samples random patches without replacement, following uniform distribution.

2. **MAE encoder.** The encoder is a ViT applied on visible, unmasked patches, therefore, it embeds the patches by a linear projection with added positional embeddings, then it process the resulting set via series of transformer blocks. Depending on the masking percentage of the image, the encoder operates only on a small subset, not in the whole image, thus, for example given a mask of 75%, the subset sample will be 25% allowing the encoder to be very large with a fraction of memory.

3. **MAE decoder.** The output of the encoder is the input of the decoder plus the masked tokens, here each mask token is a shared and learned vector that indicates the presence of a missing patch to be predicted. The approach in (He et al., 2021) add positional embeddings to the tokens in the full set in order to have the panorama of their location in the image. Additionally, in this architecture, the decoder has another series of transformers blocks. For the SSL methodology, it is only used during the pre-training phase to perform the reconstruction of the image, this means, the decoder can be independent of the encoder design.

4. **Reconstruction target.** In MAE the reconstruction approach is oriented to reconstruct the input raw pixels for each masked patch. At the end of the architecture, in the decoder output, it is generated a vector of pixel values representing a patch, this output is reshaped to form a reconstructed image. The calculated loss function computes the *mean squared error (MSE)* between the reconstruction and the original input in the pixelwise context and it is just computed on the masked patches.

5. **Implementation.** The implementation presents a straightforward approach to generate reconstructed images: first MAE generate a token for every input patch, next it randomly masked the list of tokens and based on the masking ratio, it removes the masked portion from the list allowing the encoder to deal with a small subset of the image's patches. At the encoder's output the encoded patches and the masked tokens are appended to be processed by the decoder, therefore, the decoder process the whole list of tokens. MAE authors (He et al., 2021) call this implementation a shuffling-unshuffling set of operations.

3.1.2 Masked Feature Prediction

Masked feature prediction (Wei et al., 2021) is a video oriented approach similar to MAE in its masking strategy to try to predict and reconstruct a target image from a given input. The novel concept is the prediction of features instead of raw pixels, modifying the architectures and providing new finding while dealing with features like *Histogram Oriented Gradients (HOG)*. Although, they are oriented to get a reconstruction from video sequences, for the application of the present project, it is not relevant to deal with this approach, however, the concept of feature predictions, attains the attention for the *MAE+MFP* study.

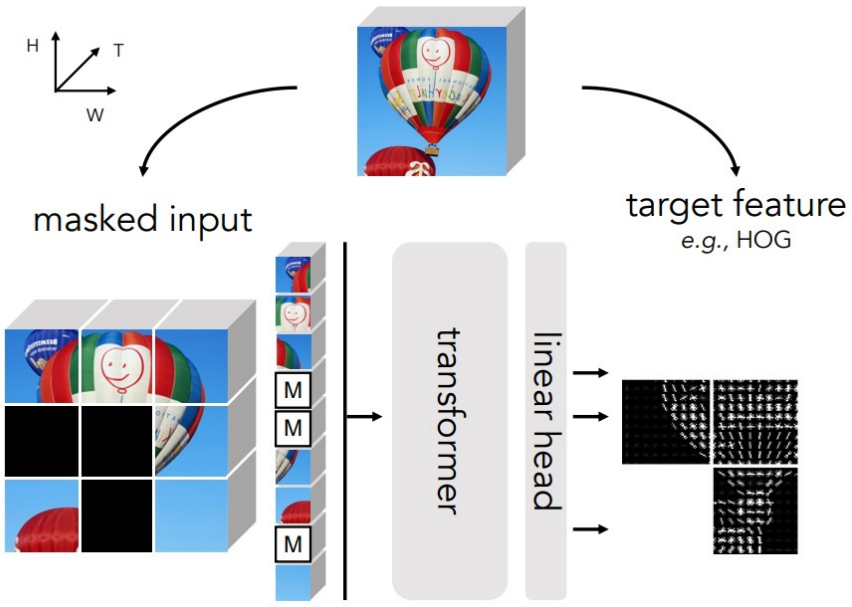


Figure 3.2 Mask-Feat architecture (Wei et al., 2021)

The approach performs a masked visual prediction task by randomly masking few space-time cubes of a video, then predicts the masked ones given the non-masked ones. The flowchart is similar to MAE's one, it tokenizes the input and predict features from the masked area. The choice of the target feature will be definitive on the task. This flowchart presented in (Wei et al., 2021) sets a video which is first divided into space-time cubes which then, are projected to a sequence of tokens. The masking out operation is performed by randomly masking out some cubes and replace them with a mask token. The prediction is performed by taking the token sequence after the mask token replacement, with positional embedding added, then processed by the transformer. The output tokens are projected by the prediction by a linear layer resulting in a feature representation of the 2D spatial patch temporally centered in each masked cube (Wei et al., 2021).

Since the present study is not interested in analyzing a set of video cubes, parts of this approach will be ignored, hence, the feature prediction concept is relevant for the analysis. As it was mentioned in the MFP approach, the definition of the feature is of vital importance and will be dependant on the task and the goal expected for the project.

Histogram Oriented Gradients

As a part of representation learning, histogram oriented gradients is a feature descriptor used in computer vision and image processing to detect objects from a given image. The approach is based on a gradient orientation computation similar to Sobel operator, also a feature descriptor. HOG descriptor focuses on the structure of the object and it generates histograms using the magnitude and orientations of the gradient.

First introduced in (Dalal and Triggs, 2005), they are presented as evaluations of well-normalized local histograms of image gradient orientations in a dense grid, based on the idea that images can be characterized by the distribution of local intensity gradients or edge detection. Their approach is implemented by dividing the image window into small spatial regions called *cells*. For each accumulating a local 1D histogram of gradient directions or edge orientations over the pixels of the cell, thus, the combined histogram entries form the representation. An additional contrast normalization is done by accumulating a measure of local histogram "energy" over a larger spatial regions defined as blocks, then using the results to normalize all of the cells in the block.

The main parameters are defined as orientations, pixels-per-cell and cells-per-block

which will define the dimensionality of the resulting vector. The flowchart can be divided in 4 stages (Rosebrock, 2014):

1. **Normalization.** A relevant step in the HOG calculation, although it is optional, in deep learning models it can be useful to improve the performance of the models.

2. **Gradient computation.** The gradient computation is performed in x and y directions by the application of a convolution operation

$$G_x = I \cdot D_x \quad and \quad G_y = I \cdot D_y$$

where I is the input image, D_x the filter in x-direction and D_y the filter in y direction.

Then, the final gradient magnitude representation is computed:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Finally, the orientation of the gradient for each pixel is computed.

$$\theta = arctan2(G_y, G_x) \tag{3.1}$$

An important element of HOG in its structure is the bin, which is given by the orientation and the weight added to the given bin is based on the magnitude.

3. **Weighted votes in each cell.** The image is divided into cells and blocks, the cell is a rectangular region defined by the number of pixels in each cell. For each cell in the image, the number of orientations is defined because it controls the number of bins in the resulting histogram. The gradient angle range goes from 0 to 180 in an unsigned scope, this is because unsigned orientations perform better with the applications where HOG are used. Therefore, the gradient magnitude at each given pixel define the weight.

4. **Contrast normalization** A local normalization is applied to deal with illumination and contrast, the operation groups the cells together into larger blocks which overlap each other, the units after this operation are cells. Figure 3.3 shows the overlapping operation, grouping cells into overlapping blocks.

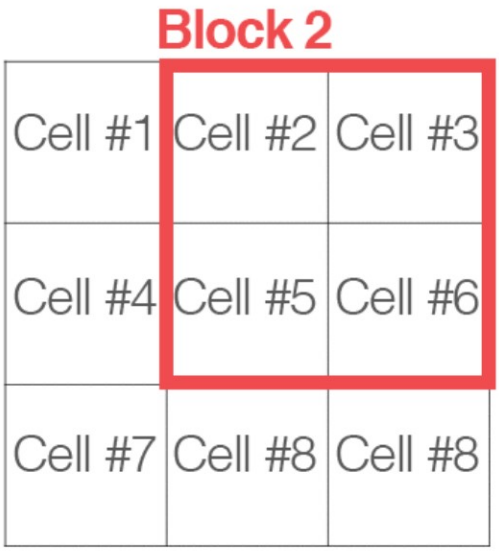


Figure 3.3 HOG generation of blocks (Rosebrock, 2014)

For each cell in the block the corresponding gradient histograms are concatenated, then a normalization L1 or L2 is applied. Finally after the blocks are normalized, the resulting histograms are concatenated representing the final vector.

3.1.3 MAE + MFP, a hybrid architecture

The novel methodology presented in this project, involves several architectures, models and representations to get with new findings and knowledge about the performance of this architecture while dealing with Earth observation oriented imagery. Therefore, the usage of a process such as SSL within ViT transformers as the core architecture in the model using HOG as representations for the target images to extract information and make predictions, leads to the definition of a model which can generate a framework to reconstruct images from masked inputs using representation learning.

Using masked image modelling (Xie et al., 2021) as the baseline for the architecture model to extrapolate it to an asymmetric encoder-decoder architecture (He et al., 2021) which has as a target a feature (Wei et al., 2021), is the main framework (a hybrid architecture) of this work. Additionally, it is evaluated by the main methodology using self-supervised learning.

MAE + MFP with HOG

The core architecture of the present project is a combination of MAE and MFP architectures to generate a new approach: the prediction and reconstruction of masked patches from a visible input image to get with a reconstructed feature represented image using HOG. This is performed with the goal of acquire new knowledge and finding about its performance using SSL with two Earth oriented datasets.

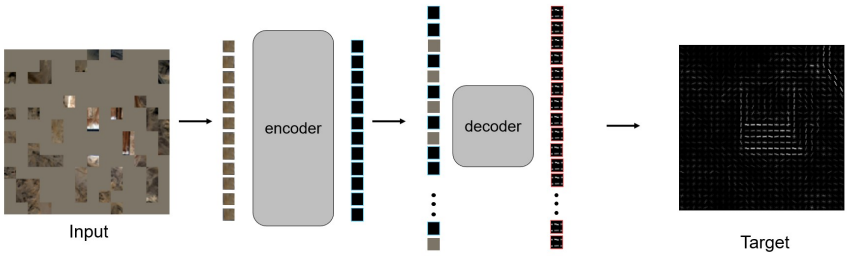


Figure 3.4 MAE+MFP architecture

The flowchart is the same as the used in MAE architecture but implementing the feature prediction concept from the MFP architecture.

1. **Masking.** Similar to MAE and MFP an input visual image is masked out accordingly to a masking ratio, the percentage is defined to be processed by the encoder. Thus, usual rates of approximately 70% of the masked input are used, as the ViT small subset is used, the input image will be divided into specified measures for the patches:

   - The ViT patch size is set for 16 pixels, as defined in (Dosovitskiy et al., 2020)
   - The grid size for each input image is set for 14 $\times$ 14 patches

   Therefore, only the small subset of patches is taken into the encoder.

2. **Encoder architecture** Similar to MAE architecture, a vision transformer composes the encoder and encode the visible patches following an algorithm defined in the code. However, here the computation is oriented to HOG feature representations, thus, the encoder generates new token representations in the form of feature oriented encoded tokens.

3. **Decoder architecture.** At the input of the decoder, the feature oriented tokens and the masked tokens are appended together and introduced into the lightweight decoder, here a positional embedding is added such as in MAE's approach and a loss function is calculated from the reconstruction.

4. **Reconstruction target.** The final output is a reconstructed image composed by a vector of HOG feature values represented in each patch and, as a whole, the final reconstruction of the feature representation.

This architecture is the used in the complete analysis by means of self-supervised learning.

3.2 Downstream transfer learning

The downstream task may be performed by several ways to get the accuracy of the model, usually, they variate between them by how the trained model from the pre-training phase is treated. The performance evaluation by quantitative means may be executed by two different methods using the output from pre-training, either by linear classification or fine-tuning. Thus, with an ablation study it can be obtained a deeper understanding of the behavior of the model and the methodology.

3.2.1 Linear Classification

To explain linear classification or linear probing, it is necessary to consider the common scenario in deep learning in which the goal is the classification of the input data X to produce an output distribution over D classes. The last layer of the model is a densely-connected map to D values followed by a softmax function, then it is trained by the minimization of cross-entropy (Alain and Bengio, 2016). At every layer the features H_k are taken from that superficial layer and perform a prediction of the correct labels y using linear classifier.

By the using of this layer, the encoder is freezed up and only the linear layer is trained, this evaluation measures how linearly separable the embeddings produced by the pre-trained model are (Wang et al., 2022b).

3.2.2 Fine-tuning

During fine-tuning evaluation, the layers of the model are unfreezed, therefore, all the parameters of the pre-trained model are initialized, ready to be used during the stage. Usually, in massive-scale datasets the results tend to be better in terms of accuracy (Yu, 2016).

3.3 Model backbones: Vision Transformers (ViT)

In order to understand vision transformers it is necessary to get the idea about what a transformer is, and the core idea of a transformer is the Attention (Vaswani et al., 2017). The transformer is a model architecture which avoids the usage of recurrence in the neural networks (Canziani et al., 2016), and instead, relies on an attention mechanism to draw global dependencies between input and output (Vaswani et al., 2017). One of the advantages is that it allows an improvement in parallelization, thus, less costly computationally speaking.

The core architecture used in the present work is the Vision Transformers architecture, instead of relying on CNNs it explores the application of the Transformer's structure to sequences of images patches to solve tasks such as classification (Dosovitskiy et al., 2020).

For transformers applied to image, the method split an image into patches and provide the sequence of linear embeddings of them as an input to a transformer. Then, image patches are treated as the same way as tokens (words) in a Natural Language Processing *(NLP)* application. Since the results in (Dosovitskiy et al., 2020) provides already a picture of its behavior, on small-scale datasets ViT provide a poor performance compared to CNNs. However, for large-scale datasets the picture shows a different approach, they achieve a good performance when pre-trained at sufficient scale and transferred to tasks with fewer data points.

The designed architecture receives as input a 1D sequence of token embeddings. To be able to process 2D images, where CNN perform it in a straightforward way, the image is reshaped as follows:

$$\mathbf{x} \in \mathbb{R}^{H \times W \times C} \quad \rightarrow \quad \mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)} \tag{3.2}$$

As the equation 3.2 describes, the image is reshaped into a sequence of flattened

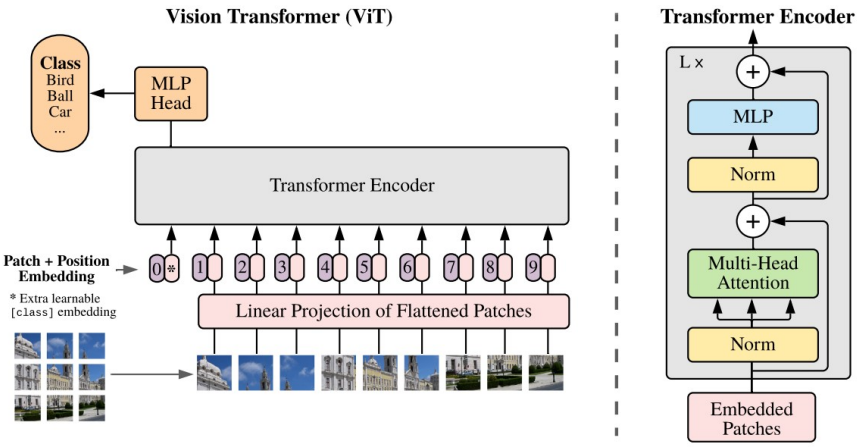


Figure 3.5 Vision Transformer, model overview (Dosovitskiy et al., 2020)

2D patches, where (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches. The transformer uses constant latent vector size D through all of its layers, so patches are flattened and mapped to D dimensions with a trainable linear projection. The model refer as output of this projection as the patch embeddings (Dosovitskiy et al., 2020).

$$\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{x}_p^1\boldsymbol{E}; \mathbf{x}_p^2\boldsymbol{E}; ...; \mathbf{x}_p^N\boldsymbol{E}] + E_{pos}, \quad \boldsymbol{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \boldsymbol{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{3.3}$$

Therefore, a learnable embedding is applied to the sequence of embedded patches $(\mathbf{z}_0^0 = \boldsymbol{x}_{class})$, whose state at the output of the transformer encoder $(\mathbf{z}_L^0)$ serves as the image representation $\boldsymbol{y}$. During pre-training and fine-tuning, a classification head is attached to $\mathbf{z}_L^0$. The classification head is implemented by a Multi-Layer Perceptron with one hidden layer at pre-training phase and by a single linear layer for fine-tuning in the downstream task phase (Dosovitskiy et al., 2020).

To retain positional information and avoid losing the distribution logic in the 2D image, 1D positional embeddings are added to patch embeddings, thus, the output vectors serves as input to the encoder. The transformer encoder consist of alternating multi-head self-attention and MLP blocks within its respective layer normalization and residual connections.

$$\mathbf{y} = LN(\mathbf{z}_L^0) \tag{3.4}$$

4 Experiments

During the experiments phase the complete process of SSL was performed to get new findings and knowledge about how does histogram oriented gradients (HOG) behave as a target to pretrain models on EO data. Therefore, test their performance with a classification task employing a database with similarities during the transfer task.

4.1 Earth Observation Datasets

4.1.1 SSL4EO-s12

As self-supervised learning methodology requires in its first step a massive-scale unannotated dataset it is necessary to provide the process with one that satisfied this characteristic, such is the case of Self-Supervised Learning for Earth Observation-Sentinel 1/2 (Wang et al., 2022c) database.

SSL4EO-S12 (Wang et al., 2022c), provides a suitable scenario for the self-supervised pre-training stage. One of its prominent characteristics relies in its large-scale oriented satellite imagery quality, providing one of the basic requirements for a spaceborne imagery analysis. Then, it also has the property of be a global coverage dataset allowing a complete analysis of different locations around the world with a multi-temporal and multi-sensor feature based on *Sentinel 2 (1C and 1A)* levels and *Sentinel 1 SAR* feature (Phiri et al., 2020). In the present project, we use the following features from this dataset:

- 3 million Sentinel-2 (multi-spectral, level-1C and level-2A) and Sentinel-1 (SAR) images
- Images are from 250K locations sampled around the globe
- The patch scale is 264x264 pixels
- As we use multi-spectral images, they are distributed over 13 bands

- At each location, 4 images are obtained from four annual seasons separated by 3 months to get seasonal variation over the year

4.1.2 EuroSAT

Regarding the transfer learning phase, EuroSAT dataset is used as the main subject of study. The main reason of the creation of EuroSAT (Helber et al., 2019), (Helber et al., 2018) is the addressing of the challenge of land-use and land-cover classification using Sentinel-2 imagery. The dataset is curated for its usage on 13 spectral bands divided into 10 different classes with approximately 27'000 labeled and georeferenced images. The main motivation, is the application of several domains with satellite imagery resources such as agriculture, climate change, urban development, environmental monitoring, etc., and as (Helber et al., 2019) cites. In order to use such capabilities it is necessary to process and transform images into a structured semantics.

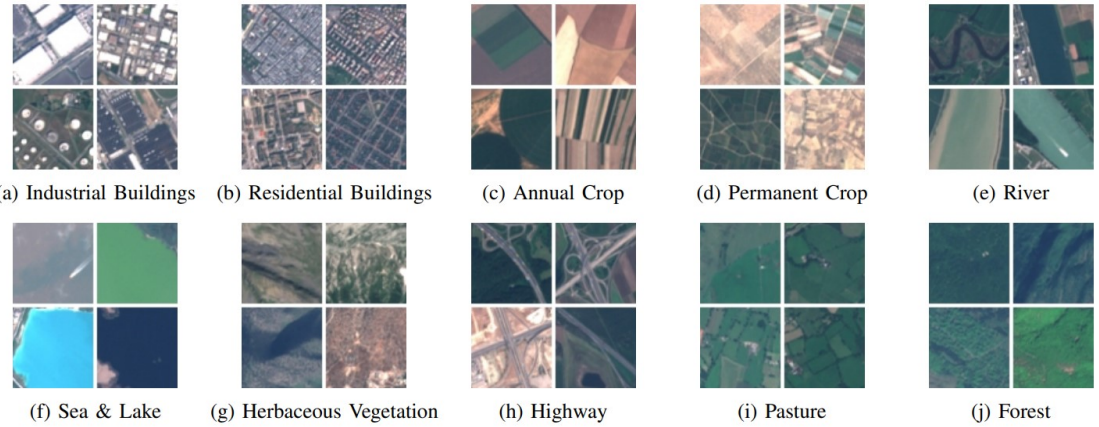


Figure 4.1 Eurosat sample image patches (Helber et al., 2019)

An overview of the images used for this dataset is oriented to 10 different classes released in sample image patches of 64x64 pixels, each class consists of between 2000 and 3000 images surrounding between 10 m and 60 m of spatial resolution. A panorama of the studied classes is presented in Figure 4.1, while a general picture of the used bands is presented in Table 4.1. Is worth to mention that the bands and resolution in the following table also holds for SSL4EO-s12 dataset.

Table 4.1 Sentinel 2 Multispectral Imagery (Helber et al., 2019)

Band	**Spatial resolution** m	**Central wavelength** mm
B01 - Aerosols	60	443
B02 - Blue	10	490
B03 - Green	10	560
B04 - Red	10	665
B05 - Red edge 1	20	705
B06 - Red edge 2	20	740
B07 - Red edge 3	20	783
B08 - NIR	10	842
B08A - Red edge 4	20	865
B09 - Water vapor	60	945
B10 - Cirrus	60	1375
B11 - SWIR 1	20	1610
B12 - SWIR 2	20	2190

4.2 Self-supervised pretraining

As was mentioned in Chapter 3, self-supervised learning is mainly divided into two phases: pre-training and transfer learning tasks. For the experimental phase we performed self-supervised pre-training with our hybrid *(MAE+MFP)* architecture, then we transferred for the classification task partially this architecture to the EuroSat dataset.

During self-supervised pre-training, a modification of the codes presented at masked autoencoders study (He et al., 2021) was carried out. Then, in a similar manner we took the HOG development in masked feature prediction study (Wei et al., 2021) modifying it in accordance to the necessities of *MAE+MFP* model.

4.2.1 Data preparation

The evaluation is performed on the Sentinel 2-c multi-spectral images which are then pre-processed before start the pre-training process, the preprocessing is performed by the normalization of the data, this with the goal of improving the performance and stability of the model (Google, 2022). The normalization is carried out by computing the mean and standard deviation of each of the 13 bands in the dataset, moreover, a

second normalization stage is performed on the HOG layer to observe the behavior of the performance of the model under different circumstances.

For the data augmentation and transformation we set a batch for the images within a Tensor of **(B, S, C, H, W)**, where **B** is a number of images in the batch, **S** the season of the image, **C** is the number of channels representing one of the 13 bands, **H** and **W** are the height and width of the image. Then, the data is adjusted to the requirements set for *ViT small patch 16* architecture which is defined by Figure 4.2. The transformations applied for augmenting the data follow the concept of vision transformers, where images are split into fixed-size patches, linearly embedded and set with a positional embedding.

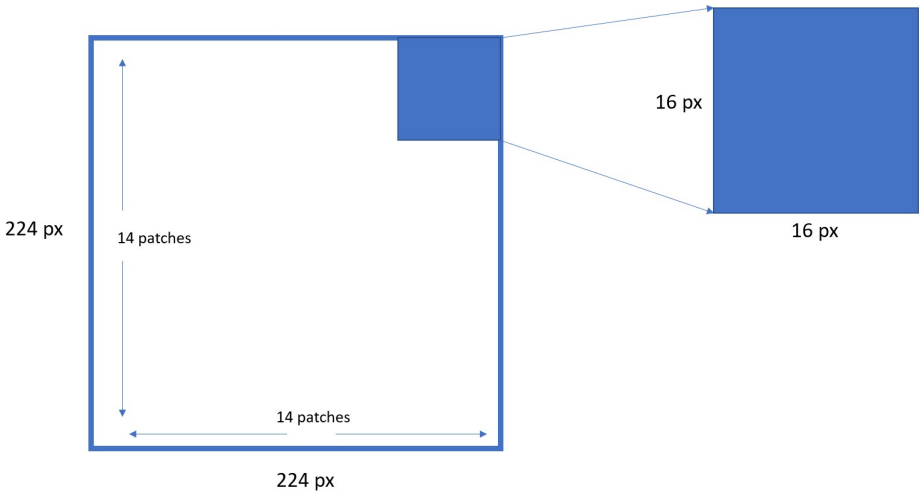


Figure 4.2 ViT small patch 16 grid size and pixels definition

Finally, for the dataloader, when it deals with a massive amount of images, a new strategy may be addressed: the LMDB (Lightning Memory-Mapped Database) manager (Chu, 2015). It is a database management library modeled on a previous manager called BerkeleyDB API but simplified, its main feature is the high performance and memory-efficiency based on its read-write mode strategy to deal with the data, thus, a good candidate to deal with massive-scale datasets.

Model

In the present work the architecture is a hybrid design of MAE and MFP, where it is not only necessary to define raw pixels but also the definition of the HOG feature descriptor.

HOG Layer. Histogram oriented gradients are composed by the number of bins and the number of cells, the bins corresponds to the phase and in for the experiments it is taken a phase of 20°. Then to get a complete distribution of angles, this means 180° it is necessary to get 9 bins. For an easy management of the data, the number of cells is set to a default of 8 and the Gaussian kernel to 16. The definition of the weights for either x and y directions is set within the function of the HOG layer and a normalization just for HOG gradients is also defined in this function, reshaping the HOG feature to later generate a feature for each batch of images. For the final output that provides the construction of HOG, a gradient in x and y is convolved in function of its respective weights. Additionally, the normalization is defined and the phase is calculated from the gradients and the number of bins similarly computed as in equation 3.1.

For the final output two outcomes are generated:

1. Output 1D. For the later training and acquisition of the loss for HOG feature
2. Output 2D. For the image reconstruction in HOG terms

Definition of parameters. The masked autoencoder is mainly defined by the characteristics form the transformer used in this architecutre: the ViT small transformer, its characteristics are enumerated in the following table:

Table 4.2 ViT small patch 16 characteristics (Dosovitskiy et al., 2020)

Model	Layers	Hidden size D	Heads	Image size (px)
ViT small	12	384	6	224×224

Encoder-decoder specifics and weights initialization. At this stage, the patches and positional embeds are defined in function of the parameters. A classification token for the encoder and a mask token for the decoder are also defined for later processing. Additionally, depending of which feature is the model dealing with (pixelwise or HOG) and the number of bands to be trained (3 for RGB or 13 for MSI), the parameters are defined to satisfy this variations.

Regarding the initialization of the weights, they are processed by the positional embed-

ding with the posterior calculation of the mean and standard deviation.

Encoder Tools defined in the code as patchify and unpatchifiy are oriented for a better handling of the process, besides before getting into the encoder phase, a random masking is performed per-sample. In the encoder, the patches and the masking ratio are taken as input of the function, then a positional embedding is added to not lose the position of the patches in the image. A classification token is also added and the transformer blocks from the ViT small parameters are applied to the image. The output generates a transformed (encoded) image, a mask and the IDs of each patch.

Decoder. In the decoder phase the mask tokens are added within the encoded patches, thus, they are appended to the sequence, a positional embedding is added and the transformer blocks are applied. Finally, a predictor projection is also applied to the patches which comes from the definition of the decoder specifics to generate a single output of the patches of the images.

Pretraining

The pre-training phase sets the final parameters to be processed by the system which will run the training batches prepared for this process, we used 4 NVIDIA A100 GPUs' for training (Advance Simulation (IAS), 2022). During the training, key concepts are relevant to understand the performance and behavior of the model.

Loss. The loss represents the penalty for a bad prediction in the model, in this thesis we use the L2 Loss function namely Least Square Errors *(LSE)* which is used to minimize the error as expressed in the following equation:

$$L2 = \sum_{i=1}^{n} (y_{true} - y_{predicted})^2 \tag{4.1}$$

Optimizer The optimizer is used improve the performance of the model's training by changing the attributes of the neural network to attempt a reduction of the loss. During the pretraining and transfer learning tasks, we use the Stochastic Gradient Descent *(SGD)* algorithm which updates the model's parameters frequently to get a high variance in loss functions at different intensities.

Epoch For the experiments in the present analysis, a sample of number of epochs of 100 was taken to have representative results in the self-supervised learning process. The execution for self-supervised pre-training as well as the downstream task was

performed by using the Juwels Supercomputer resources (Advance Simulation (IAS), 2022).

Learning rate The learning rate controls how much to change the model in response to the estimated error each time the model weights are updated. It will largely influence the performance of the model, not just during self-supervised pre-training where the loss is the most important output, but also during transfer learning where the accuracy will be heavily impacted by the definition of this parameter.

4.3 Downstream Task

The downstream task is the second phase of SSL, its main objective is to transfer the "knowledge" learned in the pre-training phase to another dataset with similar characteristics, it is also known as transfer learning. We applied it for a classification task in the EuroSat dataset.

4.3.1 Data preparation

The input data is normalized by setting all the images to a similar scale, then optimize the training during the downstream task. The full EuroSAT dataset is split into 80% training and 20% validation subsets.

The learning rate is set to 0.1 in most of the cases with the exception of linear classification study when it is changed to 0.01. The classification training is performed under a sample of 100 epochs and we use the Stochastic Gradient Descent *(SGD)* optimizer which updates the model's parameters, frequently to get a high variance in loss functions at different intensities.

4.3.2 Training classification

Finally, the main flowchart provides the accuracy of the model's classification, a series of strategies are executed to address the final goal, the pipeline's structure is similar to the pre-training phase with punctual variations on specific parts of the architectures.

The task starts with the split of the data, one subset for training and one subset for validation, then a model to be used for the training is to be set. The ViT smal transformer architecture is used as the backbone of the process with the specification

of the number of classes and the number of channels. Then, depending on the mode of the training classification, a pre-trained model is loaded, therefore, depending on the mode, determined layers of the model may be frozen or not. Finally, the classification output is generated.

To understand better the variation of the strategies to be used to get with the training accuracy of the model, the following diagram explains the main flow and the variations on this stage.

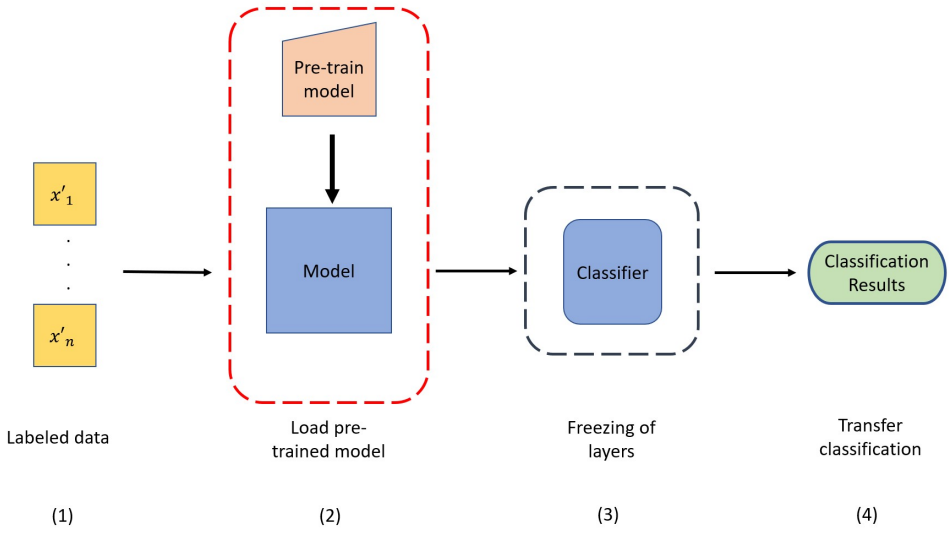


Figure 4.3 Main pipeline of training code for Downstream task

1. **Labeled data**. The labeled data from the EuroSat dataset is loaded into the model and split into subsets for training and evaluation, the data is prepared using a defined dataloader.

2. **Pre-trained model**. At this stage it can be set the strategy to start the training process and get the classification.

    - If no pre-trained model is added, this means no trained weights, then strategy taken is called either random initialization or supervised learning. Additionally, this process is not considered into the self-supervised learning methodology, both are independent strategies.

    - If a pre-trained model is added, it is considered part of SSL methodology and can be either Linear Classification or Fine-tuning.

3. **Classifier** This stage also influences which strategy is being addressed to get with the classification task.
    - If the layers are frozen out but the head, this means, just dealing with the superficial layers of the model, the strategy depending on the previous stage, can be either Linear Classification when a pre-trained model was added or Random Initialization when no pre-trained model was added.
    - If the layers are not frozen at all, this means, dealing with all model's layers, the process can be either Fine-tuning if a pre-trained model was added or Supervised Learning if no pre-trained model was added.
4. **Transfer classification**. Finally, the classification results are obtained, which are evaluated in terms of accuracy.

4.4 Ablation study

The ablation study performed for the present thesis explores different ways of removing/modifying parameters across the phases of self-supervised learning. The scenarios taken to perform this study are addressed in the following table.

Table 4.3 Ablation studies

Variation	Part on the process	Modifications
Masking ratio	Modified in the Pre-training phase as part of the input to the model's encoder, this variation will change the ratio of the image to be masked out	• Masking ratio of 0.7 • Masking ratio of 0.5 • Masking ratio of 0.2

Continued on next page

Table 4.3 – continued from previous page

Variation	Part on the process	Modifications
Normalization	Modified in the Pre-training phase at different parts of the codes. The normalization is carried out for two sides, one for the general process of the images and one for the definition of the HOG feature. Additionally, the ablation study also applies for which target is being analyzed: Raw image or HOG image	• (A) **RAW** no-normalized • (B) **RAW** normalized • (C) **HOG** normalized + hog_normalized • (D) **HOG** hog_normalized • (E) **HOG** normalized • (F) **HOG** no-normalized

5 Results

In this chapter, the results will be divided into two sections, self-supervised pre-training and transfer learning phases. For self-supervised pre-training, two important aspects are presented in this section: the loss of the model with respect different masking strategies and feature normalization, and the image reconstruction in the RGB spectrum and multi-spectrum images (MSI).

In the case of the transfer classification task the results are more diverse, the accuracies graphs are included within the complete ablation study, evaluation of the model as well as diverse confusion matrices, and last but not least, the mis-classified images by the model in the most relevant cases.

5.1 Self-supervised pre-training

5.1.1 Training loss

The training loss is analyzed by two different aspects of the ablation study, the variation of the features normalization and the masking ratio variation. For the sake of the study a complete analysis by this two studies is performed with a masking ratio of 0.7 and the 6 different normalization scenarios presented in Table 4.3, while for the remaining masking ratios (0.5 and 0.2) the normalization scenarios applied will be B, C and D.

As Fig. 5.1 shows, the training loss tends to converge towards zero, however, at first epochs is when this tendency is more significant due to the training process and, when the model eventually goes through the training process, the change is not that notorious and it is reduced epoch by epoch. This can also explain why is chosen a sample of 100 epochs, more samples would be not that relevant as the loss changing rate is less and less within the epochs. Additionally, it is notorious that the *HOG norm+hog_norm* scenario presents a bigger value in its training loss than the rest of the other scenarios. The reason may be because since it presents two normalization applications, one for all

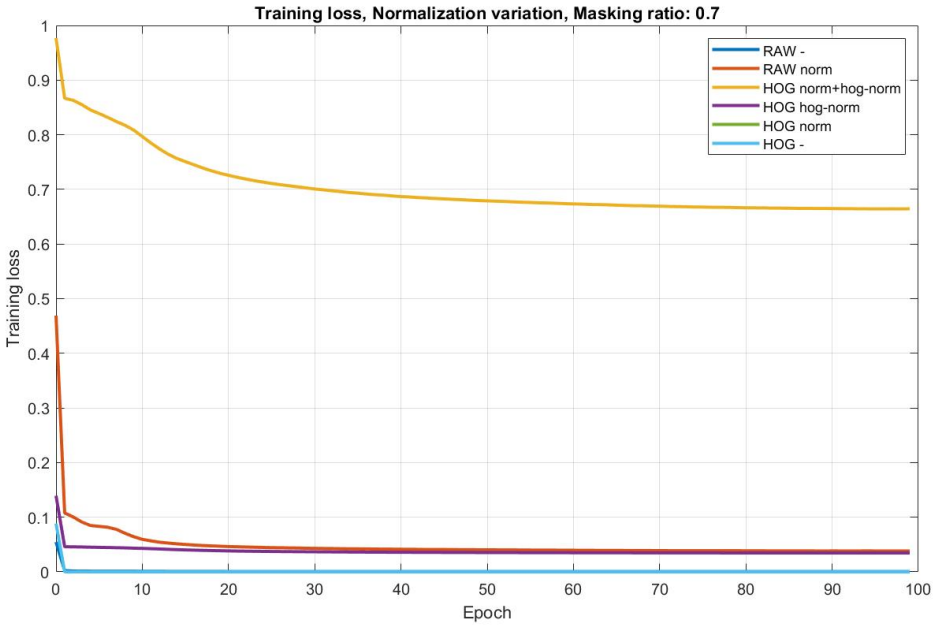


Figure 5.1 Training loss for masking ratio of 0.7

the images in the process and one for the featured images, it affects the performance of the final training loss. However, it will be in the image reconstruction as well as in transfer learning process when the classification output seems to be not directly affected by this loss.

Figures 5.2 and 5.3 presents a similar behavior with respect Fig. 5.1, the variations are not noticeable in the graphs, therefore, Table 5.1 shows the final training losses for each case in this ablation study, the normalization scenarios are presented taking the terminology in Table 4.3.

Table 5.1 Training loss: ablation study

Mask/Norm	**A**	**B**	**C**	**D**	**E**	**F**
0.7	4.55E-04	0.038	0.664	0.034	5.74E-06	5.75E-06
0.5		2.78E-04	0.593	0.030		
0.2		1.72E-04	0.525	0.026		

As the table suggest, the lower the masking ratio the lower the training loss, this is mainly because there are less masked patches to train and more information taken as a

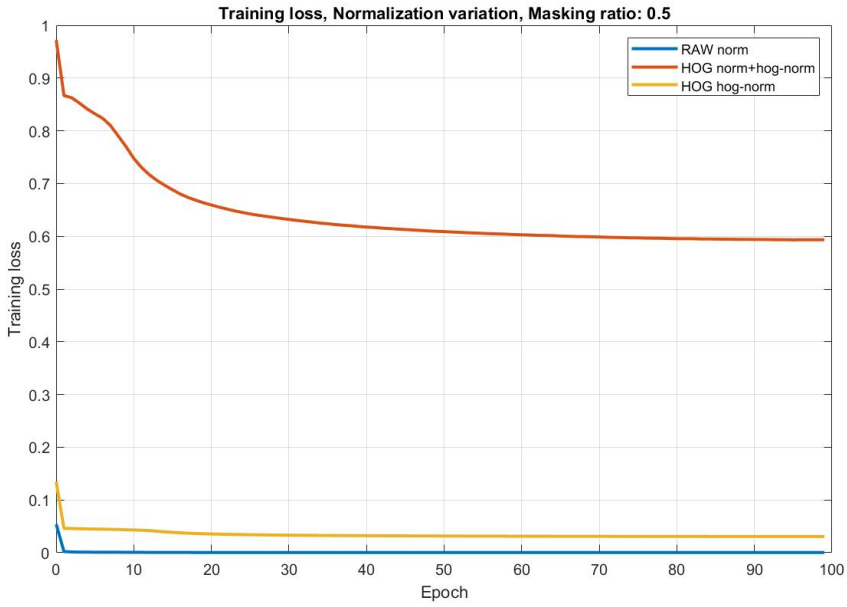


Figure 5.2 Training loss for masking ratio of 0.5

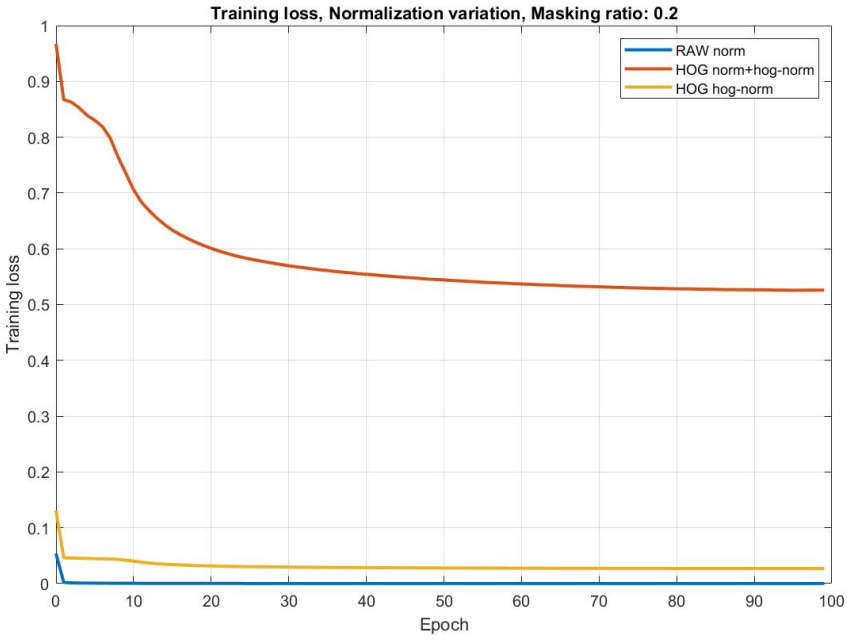


Figure 5.3 Training loss for masking ratio of 0.2

ground truth to learn from. Additionally, from the normalization study it can be seen that the cases where the general normalization was applied or no normalization at all present the lowest training loss. Therefore, it can be deducted that the normalization do not significantly improves the training loss performance, however, the case may be different for the image reconstruction as well as the transfer classification results.

5.1.2 Image reconstruction

The pipeline is addressed in a similar manner to the training loss analyzed before, a complete analysis of the masking ratio of 0.7 with all the normalization studies is shown, while for the masking ratios of 0.5 and 0.2 the most relevant image reconstructions are presented. Since it is also intended to get a comparison with a RGB visualization, it is also included to get a picture of the image reconstruction concerning the visual spectrum.

As it is presented in Fig. 5.4, different classes from the SSL4EO dataset were chosen to give a representation of the image reconstruction with a masking ratio of 0.7, these reconstructions were performed using a no-normalized ablation study. As the RGB analysis is not the main focus of the present work, a masking ratio of 0.7 for RGB is the only used since it also provides the necessary results for its comparison with HOG image reconstructions. For each image it is shown the original image *(left)*, a masked input *(middle left)*, the image reconstruction without the original image patches *(middle right)*, and the reconstruction + visible image *(right)*.

For HOG image reconstruction in Fig. 5.5, the histogram oriented gradients are taken as a target for the reconstruction, the distribution of the images goes on a similar manner to the RGB reconstruction, adding the target HOG at the left-most side of the columns. The distribution of the color is due the usage of the library to perform the visualization of the gradients. The intensity distribution goes in function of the color, the yellow represents the pixels with the major intensity while blue/dark blue color represents the pixels with the less intensity.

From the HOG image reconstructions it can be spotted that the reconstructed gradients follows the behavior of the target HOG images, there are minor variations with variations in each class but they achieve its function, the prediction of the feature target and its reconstruction in this terms.

In the Fig. 5.6, it can be visualized the difference on the same masking ratio of 0.7 with the other normalization scenarios. From this figure it can be observed that each row

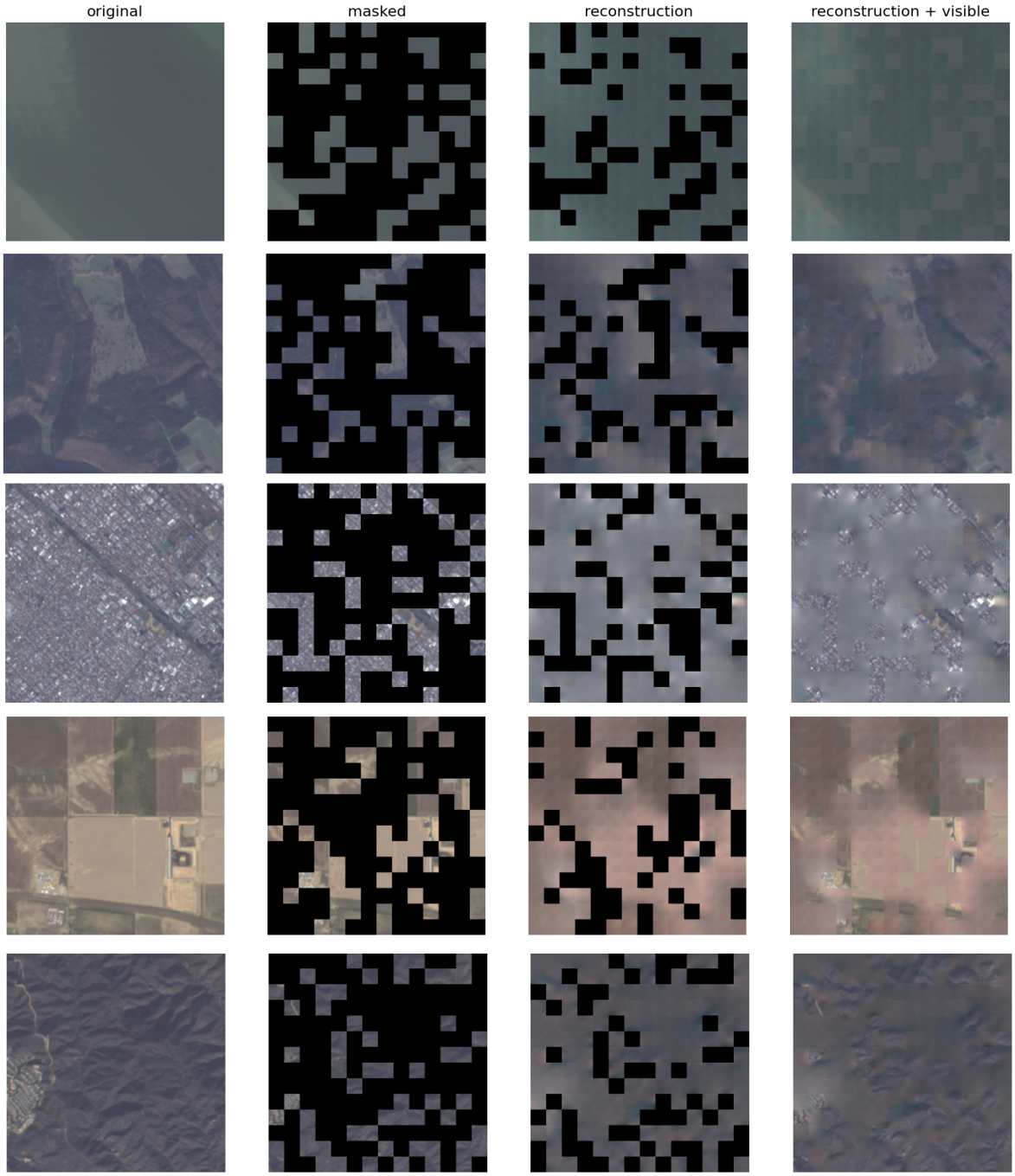


Figure 5.4 RGB image reconstruction

corresponds to each normalization scenario in HOG image reconstruction, the first row *(upper)* corresponds to norm+norm-hog (C), the second row *(mid-upper)* is norm-hog

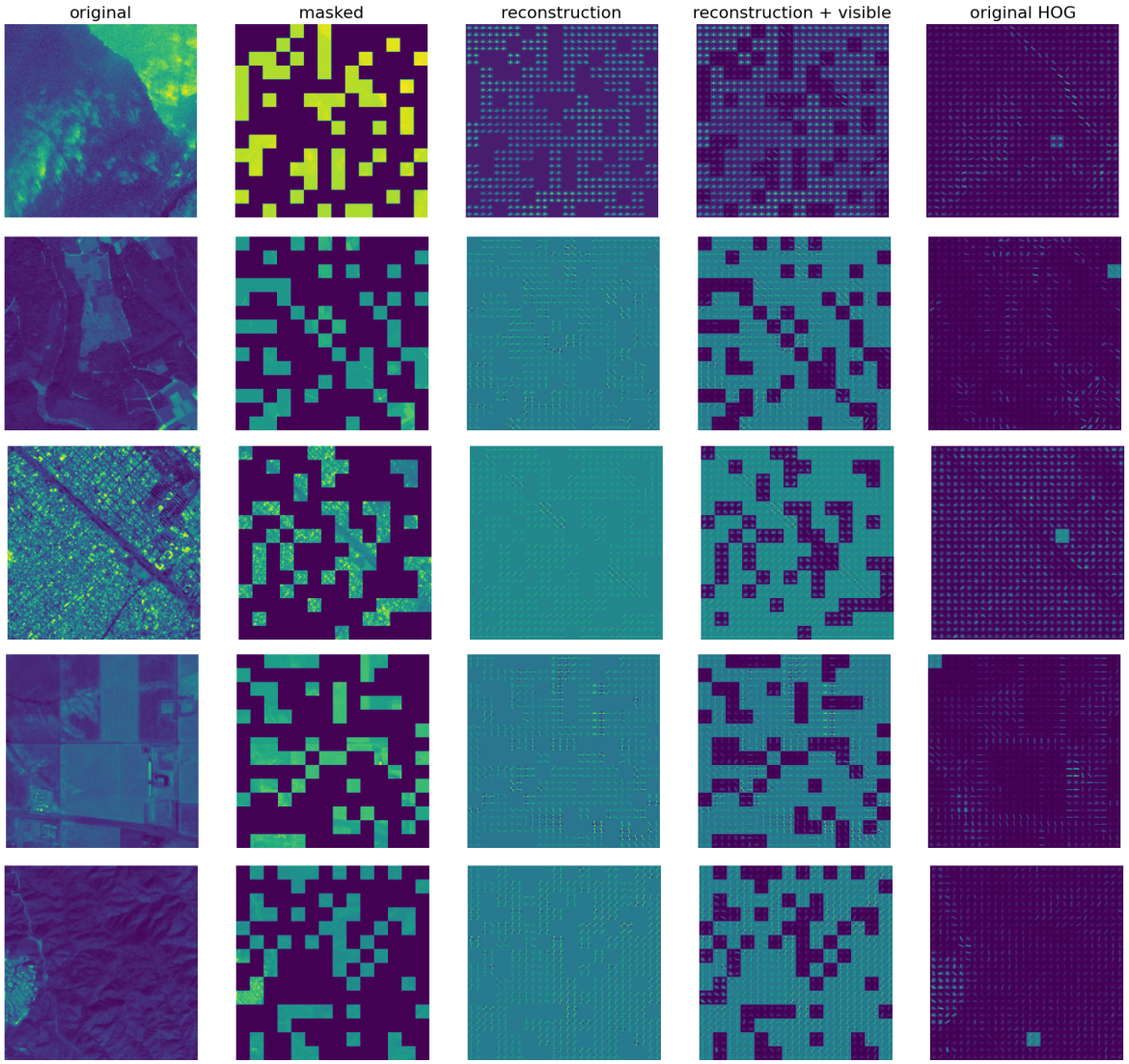


Figure 5.5 HOG image reconstruction, mask: 0.7, norm+hog-norm

(D), the third one *(mid-bottom)* addresses scenario norm (E), while the last row *bottom* is the one without any normalization (F).

From Fig. 5.6, the only scenario suitable for HOG image reconstruction is the one with both normalizations while the rest of the scenarios shows unrecognizable patterns due to the intensity of the gradients. This just affects the image reconstruction, later in the transfer classification task it will be observed that all the scenarios provide a relevant classification performance for the EuroSat dataset.

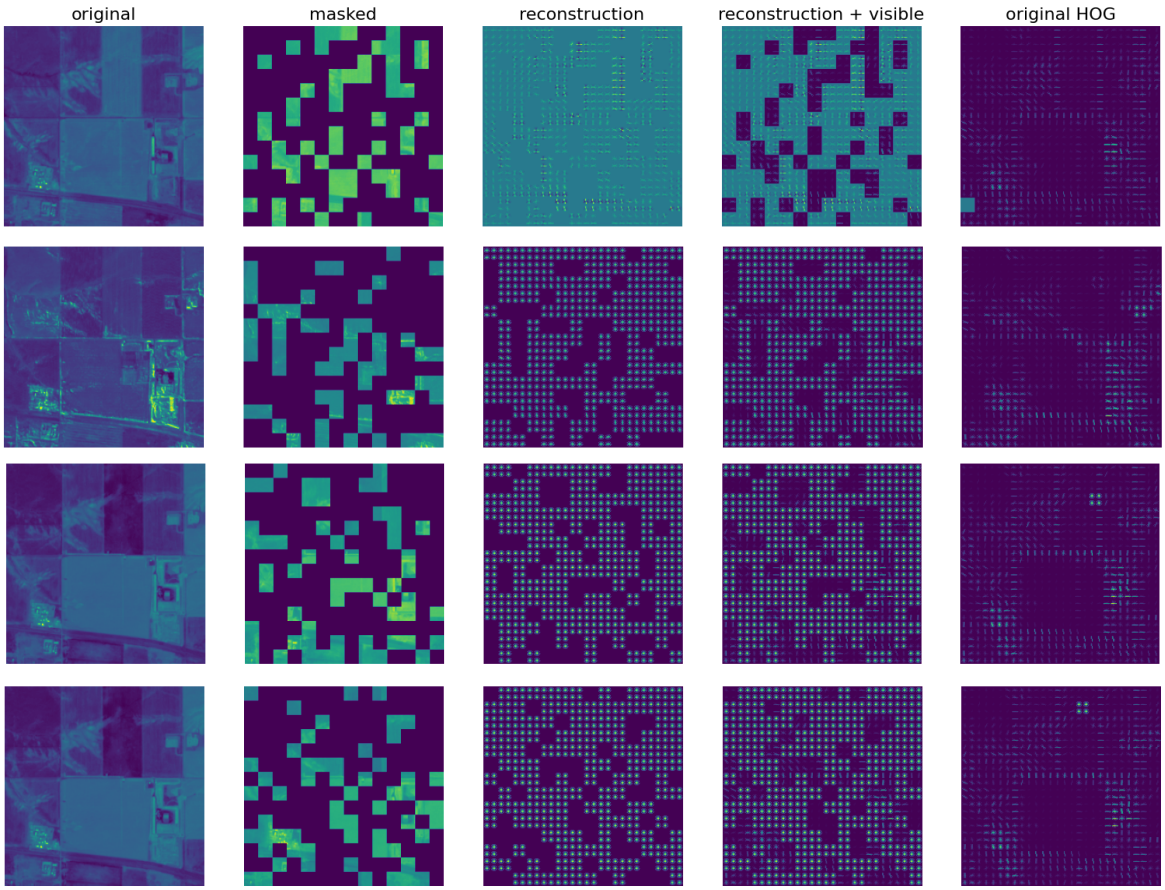


Figure 5.6 HOG image reconstruction, normalization study comparison

The last analysis in this section is the variation of the mask and its reconstruction in the HOG feature. The variations are oriented to three different masking ratios: 0.7, 0.5 and 0.2 with the normalization study of *norm+hog_norm* as it was presented in Fig. 5.5. In Fig 5.7 the masking ratio of 0.7 *(upper)* is the one with the major masking coverage. Thus, it is the strategy that requires more reconstruction coverage as it is shown in the reconstruction column, the masking ratio of 0.5 *(middle)* is half covered, thus, partly dealing with the original image patches and predictions for the posterior reconstruction. While the one with less coverage ratio is masking of 0.2 *(bottom)* where it can be appreciated the most patches of the original images with a few masked, therefore, few patches to be reconstructed.

We can observe that the reconstruction performance it is not affected significantly by the masking ratio, if they are compared with the target feature *(original HOG)*, the

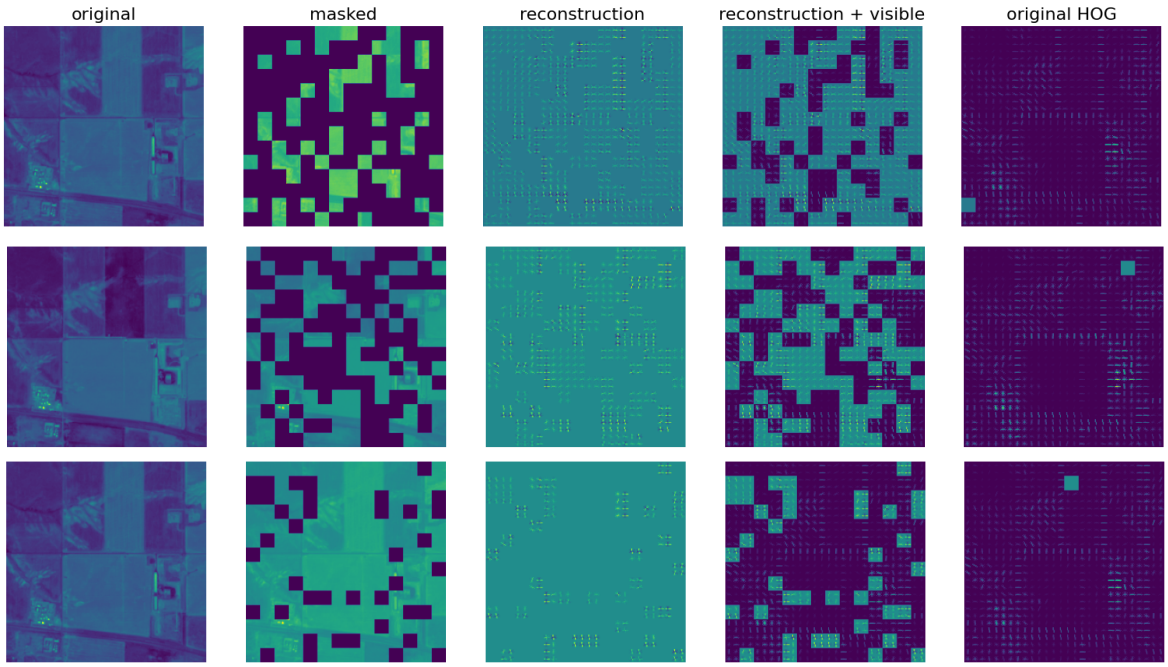


Figure 5.7 HOG image reconstruction, masking ratio comparison

(reconstruction + visible) images show the predicted gradients following or attempt to follow the ones presented in the target, no matter if the original images are 70% or 20% covered.

5.2 Downstream Task

This section presents the classification task results and are divided in three phases: the classification accuracies graphs that show the final achieved values, a deep analysis and comparison of the results provided by the ablation study and evaluation strategies, and the analysis of the classification by means of a confusion matrix.

5.2.1 Classification accuracies

Th graphs included in this section represents the behavior of the classification accuracies with respect the epochs and are presented in the different performed scenarios. The highest values from the normalization study are taken with the variation of the masking strategies taken during the pre-training phase.

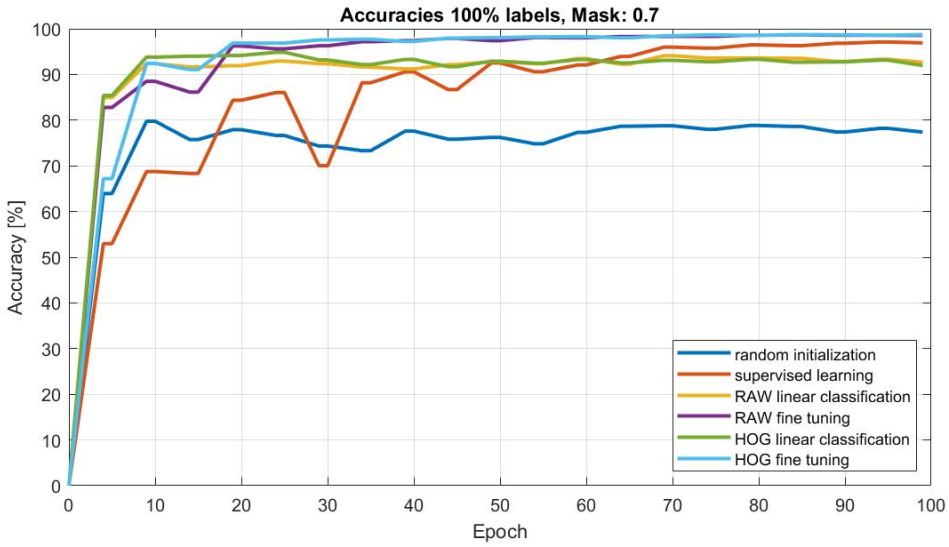


Figure 5.8 Classification accuracies with 100% of the labels and masking ratio of 0.7

In Fig. 5.8, the accuracy with the highest value is represented by the HOG fine tuning process while in a similar behavior is the RAW fine tuning accuracy with a slightly less final value after the training. Then, the supervised learning process shows a relatively high accuracy surpassing the linear classification results, providing enough evidence about its benefits and why it is still a good training strategy but still below SSL. Additionally, the linear classification performances achieve accuracies above 90% demonstrating that are also reasonable strategies to follow. Finally, the random initialization shows the worst performance by achieving a final accuracy below 80%.

Fig. 5.8 also demonstrates the initial bad performance of supervised learning which eventually surpasses the linear classification results after 100 epochs, moreover, the behavior of all the training strategies taken initially achieve after 5 or 6 epochs accuracies above 50%. Therefore, they increasing its performance with some variations until the final epoch stated at 100. Hence, the best way for classification training for the Eurosat dataset was fine-tuning with 100% of the labels using self-supervised learning.

From Fig 5.9 to Fig. 5.11 the performance decreases within the usage of the labels. With 50% of the labels the scenario seems similar with respect the one with the 100% of the labels. However, linear classification results approaches to the ones obtained by supervised learning. At 10% of the labels the scenario changes, the accuracies become more unstable and the linear classification results outperforms the supervised

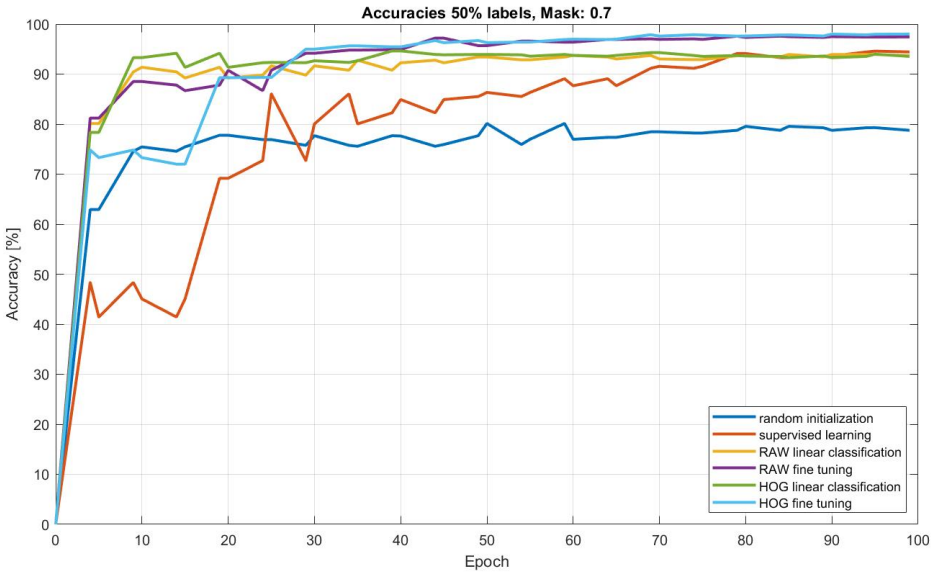


Figure 5.9 Classification accuracies with 50% of the labels and masking ratio of 0.7

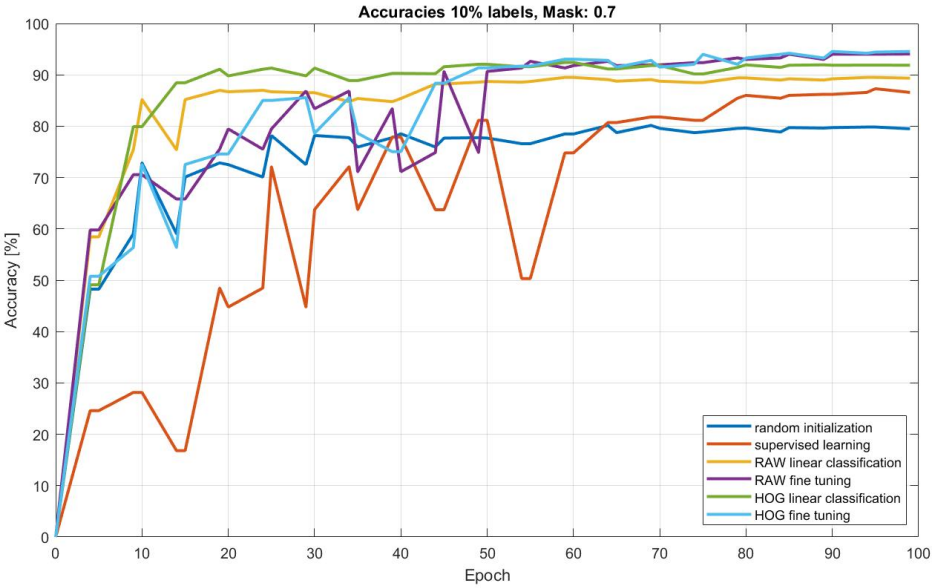


Figure 5.10 Classification accuracies with 10% of the labels and masking ratio of 0.7

learning ones while with 1% of the labels the scenario changes drastically, fine-tuning results are outperformed by linear classification results and even supervised learning is

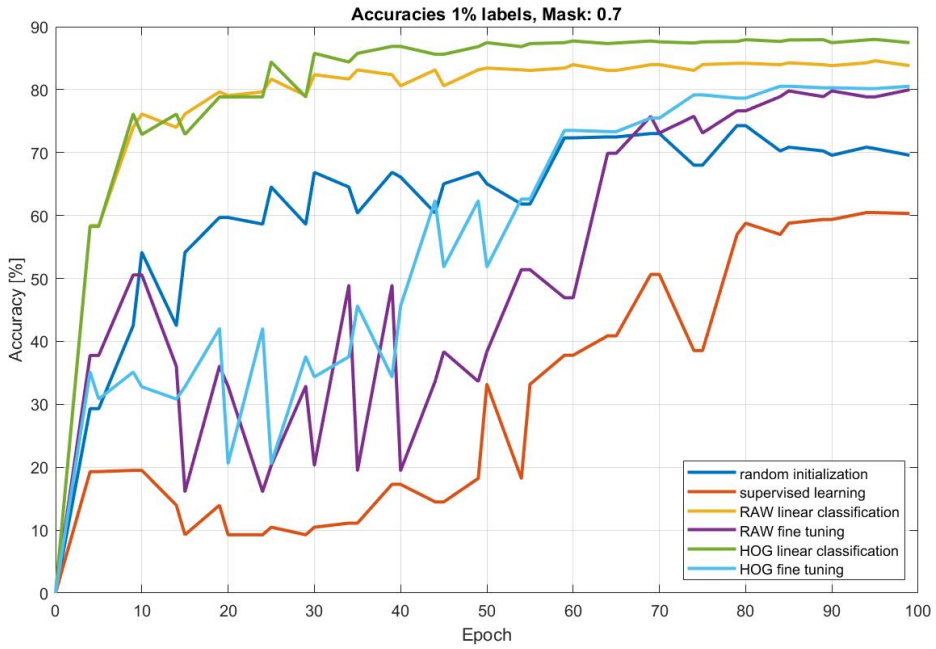


Figure 5.11 Classification accuracies with 1% of the labels and masking ratio of 0.7

outperformed by the random initialization approaches.

The less percentage of the labels is used, the worse is the fine-tuning and supervised learning performance, while linear classification results remain mostly unaltered as well as the random initialization results. The general performance of the classification task is significantly worse if the percentage of the labels is decreased.

If Fig. 5.8 is compared with Fig. 5.12 and 5.13, the results are almost the same since the behavior of the accuracies do not change significantly, therefore, it can be seen that the masking ratio does not affect in a significant way the final output of the classification task.

5.2.2 Evaluation strategies and ablation study

One of the strategies we have to analyze the performance of the model in terms of accuracy is performing a variation of the labels percentage and an ablation study.

Ablation study is performed to understand the causality in the system it is being analyzed, in other words, is a tool to better understand the behavior of the system, either

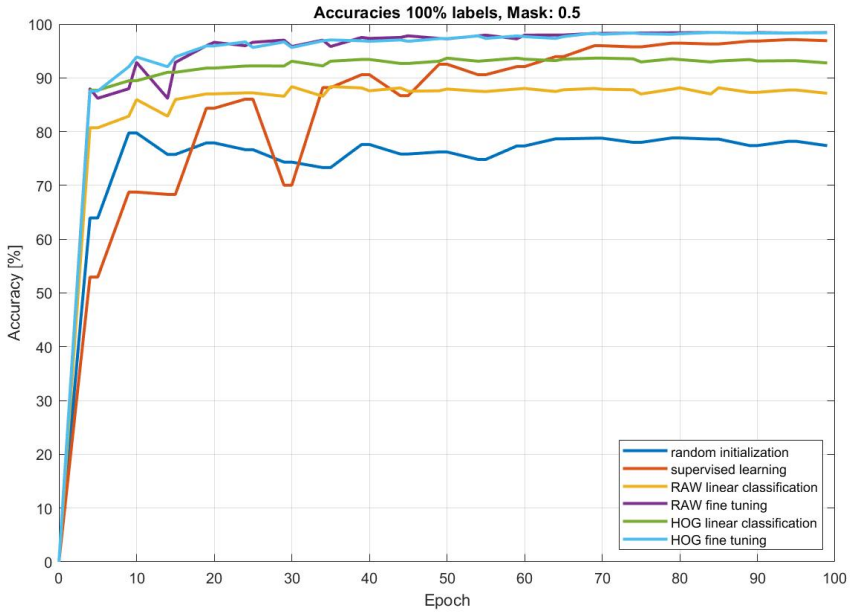


Figure 5.12 Classification accuracies with 100% of the labels and masking ratio of 0.5

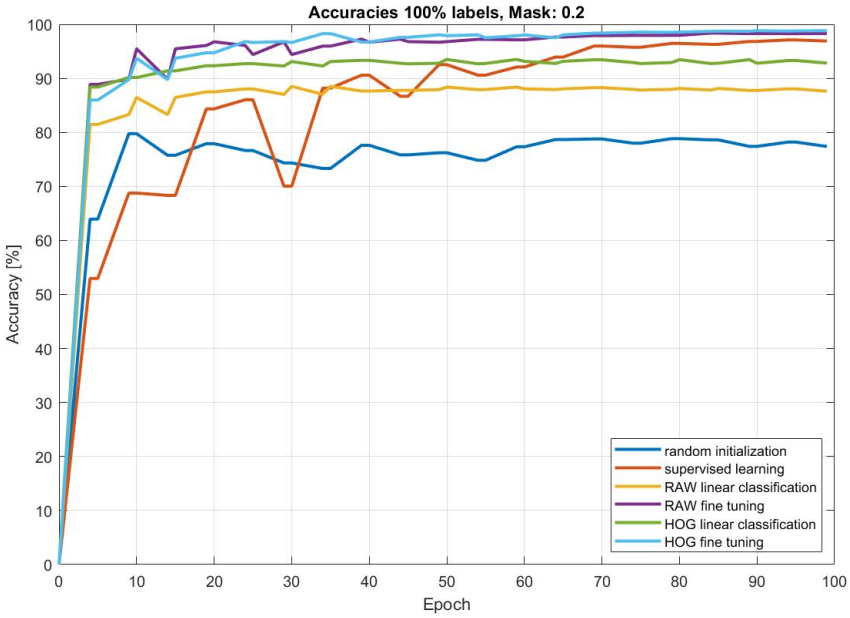


Figure 5.13 Classification accuracies with 100% of the labels and masking ratio of 0.2

by removing or perform variations in its execution. As it was presented during self-supervised pre-training for the results, some variations minimally affect the behavior of the results, while others provide a considerable variation. The two ablation studies in the present thesis are the variation of the normalization strategies taken from the pre-training phase and the variation of the masking ratio from the pre-training phase.

Additionally, in this thesis we perform an evaluation strategy namely the variation of the percentage of the labels of the Eurosat dataset, which have also an impact on the performance of the model. Moreover, the use of supervised learning and random initialization are also considered as evaluation measures to analyze the performance of the model.

Table 5.2 Labels' percentage variation

		100%	**50%**	**10%**	**1%**
Random Initialization		0.7840	0.7929	0.795	0.7062
Supervised Learning		0.969	0.9459	0.8729	0.6031
Linear Classif.	**RAW**	0.9267	0.9367	0.8935	0.8459
	HOG	0.9196	0.9398	0.9187	0.8798
Fine Tuning	**RAW**	0.9857	0.9742	0.9405	0.7998
	HOG	**0.9872**	0.9796	0.9442	0.8053

In Table 5.2, the highest accuracies from each label variation are included, no matter the normalization strategy taken. It can be seen that the highest classification accuracy is from the 100% of the labels using fine-tuning with HOG feature analysis followed by a similar value by the same strategy and label assignment but using RAW pixels instead of HOG gradients. Then, the third highest value comes from the supervised learning algorithm which increases its performance in function of the percentage of the labels providing a key aspect of the behavior of pure supervised learning. It performs better when there are more data available and even outperforms linear classification strategy taken from SSL.

From the side of linear classification, it outperforms fine-tuning when dealing with 50% of the labels by 1 or 2 points suggesting that with less data available it can be a strategy to follow to achieve reasonable good results. Finally, random initialization behaves as expected performing with low accuracies in the entire labels variation but still outperforming supervised learning when very few data is available, as in the case of 1% of the labels.

Table 5.3 Normalization variation

		norm	norm.hog	n+norm.hog	-
Linear Classif.	**HOG**	0.8598	0.9398	0.9125	0.8794
Fine Tuning	**HOG**	0.9661	0.9824	**0.9872**	0.9701

In table 5.3 the panorama changes, the ablation study is now focused on the performance over the normalization variations. Here the highest accuracy is presented and it now corresponds to the *norm+norm_hog* form, thus, it achieves its goal: it improved the performance of the model by 1 or 2 percentage points with respect other strategies or no normalization at all. In linear classification also adopting these variations provide a better performance with respect the scenarios without it.

Table 5.4 Masking Ratio

		M: 0.7	M: 0.5	M: 0.2
Linear Classif.	**HOG**	0.9196	0.9277	0.9283
Fine Tuning	**HOG**	0.9872	0.9838	**0.9877**

Performing a variation of the masking ratios also provides an interesting approach for the ablation study in MIM. As it can be observed from Table 5.4, all the masking ratios were performed under 100% of the labels and the highest values from the normalization variation are taken. We can see a new highest accuracy values for the whole study, with the same fine-tuning strategy, same normalization variation but a masking ratio of 0.2 the accuracy achieved is 0.5% greater than the one with a masking ratio of 0.7%. This could be because it requires less effort in the prediction for the model as it is just needed to reconstruct/predict 20% of the mask. However, it can also be observed that still a masking ratio of 0.7 is still generating the highest accuracies and probably due to the asymmetrical structure of *MAE+MFP* mainly talking about the encoder. For 0.7 masking ratio, the visible patches to be processed are just 30%, for a masking ratio of 0.2, the visible patches represents most of the image and needed to be processed by the encoder. Thus, this variation shows that both, advantages and disadvantages could be faced by performing a masking ratio variation.

5.2.3 Confusion Matrices

This section is divided in two phases, one showing the confusion matrices obtained with 100% of the labels and one showing the ones with 10% of the labels, this with the goal of presenting its performance when performing the classification of the 10 classes from EuroSat dataset. Additionally, a more accurate table presenting each class accuracy is addressed for both scenarios. Moreover, the normalization study is also presented in this analysis, where the highest accuracy scenarios such as *RAW norm*, *HOG norm_hog* and *HOG norm+norm_hog* comes into the analysis, as well as the outputs generated by using supervised learning with this classification tool.

Confusion Matrices 100% of labels

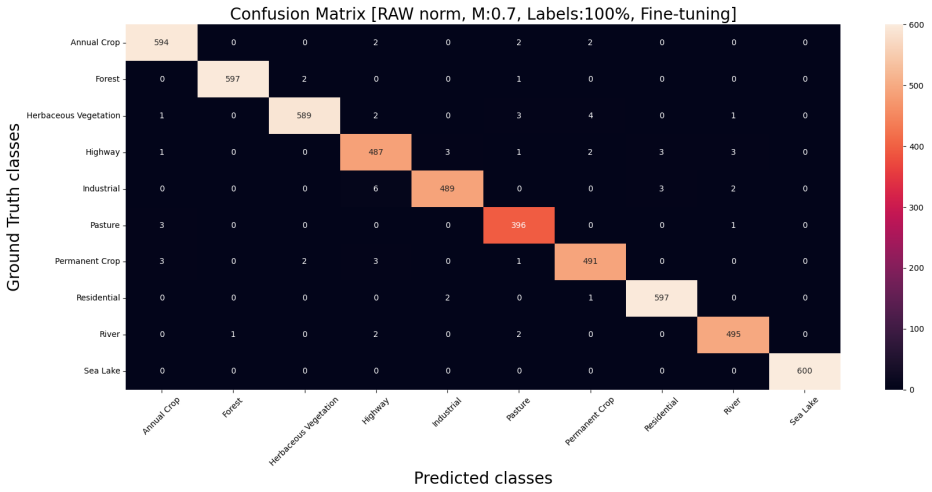


Figure 5.14 Confusion Matrix for 100% of labels, RAW (norm)

The confusion matrices explains how good was the classification with respects the classes it intended to classify, thus, it is compared the predicted images with respect the ground truth images. They are compared in a matrix where, if they are the same images, they match in the diagonal and are catalogued as correctly classified. In a perfectly classified model the diagonal should be filled up with numbers while the other parts of the matrix would be in zeros, however, as this is not the case, it will be found some misclassifed images represented outside the main diagonal. From Fig. 5.14 to 5.17 the main diagonal is mainly populated with correctly classified images,

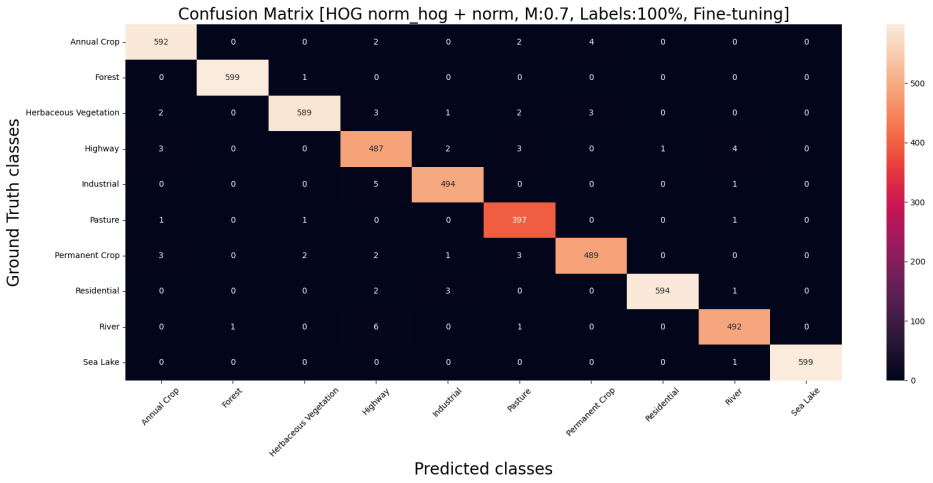


Figure 5.15 Confusion Matrix for 100% of labels, HOG (norm+norm.hog)

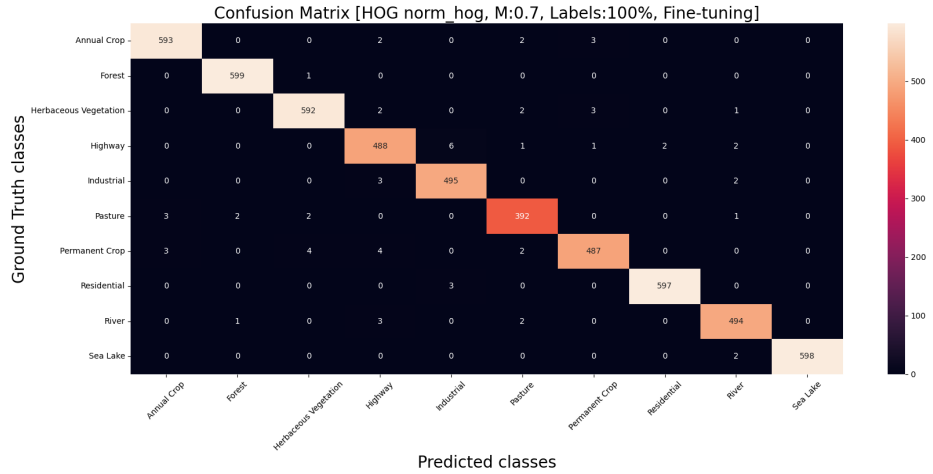


Figure 5.16 Confusion Matrix for 100% of labels, HOG (norm.hog)

while the rest of the misclassified ones are spotted randomly in the matrix in a very low rate.

In Table 5.5 the accuracies per class are compared with respect each normalization case, the highest ones for each feature, raw pixels as well as HOG features. Additionally, supervised learning class classification is included to get a better picture about how

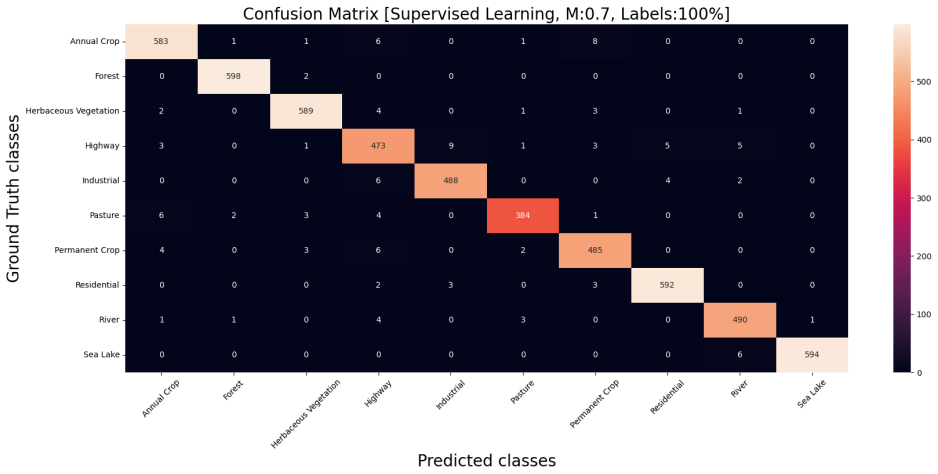


Figure 5.17 Confusion Matrix for 100% of labels, supervised learning

Table 5.5 Confusion Matrix: Accuracies per class for 100% of the labels

-	RAW norm	HOG norm+norm.hog	Sup. Learning
Annual Crop	99.0	98.83	97.16
Forest	99.5	99.83	99.67
Herbaceous Vegetation	98.16	98.67	98.16
Highway	97.4	97.6	94.6
Industrial	97.8	90.0	97.6
Pasture	99.0	98.0	96.0
Permanent Crop	98.2	97.4	97.0
Residential	99.5	99.5	98.67
River	99.0	98.8	98.0
Sea Lake	100.0	99.67	99.0

does it perform. In general, it can be said that there are two classes that performs better than the others, *Forest and Sea Lake*, they have the highest accuracies and even for RAW pixels *Sea Lake* achieves a perfect classification, while for *Forest*, HOG and supervised learning provides a nearly 100% accuracy. Moreover, the performance in general is a good one as it can also be observed at the confusion matrices, indicating that self-supervised learning is accomplishing its task to provide a clear classification scenarios of a space-borne dataset. For a 10% labels variation the scenario looks as

follows.

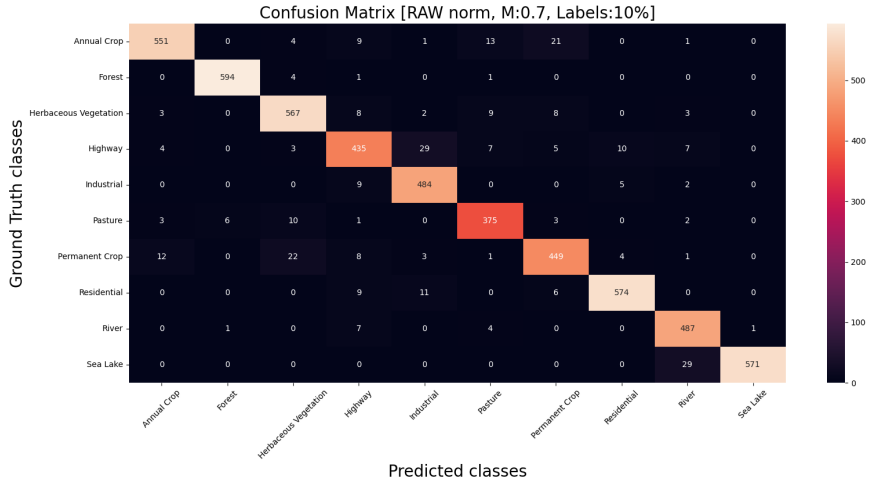


Figure 5.18 Confusion Matrix for 10% of labels, RAW (norm)

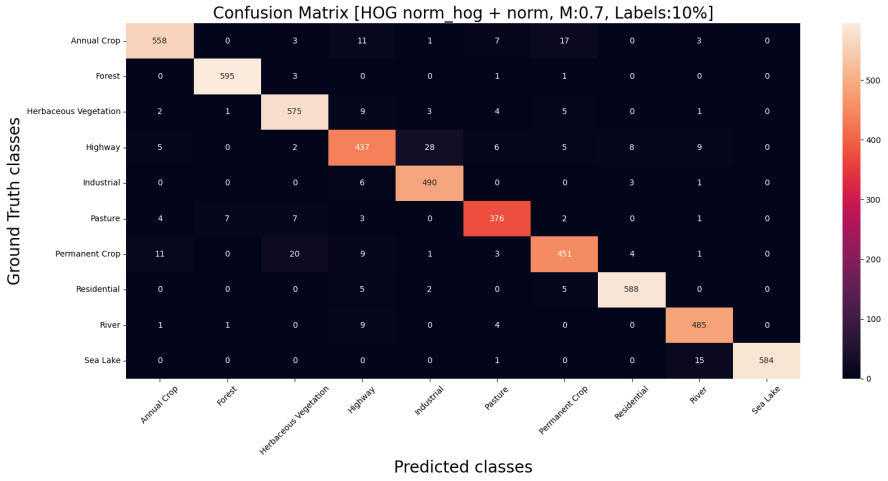


Figure 5.19 Confusion Matrix for 10% of labels, HOG (norm+norm.hog)

By the comparison between the confusion matrices with 100% of the labels and the ones with 10%, we can observe that the performance is lower as it was already analyzed in Table 5.2, this behavior also affects the classification capability of the classes and the distribution of missclassified images. We can observe from Fig. 5.18 to Fig. 5.21 that

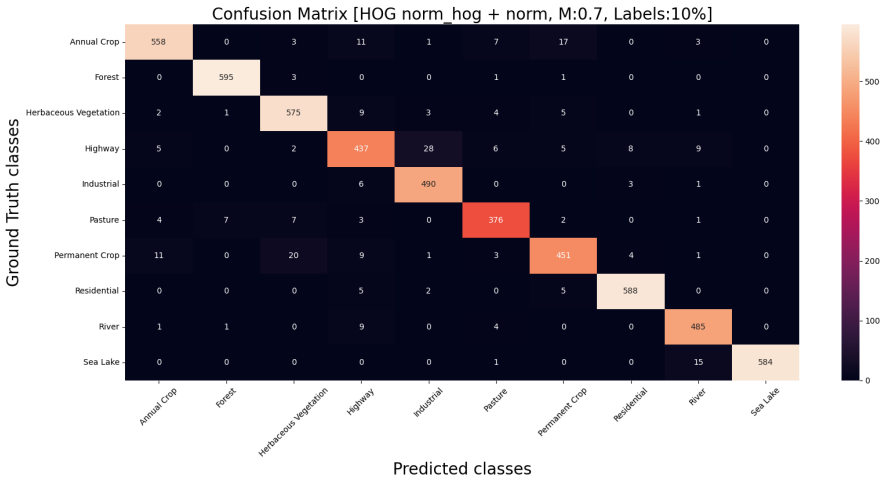


Figure 5.20 Confusion Matrix for 10% of labels, HOG (norm.hog)

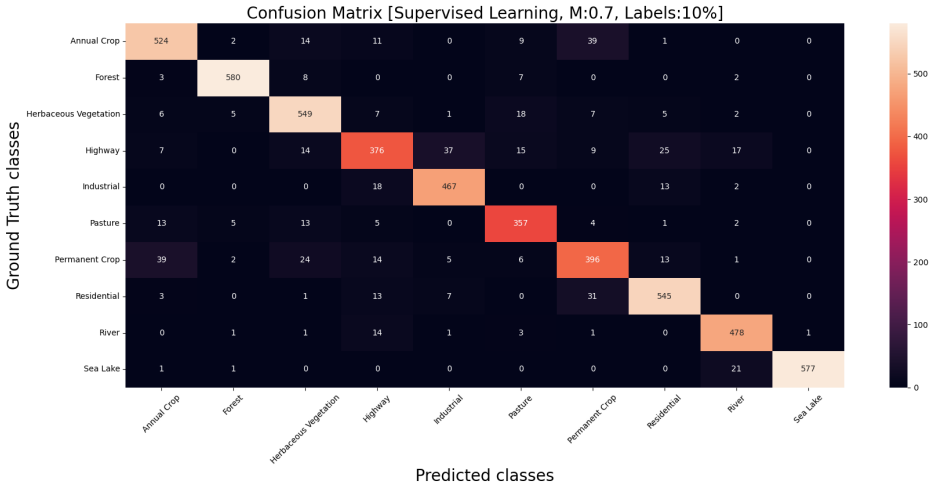


Figure 5.21 Confusion Matrix for 10% of labels, supervised learning

there are less correctly classified images in the main diagonal and more misclassifed values distributed in the confusion matrices. Thus, the accuracies per class could be seen as Table 5.6 suggests.

From the analysis of 10% of the labels, the performance is definitely lower, in some classes the classification does not reach the 80% of the accuracy and the highest one

Table 5.6 Confusion Matrix: Accuracies per class for 10% of the labels

-	RAW norm	HOG norm+norm.hog	Sup. Learning
Annual Crop	91.83	93.0	87.33
Forest	99.0	99.17	96.67
Herbaceous Vegetation	94.5	95.83	91.5
Highway	87.0	87.4	75.2
Industrial	96.8	98.0	93.4
Pasture	93.75	94.0	89.25
Permanent Crop	89.8	90.2	79.2
Residential	95.67	98.0	90.83
River	97.4	97.0	95.6
Sea Lake	95.17	97.33	96.17

barely surpasses the 99%. Additionally, the best classified class is the *Forest* one in a similar way as with 100% of the labels in Table 5.5, thus, it can be suggested that forestry patches are easily spotted and classified. Moreover, classes like *Highway and Permanent crop* are usually misclassified and the study suggest that they are easily addressed as other classes, for the case of the *Highway* that usually is included among other environments/classes and *Permanent Crop* which may be confused for *Annual Crop*.

Additionally, from Table 5.6 we can see that with 100% of the labels the RAW normalization tends to have better accuracies while with 10% is the case of HOG norm + norm.hog.

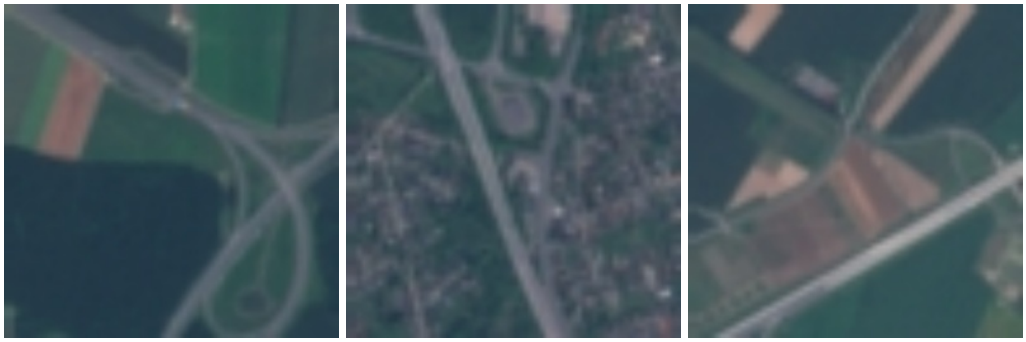
Finally, the variation of the labels is a significantly indicator that it affects how well a classification task is performed in SSL method, suggesting that the 100% of labels approach is always better to be used rather that the ones with less labels.

5.2.4 Misclassified Images

As a result for the confusion matrices we can spot some images that are misclassified, thus, the accuracy is not completely perfect and there are cases where it is hard to get with an accurate classification regarding the ground truth given by the original dataset. We will address the misclassified images by the supervised learning performed in masked autoencoders in comparison with our hybrid model, where the images were

correctly classified. Hence, we will observe the difference and performance between two strategies and which images are capable to be correctly classified if a hybrid strategy using feature descriptors such as HOG gradients are used for the analysis in the downstream task.

The following images are under the following criteria: they are misclassified in supervised learning but correctly classified by the hybrid model MAE+MFP using fine-tuning.



(a) (b) (c)

Figure 5.22 (a) Highway misclassified as permanent crop (b) Highway misclasified as permanent crop (c) Highway misclassified as Industrial

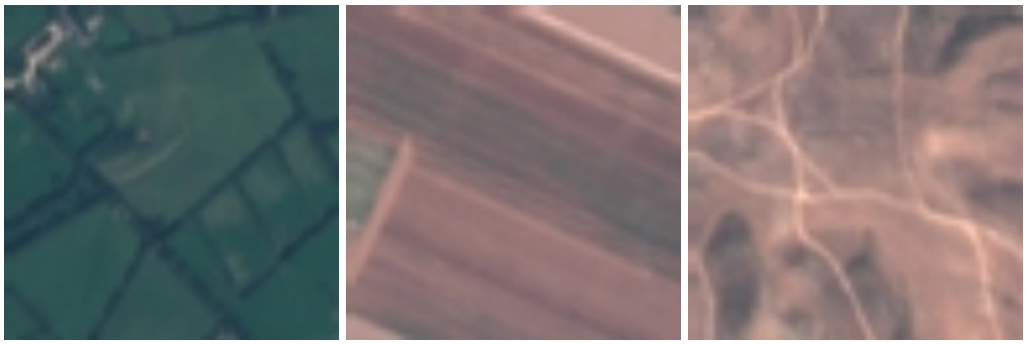
Highways are the most misclassified classes in the analysis (Fig. 5.22), partly because they form part of other classes and tend to combine together with other classes usually with crops, hence, several of them will be misclassified depending on their location. However, as the hybrid model MAE + MFP is capable to use feature descriptors, it will take the most relevant information about the details of the image and discard other non-relevant information such as the color in pixelwise examples.

In Fig. 5.23, we can observe another set of misclassification examples, industrial and residential zones are also subject of incorrectly labeling that can be assigned to other labels such as rivers or highways due to their paths such as main avenues or in case of (b) a misclassification for permanent crop, most probably due to the distribution and color of the image. Otherwise when worked with our hybrid model, the most important details are spotted, therefore, it can be classified correctly as a residential zone due to the distribution of the houses which were not spotted using MAE with supervised learning.

Another example (Fig. 5.24) is a set of cases when there are similarities between the

(a) (b) (c)

Figure 5.23 (a) Industrial zone misclassifeid as river (b) Residential zone misclassified as permanent crop (c) Industrial zone misclassified as highway



(a) (b) (c)

Figure 5.24 (a) Pasture misclassified as Herb. Vegetation (b) Annual crop misclassified as permanent crop (c) Herb. vegetation misclassified as permanent crop

land type where it is easy to get a misclassification due to the management of the vegetation or distribution/class of plants that are used in the acquired patches. Such is the case of the relation of the crops with pasture and herbaceous vegetation, where in (a) the pasture is spotted as vegetation, in (b) we can observe a confusion between permanent and annual crop, and in (c) the vegetation is taken as a permanent crop. However, this is addressed by the usage of the hybrid architecture which is capable to address the details of the patches and can determine the correct class of the acquired image.

Finally, in Fig. 5.25, we present some special misclassifications, in (a) it is easily misunderstanding by the MAE supervised learning algorithm the labelling of permanent crop

(a) (b) (c)

Figure 5.25 (a) Permanent crop misclassified as highway (b) Sea lake misclassified as river (c) River misclassified as highway

with highway, probably there is a way used by humans to manage the crop growing. However, it is not enough to being classified as such, then, using the HOG feature descriptor it can be labelled as permanent crop. In (b) we face an interesting example of an uncertainty between a lake and a river, the supervised learning classification set is as a river, probably because some examples of wide rivers. The spotting of edges performed by the hybrid model is capable to determined that it is actually a sea lake. Ultimately, in (c), we can see an interesting example where two classes are easily spotted, we can observe a river and a highway, so which is the correct label in this case?, according to a MAE model it is a highway. However, for an edge detection model such as MAE + MFP and for the human brain we can state that the river is most predominant in the current patch, hence, we can consider it as a river image.

By this last example, we can observe that the usage of an edge detection model has a big influence over what can be correctly classified taken into consideration the main details given by the edges. Thus, the change of light intensity in the acquired patches, this is important because it works as a filter for non-useful information that otherwise could interfere with the final classification.

6 Conclusions

For the present work, a masked autoencoders based on feature descriptors using histogram oriented gradients *(HOG)* is used to perform a pretraining of a spaceborne imagery oriented to Earth observation in the field of multispectral images *(MSI)* by means of self-supervised learning *(SSL)*. The proposed model encompasses new approaches to get with the classification by using vision transformers *(ViT)* as a backbone for the masked image modelling *(MIM)* based architectures, which have an asymmetrical characteristic to deal with the massive-scale datasets. Moreover, the usage of histogram oriented gradients allows a better acquaintance of edges in the images, a better distribution of the light intensity for the determination of the labels assigned to the images and an improvement of the performance when training the images by means of SSL.

The new findings in this work show that the performance is improved in comparison with the state of the art by achieving a classification accuracy of 98.72% which overrides the previous values achieved by pixelwise image oriented studies. Thus, we can state that feature descriptors improve the performance of the model's training as well as the classification tasks. Additionally, by a comparison of a classification by supervised learning with the presented hybrid architecture, the proposed model is capable to deal with ambivalent scenarios where the classes are not at all clear or there are two or more classes involved.

During our study we could observe that the final outcomes in terms of performance in comparison with Masked Autoencoder *MAE* model study do not present a big improvement but a slight improvement in the final accuracies. Hence, the pixelwise analysis already cover a good classification accuracy for satellite land-cover imagery.

Although the improvements of the present work are not significant in terms of performance of the model, it opens a new gate for future investigations on the field of feature descriptors, where a wide spectrum of analysis is open. HOG is just a way to deal with multispectral imagery, however, more descriptors like canny edges or

scale-invariant feature transforms can offer the possibility to outperform last results and get new findings in remote sensing for Earth observation.

List of Figures

List of Tables

Bibliography

[1] Jia Deng et al. "ImageNet: a Large-Scale Hierarchical Image Database." In: *IEEE Conference on Computer Vision and Pattern Recognition* (June 2009), pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on p. 1).

[2] Yi Wang et al. "Self-supervised Learning in Remote Sensing: A Review." In: *IEEE Geoscience and Remote Sensing Magazine* (2022). DOI: 10.48550/ARXIV.2206.13188 (cit. on pp. 1, 2, 3, 4, 8, 25).

[3] Xiao Xiang Zhu et al. "Deep learning in remote sensing: a review." In: *CoRR* abs/1710.03959 (2017). arXiv: 1710.03959. URL: http://arxiv.org/abs/1710.03959 (cit. on pp. 1, 12).

[4] Yann LeCun, Y. Bengio, and Geoffrey Hinton. "Deep Learning." In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539 (cit. on p. 1).

[5] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. "Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives." In: *CoRR* abs/1206.5538 (2012). arXiv: 1206.5538. URL: http://arxiv.org/abs/1206.5538 (cit. on p. 1).

[6] Linus Ericsson et al. "Self-Supervised Representation Learning: Introduction, advances, and challenges." In: *IEEE Signal Processing Magazine* 39.3 (2022), pp. 42–62. DOI: 10.1109/msp.2021.3134634 (cit. on p. 1).

[7] Zibei Wang. "Self-supervised Learning in Computer Vision: A Review." In: (2022). Ed. by Qi Liu et al., pp. 1112–1121 (cit. on pp. 1, 2).

[8] Yi Wang et al. "SSL4EO-S12: A Large-Scale Multi-Modal, Multi-Temporal Dataset for Self-Supervised Learning in Earth Observation." In: (2022). DOI: 10.48550/ARXIV.2211.07044. URL: https://arxiv.org/abs/2211.07044 (cit. on pp. 2, 8, 9, 29).

[9] Y. Lecun et al. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791 (cit. on p. 3).

[10] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition." In: *Neural Computation* 1.4 (Dec. 1989), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541. eprint: https://direct.mit.edu/neco/article-pdf/1/4/541/811941/neco.1989.1.4.541.pdf. URL: https://doi.org/10.1162/neco.1989.1.4.541 (cit. on p. 3).

[11] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929 (cit. on pp. 3, 8, 9, 18, 24, 26, 27, 33).

[12] Xiao Liu et al. "Self-supervised Learning: Generative or Contrastive." In: *CoRR* abs/2006.08218 (2020). arXiv: 2006.08218. URL: https://arxiv.org/abs/2006.08218 (cit. on p. 4).

[13] Dana H. Ballard. "Modular Learning in Neural Networks." In: *Cognitove Modeling* (1987), pp. 279–284 (cit. on p. 4).

[14] Zhenda Xie et al. "SimMIM: A Simple Framework for Masked Image Modeling." In: (Nov. 2021) (cit. on pp. 7, 14, 17, 23).

[15] Kaiming He et al. "Masked Autoencoders Are Scalable Vision Learners." In: *CoRR* abs/2111.06377 (2021). arXiv: 2111.06377. URL: https://arxiv.org/abs/2111.06377 (cit. on pp. 7, 9, 18, 19, 20, 23, 31).

[16] Chen Wei et al. "Masked Feature Prediction for Self-Supervised Visual Pre-Training." In: *CoRR* abs/2112.09133 (2021). arXiv: 2112.09133. URL: https://arxiv.org/abs/2112.09133 (cit. on pp. 7, 9, 18, 20, 21, 23, 31).

[17] Fuzhen Zhuang et al. "A Comprehensive Survey on Transfer Learning." In: *CoRR* abs/1911.02685 (2019). arXiv: 1911.02685. URL: http://arxiv.org/abs/1911.02685 (cit. on p. 8).

[18] Patrick Helber et al. "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification." In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2019) (cit. on pp. 8, 9, 30, 31).

[19] Patrick Helber et al. "Introducing EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification." In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2018, pp. 204–207 (cit. on pp. 8, 9, 30).

[20] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge." In: *International Journal of Computer Vision* 115 (Sept. 2014). DOI: 10.1007/s11263-015-0816-y (cit. on p. 11).

[21] Carl Doersch, Abhinav Gupta, and Alexei Efros. "Unsupervised Visual Representation Learning by Context Prediction." In: (May 2015). DOI: 10.1109/ICCV.2015.167 (cit. on p. 11).

[22] Priya Goyal et al. "Scaling and Benchmarking Self-Supervised Visual Representation Learning." In: (May 2019) (cit. on p. 11).

[23] Mathilde Caron et al. "Leveraging Large-Scale Uncurated Data for Unsupervised Pre-training of Visual Features." In: *CoRR* abs/1905.01278 (2019). arXiv: 1905.01278. URL: http://arxiv.org/abs/1905.01278 (cit. on p. 11).

[24] Ishan Misra and Laurens van der Maaten. "Self-Supervised Learning of Pretext-Invariant Representations." In: *CoRR* abs/1912.01991 (2019). arXiv: 1912.01991. URL: http://arxiv.org/abs/1912.01991 (cit. on p. 12).

[25] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning." In: *CoRR* abs/1911.05722 (2019). arXiv: 1911.05722. URL: http://arxiv.org/abs/1911.05722 (cit. on p. 12).

[26] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations." In: *CoRR* abs/2002.05709 (2020). arXiv: 2002.05709. URL: https://arxiv.org/abs/2002.05709 (cit. on p. 12).

[27] Jean-Bastien Grill et al. "Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning." In: *CoRR* abs/2006.07733 (2020). arXiv: 2006.07733. URL: https://arxiv.org/abs/2006.07733 (cit. on p. 12).

[28] Priya Goyal et al. "Self-supervised Pretraining of Visual Features in the Wild." In: *CoRR* abs/2103.01988 (2021). arXiv: 2103.01988. URL: https://arxiv.org/abs/2103.01988 (cit. on p. 12).

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Commun. ACM* 60.6 (2017), 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386 (cit. on p. 13).

[30] Zhicheng Zhao et al. "When Self-Supervised Learning Meets Scene Classification: Remote Sensing Scene Classification Based on a Multitask Learning Framework." In: *Remote Sensing* 12.20 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12203276. URL: https://www.mdpi.com/2072-4292/12/20/3276 (cit. on p. 13).

[31] Shuai Zhang et al. "Rotation Awareness Based Self-Supervised Learning for SAR Target Recognition." In: (2019), pp. 1378–1381. DOI: 10.1109/IGARSS.2019.8899169 (cit. on p. 13).

[32] Suriya Singh et al. "Self-Supervised Feature Learning for Semantic Segmentation of Overhead Imagery." In: (Sept. 2018) (cit. on p. 13).

[33] Chao Tao et al. "Remote Sensing Image Scene Classification With Self-Supervised Paradigm Under Limited Labeled Samples." In: *IEEE Geoscience and Remote Sensing Letters* PP (Oct. 2020). DOI: 10.1109/LGRS.2020.3038420 (cit. on p. 13).

[34] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. DOI: 10.48550/ARXIV.2103.14030. URL: https://arxiv.org/abs/2103.14030 (cit. on p. 14).

[35] Jinghao Zhou et al. "iBOT: Image BERT Pre-Training with Online Tokenizer." In: *CoRR* abs/2111.07832 (2021). arXiv: 2111.07832. URL: https://arxiv.org/abs/2111.07832 (cit. on p. 14).

[36] Deepak Pathak et al. "Context Encoders: Feature Learning by Inpainting." In: *CoRR* abs/1604.07379 (2016). arXiv: 1604.07379. URL: http://arxiv.org/abs/1604.07379 (cit. on p. 14).

[37] Mark Chen et al. "Generative Pretraining From Pixels." In: Proceedings of Machine Learning Research 119 (2020). Ed. by Hal Daumé III and Aarti Singh, pp. 1691–1703. URL: https://proceedings.mlr.press/v119/chen20s.html (cit. on p. 14).

[38] Olivier J. Hénaff et al. "Data-Efficient Image Recognition with Contrastive Predictive Coding." In: *CoRR* abs/1905.09272 (2019). arXiv: 1905.09272. URL: http://arxiv.org/abs/1905.09272 (cit. on p. 15).

[39] Trieu H. Trinh, Minh-Thang Luong, and Quoc V. Le. "Selfie: Self-supervised Pretraining for Image Embedding." In: *CoRR* abs/1906.02940 (2019). arXiv: 1906.02940. URL: http://arxiv.org/abs/1906.02940 (cit. on p. 15).

[40] Peiyan Guan and Edmund Y. Lam. "Cross-Domain Contrastive Learning for Hyperspectral Image Classification." In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–13. DOI: 10.1109/TGRS.2022.3176637 (cit. on p. 15).

[41] Xuying Wang et al. "GSC-MIM: Global semantic integrated self-distilled complementary masked image model for remote sensing images scene classification." In: *Frontiers in Ecology and Evolution* 10 (2022). ISSN: 2296-701X. DOI: 10.3389/fevo.2022.1083801. URL: https://www.frontiersin.org/articles/10.3389/fevo.2022.1083801 (cit. on p. 15).

[42] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection." In: 1 (2005), 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177 (cit. on p. 21).

[43] Adrian Rosebrock. "Histogram of Oriented Gradients and Object Detection." 2014 (cit. on pp. 22, 23).

[44] Guillaume Alain and Yoshua Bengio. "Understanding intermediate layers using linear classifier probes." In: (2016). DOI: 10.48550/ARXIV.1610.01644. URL: https://arxiv.org/abs/1610.01644 (cit. on p. 25).

[45] Felix Yu. "A Comprehensive guide to Fine-tuning Deep Learning Models in Keras (Part I)." 2016 (cit. on p. 26).

[46] Ashish Vaswani et al. "Attention Is All You Need." In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (cit. on p. 26).

[47] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. "An Analysis of Deep Neural Network Models for Practical Applications." In: *CoRR* abs/1605.07678 (2016). arXiv: 1605.07678. URL: http://arxiv.org/abs/1605.07678 (cit. on p. 26).

[48] Darius Phiri et al. "Sentinel-2 Data for Land Cover/Use Mapping: A Review." In: *Remote Sensing* 12.14 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12142291. URL: https://www.mdpi.com/2072-4292/12/14/2291 (cit. on p. 29).

[49] Developer Google. "Normalization." 2022 (cit. on p. 31).

[50] Howard Chu. "Lightning Memory-Mapped Database Manager (LMDB)." 2015 (cit. on p. 32).

[51] Institute for Advance Simulation (IAS). "Juelich Supercomputing Centre (JSC)." 2022 (cit. on pp. 34, 35).