Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Datateknik
2022 | LIU-IDA/LITH-EX-A--2022/019--SE

Teaching an AI to recycle by looking at scrap metal

– Semantic segmentation through self-supervised learning with transformers

---

Lär en AI att källsortera genom att kolla på metallskrot

Edwin Forsberg
Carl Harris

Supervisor : Felipe Boeira
Examiner : Ola Leifler

External supervisor : Björn Algers & Anton Karlsson

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Abstract

Stena Recycling is one of the leading recycling companies in Sweden and at their facility in Halmstad, 300 tonnes of refuse are handled every day where aluminium is one of the most valuable materials they sort. Today, most of the sorting process is done automatically, but there are still parts of the refuse that are not correctly sorted. Approximately 4% of the aluminium is currently not properly sorted and goes to waste. Earlier works have investigated using machine vision to help in the sorting process at Stena Recycling. However, consistently through all these previous works, there is a problem in gathering enough annotated data to train the machine learning models. This thesis aims to investigate how machine vision could be used in the recycling process and if pre-training models using self-supervised learning can alleviate the problem of gathering annotated data and yield an improvement. The results show that machine vision models could viably be used in an information system to assist operators. This thesis also shows that pre-training models with self-supervised learning may yield a small increase in performance. Furthermore, we show that models pre-trained using self-supervised learning also appear to transfer the knowledge learned from images created in a lab environment to images taken at the recycling plant.

Acknowledgments

We want to express our gratitude to our examiner Ola Leifler for the opportunity and guidance, always pushing us to be better.

We would like to direct our gratitude towards our supervisor Felipe Boeira at LiU and Anton Karlsson and Björn Algers at Combitech for the support and your insightful discussions.

We also would like to thank Malin Lindberg and team Tofu at Combitech for giving us the opportunity and making us feel welcome.

Lastly, we want to say thanks for all the support of our respective families and friends.

Contents

List of Figures

List of Tables

1 Introduction

The task of sorting waste is something that most people do in their everyday life, but it does not stop at the waste bins. Large companies and organisations further sort the waste and reintroduce it into society. One of the largest recycling companies in Sweden is Stena Recycling and one of the materials that they sort is aluminium. Recycling aluminium contributes to reducing climate emissions, as recycling 1 tonne of aluminium reduces carbon dioxide emissions by 13 tonne [36]. This reduction is roughly equivalent to the total emissions caused by three average Swedish citizens in a year [11]. The sorting process at Stena Recycling is mostly automatic and aluminium is one of the most valuable materials that they sort and sell to their customers. Because of this, Stena recycling does not want aluminium that could be sold to customers to go to waste and therefore it is essential that the material is properly sorted. This is why there exists an interest in automatically validating that the sorting is correct. Earlier studies have shown that this is possible using machine vision techniques [2].

One particular approach to achieve this is to do semantic segmentation. Semantic segmentation is the process of classifying each pixel into categories of different objects, thus semantically segmenting the picture. For instance, highlighting aluminium in a picture of scrap metal [21].

The data required for this task needs to be labelled manually, which is costly and time-consuming since models often train on datasets with hundreds of thousands of examples or more. To alleviate this, self-supervised learning (SSL) has been studied more and more which gives the ability to pre-train models on unlabelled data. However, most studies in the field of SSL focus more on general tasks and benchmark datasets, instead of a single specific application [9, 8, 50, 51].

This thesis explores the viability of pre-training a transformer model using SSL to do semantic segmentation on pictures containing scrap metal and thus separating aluminium, non-aluminium objects and background. Practically this could be used to inform operators about potential issues in the sorting process. In turn, this would lead to less waste and increased efficiency.

1.1 Motivation

The motivation behind this thesis is to explore if advancements in machine learning, can be used to give more information to operators at the recycling plant so that they can make better

decisions. Earlier studies at Combitech have shown promise using machine vision for this task, however, significant hurdles remain before they can be used in practice. This thesis investigates the possibility of addressing these issues by applying recent advancements in machine learning.

Two of the main issues within the field of machine vision are that pictures are large and it is not obvious which pixels are related to one another. One solution to this is to use convolutional neural networks which assume that pixels close in the image are more related to each other [40]. However, sometimes faraway pixels are essential to each other. To solve this problem transformer models can be used which is an architecture taken from the field of natural language processing. Transformers use a mechanism called attention to determine what part of the data is most related to other parts of the data [43].

A problem that occurs in almost all areas of machine learning is acquiring good training data that is correctly annotated. Finding training data that can be used to train a model good enough to generalise and work on unseen data is cumbersome and not always possible to find. For example, it is easy to acquire images of scrap metal from the recycling plant. However, the process of annotating the images requires manual labour since areas of the image need to be specified as aluminium or other materials.

A proposed method for overcoming the problem of small or low-quality datasets is to use SSL. This method attempts to extract information from unlabelled data. Within machine vision, early attempts of SSL manifested in removing a part of an image and having the model recreate it and thus learning something about the image in the process. For this task there is no need for a manually created label of what is correct, the original image is the point of comparison [51].

Using a camera to distinguish aluminium from other refuse is not the only option. Studies trying to sort scrap metal have been conducted where both cameras and inductive sensors have been used [27]. However, using only a camera seems to show the most promising results over inductive sensors which is the reason why it is investigated in this thesis. It is also cheaper than using other sensors and more practical to implement in a real-world application.

1.2 Context

Combitech is the company that has commissioned the thesis work. They are a technology-focused consulting company that has worked with Stena Recycling for several years, in particular with Stena Recycling's plant in Halmstad which among other things sorts aluminium. These projects have been focused on producing and investigating solutions for metal refuse detection and estimation. Prior to this thesis, there have been projects involving machine vision solutions, however, each such project faced the issue of needing manually annotated datasets that have been deemed infeasible by Combitech and Stena. As Combitech is a consulting company with other machine vision projects, they are interested to investigate if SSL could be used in general for machine vision projects and use this project to test its viability.

The current method employed by Stena Recycling is that aluminium is mostly automatically sorted out. The aluminium which is not removed from the other material is gathered together with other material and sold in bulk for a lower price. This aluminium might not be used in applications specific to aluminium and thus it is in a sense wasted. From empirical studies conducted by Stena, it has been shown that the material after sorting still contains approximately 4% aluminium. Stena's ambition is to reduce this down to 2.5% because this would improve the resulting alloy the recycled material is used for. A potential solution is to use an automated system powered by machine learning to detect when aluminium is present on the conveyor belt after sorting and signalling to the personnel so that the material can be removed.

This thesis explores if using transformer models pre-trained with SSL could be used to achieve this. Along with exploring how different SSL frameworks, different amounts of data

and using lighting adjusted data might affect the performance. SSL in particular is interesting as it is costly and difficult to create annotated data of high enough quality in a sufficient quantity, which SSL could potentially alleviate.

1.3 Aim

This project aims to investigate the viability of using SSL to train a transformer model to detect aluminium in pictures of scrap metal. In addition, it aims to investigate if SSL can be useful to limit the need for annotated data. In addition the aim of this project is not to find the optimal possible performance for each analysed architecture and framework, but rather an exploration of general differences and trends. The focus lies in finding implementations that are good enough to be said to be representative.

1.4 Research questions

The research questions investigated in this thesis are:

1. How could a transformer model in conjunction with SSL be viable to assist with recycling aluminium at a recycling plant?
2. How is the behaviour and performance affected by using different SSL frameworks?
3. Does pre-training using SSL yield an improvement for classifying aluminium and non-aluminium objects?
4. How does the amount of annotated training data affect the performance?

2 Theory

Here the concepts used in the thesis and related work will be briefly explained. It is structured in such a way that the reader should be able to read only the sections necessary to understand the thesis work given their specific background knowledge within the field. As such each section is written in a way that is as self-contained as possible.

2.1 Fundamental concepts in machine learning

To be able to grasp the machine learning models and frameworks that are used and discussed in this project, it is important to have some knowledge of the fundamentals of machine learning. In this section fundamental concepts within machine learning are presented, these are necessary to understand later discussions, but the aim is to give an intuition rather than a detail-oriented description.

2.1.1 Neural networks

A neural network in machine learning is a collection of neurons loosely inspired by the neural networks found in biological brains. Neural networks are typically constructed by neurons arranged into layers. The most common kind of neural network is a feedforward network in which neurons are only connected forward to the next layer. Input neurons that feed into a number of intermediate layers of neurons which are called hidden layers, eventually feed into a set of output neurons. The strength of these connections, typically called weights, determine the behaviour of the network and the process of finding the optimal weights is the heart of machine learning. Figure 2.1 shows a representation of a typical neural network [44].

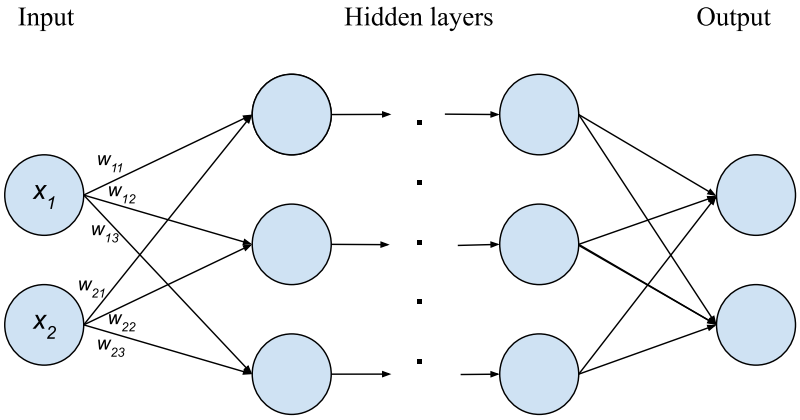

Figure 2.1: Illustration of a neural network architecture, w is the weights of the network, only the first weights of the network are labeled in the figure.

The value of each neuron is calculated using the previous layer. The output from each neuron in the previous layer is multiplied by the weight of the connection it has to the given neuron in the following layer as can be seen by the product $x_i w_{ij}$ in Equation 2.1. These products are summed together with a bias term which is a trainable weight that does not depend on input. Lastly, the sum is used as input to an activation function σ in Equation 2.1. An example of an activation function is the ReLU activation function which can be seen in Equation 2.2 where y is the output and x is the input [1].

$$y = \sigma(\sum_{i=1}^{m} x_i(\sum_{j=1}^{k} w_{ij})) + bias \tag{2.1}$$

$$y = \begin{cases} 0 & \text{if } x \leqslant 0 \\ x & \text{if } x > 0 \end{cases} \tag{2.2}$$

The neurons from one layer can either be connected to every neuron in the next layer or only to some of the neurons in the next layer. If every neuron in one layer is connected to every neuron in the next layer the network is said to be a fully connected neural network. One example of a task that a neural network can be used for is classification where each output neuron can represent a specific class and the output of each neuron represent the probability of it being that class [44].

2.1.2 Training

The process of training neural networks in general is the process of finding the correct weights for the connections. This is done iteratively by feeding input into the network and comparing it to the desired output. Connections that contributed to the correct answer have their weights increased, while connections that contributed to incorrect answers have their weights decreased [20].

The weights are updated by comparing the desired output to what was produced and analyzing what neurons contributed positively or negatively. In Figure 2.2 there is a simple example of three input neurons with one output neuron. In this example the network is in the process of being trained, it has received an input of a high value at the top neuron, a large negative value at the middle neuron and a small negative value on the bottom neuron. These inputs are multiplied with the weights representing the strength of the connections between the neurons. These products are summed together and can be thought of as the contribution

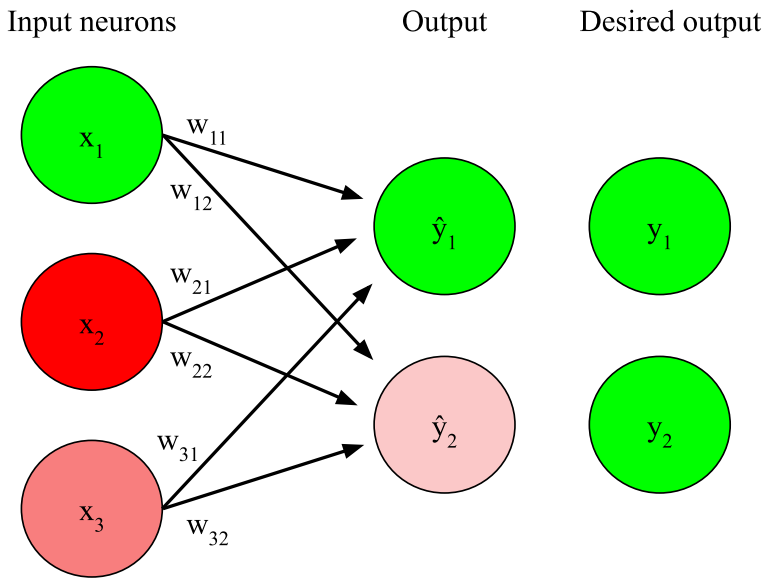

Figure 2.2: An example of a small network being trained. The input has a large positive value on the top neuron, indicated by the strong green color, a large negative value on the middle neuron and a small negative value on the bottom neuron.

of each neuron to the activation of the neuron in the next layer they are connected to. For instance, the weights of the connections between the nodes in the input layer and in the output layer are such that their sum is positive e.g. $\sigma(x_1w_{11} + x_2w_{21} + x_3w_{31}) = \hat{y}_1 > 0$, while the bottom neuron in the output layer is negative i.e. $\sigma(x_1w_{12} + x_2w_{22} + x_3w_{32}) = \hat{y}_2 < 0$. By comparing the desired output to the produced, it is noted that output Neuron 2 would ideally have a positive output. Thus, the connections that produced negative values into the output neuron will be decreased in proportion to their negative contribution. The connections that produced the positive values should be increased.

This process can be scaled up to large networks with many layers. For instance, in a network with three layers, the corrections for the output layer and the hidden layer can be found as previously described. The remaining layer can be found by using the hidden layer as if it were the output layer and adjust according to the desired values of the hidden layer. This is repeated for every layer. This process of letting the changes propagate backwards is called backpropagation [20].

Backpropagation produces the gradient for each weight through this method and can as such be thought of as how each weight ought to be updated. So that the discrepancy between the predicted output and desired output is minimized. The way the discrepancy is formally measured is through loss functions [20].

2.1.3 Loss functions

Loss functions in machine learning are used to compare the output of a model against a ground truth which in turn is used for updating the weights of the model. There exist several different loss functions that can penalise different aspects of the results depending on the end task of the model. One common loss function used for classification tasks is the cross-entropy loss function, which is taken from the field of information theory. Cross-entropy loss compares a probability output of a model against the actual class for each possible class. The function multiplies the distribution of the classes for the ground truth with the predicted probability distribution of the model and takes the sum of those products. The equation for

cross-entropy loss can be seen in Equation 2.3 where i is the class, m is the number of classes, $p(x_i)$ is the probability from the ground truth and $q(x_i)$ is the probability output from the model [7].

$$Loss = -\sum_{i=1}^{m} p(x_i) log(q(x_i)) \tag{2.3}$$

2.1.4 Optimisers

Training a neural network is the process of optimising the trainable weights of a network so that it minimises the loss function. Through this perspective, one can view training as exploring the state space for the model constructed through all the possible combinations of values the weights can take. Commonly, this is visualised as traversing a surface where the height symbolises the loss function of the model evaluated at that particular weight configuration with the other dimensions being all the trainable parameters. Thus, the objective is to find the local minima. It is unfeasible to try every combination and that is why a version of gradient descent is typically used. Gradient descent is the process of using the difference between the predicted output and the desired output to inform what direction and how far to move in the state space [18].

Gradient descent passes each data point from the dataset through the network to get the gradient used to step in the state space. The drawback with this is that it is very computationally intensive to use the whole dataset for each step, especially as the size of the dataset increase. Stochastic gradient descent (SGD) addresses this by using a smaller amount of data points to calculate the gradient, this is sample is usually called a batch. It rests on the assumption that a sample of the data points is sufficiently representative of the dataset at large which enables it to approximate the gradient for the whole dataset. This usually results in more steps, but is a lot faster since each step requires many times less computation [5].

A common obstacle that an optimiser encounter is ravines, which is where the surface varies much more along one dimension than the others. As a consequence of the topology, the optimiser's path will bounce back and forth between the walls of the ravine since the gradient along that dimension will be much larger than in the direction along the ravine. Several methods have been proposed to address this, for example through encoding momentum [25].

Having a momentum encoder, as in the case of the Adam optimizer, can intuitively be seen as adding weight and momentum to the path the optimizer takes, thus oscillating back and forth is not as prevalent, since switching direction rapidly changes the momentum [25].

To enable more parallel computation higher, batch sizes can be used so that more hardware can be utilised at once. To compensate for a larger batch size the learning rate is scaled since more data should result in larger steps as there is more information on which to make the decision. In practice, however, the learning rate can not be scaled too large and to address this layer-wise adaptive rate scaling (LARS) was proposed [49]. This technique allows each layer of the model to have its own learning rate based on the norm of its weights and the norm of its gradient as these can vary by a lot between layers. This enables the learning rate to be more dynamic which allows for larger batch sizes which in the end allows for a faster training process [49].

2.1.5 Learning rate scheduler

In the beginning of training a model, it is likely far away from a desirable local minima as its parameters are randomly instantiated. Thus it is often beneficial to take large steps in the configuration space. However, as the training continues, eventually the models parameter vector will settle close to a local minima as each step tend to bring it closer to a favourable state. As it gets closer it tends to be beneficial to move with smaller steps as otherwise the

model might pass by the local minima and move back and forth around the desired spot without reaching it [26].

To achieve this behaviour a learning rate scheduler is often used, which produces a variable learning rate. Often a mathematical function is used to create the schedule, typical functions are the cosine function or a linear function. This in combination with using an optimiser often results in shorter training times [30].

2.1.6 Overfitting and regularisation

Overfitting is a phenomenon within machine learning, where a network has a very low loss when applied to its training data i.e., examples it has trained on, but has a high loss on validation data or test data i.e., examples that it hasn't trained on. One can intuitively think about it as memorising instead of learning. Since the training of the network is minimising the loss function over all the pairs of input and output in the training data it is not explicitly minimising the loss outside of the training data. In the extreme case a perfectly trained network, from a training loss perspective, could trivially be a mapping for every input point to output point in the training data, essentially just a list memorising the training data [42].

This is an issue because any dataset is likely to contain some degree of noise and it is likely not perfectly representative of the hypothetical dataset containing all possible input-output pairs that could exist. To mitigate overfitting, regularisation of some form is often used [42].

$$loss_{total}(x) = loss(x) + \lambda \sum w_i^2 \tag{2.4}$$

Regularisation in machine learning often uses some measure of the complexity of the model with the objective to minimise it. From this point of view, it is preferable to have smaller weights in general as this would be a less complex model. Such a model is expected to generalise better than one having a few important weights highly attuned to the training data. The λ parameter controls how the fitness and complexity of the model should be weighed against one another. A high value on λ will highly penalize complex models while a small value will permit complex models and increase the importance of a tight fit onto the training data [42].

2.1.7 Embeddings and representations

In machine learning, it is often of interest to reduce the number of dimensions of input data, both to reduce computational cost and to extract information. Within natural language processing, a dictionary is often produced as a list of each word that occurs in a corpus. To encode a word the index it has in that list is used instead of a string of characters. However, this encoding, called one-hot encoding is not compact nor easy to do operations on as they are typically implemented as a vector with the length of the number of words in the dictionary, with one element being one and the rest being zero [28]. Instead, the vector is mapped into a latent space with much fewer dimensions. This space is constructed such that similar words are closer to each other than dissimilar words. Dimensions in the embedding space often correspond to specific features of the data. Finding such a mapping is not trivial, but is learned in the same manner as other layers are learned in a neural net [28]. This idea of representing data in latent dimensions is general and applies to machine vision as well. In some contexts, it is more common to use the term representation to refer to the same concept.

2.2 Self-supervised learning

Within machine learning, there are different methods to train the models. When using manually labelled datasets for training it is typically called supervised learning, this most often takes the form of a dataset being manually annotated with the information that the model is

predicting. For example, a classifier would have the class labels added to a dataset so it can compare its prediction to the true label [51].

SSL is a technique where the correct prediction already exists in the dataset without the need for human annotation. A common technique is to remove part of the data and have the model reconstruct it. Now there exists a natural point of comparison in the original data so the model can be trained more cheaply since it does not need annotated data [51].

The removal and reconstruction of data have been a very successful pretraining task for SSL in the field of natural language processing, resulting in pretraining frameworks such as the ones used for BeRT [15] and GPT [6]. This approach has not yet been shown to be very effective in machine learning. Instead, representation learning by augmenting pictures has been shown to be effective as a pretraining task [19, 50, 8, 9]. The general approach can be briefly characterized as taking an input image, applying some transformation to it, generating a representation for it by using the neural network and then comparing it to the representation generated for the same input image with a slightly different transformation applied to it. The task is ensuring similar images are mapped to similar representations. A common issue, usually called a collapse, is often faced when applying this general approach is that the trivial solution to guaranteeing that similar images are mapped to similar representations is a constant function [19, 50, 8, 9].

2.2.1 Contrastive and non-contrastive learning

SSL methods often focus on representation learning, which is the task of finding a way to represent the input in some latent dimension of features. The goal of representation learning is that input that is semantically similar will result in similar outputs while input that is semantically dissimilar results in dissimilar outputs. Contrastive learning frameworks use positive examples to push representation together and negative examples to pull representation apart [10].

One common technique for implementing contrastive learning is to use siamese networks. Siamese network often consists of two networks that are identical but are fed different inputs that are then compared. The input used is often grouped in pairs with either positive samples or negative samples together with a "normal" sample. The positive samples are supposed to be similar to the normal sample and are often produced by doing some sort of augmentation to the normal sample. Negative samples are often chosen by uniformly sampling the dataset. These samples are used so that the models do not collapse and always produce the same constant for every input. Since without negative samples the model can always yield the same output regardless of the input since the images are supposed to be similar and the output is now the same. However, finding good negative examples is not trivial since the definition of a negative example that is correctly dissimilar is vague [23].

Non-contrastive learning fixes this problem since it does not use negative examples while still comparing the embedding of two networks during training. To avoid collapse several different approaches have been published and some of the most common approaches include clustering and having different weights for the siamese networks [9, 8].

2.2.2 DINO

DINO is a framework developed by Facebook for non-contrastive SSL. DINO implements two networks with the same architecture but different parameters labelled student and teacher. An illustration of DINO can be seen in Figure 2.3 where Z_s and Z_t is the feature representation produced by the student and teacher network respectively. During training the gradients only backpropagate through the student network while the teacher's weights are assigned through an exponential moving average of the student's weights. The equation describing the weight transfer can be seen in Equation 2.5 [9].

$$\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s \tag{2.5}$$

During training the same image is sent through the teacher and the student, however, one of the networks gets an augmented version of the image. The output of the two networks is a representation vector each which is then used to calculate the cross-entropy loss. To avoid collapse the teacher network applies a centering and sharpening to the network output. The centering prevents the probability distribution output of the teacher network from spiking but encourages the probability distribution to collapse to the uniform distribution. While sharpening during the softmax step, does the opposite to balance out the encouragement of collapse from the centering [9].

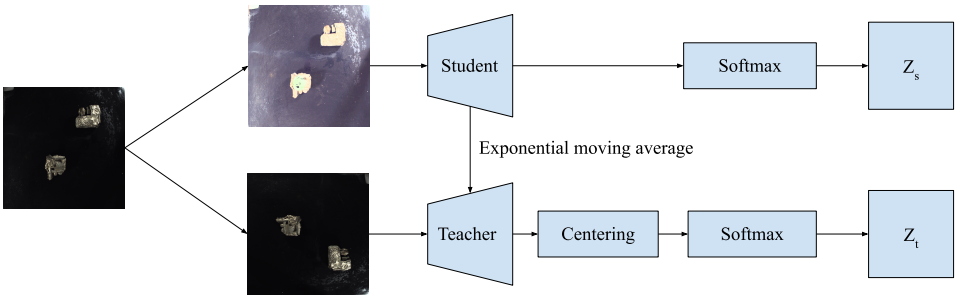

Figure 2.3: DINO architecture. Illustration inspired by figures in the paper introducing DINO [9]

2.2.3 SwAV

SwAV is a non-contrastive SSL framework introduced in June 2020 [8]. SwAV produces feature representations for two images that are augmented versions of the same image. What differentiates SwAV from most SSL frameworks is that it uses clustering to separate representations. During training, a unit sphere consisting of K different clusters is computed. Each representation vector is used to assign the image to these clusters. The network swaps the clustering assignment of the two images and from this tries to predict the representation vector that created the cluster assignment. It is the result of these predictions that are then used in the loss function. Which, in turn, is used to calculate the gradients and update the weights of the models. SwAV avoids the trivial solutions by enforcing that the distributions of cluster assignments is even among all clusters. This prevents the model from always predicting the same clusters for every representation [8]. An illustration of the SwAV architecture can be seen in Figure 2.4, where f is the network being applied to both images, Z_1 and Z_2 are the features representation of the images. Q_1 and Q_2 are the clustering assignments based on the feature representations Z_1 and Z_2 combined with C, which is the vector representation of the K clusters [8].

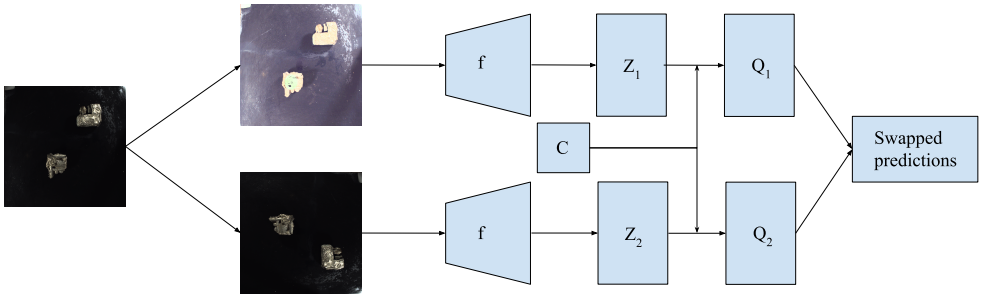

Figure 2.4: SwAV architecture. Illustration inspired by figures in the paper introducing SwAV [8]

2.2.4 Barlow Twins

Barlow Twins is a non-contrastive SSL framework proposed in March 2021 [50]. The framework avoids collapse by using a novel loss function. Similarly to previously discussed SSL frameworks, Barlow Twins applies augmentations to a batch of images and produces feature representations for these. The representations of these images are used to compute a cross-correlation matrix, which is calculated by multiplying the representations and normalising the results. An illustration of the architecture for Barlow Twins can be seen in Figure 2.5 where f is the network, Z_1 and Z_2 are the representations of the images. The values of the matrix are between -1 and 1 and the goal of this matrix is to maximise the diagonal elements while having all other values as close to 0 as possible. This is because the closer to 1 these elements are, the more similar the representations are. This loss function compares the cross-correlation matrix against an identity matrix which is the optimal result of the cross-correlation matrix. Thus the invariant information is maximised and the redundant information is minimised. This avoids collapse since the trivial solution of assigning the same feature representation for every input produces only redundant information. Barlow Twins have been observed to be affected significantly by what augmentations are chosen, this can be used to control what type of information the representation should find [50].

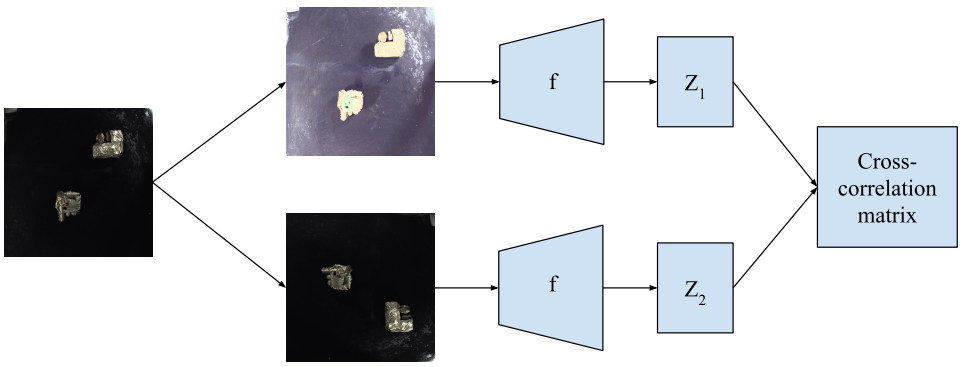

Figure 2.5: Barlow Twins architecture. Illustration inspired by figures in the paper introducing Barlow Twins [50]

2.2.5 Vissl

Vissl[1] is an open source library developed by Facebooks AI research (FAIR) team. It is built using the Pytorch library and implements the SSL frameworks proposed by the FAIR team. Among the frameworks implemented are DINO, MoCo, SwAV, SimCLR and PIRL.

2.3 Architectures

In the thesis, several model architectures were used. As the full system consists of an encoder and decoder a suitable architecture for each was needed. In this section, the architectures used and considered in the project are explained.

2.3.1 Transformer architectures

The transformer model is a machine learning model that has its origins in natural language processing (NLP) and is used to handle sequential input data. This data is typically a sequence of word embeddings. Transformers use a mechanism called attention to relate the sequential input to each other. As it comes from the NLP subfield of machine learning, it was first introduced as a way to relate words in translations from different words to one another. Thus capturing what data points in the input had the largest effect on the output, i.e. what to pay attention to in a sense. When the concept is applied to machine vision the implementation is changed slightly, but the intuition behind it remains useful for understanding it. In machine vision applications it captures the relationship between pixels that are part of the same object. Specifically, transformers use multi-headed attention which applies the attention mechanism several times in parallel capturing different facets of the information and how different patches relate to one another [46].

One of the first uses of the attention mechanism within the machine vision field was the Vision Transformer (ViT) [16]. It splits up an image into patches of pixels, grouped together in squares. Originally the patches were 16 by 16 pixels in size. These patches are embedded into a learned latent dimension and in turn, used analogously as word embeddings in the transformer models from the NLP field.

Swin Transformer

Shifted windows (Swin) Transformer is a transformer architecture developed by Microsoft and used for vision tasks. It was first proposed in march 2021 and established a new state of the art for several vision-related tasks, among them semantic segmentation where it achieved the highest metric on the ADE20K dataset [34]. The Swin Transformer is a hierarchical transformer architecture which uses image patches of varying size in hierarchical layers where the goal is to capture both global and local features.

The model takes in an image which is split into several windows and processed by the first layer. The results of this is sent to the next layer which splits it up further. Since the window size between the layers vary while the number of patches in each window is constant, the patches get merged to fit inside the windows. In this architecture the layers with smaller window sizes are used to capture smaller features while the layers with larger window sizes are used to capture more global features. In the Swin Transformer architecture, attention is a key component and is computed inside each window. To propagate information between windows they are shifted so that they overlap. This is so that pixels that are part of the same object, but not inside the same window share information [34].

The use of windows in the image representation is the biggest difference between the Swin Transformer architecture and the ViT architecture is only split up into patches and not into windows of varying size.

[1] vissl.readthedocs.io

2.3.2 UPerNet

UPerNet is a decoder framework that stands for unified perceptual parsing network. The goal of UPerNet was to create a decoder that could work on several different machine vision tasks at once, such as classification, material detection and part detection to name a few. UPerNet utilizes several different levels of feature maps from the backbone to capture several layers of semantic information which is then for most tasks fused. An illustration of this is shown in Figure 2.6 [48].

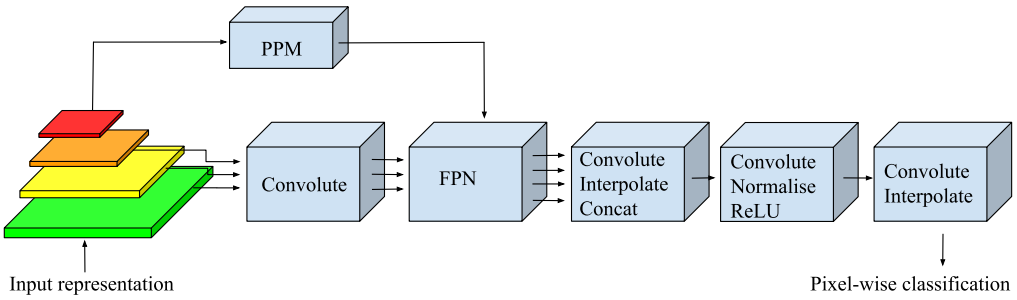

Figure 2.6: UPerNet architecture. Illustration inspired by figures in the paper introducing UPerNet [48]

The last embedding layer of the backbone is first passed through a pyramid pooling module. The PPM splits the embedding with a pooling layer, into four different scales of representation where it goes from a more global representation to more detailed representations. Pooling samples part of the input to create more abstract representations of the input. These representations are then passed through a convolutional layer, normalised, passed through a ReLU activation function and then interpolated to be the original size before being concatenated together. The concatenated result is then sent passed through a convolutional layer, normalised and passed through a ReLU activation function before being returned [52]. A representation of the PPM can be seen in Figure 2.8.

The result of the PPM is then used together with the output of the other embedding layers from the backbone in a feature pyramid network (FPN). Firstly, all the intermediate results from the embedding layers except the last one are passed through a convolutional layer. After this, the result of the PPM is interpolated to have the same dimensions as the second to smallest embedding layer. The process of interpolating and adding together layers is done iteratively until every embedding layer contains an interpolated version of all the previous embedding layers [31]. A representation of this process with two layers can be seen in Figure 2.7. These new embedding layers are now each passed through a convolutional layer, interpolated to be the same size as the largest embedding, and fused together by being concatenated together. This new result is then passed through a convolutional layer, normalised and passed through a ReLU activation function before being passed through a final convolutional layer and interpolated to fit the target size [48].

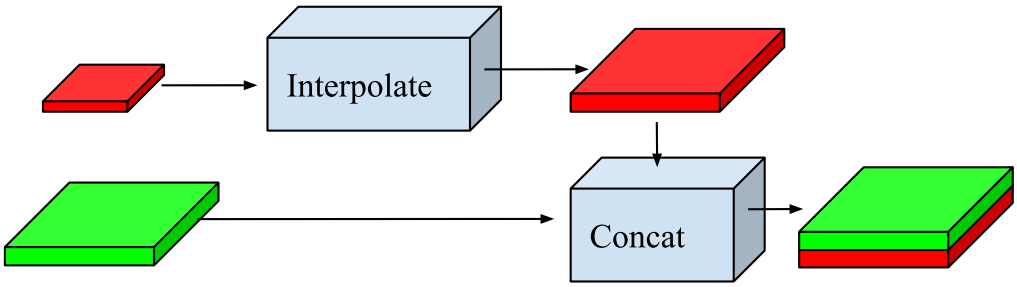

Figure 2.7: FPN interpolation and concatenation step for two embeddings. To concatenate two image embeddings of different scale, the smaller is scaled up.

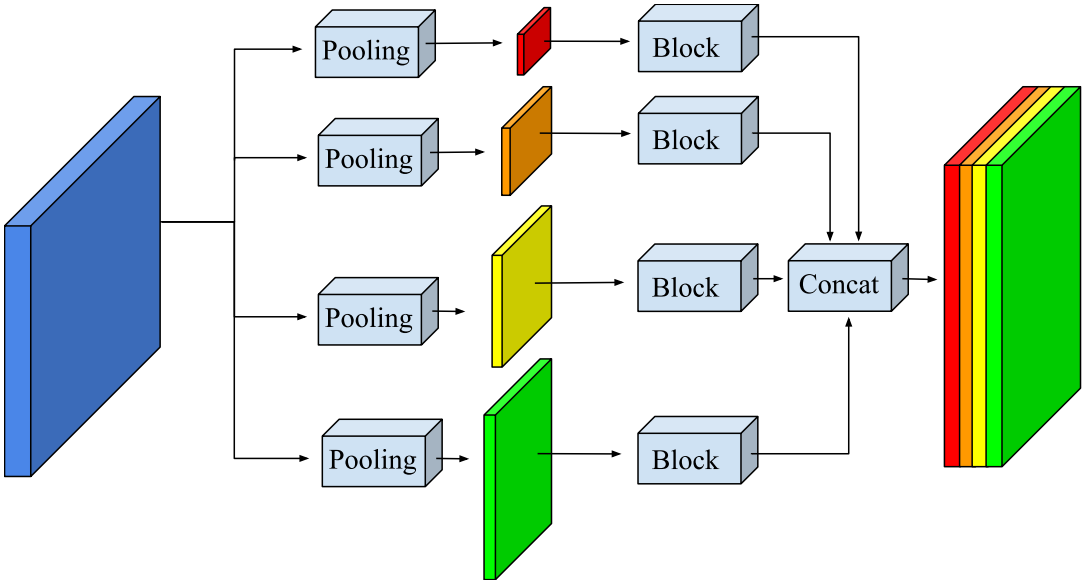

Figure 2.8: PPM architecture. A feature representation gets split into four embeddings of different sizes by a pooling layer to represent different levels of abstract features. These embeddings are sent through blocks consisting of a convolutional layers, after which they are interpolated to the desired output size. Lastly, they are normalised, and passed through a ReLU activation function before being concatenated. Illustration inspired by figures in the paper introducing the PPM architecture [52].

2.4 Semantic segmentation

Semantic segmentation is the task of recognising parts of an image as objects and labelling them with a semantic class. This is done by training a machine learning model which in turn gives a discrete label for every individual pixel of an image input from a fixed set of labels. Semantic segmentation does not take into account if objects are separate, for example, if an image contains multiple cars, pixels of these two separate objects are labelled by the same category despite being separate objects, as they are the same category of object [12].

2.5 Intersection-over-union

Intersection-over-union (IoU) is a metric for comparing how well two segmentations overlap and is one of the more common metrics for semantic segmentation. IoU is calculated by taking the area of the overlap between the predicted segmentation and the ground truth divided

by the area of the union of the two pictures, the equation for IoU can be seen in Equation 2.6. As such the IoU is a value between 0 and 1, with 0 meaning there is no overlap and 1 meaning there is a perfect overlap. The mean intersection-over-union (mIoU) is defined as the IoU of each class divided by the number of classes. This gives a measure of how correct the predicted segmentations are [39].

$$IoU = \frac{Area\,of\,overlap}{Area\,of\,union} \tag{2.6}$$

2.6 Data augmentation

Data augmentation is the act of transforming data in some specific way. In the context of machine vision, it is typically image transformations, such as cropping part of the image, changing the colour channels, rotating the image, and adding noise. What exact augmentations are appropriate to apply depends on the dataset and the task.

Data augmentations are typically used in SSL to create positive examples. In other words, by taking a source image and applying some transformation to it, a semantically similar image is obtained. Two images can be thought of as semantically similar if they contain the same high-level meaning to a human. For example, if an image of a dog is cropped and two augmented pictures are produced, then one might contain the fur from the leg of the dog and the other contains the shape of its ears. In this case, a human would be able to tell that the images contain similar semantics even though they do not overlap. However, in some cases, the semantics can be destroyed by augmentation. For example, if the images in the dataset contain very fine detail then a blurring augmentation might remove the semantic information that separates images from each other in the dataset.

What information is important depends on what problem is being solved and the method used for solving it. For example, if the task is classifying pictures, then a rotation would not change what label a given image should be assigned, i.e. an image of a dog rotated 90 degrees should not be assigned a different label just because it is rotated. In contrast, if the task is semantic segmentation then a rotation would destroy the information about where in the image each semantic object is. In other words, if an image is rotated then the ground truth for what label each pixel should receive is no longer correct as the input pixels have moved according to the rotation. This is important to consider when pretraining for semantic segmentation as the representation should be closely related to what semantic segmentation will be mapped to it.

3 Related work

To give a sense of the state of the field a summary of related work is delivered here. Specifically, three studies investigating classification of scrap metal using different approaches were interesting to compare to our study. As well as a summary of how SSL has been used on adjacent tasks within machine vision.

N. Smirnov and A. Trifonov [41] discusses classification of scrap metal from train cars, they used a consumer-grade digital colour camera and a convolutional neural network (CNN). A convolutional neural network is a widely used type of architecture within machine vision as it leverages the locality bias present in images, i.e. pixels close together have more in common. The method used in this paper is similar to the method used in this project, however, the specific task is slightly different as it sorts entire pictures and it does it by type of scrap metal rather than sorting by the materials [41]. There have also been studies [27] exploring the possibility of using inductive sensors to classify scrap metal in conjunction with a colour camera. The colour camera data is not used with any machine learning technique, just a separation of the colour channels [27]. In a precursor [2] to this project, a study was conducted with a similar setup to the one in this project. However, this precursor explores different methods and focuses on separating metals that are ferrous from non-ferrous, while this project will mainly focus on separating aluminium from other metals as this was a flaw earlier [2].

3.1 Machine vision applied to sorting scrap metal

Machine vision has been applied to the task of sorting or classifying scrap metal in several studies. Both traditional machine vision techniques and techniques based on artificial intelligence have been investigated. The possibility of using a colour camera in conjunction with an inductive sensor array to sort scrap metals by the material was investigated in 2005 by M. Kutila, J. Viitanen and A. Vattulainen [27]. This paper explored the use of traditional techniques of machine vision, using a colour analysis based on the three colour channels of an RGB image. With this technique they could separate metals by colour, for example, they separated reddish metals like brass and copper from silver-grey metals like zinc and aluminium. This technique could separate metals with dissimilar colours well but struggled with metals of similar colours. For example, aluminium from stainless steel had low accuracy. In addition to the colour camera, an inductive sensor array was used to measure the electrical character-

istics of the metals to distinguish them. However, these measurements were not consistent as the shape and occlusion of objects made them unreliable and not as useful as the visual information [27].

Deep learning has also been applied to classifying scrap metals. In 2021, N. Smirnov and A. Trifonov [41] researched the possibility of automatically classifying the content of incoming train cars to a recycling plant using CNNs [41]. However, the task was not classifying images of metal at a conveyor belt pixel-by-pixel, but rather classifying train cars filled with different kinds of scrap metal, divided by shape, size and other characteristics. Thus, it is a pure classification task in contrast to semantic segmentation. The input data was gathered from consumer-grade digital colour cameras with a resolution of 214 by 214 pixels taken at an angle to the train car. This data was used to train four different models based on CNN and pre-trained on ImageNet [14] a dataset commonly used for pre-training neural networks used on general machine vision tasks and not specifically scrap metal. The models tested were ResNet152V2, InceptionResNetV2, DenseNet201 and NASNetLarge which produced results that shows promise [41].

Machine learning has also been investigated to sort scrap metal by modelling it as a semantic segmentation task. In 2020 F. Almin [2] carried out a thesis work for Combitech which was a precursor to this thesis. In that thesis project, the possibility of training a CNN to detect ferrous and non-ferrous materials was explored. The data was sampled using a Dalsa Genia Nano camera which is an RGB camera and is similar to the ones used in this thesis project. The dataset was manually annotated by segmenting images of just one category, either ferrous or non-ferrous and marking it against the background. The images containing both ferrous and non-ferrous materials were created by stitching together images containing only ferrous and only non-ferrous materials [2].

F. Almin's results give a baseline to compare the results of this thesis against when doing scrap metal sorting using computer vision as the setups are very similar although the task of separating copper from iron is likely easier than separating aluminium from mostly grey metals as indicated by the 2005 study using the colour camera [27]. A 2021 study comparing transformers to CNNs on image classification tasks suggests that transformers can perform excellently when compared to CNNs with less training time. This motivates the use of transformers to explore a potential improvement to using CNNs [53].

3.2 SSL for machine vision

The use of SSL for pre-training models has in recent years become more prominent, in particular in the subfields of natural language processing and machine vision. The advancements have been rapid and the state of the art is constantly improving. Using SSL is a tool to use unlabeled data to avoid the need for large annotated datasets [33]. Over the years as new and improved SSL techniques have been presented models that have been pre-trained using SSL have shown to be on par with models trained using supervised learning and even sometimes exhibit properties that the supervised trained model does not. One example of this is in the paper presenting the DINO SSL framework [9] where the models pre-trained using SSL were better at finding scene layouts and object boundaries on images. This was shown by producing an attention map which highlights what parts of an image the model find is valuable semantic information [9].

The previous standard for doing SSL for machine vision was to pre-train models on pre-text tasks which is a task that the model is supposed to solve during training. One of the most common pre-text tasks is the jigsaw puzzle task where an image is split up into several rectangles which are all then randomly placed. The goal of the task is to move the pieces such that the original picture is restored [24]. The highest performing SSL framework using the jigsaw tasks as the pre-text task is PIRL from 2019 [35]. One of the problems of using these is finding the best pre-text task for the end task of the model. Some pre-text tasks performed better

for classification and some worked better for semantic segmentation. Later approaches have used SSL to compare representation between images instead of pre-text tasks. Here the goal is that images with similar semantic information get similar representations, while images that do not share semantic information get dissimilar representations [24]. Some of the most recent frameworks using these techniques are SwAV from 2020 [8], DINO [9], SimCLR [10] and Barlow Twins [50] from 2021. These have shown a significant increase in performance in comparison to the previous pre-text SSL frameworks and have even sometimes outperformed supervised models on the same task after finetuning [10].

4 Method

This chapter presents the methodology used for collecting images, using the frameworks, training the models and evaluating the models, as well as conducting the interview and constructing a context for the thesis work. As the training is divided into two separate phases, the pre-training and the downstream training the sections have been structured to reflect that.

As machine learning models have several hyperparameters that control the behaviour, it is important to take them into account. Whenever possible and reasonable the established default techniques and values have been used, these are typically given by the original papers or by a widely used implementation such as a library. In some cases, however, there is a significant difference in either application or end task that the defaults cannot be argued to be appropriate. In these cases, a small parameter search has been conducted to guarantee that the results reflect the performance that can be reasonably expected.

4.1 Architecture selection

In this section, the method for choosing the architecture for the models and the necessary parameters and techniques. For clarity, the term *architecture* is used to refer to a design, while the term *model* is used to refer to an instance of an architecture. The overarching configuration consists of two parts, an encoder and a decoder. This is similar to how recent state of the art has structured their overall architecture [34]. An advantage of this is that the encoder part can be pre-trained, in contrast to an end-to-end system in which it would not easily be possible.

4.1.1 Encoder

Two architectures were investigated as potential implementations of the encoder. These were the ViT architecture and the Swin Transformer architecture. The motivation behind investigating these architectures was as follows: ViT was the earliest implementation of the transformer technique in a machine vision context and was therefore interesting as a representation of transformer architectures without any later enhancements [16].

The Swin Transformer architecture was interesting to investigate because its design seemed to lend itself particularly well to the dataset and task. Since the Swin Transformer architecture processes the image on several levels of detail it seemed advantageous to use on scrap metal as the image contains information about the shape of the object on a high level,

but also more fine-grained details such as the texture on the object. It also had state-of-the-art performance on several datasets within the task of semantic segmentation [34]. These reasons lead to the Swin Transformer being selected as the architecture for the encoder.

Size of model

As the number of trainable weights in a model increases behaviour changes. It is therefore important to have a roughly consistent number of weights so that a comparison between models fairly compare the architecture and not simply the sizes of the models.

A model that is too large will require excessive amounts of training time and computational resources, while a model that is too small might not perform well enough to be representative of the architecture. ResNet-50 is a common convolutional model that is used as a standard architecture to evaluate SSL frameworks in several papers [50] [8] [19] [9]. The number 50 in ResNet-50 refers to the number of layers in the model and is similar in size to the Swin-T model [34]. To be consistent with the de facto standard, Swin-T was selected as the encoder model so that the size would be appropriate.

Augmentations

During the training of the models in this project, data augmentation has been used to increase the amount of training data and to make the models more robust and better at handling deviation in the data. Choosing the augmentations is not obvious and heavily depends on the end task of the model. Some of the most common data augmentations used during SSL are augmentations that flip the image data along either the vertical or horizontal axis [8] [50] [9]. This augmentation makes sense when the end task is classification since semantically the image is the same class regardless of whether it is shown upside down or not. However, for semantic segmentation, the positional information is much more important since the goal is not to classify the entire picture but instead classify find the pixels that are a certain class. In light of this, no augmentations that alter the position of objects in the data have been used during pre-training. However, for downstream training the encoder and decoder no comparison between representation is made and thus image rotation and horizontal flip were applied during the downstream training to increase the amount of data.

Other common data augmentations that do not rotate or flip the images used for training the models in this project were solarize, gaussian blur, and colour distortion. Solarize inverts all pixel values above a certain threshold, gaussian blur blurs the images and colour distortion randomly distorts the hue, saturation, and brightness or converts the image to a grayscale image. These augmentations were made both during the pre-training and the downstream training. The goal of these augmentations is to make the model less sensitive to the lighting conditions and glare that might vary over the images. Table 4.1 presents the augmentations and their parameter values during pre-training for the three frameworks.

Framework	MultiCrop (number of crops)	Colour distortion (strength)	Gaussian blur (probability, min radius, max radius)	Solarize (probability)
SwAV	2	1.0	0.5, 0.1, 2.0	-
DINO	2	0.5	1.0, 0.1, 2.0	(0, 0.2)
Barlow twins	2	0.5	1.0, 0.1, 2.0	(1.0, 0.1)

Table 4.1: Describes the augmentations and augmentation parameters used during pre-training for SwAV, DINO and Barlow twins.

4.1.2 Decoder

For determining what decoders to use for the models' several decoder architectures were investigated to determine which was most fitting. Different methods for extracting the representation vector from the encoder were also investigated to find the most suitable combination, for example combining the output of several layers into one cohesive representation. All comparisons were made by training with the downstream training dataset and evaluating on the downstream validation dataset.

Three decoders were tested which were; C1, Setr and UPerNet. These decoders vary in both structure and size with the C1 decoder being the smallest with 1,328,067 parameters, UPerNet having 1,882,851 parameters and Setr having 3,538,371 parameters in the implementation used in this thesis work. C1 was investigated since it was the decoder used at Combitech in previous work [2] and therefore would be a good baseline to compare the other decoders against. Setr and UPerNet were investigated because they were both compared in the paper introducing the Swin Transformer [34]. Setr was the state of the art architecture previous to the Swin Transformer article. As it achieved the highest mIoU on the ADE20K dataset, which is one of the canonical dataset new architectures are compared on. However, with the publication of the Swin Transformer paper UPerNet became the new state of the art as it outperformed Setr [34].

Implementation of C1 was taken from Combitech which had used C1 in previous projects. Setr was implemented by following the explanation of the architecture in the original paper for Setr [53]. Lastly, the UperNet decoder was implemented by following the explanation of the UperNet and PPM architecture from the original papers [48] [52]. Each decoder was trained for 50 epochs on the downstream task with an otherwise identical model without any pre-training. The downstream training was performed using an identical setup except for the decoders used and the resulting IoU and mIoU on the validation data were used to compare the three decoders. In these results, the UPerNet decoder performed significantly better than the C1 and Setr decoders and was thus chosen as the decoder to use.

The parameters used for the final UPerNet decoder that was used were mostly the same used in the original paper [52]. The parameters that could not be left as default from the paper were embedding size and the hidden dimension. This was necessary for the dimensions of the representation from the encoder to match up with the internal structure of the decoder. Thus, the embedding size was set to 768 as it was inherited from the decoder which in turn sets the hidden internal dimension to 96 as it is halved three times.

4.2 Data collection

In this section, the method for the data collection is described. There is one primary distinction between the data used. That is whether it is used to pre-train with using some SSL framework, or if it is used for the downstream training. The data used for pre-training was collected without a ground truth annotation from cameras mounted above the conveyor belt at Stena Recycling's plant in Halmstad. The data used for downstream training was created by taking material of each category from the recycling plants and photographing these separately to later stitch together. The method for collection is further explained in Section 4.2.1.

4.2.1 Pre-training dataset

The dataset used for pre-training the models was collected at Stena Recycling's plant in Halmstad between November 2021 and February 2022. This dataset was created by the external supervisors of this thesis at Combitech. The images in the dataset were collected using a Basler acA2440-20gc GigE camera[1] that was mounted above the conveyor belt that the scrap metal travels on after it has been sorted of its aluminium content. The camera took images

[1] https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca2440-20gc/

with an interval of initially 5 seconds and later on every 2 seconds. Pictures were taken most weekdays except for Fridays between November 2021 and February 2022 and the images were sorted into folders for each separate day. In total 318000 images were collected and used to pre-train with.

The conveyor belt was not always active and did not always contain any material while the camera was running, thus often capturing images without any refuse. To remove these images a similarity metric was used to find when the conveyor belt had stopped and there was no change in the images being taken. This dataset was split up into a training and validation set where the validation set consisted of all the images taken during the 2nd of February 2022, these images were later used to do inference on. The training dataset was all the other images captured.

There was also some preprocessing performed on the dataset in the form of cropping. The Swin Transformer model that was used required the images to be of a specific dimension. In the original paper, the two dimensions used were 224x224 pixels and 384x384 pixels. The images that were provided by Combitech had already been downscaled to 700x700 pixels and to maximize the data, each 700x700 pixel image was cropped into two 384x384 pixel images. Most of the refuse in the images were located in the middle part of the conveyor belt with respect to the y-axis. Because of this the top and bottom 158 pixels were decided to be cropped out so that only the middle part of the images in the y-axis was used. After this, the images were cropped in the x-axis so that the first image contains the first 384 pixels and the second image contains the last 384 pixels of the image. An image from the pre-training dataset is shown in Figure 4.1. As each original image was thus cropped into two parts, the total number of images was increased to 636000. Since $384 * 2 = 768$ there is some overlap between the images. However, since the two crops are either in the training set or in the validation set, but not in both, thus there is no contamination between the training and validation set. Having some of the information duplicated between images should not affect the performance of the model as this is analogous to having the same data appear in a later epoch which is the usual case.

Figure 4.1: Image from the pre-training dataset

4.2.2 Downstream dataset

The dataset used for fine-tuning and evaluating the decoder and encoder on the downstream task was collected by Combitech prior to the master thesis. As separating aluminium from other scrap metals is not trivial without expert knowledge, certain precautions had to be

taken to ensure objects were matched correctly. The material was delivered by Stena Recycling, separated into labelled buckets, so that all aluminium was together and vice versa.

The images were captured using a Dalsa Genie nano c2450 camera[2] of the same model as the one used at Stena Recycling's plant and were 616x514 pixels. The material was put onto a black sheet a few objects at a time, all from the same category. Using the black sheet as a background the images were segmented into bright pixels and dark pixels automatically, corresponding to the background and the material respectively. This is the source for the map used as ground truth for what part of an image belongs to a given class.

These images were split up into a training, validation and test dataset with the ratio of roughly 70/10/20 per cent respectively for images containing aluminium as well as images containing other material. The datasets contained 197, 27 and 60 images of other material for the training, validation and test dataset respectively and 217, 33 and 70 images of aluminium for the three datasets respectively. As it was deemed important that each image contained both aluminium and other material, the number of stitched images was constrained to the lower bound in each case. Thus, producing a training dataset with 197 images, a validation dataset with 27 images and a test dataset with 60 images. The images containing aluminium contained three different types of aluminium objects; big blocks, crumpled material and crushed cans. The images containing other material could be classified into the three different categories of; larger objects, medium objects and small objects. The split of the images into the datasets was done so that every dataset contained material from all the categories from both aluminium and other material. The training dataset was used to train the models while the validation dataset was used to evaluate different decoders, learning schedulers and other implementation details. Lastly, the test dataset was used for the final evaluation of the models.

To create a dataset that was more representative of the real-world situation the problem was aimed at, two things were needed. Firstly, the images were stitched together using the automatically created segmentation masks, so that each image could contain both aluminium and other materials, rather than just one or the other. Secondly, the stitched-together material was placed onto background images of the conveyor belt when it was running empty at the recycling plant. This gave images that look more similar to the images taken directly above the conveyor belt at Stena Recycling. However, since the images containing material and the images of the conveyor belt were taken in different settings the lighting does not match up perfectly and makes it look artificial. The automatic masking of the material also resulted in some shadows being interpreted as material that made it into the stitched images which it should not have.

After creating the stitched images they are however still of size 616x514 while they need to be of size 384x384 pixels as mentioned in Section 4.2.1. This was achieved by doing five crops out of each image with four of the images being cropped in the four corners of the image and the last being cropped out of the middle of the image. The reason for doing this crop is that the downstream dataset is very limited and no parts of the images should go to waste which does not happen when doing these crops. This resulted in overlap between the crops. An example of an image used for downstream training is shown in Figures 4.2 and 4.3 where Figure 4.2 is the stitched image that is fed into the model and Figure 4.3 is the mask that the output of the model is compared against. The black pixels in Figure 4.3 is the background, the grey pixels are other material and the white pixels are aluminium.

To determine the possibility of creating a downstream dataset where the lighting was more similar to the pre-training dataset, a dataset where the images were pre-processed was created. In this dataset, the background images' gamma values had been increased by 60% and the saturation of the images of materials had been decreased by 25%. An example of the images in this dataset is shown in Figure 4.4. Increasing the gamma values of the images containing background decreases the exposure of the background and lowering the saturation

[2]https://www.edmundoptics.eu/p/c2450-23-color-teledyne-dalsa-genie-nano-5gige-poe-camera/43714/

on the images containing material limits how much it stands out from the background. The values were determined by testing different combinations of gamma and saturation values and selecting the values which made the images look most similar to the pre-training image shown in Figure 4.1.

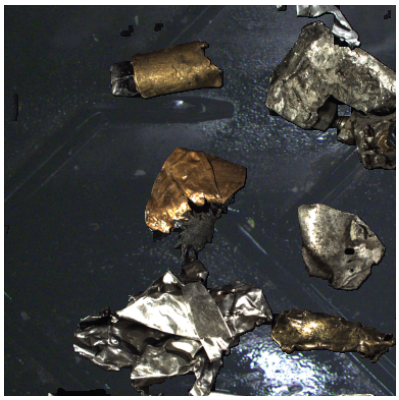

Figure 4.2: A sample from the downstream dataset

Figure 4.3: Ground truth mask of Figure 4.2 and 4.4

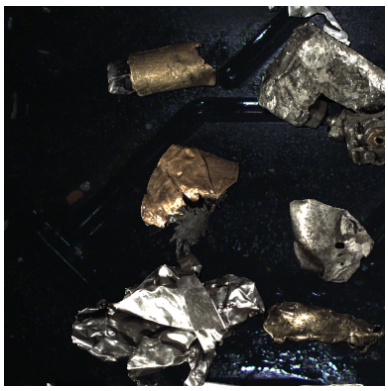

Figure 4.4: Image from the downstream dataset where the gamma values on the background have been increased and the saturation on the material have been decreased

4.3 Frameworks

In the literature study, several SSL frameworks used within the machine vision field were of interest in the context of the thesis, but a selection had to be made to keep the scope limited. The main differences between different frameworks related to the task at hand appeared to be the techniques used to avoid collapse, which in turn had a large effect on the required minimum batch sizes and thus the memory needed during run time. The second major factor considered when selecting the frameworks was the performance on benchmark tasks, as the performance of the framework will be of interest.

The three frameworks chosen to investigate were DINO, SwAV and Barlow twins. This was because these are some of the newest SSL frameworks and they have proven to perform well on benchmark tests. These three frameworks solve the problem of collapse in three different and interesting ways which made them interesting to compare. DINO solves it with a momentum encoder, SwAV with clustering and Barlow twins solve it with its unique loss function. Another factor for choosing these frameworks is that they do not use negative examples which reduces memory usage [9, 50, 8].

The hyperparameters of these frameworks were left as default whenever possible. However, the parameters that were changed were related to the augmentations, which have been covered in Section 4.1.1, and the dimensions of the projection heads, which are only used during the pre-training to produce the representation vectors. This was because the default values were set for the ResNet-50 architecture. The projection head dimensions was; [768], [768, 2048, 128], [768, 4096, 4096, 8192, 8192, 8192] for DINO, SwAV and Barlow Twins respectively.

4.4 Hardware

Here the hardware used to perform the training is briefly described, for replicability purposes. A supercomputer cluster called Berzelius was made available for this thesis work. It is run by the national supercomputer centre in Sweden (NSC). NSC offers the possibility to process large computational problems. Berzelius in particular is designed to handle the particular characteristics of machine learning and has been made available for use in this project [47].

The computational nodes of BerzeLiUs contain 8 NVIDIA®A100 Tensor Core GPUs with 40 GB on-board HBM2 VRAM, 2 AMD Epyc™7742 CPUs, which have 64 cores and 128 threads [47]. This enables large models to be trained as a smaller GPU might not be able to fit the model and data in its VRAM.

4.5 Training of models

The training of the models was structured into two parts. First is the pre-training part, within which the encoder part of the model was trained using one of the selected SSL frameworks, DINO, SwAV or Barlow Twins. The objective of this training task is to find a good representation for each image according to the SSL framework being used. This part is the larger of the two, by a factor of about 20:1, as the dataset used in the pre-training is several magnitudes greater than the annotated dataset used for training on the downstream task. The second part of the training is the training on the downstream task. During this training, the entire model is trained to semantically segment images of scrap metal.

To illustrate the differences and similarities between the two parts of the training, a diagram for the pre-training is shown in Figure 4.5 and for the downstream training in Figure 4.6.

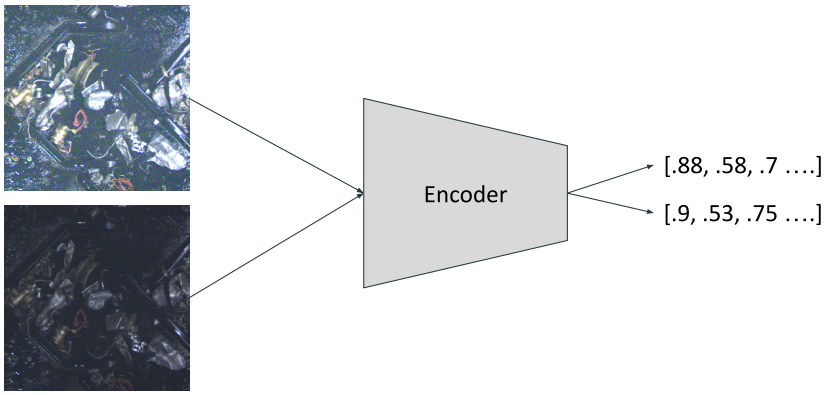

Figure 4.5: A high-level overview of the pre-training process, with two augmented versions of the same image which the encoder uses to produce representation vectors from.

In the pre-training part, there is no ground truth for semantic segmentation and thus the network is performing the task of representation learning instead. This means the task is to

produce a vector representing the input image and thus encoding the information about it in a latent space.

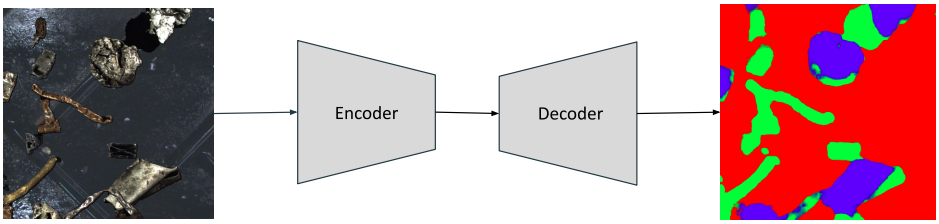

Figure 4.6: A high level overview of the downstream training process, an image is used as input and a prediction is returned.

In the downstream part, the task is semantic segmentation as there now is a ground truth for that available. This necessitates adding a decoding part to the model's architecture, which decodes the representation vector and extracts the information encoded in it. It uses this to produce a semantic segmentation reflecting where in the image different metals are located.

4.5.1 Pre-training

The pre-training of the models was performed on BerzeLiUs, see Section 4.4 for more details. Each model used the same allocation of 4 GPUs and 6 CPUs for training. The CPUs are used for loading the batch data, by having a low number of CPUs in comparison to the number of GPUs the time to train increases and the model gets limited by how fast the data can be delivered which wastes the GPU allocation. To find a suitable ratio between CPUs and GPUs, several ratios were tested. Three CPUs for every two GPUs seemed to give the best result while not taking up too much of the allocated resources at BerzeLiUs.

Before conducting the final pre-training of the models some training had been done for around one epoch for each of the frameworks with a Swin Transformer network. This was to ensure that most of the default values were suitable and gave good enough performance in comparison to the previous results produced by the models from Combitech. Parameters that had to be changed were the number of epochs and the batch size. Since there had been some success with pre-training for one epoch, it was decided that pre-training for 20 epochs was likely to be sufficient to produce good enough results to draw conclusions.

The batch size was chosen to be 64 for each of the frameworks split up into 16 for each GPU. Batch sizes are customarily chosen to be a power of two and 64 was the largest power of two that could be chosen with the hardware used [37]. The original papers for the frameworks trained used a batch size larger than 64, however, the learning rate is scaled based on the batch size and should compensate for the smaller batch size, thus 64 was found through testing.

The state of each pre-trained model was saved into a checkpoint every epoch. These checkpoints save the values of the learned weights of the model at that time. A selection of these checkpoints were downstream trained.

The optimisers during pre-training were left as the default values from VISSL's configuration files for the frameworks. For training, three optimisers were used, AdamW was used for DiNO, SGD for SwAV and LARS was used for Barlow twins, which are the optimisers used in the papers [50, 9, 8]. The learning rates and other hyperparameters were left as default or scaled to the dataset used.

A common way to analyse the behaviour of machine vision models based on Transformer architecture is to create a map of what part of the input a model pays attention to [9] [16]. This is achieved by summing the attention paid to each datapoint from every other datapoint within the window of the transformer. By applying a threshold based on a cumulative sum the top attended data points are selected for each head. Each head attends to slightly different features and each gets a colour to show what each individual head attended to the most. As the Swin Transformer is hierarchical, each layer has a number of heads. The first layer was selected as it has the most high-level features and thus produced the most interesting attention maps. The results of these attention maps are presented and analysed to assess how the models behave.

4.5.2 Downstream training

The code for downstream training of the models was implemented in Python using the PyTorch library. For retrieving that training data from the downstream dataset to train the models, a data loader was created. The data loader loads the images from the dataset, converts them to tensors, normalise the values and then randomly applies an augmentation. These augmentations consist of applying rotation, flip, colour jitter, gaussian blur or solarization. This is a technique used to increase the amount of training data, by adding distorted versions of the original.

As described in Section 4.5.1, each model was saved into a checkpoint which stores the learned weights and the current state of the optimiser at regular intervals.

These checkpoints were then later used to initialise the weights for the downstream training. The reason for saving several checkpoints from the pre-training is so that the optimal number of pre-training epochs could be selected. One potential risk with pre-training for too long is overfitting, thus motivating saving the state of the pre-trained models at regular intervals.

In contrast to when implementing and choosing hyperparameters, there are no defaults available for the downstream training. To find an acceptable configuration a small-scale search was performed to find a reasonable optimiser and learning rate scheduler with suitable hyperparameters. The search was conducted similarly to how the decoder architecture was conducted. A limited number of options is selected to investigate based on what is commonly used within the field. The options are tested by training on the downstream task for 50 epochs and the option that produces the best mIoU on the validation dataset is selected and used in the later trials.

First, two learning rate schedulers were selected, a step learning rate, which follows an exponential curve and a learning rate scheduler which follows a cosine curve and works by reinitialising some weights at regular intervals to mitigate stagnating and getting trapped in local minima. This reinitialising process is typically called annealing. After training for 50 epochs the cosine annealing learning rate scheduler performed marginally better and thus it was chosen. Out of the hyperparameters tested having 10 epochs between restarts and a maximum learning rate of 0.001 produced the best results on the validation dataset.

With the learning rate scheduler selected the optimiser was selected by comparing SGD and AdamW. These are the optimisers used in the pre-training that are well established and implemented in PyTorch. The trial resulted in AdamW slightly outperforming SGD with the default hyperparameters. The final configuration used for the downstream training is summarised in Table 4.2.

Component	Description	Value
Decoder architecture	The decoder architecture is the part of the model that takes a representation of the image and outputs a segmentation	UPerNet
Optimiser	Function for updating the weights of a model, see Section 2.1.4 for more information	AdamW
Hidden dimension	The size of the representations feature vector	96
Learning rate scheduler	Learning rate scheduler which determine how the learning rate which is used by the optimizer should change after each epoch, see Section 2.1.5 for more information	CosineAnnealing-WarmRestarts
Epoch restarts	The number of epochs before the learning rate gets reset to the base learning rate by the learning rate scheduler	10
Base learning rate	The initial value for the learning rate scaled to a batchsize of 8, the scaling is linear	0.0001
Maximum learning rate	The maximum value that the learning rate can get by the learning rate scheduler	0.001

Table 4.2: The final configuration arrived at by the parameter and architecture search. It is used for all downstream training.

4.6 Investigating practical feasibility and potential impacts

In order to measure the feasibility and future value a future implementation could have, it is necessary to establish the context it will be implemented in and what led to it. To answer this, an interview with Stena Recycling were held to establish the current systems in place and how a solution could create value. In addition to this quantitative measures was established during the interview so that it can be determined if the desired value has been met.

4.6.1 Interview with Stena Recycling

In order to tailor the solution to the problem and to answer the research question regarding how the system can be used to create value, an interview and continuous dialogue with Stena Recycling were performed. The interview was conducted remotely by the master thesis students. One student was primarily asking the questions and the other was keeping notes and documenting the answers. There was no coding phase, but the answers were summarised by the master thesis students and approved by the interviewee. The interviewee was employed at Stena Recycling as a product owner and had been involved with the earlier collaboration between Stena Recycling and Combitech. The interview was conducted in Swedish, with the summaries being documented in Swedish during the interview and later translated into English along with the questions.

The interview began with a demonstration of the proposed solution and an explanation of the aim of the interview. Firstly, questions related to the process at Stena Recycling, how their measurements had been carried out and the material on the belt were asked. Secondly, questions regarding proposed uses of the system and what impacts a given result for Stena Recycling would have. All of the questions asked were open-ended. The interview was semi-structured, meaning that most of the questions were prepared in advance, but not all. Questions regarding the specific impact of proposed systems were not prepared as they were dependent on what systems the interviewee deemed viable. The aim of the interview was to get factual answers regarding the current operation at the recycling plant, and the hopes of what impact proposed systems could have. Additionally, the interview assisted in understanding what was of interest to Stena Recycling. This in turn would help guide the work of

the thesis, as well as help bridge the gap between metrics from the machine vision field and real-world use cases.

To see if the findings of this report could be of use to Stena Recycling, questions regarding possible use cases were asked. To bridge the gap between the findings of this thesis and what value it can bring to Stena Recycling, metrics taken from experiments need to be converted to more tangible results. What metrics are relevant depends on what use case a given solution is applied to. In light of this, two hypothetical use cases were briefly outlined so that the interviewee could assess the viability and desirability of each from Stena Recycling's point of view. These hypothetical use cases were: an information system estimating the aluminium ratio to inform manual responses and an automatic system which removes material detected to be aluminium.

1. What is the current process for measuring the amount of aluminium after sorting?
2. What costs are associated with the measuring process?
3. How does the amount of aluminium vary?
4. What is the ambition of Stena Recycling in regards to sorting the aluminium?
5. Is automatic estimation a desirable feature?
6. What margin of error is required for automatic estimation to be a viable use case?
7. Is an information system of aluminium ratio to inform manual responses viable to use in the sorting process?
8. If so, what kind of manual responses could be used?
9. What number of manual responses is acceptable to lower the aluminium percentage?
10. Is an automatic removal system viable to use in the sorting process?
11. What would be the effect of reducing the percentage of aluminium from 4% to 2.5%?
12. In the context of creating a training dataset for neural networks, what are the possibilities of annotating images from the conveyor belt, i.e. can an experienced sorter tell apart materials with high accuracy just from the photos?

4.7 Evaluation

To answer the research questions it was imperative that the resulting models were compared and evaluated in a way that enables fair and relevant comparisons of the performance and behaviour. As this allowed the effects of different SSL frameworks and limited training data to be investigated, thus answering RQ 2, RQ 3 and RQ 4. Similarly, to address the viability portion of RQ 1, the models were evaluated with the ratio estimation metric quantitatively as well as qualitatively on real-world data.

Thus, the evaluation was carried out in two paradigms. The primary evaluation was the quantitative evaluation, which measured the performance of each model in the canonical metrics within the task of semantic segmentation, class-wise IoU and mIoU. As annotated ground truth is only available in the downstream dataset, this evaluation primarily evaluates how well the models perform at the specific task of segmenting images consistent with the downstream dataset. As machine learning metrics are not directly transferable to performance in a practical system, a way to evaluate viability was constructed after discussions with Stena Recycling representatives. This statistical evaluation was performed by extracting the information that would be used in practice from the models, which in this case is an estimation of the ratio of aluminium to other material.

The secondary evaluation was a qualitative analysis of the segmentations on images both from the downstream dataset and from the pre-training dataset. In both cases, only images not used for training were used as that would pollute the results. The images were chosen with different characteristics so that large parts of the dataset would be represented. This evaluation could compare in what way the models differ in their predictions, apart from the raw performance. In addition, this evaluation allows a way to test how well models generalise to the real-world data even though in a limited capacity as no ground truth was available for proper comparison. In addition, attention maps produced by the models were analysed. This gives information on how the behaviour of the models is prior to downstream training.

4.7.1 Quantitative evaluation

To evaluate the models a collection of metrics were calculated by comparing the predictions on the test dataset with the ground truth annotations. The metrics IoU, mIoU and error of aluminium ratio estimation, were calculated based on the prediction of each model evaluated on the test data.

To evaluate the three different SSL frameworks a comparison was made on the resulting models from pre-training the architecture with each SSL framework. In addition to pre-training three models, a model with identical architecture was also trained exclusively on the downstream task to measure the effect of pre-training with SSL.

Pre-trained models of the epoch that performed the best on the downstream dataset, together with a model without pre-training were trained and evaluated on the processed downstream dataset. This was done to investigate the possibility of creating a downstream dataset that is more similar to the pre-training dataset.

To answer questions regarding viability, hypothetical use cases were constructed by the thesis students to be judged by Stena Recycling through an interview, as discussed in Section 4.6.1. To assess the real-world effectiveness of any given model the gap between practical impact and metric had to be bridged. Even though IoU is the most established and the standard metric for comparing semantic segmentation models, it is hard to connect it to practical use cases for Stena Recycling. Since they are interested in estimating the ratio of aluminium, the error of this estimation was more a relevant way to evaluate the models. However, IoU is of interest to other researchers within machine vision as well as Combitech since this is a metric that they can compare their results against. Thus, these two measures complement each other and fulfil different roles.

This error was measured by percentage by firstly calculating the difference between the predicted number of aluminium pixels in relation to pixels classified as either aluminium or other material. This ratio was then compared to the actual annotated ratio and is described in Equation 4.1 where p_a is the number of predicted aluminium pixels, p_o is the number of predicted pixels of other material, t_a is the actual number of aluminium pixels and t_o is the actual number of pixels of other material.

$$error\_percentage = \frac{\frac{p_a}{p_a+p_o} - \frac{t_a}{t_a+t_o}}{\frac{t_a}{t_a+t_o}} \tag{4.1}$$

This metric is measured over a collection of images at once since the helpful information for an operator would be a good estimate of general trends. For example, the ratio of aluminium during the last hour of operation is more useful than a rough estimate for each individual image. Thus evaluating over a collection of images rather on each individual image is more representative of how it would be used.

4.7.2 Qualitative evaluation

Along with the quantitative measures, a qualitative evaluation was performed to compare the frameworks and the model with no pretraining against models that had been pre-trained. The models were applied to four images from the pre-training dataset that had not been used in the training as well as four images from the test partition of the downstream dataset. These images were selected to represent a broad range of types of images. The first image was of an empty conveyor belt, this could be used to verify that a model does not predict material on empty images, simply because it expects there to be material in the image. The second image contained just a few medium-sized objects. The last two images contained large and small objects respectively to allow as wide a spectrum of possible images. The aim of the qualitative evaluation was to investigate if the models generalised on real-world data, but also to analyse in what way the models failed or succeeded in the task.

For the evaluation carried out on the downstream images, there was a corresponding ground truth label available against which to carry out the comparison. This was taken into account when analysing what each model predicted. When the evaluation was carried out on the pre-training images there was no ground truth available, thus it was more difficult to reliably judge what prediction was correct. However, even without the ground truth annotation, it is possible to distinguish parts of the background against the scrap metal. As such, some models might disagree on the type of metal, but recognising that it is metal and not part of the background is preferable.

This qualitative evaluation is limited in scope since the sample size is limited, this impacts the strengths of the trends involved as it is possible that any phenomenon observed in the images, might not hold for the general case. However, this type of qualitative evaluation has been used before in academic papers and the methodology applied here is similar to how previous studies regarding semantic segmentation have reviewed findings [13].

In addition to this, a qualitative evaluation of the attention maps from the pre-trained models was conducted in a similar manner. The image used to represent large objects both from the pre-training dataset and the downstream dataset was used to produce the attention maps. These maps were evaluated over a number of epochs by looking for the outlines of the objects and general trends. As some of the models had noticeable phase shifts depending on how many epochs they were trained for, the attention maps during these phase shifts were chosen to be presented in the results.

5 Results

In this chapter, the primary results of this thesis are presented. It is organised into four main sections, firstly, the result from the interview with Stena. This concerns the possible practical applications and the conditions at Stena which must be taken into account. Secondly, the loss graphs and attention maps from the pre-training are presented. Thirdly, the evaluation of the models is presented through the metrics computed and the experiments performed. Lastly, the qualitative results are presented which consist of images of the predictions from the models on both the pre-training data and downstream data.

5.1 Interview with Stena

In this section, the results from the interview with a Stena Recycling representative are presented. The answers from the interview have been summarized below under the appropriate question.

1. **What is the current process for measuring the amount of aluminium after sorting?**

   The current process for measuring the aluminium after sorting is through sampling and weighing, combined with experienced personnel who has developed an intuition for the amount of aluminium.

2. **What costs are associated with the measuring process?**

   Unclear what exact number, but highly costly.

3. **How does the amount of aluminium vary?**

   The distribution of aluminium along the conveyor belt is hard to estimate, however, it is not evenly distributed. Sometimes there is no aluminium on the conveyor belt while other times there is more. Since the refuse that is being processed comes from several sources, the material that is processed varies.

4. **What is the ambition of Stena in regards to sorting the aluminium?**

   The ambition is to lower the percentage of aluminium remaining after sorting from the 4% today to around 2.5%. Lowering it closer to 0% is not the end goal since a portion of the aluminium is embedded in other objects, too small or for other reasons not cost-effective to sort.

5. **Is automatic estimation a desirable feature?**

   An automatic estimation could be a desirable feature. Having a system that can give hourly, daily, weekly and monthly estimations of the amount of aluminium would be of value since it could be used to observe trends in the sorting chain. If the trends show an increase in aluminium then an investigation of the sorting process could be started to decrease the amount of aluminium. What is important for this to work is that it should not take up much bandwidth to transfer the information from the conveyor belt to the system since it may disrupt other parts of the process.

6. **What margin of error is required for automatic estimation to be a viable use case?**

   The margin of error that would be acceptable is 12.5%. For example, if the actual ratio of aluminium is 4%, then an acceptable result would be 4% $\pm$0.5%. This error would be calculated for the refuse processed over an hour or longer.

7. **Is an information system of aluminium ratio to inform manual responses viable to use in the sorting process?**

   Yes, it seems likely it could be a useful system. However, it needs to be accurate enough so that the operators trust the result. A false result could lead to an operator having to run 100 meters to a machine in vain. If this was the case the operators would quickly lose trust in the system and it might end up not being used. If the system were to be implemented a likely scenario would be to have a pilot study over the course of a 6- or 12-month period to validate if the results are trustworthy.

8. **If so, what kind of manual responses could be used?**

   A manual response to a report or flag in the information system could be to look over a particular machine in case there is an issue before it gets worse.

9. **What number of manual responses is acceptable to decrease the aluminium percentage?**

   The number of responses is not necessarily the issue if they are warranted, responses done on false information are intolerable. It would be possible to drive the ratio of aluminium down from 4% now, but at a cost at which it is not desirable.

10. **Is an automatic removal system viable to use in the sorting process?**

    No, there is no need to reinvent the wheel as there are other possible solutions available and in use. The possibility of implementing an efficient system, with a robot arm, for example, is not deemed viable and has been explored at Stena by the interviewee.

11. **What would be the effect of reducing percentage of aluminium, for example from 4% down to 2.5%?**

    The first effect is that it would be more profitable for Stena Recycling as aluminium is more valuable as a commodity than others that are sorted at the plant. The second effect is less waste of aluminium, since the unsorted aluminium is sold as scrap metal it is likely not going to be recycled and if it is, it will likely be through a less environmentally friendly process.

12. **In the context of creating a training dataset for neural networks, what are the possibilities of annotating images from the conveyor belt, i.e. can an experienced sorter tell apart materials with high accuracy just from the photos?**

    There is no sorter who is skilled enough to identify the material of objects just from pictures with 100% accuracy. It is not feasible.

5.2 Pre-training results

This section presents the loss over time during pre-training for the models trained using SwAV, Barlow Twins and DINO are presented. The steps in Figures 5.1, 5.2 and 5.3 are iterations during training where one iteration is 64 images. The graphs are smoothed using an exponential moving average with a value of 0.6. SwAV [8] and DINO [9] use the same loss function and thus their losses are of the same order of magnitude. Barlow Twins [50] uses a loss function that is based on a correlation matrix and is thus larger in magnitude. However, the loss is a unit-less quantity and only the relative results for a model are generally of interest. The loss is not used to compare models but rather used when updating the models' weights and thus shows the convergence of the models towards the optimal weights.

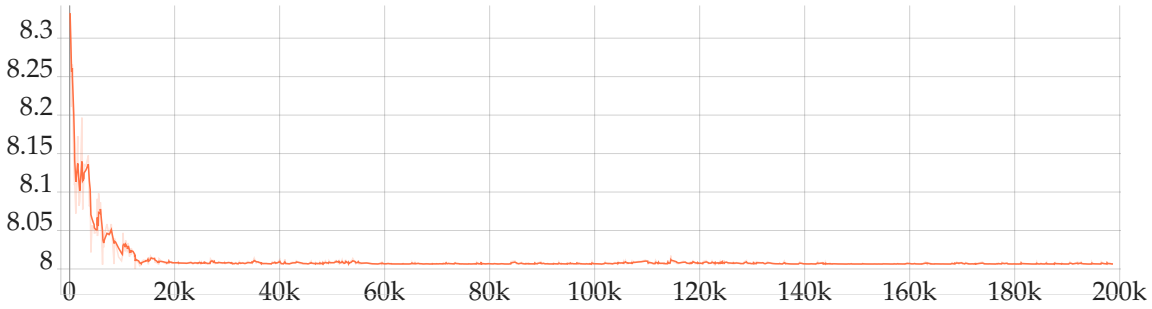

Figure 5.1: Loss for each iteration during pre-training on training data for model pre-trained using SwAV.

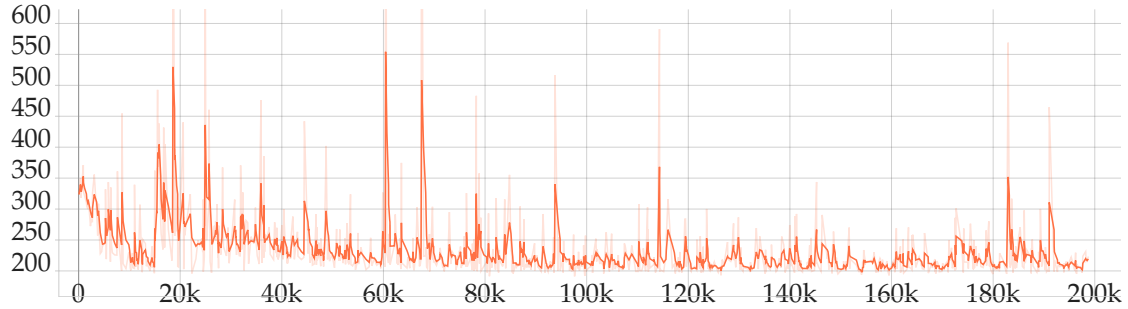

Figure 5.2: Loss for each iteration during pre-training on training data for model pre-trained using Barlow Twins.

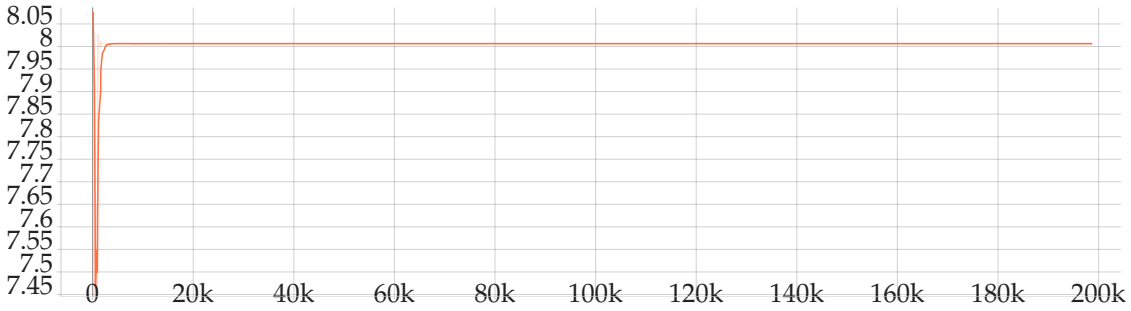

Figure 5.3: Loss for each iteration during pre-training on training data for model pre-trained using DINO.

The attention maps for each of the pre-trained models from selected epochs are presented in Figures 5.4, 5.5 and 5.6 along with the attention map of a model with no pre-training in

Figure 5.7. The attention maps show what pixels are most important for the output of the models as explained in Section 2.3.1. These models have only been pre-trained and thus there is no explicit instruction for the models to segment the image or to observe the shapes and position of the objects. The fact that the attention maps partly mirror the shapes of the objects in the images could indicate that the shapes and positional information of the objects are an important part of how the models encode the images. The attention map produced by the model which has not been trained shows that there is no discernible structure in the attention map in Figure 5.7, this is the expected result as the model's weights are randomly initialised. Initially, the attention maps for DINO and Barlow Twins in Figures 5.4 and 5.5 seem to have some structure related to the objects in the image. With more pre-training, the images become more similar to the attention map produced by the untrained model in Figure C.1. This indicates that these models have weights that are similar to randomly initialised weights.

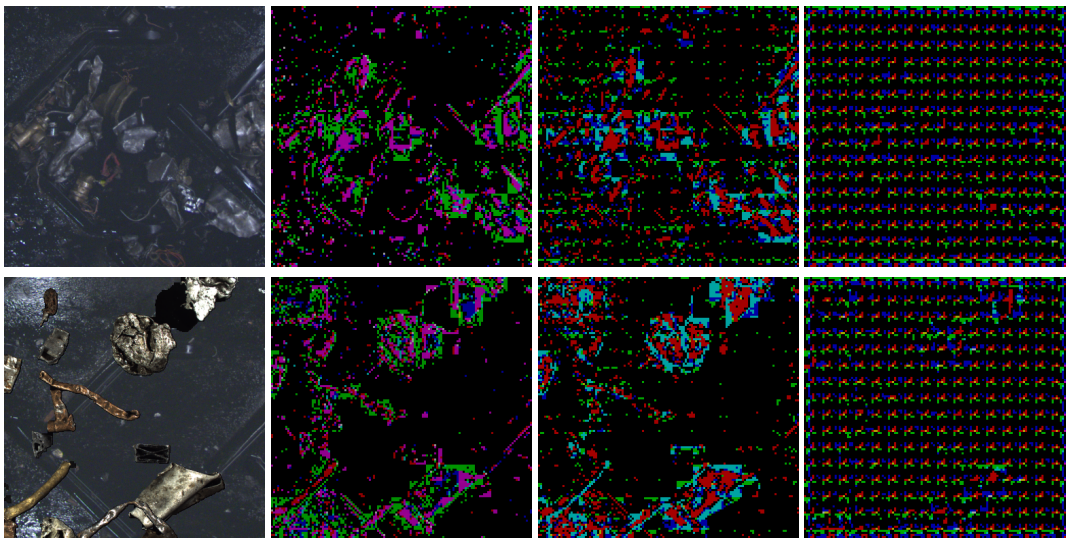

Figure 5.4: Attention maps extracted from the models pre-trained with DINO for 2, 3 and 4 epochs, respectively. Each attention head is represented by a different color, when multiple heads pay attention to the same patches the colors mix.

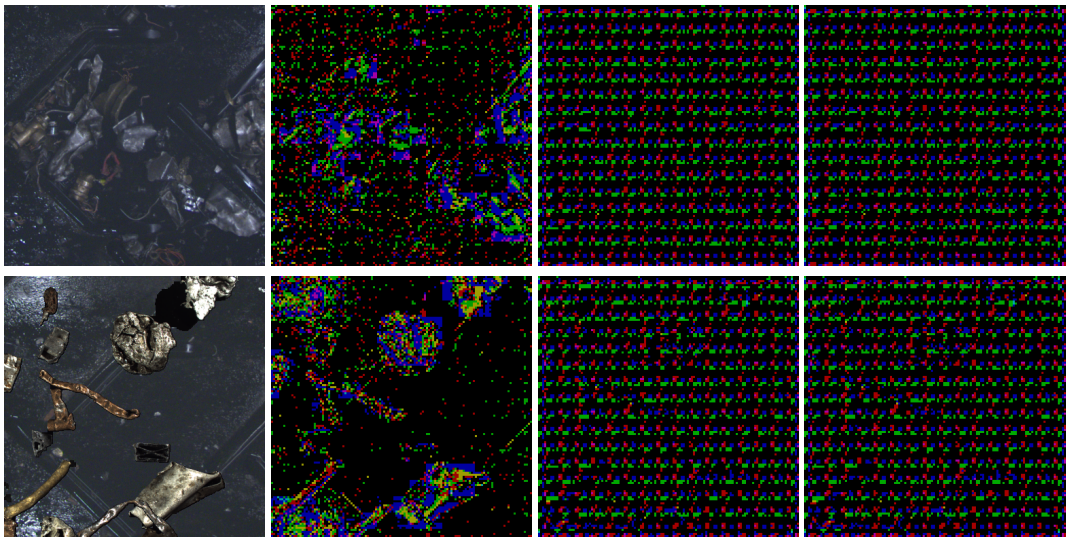

Figure 5.5: Attention maps extracted from the models pre-trained with Barlow Twins for 1, 2 and 3 epochs respectively. Each attention head is represented by a different color, when multiple heads pay attention to the same patches the colors mix.

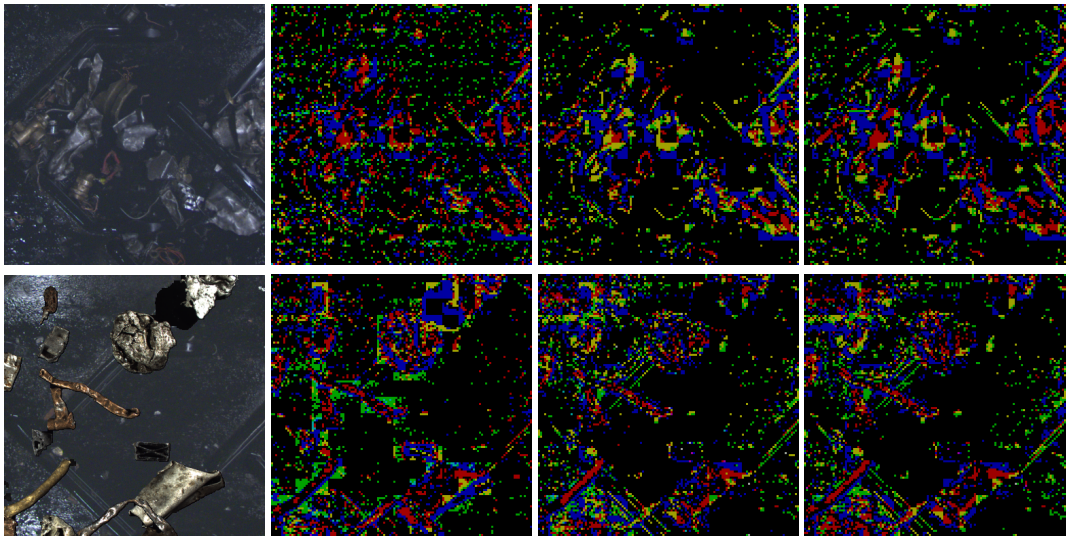

Figure 5.6: Attention maps extracted from the models pre-trained with SwAV for 1, 10 and 20 epochs, respectively. Each attention head is represented by a different color, when multiple heads pay attention to the same patches the colors mix.

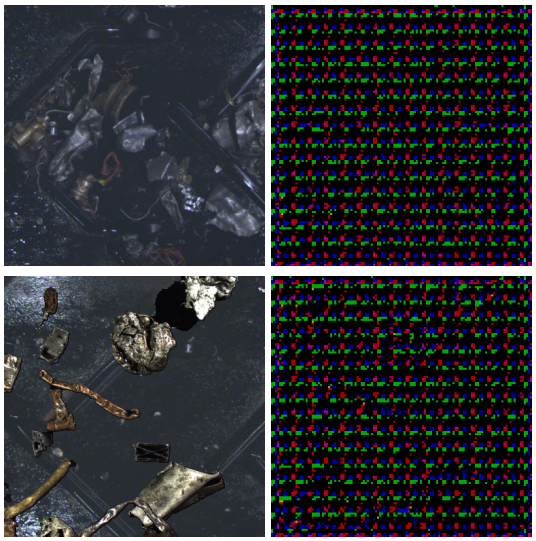

Figure 5.7: Attention maps extracted from a model which has not been pre-trained. Each attention head is represented by a different color, when multiple heads pay attention to the same patches the colors mix.

5.3 Downstream results

In this section, the results from the downstream trained models on the test dataset are presented. Tables 5.1, 5.2 and 5.3 presents the results from models on the test data that have been pre-trained using SwAV, DINO and Barlow Twins respectively for 1, 3, 5, 10, 15 and 20 epochs and then downstream trained for 100 epochs. The background, other and aluminium columns in the tables are the IoU for the background, other material and aluminium respectively. The IoU is from the best downstream training epoch on the test dataset. The mIoU is the average of the IoU for each class as described in Section 2.5 and the error value is the error of the ratio estimation of the amount of aluminium as described in Section 4.7, for the epoch model with the best IoU value.

From the results shown in Figures 5.4 and 5.5, it is clear that the attention map of the models pre-trained with Barlow Twins and DINO deteriorates over time. Since the attention of these models does not seem to focus on the materials in the picture after just a few epochs of pre-training, it is expected that the best performing versions of these models on the downstream task would be the models pre-trained 1 or 3 epochs. This seems to be the case in Table 5.2 for the model pre-trained with Barlow Twins since the best version of the model seems to be the one pre-trained for 1 epoch and the mIoU gets worse with later pre-training epochs. However, this does not seem to be the case for the model pre-trained with DINO in Table 5.1. The mIoU does not have the downward trend in mIoU as the models pre-trained with Barlow Twins, instead, the best version is the one pre-trained for 20 epochs. As the attention map for the model pre-trained with SwAV in Figure 5.6 is more stable it is expected that the mIoU in Table 5.3 which seems to be the case. Table 5.4 presents a summary of the best models together with the best model that have not been pre-trained and only downstream trained.

DINO

Pre-training epoch	Error (%)	Background	Other	Aluminium	mIoU
1	2	0.9597	0.7474	0.7800	0.8290
3	**0.5**	**0.9697**	0.7545	0.7848	0.8363
5	1.6	0.9483	0.6992	0.7742	0.8072
10	2.8	0.9548	0.7432	**0.7956**	0.8312
15	3.7	0.9571	0.7446	0.7899	0.8305
20	5.6	0.9626	**0.7620**	0.7926	**0.8391**

Table 5.1: Downstream results for models pre-trained with DINO for 1, 3, 5, 10, 15 and 20 epochs. The first column denotes how many epochs a model was pre-trained for.

Barlow Twins

Pre-training epoch	Error (%)	Background	Other	Aluminium	mIoU
1	**0.4**	0.9585	**0.7467**	**0.7980**	**0.8344**
3	2.1	0.9595	0.7193	0.7562	0.8117
5	1.4	0.9521	0.6997	0.7646	0.8054
10	1.6	0.9594	0.7213	0.7738	0.8181
15	5.9	**0.9603**	0.6933	0.7563	0.8033
20	2.9	0.9601	0.6753	0.7302	0.7886

Table 5.2: Downstream results for models pre-trained with Barlow Twins for 1, 3, 5, 10, 15 and 20 epochs. The first column denotes how many epochs a model was pre-trained for.

SwAV

Pre-training epoch	Error (%)	Background	Other	Aluminium	mIoU
1	0.6	0.9600	0.7594	0.8068	0.8420
3	0.9	0.9694	**0.7859**	**0.8176**	**0.8576**
5	2.9	0.9572	0.7557	0.7971	0.8367
10	**0.5**	0.9655	0.7703	0.7960	0.8439
15	2.0	0.9699	0.7656	0.8069	0.8475
20	6.9	**0.9701**	0.7726	0.8095	0.8508

Table 5.3: Downstream results for models pre-trained with SwAV for 1, 3, 5, 10, 15 and 20 epochs. The first column denotes how many epochs a model was pre-trained for.

Summary of best models

SSL Framework	Error (%)	Background	Other	Aluminium	mIoU
Barlow Twins	**0.4**	0.9585	0.7467	0.7980	0.8344
DINO	5.6	0.9626	0.7620	0.7926	0.8391
SwAV	0.9	**0.9694**	**0.7859**	**0.8176**	**0.8576**
No pre-training	0.9	0.9579	0.7565	0.8075	0.8406

Table 5.4: A summary of the best models produced by each setup on the downstream dataset.

The models from the best pre-trained epoch according to Tables 5.1, 5.2 and 5.3 trained on 50% and 20% is presented in Tables 5.5 and 5.6. The decreased performance compared to Table 5.4 is expected since less training data means that there are fewer examples for the models to learn and generalise from. The drop in performance is not consistent between models with the model pre-trained with DINO having the largest performance drop when only 50% of the downstream training data is used. The error of the models also seems to fluctuate and does not seem to be linked with the mIoU since in Table 5.6 the model pre-trained with Barlow Twins and SwAV had similar mIoU but dissimilar error. Also notable is that DINO had an error above 12.5% in Table 5.6 which is significantly higher than the other models trained with 20% of the downstream training data. For the other models, there is not such a significant increase in error despite the mIoU going down.

Trained on 50% of the training data

SSL Framework	Error (%)	Background	Other	Aluminium	mIoU
Barlow Twins	5.8	0.9593	0.7042	0.7453	0.8029
DINO	10.9	0.9222	0.5299	0.6004	0.6841
SwAV	5.0	0.9570	0.7150	0.7570	0.8097
No pre-training	**1.1**	**0.9586**	**0.7420**	**0.7595**	**0.8201**

Table 5.5: Downstream results for the models trained with 50% of the downstream training dataset.

Trained on 20% of the training data

SSL Framework	Error (%)	Background	Other	Aluminium	mIoU
Barlow Twins	**0.8**	**0.9419**	0.6266	**0.6901**	0.7529
DINO	23.3	0.9241	0.4939	0.5827	0.6669
SwAV	4.7	0.9403	**0.6651**	0.6728	**0.7594**
No pre-training	1.1	0.9391	0.6315	0.6891	0.7532

Table 5.6: Downstream results for the models trained with 20% of the downstream training dataset.

In Table 5.7 the results of the best models from Tables 5.1, 5.2 and 5.3 along with a model without pre-training, trained on processed downstream training dataset is presented. The results show a drop in performance for all models that have been pre-trained while the model without pre-training got a small increase in mIoU.

Trained on processed data

SSL Framework	Error (%)	Background	Other	Aluminium	mIoU
Barlow Twins	4.0	0.9457	0.7142	0.7712	0.8103
DINO	17.4	0.9429	0.6178	0.6793	0.7467
SwAV	1.8	0.9588	0.7511	0.7979	0.8359
No pre-training	**1.5**	**0.9566**	**0.7650**	**0.8048**	**0.8421**

Table 5.7: A summary of the models trained on the processed downstream dataset.

5.4 Qualitative results

This section presents the prediction of the models investigated on the validation dataset from the pre-training datasets and downstream dataset, presented in Figures 5.8 and 5.9 respectively. The figures in this section have the original image on the top row with the prediction for each model below as well as the ground truth for Figure 5.8. In the predictions and the ground truth, each red pixel is background, blue pixels are aluminium and green pixels are other materials. The models that were used for the prediction in Figure 5.8 are the best models that were presented in Table 5.4.

5.4.1 Downstream data

The results show that most models performed very similarly, but a difference that can be seen between the models is regarding small objects and patches of contrasting material on top of larger objects. For example, in the first image in Figure 5.8, which is mostly empty, there are some small artefacts on the left side of the ground-truth label, these are assigned different labels by the models. These artefacts are a result of the annotation process which automatically annotates objects against a uniform background, but occasionally part of the black background is falsely annotated as material. An example where the models all predict these artefacts the same as the ground truth is in the third image. In this image, there are two large aluminium objects with a dark patch in between which is annotated as aluminium in the ground truth. All the models infer that this dark patch is more likely to be aluminium than other material, in accordance with the ground truth.

Another interesting observation regarding the difference in model prediction and the ground truth can be seen in the upper-middle part of the second image. In the image there is a crushed aluminium soft drink can with a yellowish tint and a small greyish object of crumpled metal underneath. These objects are completely made up of aluminium as can be seen in the ground truth labelling in the second row. However, each model segments this as a mix of both aluminium and other material.

An additional example of where occlusion leads to error in the models' predictions can be seen close to the bottom of the third image where there is a thin object made of a grey material that occludes a wider aluminium object which is also grey. This occlusion exemplifies an error which was observed from each model. The objects themselves are correctly classified as aluminium and other material respectively, but the occlusion is missed and the part of the thin object that is on top of the larger object is incorrectly classified as aluminium.

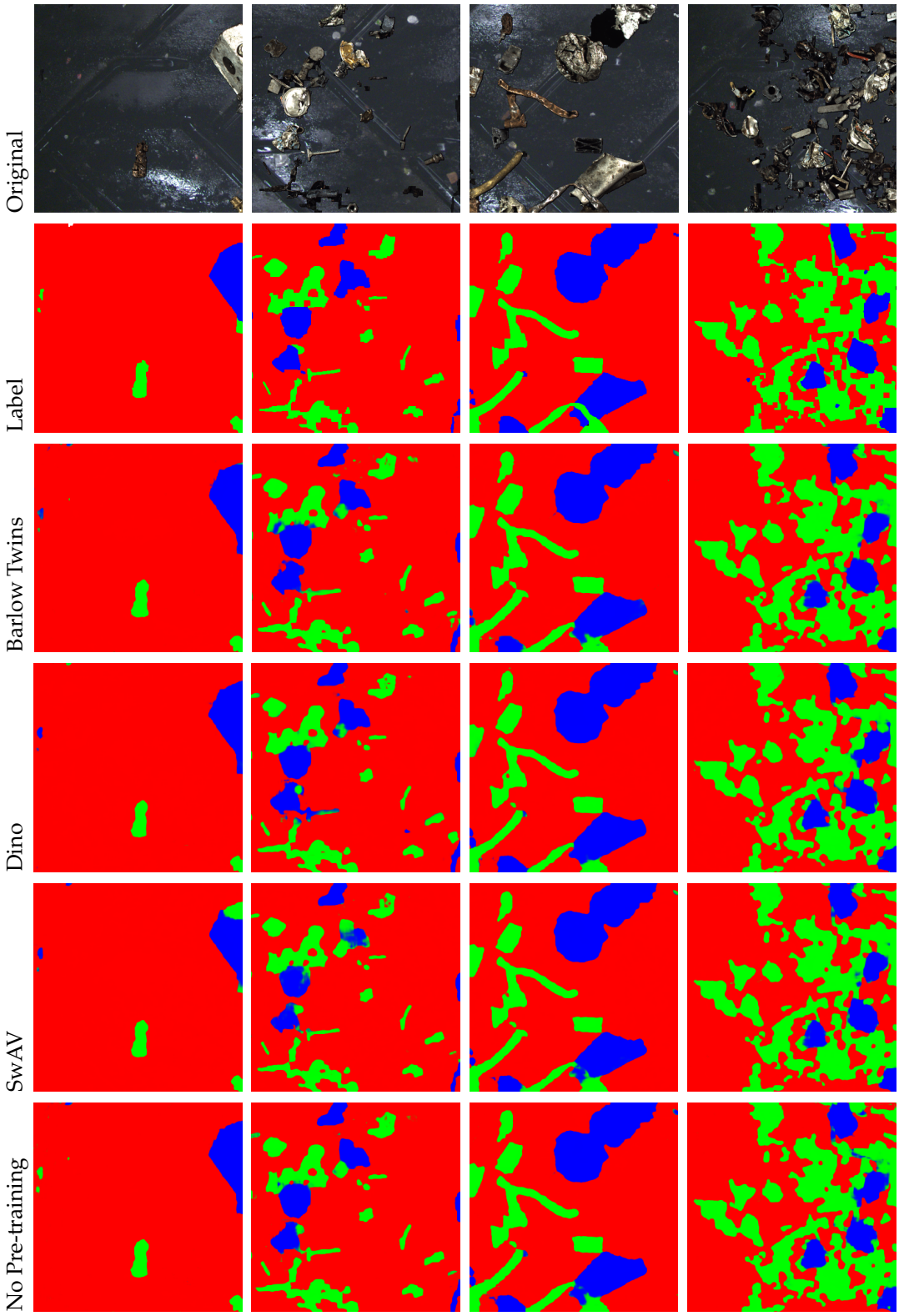

Figure 5.8: The first row consists of images taken from the test partition of the downstream dataset. The second row contains the ground truth annotations created through the stitching process. Each subsequent row contains the segmentation produced by a model on those images. The third row contains predictions by the model pre-trained with the Barlow Twins SSL framework. The fourth row is DINO, the fifth SwAV and the final row contains the predictions of a model that was not pre-trained.

5.4.2 Pre-training data

For the predictions in Figure 5.9 the best pre-training epoch that produced the finetuned models with the highest mIoU from Table 5.4 was chosen. To find the best downstream epoch to use several was tested and the best was presented. These epochs were: 24 for no pre-training, 94 for DINO, 24 for Barlow Twins and 86 for SwAV. Since there is no ground truth there is no reliable way to determine the correct annotation. However, it should be possible to determine what parts of the images are not metal objects but background. As such it is possible to analyse whether the objects can be detected against the background.

By applying this argument it can be argued that the predictions produced by DINO and SwAV pre-trained models as well as the model without pre-training struggle to correctly annotate the background. In contrast, Barlow Twins seem better adapted to classifying the background and the objects.

The third image seems to produce the least noisy predictions as it is primarily the objects which are annotated as either aluminium or other material and the conveyor belt is primarily classified as background.

The trends and differences in the behaviour of the models are not always consistent between different images. For example, the model pre-trained with SwAV appears to be able to handle the glare in the first image, which is largely empty, while the model pre-trained with DINO and the model without pre-training make more significant misclassifications. However, the third image has the reverse effect on these models. The SwAV pre-trained model produces the most noise and presumably incorrect prediction when compared to the model pre-trained using DINO. What is consistent, however, is that the model pre-trained with the Barlow Twins framework appears to perform significantly better.

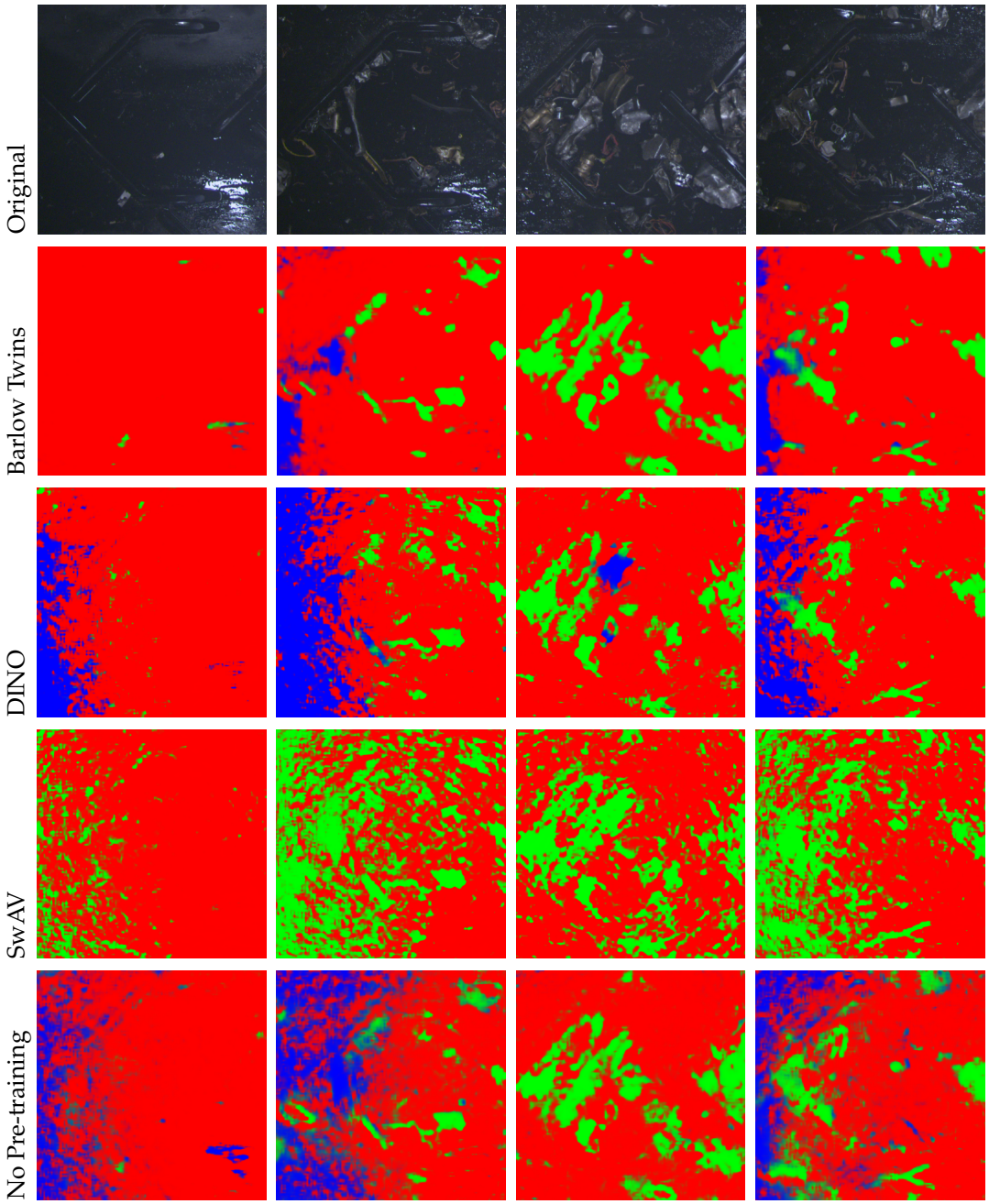

Figure 5.9: The first row of images are the 4 selected according to 4.7, the first is empty, the second contains a few objects, the third contains large objects and the fourth contains small objects. Each row contains the segmentation produced by a model on those images. The second row contains predictions by the model pre-trained with the Barlow Twins SSL framework. The third row is DINO, the fourth SwAV and the final row contains the predictions of a model that was not pre-trained.

6 Discussion

The discussion is split into three parts. Firstly, the results are discussed and analysed. Secondly, the methodology applied to produce those results are discussed to highlight potential weaknesses and strengths of the results. Lastly, the work is put into a broader societal context and the potential impact of the work and how it relates to society at large is discussed critically.

6.1 Results

The results from the thesis are discussed in this section. It is structured into subsections, pertaining to each self-contained part of the results.

6.1.1 Interview with Stena Recycling

The interview elucidated what potentially could be of value in a practical application of the technical methods used in this thesis. It also bridged the gap between metrics used in machine learning to evaluate models and metrics that would measure the impact of a practical application.

Potential use cases were discussed in the interview so that the design could be focused on the desired outcomes and so that criteria could be defined for when to consider a system viable. The primary use case consisted of estimating the ratio of aluminium to other metals and using this information to alert operators if there were abnormal amounts of aluminium going unsorted over a period of time. From the answer from Question 10 using the techniques investigated in this thesis for an automatic sorting system does not seem to be a desirable use case for now. This might be because large parts of the sorting process as a whole are already automated and it was not deemed likely by the interviewee that a new automatic sorting system would perform better than the existing systems.

Stena Recycling aim to reduce the amount of aluminium left after sorting and an information system is not an active means for achieving this, but it could contribute indirectly. For example, an information system could help in alerting problems in the sorting process earlier as it can show trends as per the answer to Question 5. An information system could also be used to observe the behaviour of new sorting processes and can thus be part of evaluating new sorting processes that could help reduce the amount of aluminium not properly sorted.

Specifically, the use case of estimating the ratio of aluminium was intriguing as this type of information was deemed actionable and useful by the Stena Recycling representative as seen in Section 5.1 Question 5 and Question 7. An advantage of such a system could be that there are no major changes needed to how the plant operates currently. This is because there are not any new industrial sorting equipment necessary or new sorting station, but rather just more information available for the operator. The images that will be fed through the information system would also come from cameras that are already mounted and in place at the plant. According to the answer to Question 8 in Section 5.1, this information would be used to perform the same manual responses as before, but hopefully with greater accuracy and increase the speed with which issues get addressed.

The metrics typically used to evaluate the performance of models on various machine vision tasks do not necessarily translate to a certain performance on the practical task at the recycling plant, for example, the ratio estimation. To bridge this gap, Question 6 was posed, wherein the error of the estimation was discussed and what error would be acceptable. From this, an error margin of 12.5% was deemed acceptable for a system to be potentially used in practice as an information system. This specific limit was derived from the example given by the interviewee with an acceptable range of prediction between 3.5% and 4.5% if the true ratio of aluminium is 4%. Which is the current estimation of aluminium left after sorting. This gave a metric that the models could be compared against to evaluate the viability of the system which is further discussed in Section 6.1.3. However, it is important to take into consideration that 12.5% is only based on the answer from one representative at Stena Recycling. It might be the case that other workers might have other acceptable ranges for a system to be deemed usable. Using the estimation of ratio is more easily linked to the potential use case of an information system at Stena Recycling in comparison to mIoU which might be better for comparing models against previous work within the field of machine vision. Using a system to estimate the ratio, it is more desirable in practice to not give the ratio per image, but instead give an average ratio over one hour or more, as stated in the answer to Question 6. This is because the ratio of aluminium to other materials on the conveyor belt varies largely over time according to the answer given to Question 3. This means that estimating the error in time intervals that are too small, might give misleading information. Normal fluctuations in the ratio of aluminium might be misinterpreted as a problem in the sorting process.

Lastly, an interesting answer to highlight is the one to Question 12 which discusses the possibility of an experienced sorter annotating images directly from the conveyor belt. Since the context of this thesis is that annotated data is hard and costly to acquire, it is of interest to see how a more experienced sorter could annotate data. However, according to the answer, it is not feasible even for an experienced sorter to create an annotated dataset only from images. Even though it is not feasible to create an annotated dataset only from images, it would be interesting in future work to use experienced sorters to create a dataset that is more true to the real-world situation. Perhaps by placing objects directly on a conveyor belt and then marking the material and then removing the marks. By doing this there would be a reference image with marks to inform what material is what when doing the annotating, and thus no stitching would be required. The ambition was to initially do something similar for this thesis, however, because of problems with acquiring materials that do not exist in the existing datasets, it was not possible.

6.1.2 Pre-training

The results extracted from the pre-training of the models were presented in Section 5.2. These consist of graphs of the loss during training for each one of the three SSL frameworks investigated in this thesis along with their attention maps from selected epochs. The loss function is different between the three frameworks, and the loss over time does not have to indicate whether or not the model will be considered successful or not, since only the results on the downstream task matter. However, it might still be interesting to compare the graphs to see

if there exists a connection between the pre-training loss graphs and the downstream results and how the loss varies between models.

One of the largest differences between Figures 5.1, 5.2 and 5.3 is how the loss evolve during training. The magnitude of the loss varies between the three frameworks because of the different loss functions as well as the magnitude of the variance of loss over time. The loss in Figure 5.2 sometimes varies by a factor of 3 between measuring points while the loss in Figure 5.3 varies very little. The only variance happens after the second decimal and after 20 000 iterations, the loss is constant. One could think that this variance in the loss for the model trained using Barlow Twins would result in a large variance in the downstream results for different epochs from the pre-training. However, as it can be seen in Table 5.2 the mIoU does not vary to a great degree. However, when the loss spikes in Figure 5.2, the loss immediately returns to the previously observed levels. This would suggest that the behaviour that caused the spike is not present in the states saved for later downstream training, as the spikes never occur at iterations close to the checkpoints. Another explanation could be that during downstream training the models unlearn most from pre-training which would explain why the results from the downstream training for the different pre-training epochs are similar.

Figures 5.1 and 5.3 show that both models seem to converge quickly and the loss does not change significantly after around 20 000 iterations for Figure 5.1 and not at all for Figure 5.3. This could indicate that the models have found a local optimum that they cannot escape, which would mean that every model after 20 000 iterations should be the same or very similar. However, from Tables 5.1 and 5.3 the results are not constant for the models from different pre-training epochs. This means that the model changes after 20 000 iterations even though the loss does not change significantly. Thus, the loss values in the pre-training are not a reliable predictor of downstream performance.

The models pre-trained with different SSL frameworks show distinct results in the attention maps. A comparison of the resulting attention maps is presented in the Figures 5.4 5.5 and 5.6. An attention map from an untrained model has also been provided in Figure 5.7 as a baseline, which is expected to just contain noise as the weights are randomly initialised.

The attention maps are generated from models that have been pre-trained for a certain number of epochs. These epochs have been selected to showcase phase shifts in the attention maps. For example, in Figure 5.4 the epochs are selected as 2, 3 and 4 as attention maps produced by models pre-trained for longer all contain similar results as that of epoch 4. In the early epochs, it is clear that the attention generally follows the edges and objects present in the image. Later epochs seem to be more uniform and could be a sign that the representation is more focused on global features, which fits well for classification, but not segmentation tasks. It could also be interpreted as a failure in the pre-training which produces models without a useful structure.

The attention maps produced by the models pre-trained with Barlow Twins for 1, 2 and 3 epochs are presented in Figure 5.5. These images seem to indicate a similar and even earlier phase shift as the DINO variants. The repeated structure occurred after the second epoch had been completed. Although similar to the DINO attention maps, some small structures remain as large objects seem to make the patches produce slightly different attention. However, the results from after the phase shifts of the DINO and Barlow Twins pre-trained models are quite similar to the attention maps produced by the randomly initialised model in Figure 5.7. This could indicate that pre-training is not worthwhile in some cases if not proper precautions are taken, such as making sure that the data used to pre-train and to train for the downstream task are very similar or finding the optimal configuration and augmentations to use. The models pre-trained using the SwAV framework seem to not have any phase shift for any of the 20 pre-training epochs. On the contrary, the attention map seems to become marginally clearer with each epoch of pre-training.

Attention maps are not conclusive evidence but could be used as a way to assess the rough state of a pre-trained model, without having to train it on a downstream task. This is also a

technique that can be used to analyse how a model works as you get visual information of what the model focuses on.

6.1.3 Quantitative results

This discusses the quantitative results and compares the different frameworks against each other and attempts to explain or discuss some of the differences observed between them.

Starting with models trained on the downstream task using the entire downstream training dataset, the best performing SSL framework based on the mIoU value from Table 5.4 was SwAV. SwAV had the highest mIoU with DINO having the second-highest mIoU and Barlow Twins with the lowest mIoU out of the pre-trained models. The expected result regarding which SSL framework would perform best was that SwAV and DINO would perform better than Barlow Twins since SwAV and DINO perform better on the majority of the benchmark tests [9, 50, 8]. However, the choice of settings and framework is heavily dependent on the end task. A reason for SwAV performing better than the other two SSL frameworks might be that the default settings used for the frameworks suit SwAV better than Barlow Twins and DINO on this particular task. One of the biggest differences between SwAV and the other two frameworks is that SwAV uses clustering to train the networks. As explained in Section 2.2.3, clustering during training means that the images are classified by how much they belong to each cluster, where the distribution of these classifications should be evenly distributed. Determining what these clusters that the models find can be hard to determine with confidence, but it seems reasonable that some clusters are images containing large objects, small objects, shiny objects or other categories. For many of the benchmark datasets that SSL frameworks get evaluated on, there are more clearly distinct classes such as cats or dogs, but it is more difficult to find intuitive definitions of classes in the pre-training data. However, since SwAV performed the best there seems to exist a relationship in images that help the model categorise the images into clusters and thus learn good representations of the data.

The model pre-trained with DINO achieved the second-highest mIoU which neither uses clustering as SwAV nor the novel loss function that Barlow Twins use. The paper introducing DINO [9] focuses its research mostly on SSL for pre-training transformers rather than CNNs which the Barlow Twins paper [50] focuses more on. Since the research regarding DINO focuses on trying to bridge the gap between transformers and CNNs that have been pre-trained using SSL, it might be that the design of DINO is more suitable for models using transformer architectures.

During pre-training, it is important to choose the correct augmentations for the downstream task so that the models learn the best possible representations. There is no guarantee that the selected augmentations were optimal for the task for any of the SSL frameworks as this was not deemed reasonable to exhaustively test. According to the paper introducing Barlow Twins [50] the framework is sensitive to removing augmentations. This sensitivity would imply that the model trained using Barlow Twins would be more adversely affected by suboptimal augmentations. However, it is important to also note that in the paper Barlow Twins performed worse than SwAV on the ImageNet dataset and better on the COCO dataset [32] for instance segmentation [50]. So it is not unlikely that Barlow Twins just do not work as well as DINO and SwAV for this task.

The difference between the best mIoU from the models trained with DINO, Barlow Twins and SwAV differs very little, 0.0185 between the best mIoU of SwAV and DINO and 0.0232 between the best mIoU of SwAV and Barlow Twins. Since the training is stochastic this small difference might be because of the randomness during training.

As presented in Section 5.2 the quantitative trend for the model pre-trained with Barlow Twins seems to correlate with its attention map in Figure 5.5. The mIoU of the model pre-trained with DINO does not seem to have the same declining trend and instead seems to have a more stable mIoU. This would also be expected for the later epochs since the attention map in Figure 5.4 seems to be almost constant, the small change that can be seen in Table 5.1 might

be because of the stochastic training mentioned earlier as the difference is relatively small. As discussed in Section 6.1.2 the attention map of the model pre-trained using SwAV does not seem to vary significantly over time which correlates with the results in Table 5.3.

Typically pre-training improves the performance [50, 9, 8], thus it is unexpected that the mIoU from the model without any pre-training was higher than the one pre-trained with Barlow Twins and DINO in Table 5.4. The expected results were that pre-training would give the models an advantage since there is previous knowledge from images of scrap metal, however, this knowledge does not seem to translate to the downstream task. As discussed in Section 6.1.2, the models that have been pre-trained does learn some representation of the data and seems all of the pre-training models focus on the material in the images at some point in their pre-training. However, looking at the results from Table 5.2 pre-training seems to hinder the model somewhat. The datasets used for pre-training and downstream training may be too dissimilar when compared to the papers introducing the used frameworks. These are either evaluated and finetuned on a different part of the dataset used to pre-train or on a similar dataset used for a different task [50, 9, 8]. The dataset ImageNet was used for the pre-training and was also used to evaluate the pre-trained models on classification tasks in the papers [50, 9, 8]. To be able to evaluate the performance on segmentation and detection tasks another dataset had to be used as ImageNet does not contain ground truth for these tasks. The used datasets are similar in nature to ImageNet as they aim to be general datasets and not domain-specific [14, 32, 17]. As such, the variation between datasets is larger than the variation between datasets. In contrast, the downstream and pre-training datasets used in this thesis are more uniform in the sense that they all have the same kinds of objects, with a fixed camera angle and lighting conditions. Thus, the variation between datasets might have a larger effect as the variation within the datasets is small in relation to the above-mentioned dimensions. In a sense overfitting to either dataset will hinder the performance on the other as the datasets are very specific and thus difficult to generalise between. This could imply that the knowledge acquired from the pre-training data is not as applicable to the downstream task as expected and thus must be unlearned. As previously discussed, the Barlow Twins framework has been noted as more sensitive to the choice of augmentations which could imply that it learns more specific information, in contrast to SwAV which might learn more general information. This could be a factor in the difference in performance. This would imply that if the differences between the datasets could be minimised, then the DINO Barlow Twins models might yield an increase in performance, similar to how SwAV did. This means that there still is something to gain from pre-training using SSL.

The results in Tables 5.5 and 5.6 could be interpreted as evidence that supports the hypothesis that the pre-training dataset differs too much from the downstream dataset to help the pre-trained models. These tables show that the model without pre-training performed better than the ones pre-trained with DINO and Barlow Twins when only 50% and 20% of the downstream dataset were used for downstream training. This would be consistent with the hypothesis that pre-trained models have to relearn some of their knowledge. If less downstream data is available to relearn, the mistakes cannot be corrected. Thus it is more effective to not pre-train. However, the model pre-trained with SwAV performed worse than the model without pre-training when only 50% of the data was used for downstream training and better again when only 20% of the data was used. This means that the choice of SSL framework is dependent on the amount of downstream training data available. Using SwAV and Barlow Twins to pre-train the models produces higher mIoU in comparison to the model pre-trained with DINO in Tables 5.5 and 5.6. This indicates that DINO in particular is sensitive to situations where only small amounts of downstream data are available while SwAV and Barlow Twins are more stable.

A trend that exists in the result of each of the models trained and the results in Tables 5.4, 5.5 and 5.6 is that the IoU value for the background is always higher than the IoU for other and aluminium material, as well as the IoU for aluminium material always being higher than the IoU for other material. The background is almost the same between all im-

ages, the only difference is the placement of the tracks of the conveyor belt, this homogeneity of the background is probably the reason why the IoU is constantly the highest. The reason for the IoU of aluminium material being higher than other material is not as straightforward to explain, but it could possibly be because the objects that make up other material are smaller and more difficult to segment correctly.

From the interview with the employee at Stena Recycling, it was clear from Question 6 that an acceptable margin of error was 12.5% and all the models presented in the results achieved an error below this. This is important as if a system were to be implemented it is imperative that the operators trust the information given as indicated in the answer to Question 7. A dimension to consider beyond if the error is within the threshold is the types of errors that are present in the estimation. For example, one has to consider whether there is a difference between reporting a ratio too high and one too low. For example, it is conceivable that an estimation lower than expected could be interpreted as an indication that the sorting of aluminium is performing better than expected. As this is good news it might be more willingly interpreted as truthful than the opposite. Thus, it is possible that underestimating might lead to a false impression of good performance and problems could go unnoticed. Overestimating might lead to more unnecessary investigations which is intolerable according to the answer to Question 9.

The errors measured indicate that the models do not over- or underestimate to a degree greater than the acceptable margin of error. However, it is difficult to draw any conclusion regarding which of the models performed better than each other based on the error rate since it varies greatly between downstream epochs while still most of the time remaining below 12.5%. Graphs of the error from the pre-trained models from the best pre-trained epoch along with the error for the model without pre-training are presented in Appendix A.

The error obtained from the tests is primarily a measurement of their viability in real-world applications. Important to keep in mind with this error rate is that it is calculated based on images from the downstream dataset and not from actual images from the conveyor belt. Because of this, it is unlikely that these results would translate exactly into actual images on the conveyor belt, without training with a more realistic dataset.

Outside of the results previously discussed in this section, training and evaluation on the downstream dataset, but with different gamma and saturation values for the background and material was performed. The results are presented in Table 5.7 which shows that the best mIoU was lower for each model of the model that had been pre-trained when compared to the result in Table 5.4. The best performing model on this dataset was the model without pre-training which may indicate that this processed dataset was not more similar to the pre-training data than the dataset without changed gamma and saturation values. Lowering the saturation and increasing the gamma darkens the images which may lead to information being lost since objects are not as clear which would explain the decrease in mIoU. However, looking at the inference images in Figure C.1 in the appendix and comparing them against the inference images in Figure 5.8 it is clear that SwAV performs better when downstream trained on the processed images. This indicates that models pre-trained with SwAV are sensitive to changes in lighting, which could be adjusted by changing gamma and saturation values. It also indicates that there may be a possible gain in performance to altering the downstream data. If doing this makes it more similar to the images that the model will be used on in the end.

6.1.4 Qualitative results

The qualitative results explore two different dimensions of the models' performance. Firstly, in what way do the models succeed and fail to predict correctly. Secondly, in what capacity do the downstream trained models' performance translate from the downstream task to segmentation on the pre-training data. In the segmented images from the downstream tasks, there is a ground truth to compare against and the differences are mostly about which mate-

rial is classified as aluminium or other material. In contrast, the images from the pre-training dataset do not have a ground truth and as the performance appears to be much lower, there is a larger variation in behaviour and thus also likely noisier. This would be because each model varies more and thus, an observed behaviour in one picture could just be a variation within the model as opposed to a difference between the models. Thus, it is difficult to tell whether a given prediction is an outlier or an indication of a trend for real-world performance.

Downstream data

The differences observed in the segmentations of images from the downstream data are mostly about what metal is classified as aluminium or other. This is because the outlines of the objects against the background are consistent across the different models, which is in line with the relative closeness in performance in the quantitative evaluation. As presented in the results in Section 5.4, one of the differences that can be seen between the models is regarding artefacts that in reality are just black shadows and not part of any actual objects. The models presented predicted some of these patches as aluminium or other material and were not always consistent between models. Since these black artefacts are equally likely to occur within aluminium images or images of other material it is very difficult to accurately predict these pixels' class, since no actual scrap metal object is depicted by these pixels. However, it is not impossible. As mentioned in the results, there is an example where there is a dark patch of shadow between two aluminium objects. In the ground truth, this is classified as aluminium which all of the models successfully do as well. In practice it would not be desirable that shadows of objects or other artefacts would be classified as metal, this is thus a clear example of how errors introduced in the annotation of the downstream data are transferred to errors in the models. This type of error would not be detected quantitatively as long as the test data contained similar artefacts.

The results showed that the models have difficulty with occlusions of objects with dissimilar and similar colours. An example of occlusion with two objects of different colours was shown in the results where a soft drink can with a yellowish tint was on top of some crumpled up material with a greyish colour. These objects were both aluminium but were classified as a mix of other material and aluminium. The models infer that these objects are part of occlusion and thus different objects, however, it fails to correctly classify them. This could be because, in the training data, objects of the same material are rarely touched, thus when the model detects two objects close to each other it is biased towards classifying them as different materials.

The second type of occlusion with two objects of similar colours was also presented in the results. The result showed two objects with a grey colour where a thin object of other material was on top of a broader aluminium object. The models all predicted incorrectly that both of these objects were aluminium in the occlusion part and not that the thin object was of other material. It is possible that occlusions of thin or small objects on top of larger objects are more difficult, especially if they are similar in colour, as the likelihood that the perceived edge between the objects is just noise or a bend in one and the same object increases as it becomes smaller in relation.

Pre-training data

When comparing segmented images from the pre-training dataset, there are no ground truth annotations to compare with, thus this analysis was mostly comparative between the models. However, even though it was not possible to reliably tell aluminium apart from other material, it was still possible to tell scrap metal apart from the background with ocular inspection.

The variation between the segmentations of the images was larger than on the downstream data, this was likely because the models in general performed worse on the pre-training data. In addition, the types of errors observed varied more. For instance, in the first

image, there is very little scrap metal material on the image. This was intentionally chosen since the correct annotation should be almost only background. The models pre-trained with Barlow Twins appeared to perform the best, as there is very little noise in its segmentation. A minimal amount of the background is predicted as either aluminium or other material. In contrast, SwAV, DINO and the model without pre-training appear to have much more noisy segmentations as large parts of the background is predicted as either aluminium or other material, seemingly without structure. In the first image, there is also a strong glare present in the lower right corner which primarily affects the model without pre-training as it predicts it as aluminium. While the models that have been pre-trained mostly classify it as background. This could be because the glare likely appears visually similar between separate images and thus is something that is not important to encode in the representation since it is not something which can reliably be used to distinguish different images.

The edges between classified objects and the background were not as defined in these segmentations compared to those of the downstream data. This is an indication of uncertainty and likely an indication that the quantitative results would have been worse on this dataset because an object is in general composed of one material and does not change along a gradient if it does change at all. A source of this uncertainty could be the difference between the datasets, but also an effect of the pre-training images being of lower quality as they are not as well lit.

Generally in the pre-training images, the material is accumulated along the tracks of the conveyor belt with some material spread out elsewhere in the image. As can be seen in Figure 5.9 in the top row, in images 2, 3 and 4. All of the models except for the one pre-trained with Barlow Twins appeared to have noisy predictions, but be able to locate the objects to some degree. The tracks in the images provide a clear edge, which appeared in the segmentations of the three models as part of the background which was interpreted as a sign that the models could to some degree distinguish the objects from the background.

As there is no ground truth to compare against it is hard to distinguish models which are fairly close in behaviour. However, the model pre-trained with Barlow Twins' behaviour is significantly different enough that it can be said to be qualitatively better. As the practical application would be on images like these, it seems that there is an advantage to using SSL. It is important to keep in mind that it is essential for the model to not only detect objects but also to determine if they are made of aluminium or other material. This cannot be properly tested on this data as there is no ground truth. However, the pre-training images were taken after the aluminium had been sorted out, as such, it is expected that about 4% of the remaining material is supposed to be aluminium. Thus it is more likely that a correct segmentation contains primarily other material and not so much aluminium.

6.2 Method

In this section, the methodology of the thesis is discussed. The strengths and weaknesses of the method and how they affect the replicability, reproducibility and validity of the results. The discussion is broken down into sections discussing the interview, data, choice of architecture, evaluation process and lastly the sources.

6.2.1 Interview with Stena Recycling

The interviewers were not explicitly trained in any interviewing technique, thus it is possible that how the interview was conducted affected the result. Some of the questions were primarily factual, such as Question 1, which could possibly be less affected by the way they were asked. However, as the questions were open-ended, it is possible that different parts of the process might be highlighted or focused on, depending on how the interviewee interpreted it. Other questions were more broad such as Question 4 and can likely be interpreted in many different ways.

The questions regarding the viability of the hypothetical use cases are by their nature speculative. Thus, these questions do not necessarily investigate the actual likelihood of a given use case being viable in practice or not, but rather the interviewee's belief of what could be viable or interesting. This was valuable in guiding the work as it allowed it to more closely follow the interests of Stena Recycling and the practical considerations.

Another thing to consider is that, just because the interviewee determined that an information system could be viable, it does not necessarily mean it will be viable. The system has to demonstrate that it works in practice by being implemented and tested. However, the answers are useful in guiding what to investigate as possibilities.

As previously stated, the interviewers were not experienced with scientific interviewing techniques and the interview was not originally planned to take on as large of a role as it did. Because of this the interviewing methodology was not as thoroughly investigated as it could have been. A consequence of this was that no coding phase was performed. This could possibly have led to information being lost in the summarization step. Showing the summaries to the interviewee and having them approved could help avoid factual errors, but could be leading when considering the more open-ended and speculative questions.

The results could be improved if more employees were interviewed, it would be of particular interest if the employees had different roles and perspectives as this could enrich the discussion and provide more nuance, in particular to the more speculative and open-ended questions. In addition, other interviewees might suggest different error thresholds to be appropriate for the ratio estimation. This could possibly strengthen the credibility of the threshold if there emerged a consensus on what is an acceptable error.

6.2.2 Pre-training data

Since the pre-training dataset was collected over a large span of time it is likely that the material on the images collected is fairly representative of what a practical application would encounter. An issue in this data is that because of the density of the material significant occlusion is observed. A possible improvement would be to spread out the material on the conveyor belt with a heavy flap pushing material on top of other material down the conveyor belt.

However, some degree of occlusion is inherent to the problem as one object can be composed of one material on the outside and another on the inside, obscured from the view of the sensor. This type of occlusion is difficult to mitigate in a cost-efficient matter and it is likely that the aluminium material trapped in such a way would not be sorted regardless of what system is used. This can be demonstrated by the fact that Stena Recycling's ambition is not to lower the ratio of aluminium to 0% but to 2.5% as a more fine-grained sorting would not currently be cost-efficient according to Section 5.1, Question 4.

Access to the dataset is likely the biggest hurdle for a researcher looking to replicate the results of this thesis. As the images are not from a publicly available dataset, it would be difficult to replicate the dataset without access to a recycling plant. However, if there was a dataset of scrap metal from a recycling plant with similar processes publicly available, then the results would likely be similar.

6.2.3 Downstream data

The number of images available was limited by practical considerations. Thus, when splitting the dataset into the training, validation and test partitions it was important to make sure no object appeared on images in more than one partition. This requirement is likely to have been fulfilled as when the material was replaced, an image without material was taken to demarcate the flow of images of different selections of objects. However, the size of the dataset and the split meant that the validation dataset and test dataset might not be representative of the larger dataset due to not containing every kind of object.

As the aim of this dataset is to recreate the images taken at Stena Recycling's plant, several steps were taken to ensure this. The lighting was reproduced by acquiring a similar light source and by placing it in a similar location relative to the camera and objects. The same model camera was used to collect both datasets. Further experiments on the dataset were conducted with changed gamma and saturation values to try to emulate the lighting conditions at the recycling plant. Despite this, there is always some variation in conditions that can lead to differences in the images. One such difference is that the material obtained from Stena Recycling consisted generally of larger objects than some of the ones observed on the actual conveyor belt at Stena Recycling's plant.

Another source of discrepancy is that the images were stitched together. To guarantee a correct ground truth annotation, images of the material were taken separately and stitched together on top of a background of an empty conveyor belt. In the images collected at Stena Recycling's plant, material tended to cluster along the extruded wedges in the middle of the conveyor belt, but in the dataset created from the stitched images, the material is positioned more uniformly over the image. An additional effect of the stitching process is that there were some errors when creating the masks for the material in the images, which resulted in shadows being included in the image or sometimes small parts of the objects not being carried over to the stitched image. Additionally, when objects overlap in the stitched images, it is possible they do not interact how they would naturally by physically touching, as they are only layered on top of each other.

It is important to take these facts into consideration when discussing how a resulting metric on the downstream dataset would translate to use in practice. This point is illustrated with the qualitative evaluation of models on the dataset collected at Stena Recycling's plant as presented in Section 5.4. Those results showed that there is a large variance in how models with similar metrics on the downstream dataset behave on the pretraining dataset. Thus reflecting that performance on the downstream dataset might not carry over to real-life application of the models, especially in the case of models that were not pre-trained.

However, the downstream dataset is used in two different capacities. Firstly, to investigate how the performance of the models would be in a practical application, this is where the earlier weaknesses regarding validity lie. Secondly, the dataset is used to evaluate how SSL frameworks and resulting models compare against one another on the actual task of semantic segmentation without any consideration for how those results translate to practical results on images from the recycling plant. In the second sense, there are fewer weaknesses regarding validity as any discrepancy between the downstream data and pretraining data is irrelevant for that part of the evaluation.

Similarly, as for the pre-training data, it might be difficult to replicate this dataset without access to the material at Stena Recycling's plant. However, it would likely be easier to replicate this dataset as it only requires access to similar material and not the recycling plant. The downstream dataset does not require as many images as the pre-training dataset to replicate the results.

6.2.4 Architecture and hyperparameter selection

The way hyperparameters were chosen was focused on getting an acceptable performance and not on finding the optimal configuration. Thus, not all possible combinations were tried. For example, it is possible that an optimiser would perform better than the selected one given a more suitable loss function. However, it is likely that the choice of architecture has only a small effect on how the SSL frameworks perform relative to one another because during the hyperparameter search the difference between the variations tried was always small, around 1%, with the exception of the decoder choice. In this case, the difference was more substantial, but because the configuration of using a Swin Transformer as encoder and a UPerNet as decoder has produced state of the art performance it is likely that no large improvements could reasonably have been made.

The choice of encoder and decoder was motivated by the recent successes, as discussed in Sections 4.1.1 and 4.1.2. They were also chosen based on their widespread use, thus open-source implementations exist and the results can more easily be replicated. The thesis was limited to one encoder architecture because each pre-training required on the order of 150 GPU-hours which in combination with pre-training with 3 frameworks was deemed too high a cost for the potential small gain. However, it could be interesting to investigate a CNN-based architecture in future work to compare the effect of encoder architecture choice.

6.2.5 Evaluation

For evaluation of the models, mainly two metrics were used to compare the models, mIoU and error. Using mIoU to compare and evaluate the models is one of the standard metrics for comparing models on semantic segmentation which is why it was chosen [34, 22]. As previously discussed this helps link the findings of this thesis to previous work within the field of machine vision. The error of the estimation of the ratio of aluminium to other materials was chosen so that real-world impact could be inferred. A difference between the measurement of the ratio between this thesis and the actual measurement that takes place at Stena Recycling is that in this thesis the ratio is computed based on pixels while at Stena Recycling it is measured by weight. By only measuring the number of aluminium pixels in comparison to pixels of other materials, the depth and weight of the different objects are not taken into account, along with the occlusion of objects not being accounted for. For future work, a conversion between aluminium pixels and weight could be explored to measure the ratio of material in a more similar way that is being done at the recycling plant today. As the measurement is taken over a sample of images captured during a period of time on the order of an hour it should mitigate potential variances and errors in the conversion between pixels and weight, as well as errors in the estimation itself.

The models were also evaluated qualitatively by ocular inspection which is a common practice when investigating semantic segmentation as the results are visual in nature [13]. There are inherent limitations with this type of evaluation however, as only a small subsection of the images were evaluated it is imperative that this sample represent the dataset in general so that observed phenomena or behaviour are not outliers, but trends. The images were chosen with this in mind, but it is an important limitation to be aware of when evaluating the certainty of the conclusions which are based on this evaluation.

6.2.6 Sources

The sources used in this thesis consist mostly of peer-reviewed published papers from journals, books and conferences. The sources that are not from published journals are sources regarding the hardware used at Berzelius, VISSL documentation and information about Stena Recycling which does not have a peer-reviewed published source. Most sources have been found by searching through Google scholar and IEEE Xplore which are reputable tools for finding published scientific material. The sources have been chosen for their relevance to the subject and based on how new they are since techniques within the field of machine vision quickly get replaced. Because of this, sources that are as recent as possible and from the 21st century have been used in the majority of cases.

6.3 The work in a wider context

There are several important aspects to consider when relating this work to a wider context, both limitations as well as potential risks and benefits. By making the recycling process more efficient it could possibly encourage more local recycling thus having a positive environmental impact. It is also important to consider the impact applying the suggested use cases could

have on those working with it. For instance, if it will be used as a tool to alleviate workloads or to replace human workers.

The potential use case most interesting to Stena Recycling came to light during the interview in Question 7 in Section 5.1 was regarding using the models to estimate the amount of aluminium after sorting. This information could be used as a support system for the operators at Stena Recycling and would help detect problems in the sorting process. Using this system to detect unwanted trends in the sorting might lead to problems in the sorting process being detected earlier which would lead to less aluminium going to waste when problems occur. Using this information might not only lead to greater profits for Stena Recycling but might also help alleviate some of the workloads at Stena Recycling and help the workers focus more on other tasks.

These improvements could make the recycling industry more attractive, both to companies and employees, as the profit margins improve and the workload becomes lighter. This might increase local recycling, through Stena Recycling and potentially other companies, which helps the environment by less waste needing to be shipped to developing countries for further processing which may not have the capability of sorting the waste properly [38].

Since machine vision can be applied to many different tasks, for instance in the detection of cancerous cells and self-driving cars, investigating methods for alleviating the training of machine vision models may have an impact on more societal areas than just refuse sorting. The problem of obtaining enough training data is something most machine vision tasks have and using SSL to help can be used by most tasks. Helping models train might lead to safer AIs for self-driving cars and more accurate models for detecting cancerous cells.

In each of these tasks, AI could either be used as a tool to support workers by performing repetitive tasks or to extend the capabilities of humans [29, 3] or be used to replace human workers in certain professions [45, 4]. If implemented as an information system as suggested in Section 6.1.1, then AI will be used to support and extend human workers, rather than to supplant them. However, it could in theory lead to replacement in subsequent implementations as the technology is improved. It is important to note that most of the processes in Stena Recycling's plants already are automated, as long as they are cost-effective. This suggests that if a system was shown to be capable of sorting aluminium and cost-effective, it is likely it would replace manual workers. It could be argued that minimising the amount of dangerous work might be viewed as common good since the safety and health of people is premiered. However, changes in the job market can be turbulent and stressful as workers seek re-employment. It might also be positive to have heavy industrial jobs as it might be closer to their skill set or just preferable to some.

7 Conclusion

This thesis work has explored the possibility of applying recent advancements in the field of machine vision to assist in sorting aluminium at a recycling plant. In particular the possibility of pre-training with SSL was investigated as well as using recently successful transformer architectures. The motivating factor behind investigating SSL was the prospect of not requiring large annotated datasets. The difference in performance quantitatively was not as large as expected when it was evaluated on the downstream dataset. However, a significant difference in results between the models without pre-training and those pre-trained with SSL can be observed in the qualitative evaluation on the pre-training dataset, which is more representative of the practical end application. Thus, it can be argued that SSL enables models to learn from datasets created in a lab environment which can be easily produced and transfer that knowledge to real-world images. This does not seem viable without pre-training.

To summarise the most interesting findings, SSL does improve the performance as evaluated by both the qualitative and quantitative approaches. The increase in performance is heavily dependent on the choice of SSL-framework and the data used to pre-train and downstream train. SwAV achieves the best quantitative results and could therefore be seen as the best candidate if the downstream dataset is very close to the pre-training dataset. However, as Barlow Twins seems to have the best performance on the pre-training dataset, as measured qualitatively, it would be the best candidate if the downstream and pre-training datasets differed significantly.

Using error of the ratio estimation as a metric is more useful when evaluating the real-world application impacts in comparison to mIoU. As this is what will actually be used in practice. Through this metric, it should be easier to help determine whether a network is good enough to use in practice in comparison to solely relying on mIoU.

7.1 Research questions

The research questions will now be explicitly answered with conclusions drawn from the results and discussions presented in Chapter 5 and 6 respectively.

RQ 1: How could a transformer model in conjunction with SSL be viable to assist with recycling aluminium at a recycling plant?

A transformer model could be useful in recycling aluminium at Stena Recycling as an information system as discussed in Section 6.1.1. For an information system to be useful the

error rate needs to be less than 12.5% according to the answer to Question 6 in Section 5.1. This is possible for models both pre-trained using SSL and without pre-training on the downstream data. The results also show that using transformer models in conjunction with SSL is beneficial and can bring improved performance if applied correctly and with enough downstream data for training as can be seen in Section 5.3.

RQ 2: How is the behaviour and mean intersection over union metric affected by using different SSL frameworks?

From Table 5.4, it is clear that using SwAV for this particular task gives the best mIoU values. However, SwAV appears to be the most sensitive to lighting conditions because of the increase in performance from Figures 5.9 and C.1. The models pre-trained with DINO have consistent metric values when all of the downstream data is available as seen in Figure 5.1. However, it has the steepest drop in performance among the framework when less downstream data is available according to Tables 5.5 and 5.6. Barlow Twins become worse with more pre-training from Table 5.2 and the representation that is learnt after the first epoch of pre-training does not translate to the downstream task. Barlow Twins performed the worst based on the mIoU when all of the downstream data was used for training as seen in Table 5.4, but performed the best on the inference on pre-training images in Figure 5.9.

RQ 3: Does pre-training using SSL yield an improvement for classifying aluminium and non-aluminium objects?

Pre-training models with SSL yields an increase in performance over models that have not been pre-trained for classifying aluminium and non-aluminium objects, which can be seen in Table 5.4. However, as discussed in Section 6.1.3, the improvement is dependent on the SSL-framework since not every framework yielded an improvement, and the improvement that did occur was minor. This answer, however, is only based on the quantitative results which are from the downstream data and as discussed in Section 6.1.4, might not be transferable to the pre-training images.

When comparing the performance on the pre-training dataset, which is more close to what the practical end task would be, there is no quantitative results to evaluate. However, there is a clear difference in the performance qualitatively as discussed in Section 6.1.4.

RQ 4: How does the amount of annotated training data affect the performance?

As discussed in Section 6.1.3, using less downstream training data result in lowered performance for both models that have been pre-trained and models that have not been pre-trained. Along with this, models pre-trained with DINO, SwAV and Barlow Twins perform worse than the model without pre-training. Important to note is that the amount of downstream data available in total is limited when compared to most other machine vision datasets [32, 54, 17]. This limited dataset might be too small to reduce and draw any conclusions regarding the amount of training data as the full dataset might already be too minimal.

In summary, SSL has been proven to be marginally useful when looking at the metrics results and more useful when looking at it quantitatively on real-world images. Using SSL seems to be able to help the models transfer knowledge between datasets which helps alleviate the problem of acquiring annotated data. As it can transfer the knowledge learned from a dataset learned in a lab environment to real-world images. This avoids the need to annotate real-world images which is much more costly. Hopefully, this could lead to cheaper and more efficient recycling of aluminium which impacts the environment and sustainability in a positive way.

7.2 Future work

More future work needs to be done to see the impacts of the model on real-life situations since the quantitative evaluation in this thesis only occurred on the stitched together datasets. In future work, a downstream dataset that is more similar to the pre-training dataset should be created by not stitching together images. Instead, the material could be marked in advance and then placed directly on the conveyor belt and images can be taken that can then be used as a reference for images without the annotations. The possibility of annotating the pre-training data could also be explored with a similar approach. By having data that is more similar to a real-world situation, a possible information system could be more extensively evaluated and perhaps the impact of such a system could be further researched. For example, the total weight of the objects appearing on the conveyor belt could be measured over some period of time, separated into aluminium and other material. Then the produced pixel ratio of the model could be compared after a pixel-to-weight conversion. This would avoid the need for annotations, but still measures the models' performance on real-world data in the way it is aimed to be used.

Trying to add more light sources at the conveyor belt when the images for the pre-training datasets are taken would also be interesting future work. With more lighting, there would hopefully be more information to extract from the images since quantitatively all the models performed worse when trained on darker images.

The comparison between the three SSL frameworks together with transformers could also be further studied using other datasets to do semantic segmentation to see if similar trends and results occur like the ones in this thesis. This could be done in other areas of the recycling process using other materials travelling on the conveyor belt.

Bibliography

[1] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. 2019. arXiv: 1803.08375 [cs.NE].

[2] Fredrik Almin. "Detection of Non-Ferrous Materials with Computer Vision". MA thesis. Linköping University, Computer Vision, 2020, p. 61.

[3] Ahmad Arslan, Cary Cooper, Zaheer Khan, Ismail Golgeci, and Imran Ali. "Artificial intelligence and human workers interaction at team level: a conceptual assessment of the challenges and potential HRM strategies." In: *International Journal of Manpower* 43.1 (2022), pp. 75–88. ISSN: 01437720. DOI: 10.1108/IJM-01-2021-0052. URL: https://doi.org/10.1108/IJM-01-2021-0052.

[4] A. Belotserkovsky, P. Lukashevich, M. Doganli, and J. Rabcan. "A Concept of a Multi-robotic System for Warehouse Automation." In: *2021 International Conference on Information and Digital Technologies (IDT), Information and Digital Technologies (IDT), 2021 International Conference on* (2021), pp. 156–161. ISSN: 978-1-6654-3692-2. DOI: 10.1109/IDT52577.2021.9497581. URL: https://doi.org/10.1109/IDT52577.2021.9497581.

[5] Léon Bottou. "Stochastic gradient descent tricks". In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 421–436.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[7] Sebastian Bruch, Xuanhui Wang, Michael Bendersky, and Marc Najork. "An analysis of the softmax cross entropy loss for learning-to-rank with binary relevance". In: *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*. 2019, pp. 75–78.

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. "Unsupervised learning of visual features by contrasting cluster assignments". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9912–9924.

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9650–9660.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[11] European Commission, Joint Research Centre, J Olivier, D Guizzardi, E Schaaf, E Solazzo, M Crippa, E Vignati, M Banja, M Muntean, G Grassi, F Monforti-Ferrario, and S Rossi. *GHG emissions of all world : 2021 report*. Publications Office, 2021. DOI: doi/10.2760/074804.

[12] Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. "What is a good evaluation measure for semantic segmentation?." In: *Bmvc*. Vol. 27. 2013. 2013, pp. 10–5244.

[13] Dengxin Dai, Christos Sakaridis, Simon Hecker, and Luc Van Gool. "Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding". In: *International Journal of Computer Vision* 128.5 (2020), pp. 1182–1204.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[17] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. "The Pascal Visual Object Classes (VOC) Challenge". In: *Int. J. Comput. Vision* 88.2 (June 2010), pp. 303–338. ISSN: 0920-5691. DOI: 10.1007/s11263-009-0275-4. URL: https://doi.org/10.1007/s11263-009-0275-4.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016, pp. 274–279.

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. "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.

[20] Frauke Günther and Stefan Fritsch. "Neuralnet: training of neural networks." In: *R J.* 2.1 (2010), p. 30.

[21] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. "A review of semantic segmentation using deep neural networks". In: *International journal of multimedia information retrieval* 7.2 (2018), pp. 87–93.

[22] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[23] Chih-Hui Ho and Nuno Nvasconcelos. "Contrastive learning with adversarial examples". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17081–17093.

[24] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. "A Survey on Contrastive Self-Supervised Learning". In: *Technologies* 9.1 (2021). ISSN: 2227-7080. DOI: 10.3390/technologies9010002. URL: https://www.mdpi.com/2227-7080/9/1/2.

[25] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2015).

[26] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. "An alternative view: When does SGD escape local minima?" In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2698–2707.

[27] M. Kutila, J. Viitanen, and A. Vattulainen. "Scrap Metal Sorting with Colour Vision and Inductive Sensor Array". In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*. Vol. 2. 2005, pp. 725–729. DOI: 10.1109/CIMCA.2005.1631554.

[28] Omer Levy and Yoav Goldberg. "Dependency-based word embeddings". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2014, pp. 302–308.

[29] P. Lewicki, J. Tochowicz, and J. van Genuchten. "Are Robots Taking Our Jobs? A RoboPlatform at a Bank." In: *IEEE Software, Software, IEEE, IEEE Softw* 36.3 (2019), pp. 101–104. ISSN: 0740-7459. DOI: 10.1109/MS.2019.2897337. URL: https://doi.org/10.1109/MS.2019.2897337.

[30] Xiaoyu Li, Zhenxun Zhuang, and Francesco Orabona. "A second look at exponential and cosine step sizes: Simplicity, adaptivity, and performance". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 6553–6564.

[31] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature Pyramid Networks for Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[33] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. "Self-supervised learning: Generative or contrastive". In: *IEEE Transactions on Knowledge and Data Engineering* (2021).

[34] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[35] Ishan Misra and Laurens van der Maaten. "Self-supervised learning of pretext-invariant representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6707–6717.

[36] *PREVENTING 900,000 TONNES OF CO2*. URL: https://www.stenaaluminium.com/insights/preventing-900000--tons-of-co2/ (visited on 11/25/2021).

[37] Pavlo M Radiuk. "Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets." In: *Information Technology & Management Science (Sciendo)* 20.1 (2017).

[38] Barbara K. Reck and T. E. Graedel. "Challenges in Metal Recycling". In: *Science* 337.6095 (2012), pp. 690–695. DOI: 10.1126/science.1217501. eprint: https://www.science.org/doi/pdf/10.1126/science.1217501. URL: https://www.science.org/doi/abs/10.1126/science.1217501.

[39] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[40] Stuart J. Russell and Peter Norvig. *Artificial Intelligence*. 4th ed. 221 River Street, Hoboken, New Jersey: Pearson, 2020, pp. 760–764.

[41] Nikolai V. Smirnov and Aleksey S. Trifonov. "Deep Learning Methods for Solving Scrap Metal Classification Task". In: *2021 International Russian Automation Conference (RusAutoCon)*. 2021, pp. 221–225. DOI: 10.1109/RusAutoCon52004.2021.9537520.

[42] Daniel Vasicek. "Artificial intelligence and machine learning: Practical aspects of overfitting and regularization". In: *Information Services & Use* 39.4 (2019), pp. 281–289.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[44] Sun-Chong Wang. "Artificial neural network". In: *Interdisciplinary computing in java programming*. Springer, 2003, pp. 81–100.

[45] Kirti Wankhede, Bharati Wukkadada, and Vidhya Nadar. "Just Walk-Out Technology and its Challenges: A Case of Amazon Go." In: *2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Inventive Research in Computing Applications (ICIRCA), 2018 International Conference on* (2018), pp. 254–257. ISSN: 978-1-5386-2456-2. DOI: 10.1109/ICIRCA.2018.8597403. URL: https://doi.org/10.1109/ICIRCA.2018.8597403.

[46] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6.

[47] Thomas Wood. *BerzeLiUs*. URL: https://www.nsc.liu.se/systems/berzelius/ (visited on 03/09/2022).

[48] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. "Unified perceptual parsing for scene understanding". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 418–434.

[49] Yang You, Igor Gitman, and Boris Ginsburg. *Scaling SGD Batch Size to 32K for ImageNet Training*. Tech. rep. UCB/EECS-2017-156. EECS Department, University of California, Berkeley, Sept. 2017. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-156.html.

[50] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. "Barlow Twins: Self-Supervised Learning via Redundancy Reduction". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 12310–12320. URL: https://proceedings.mlr.press/v139/zbontar21a.html.

[51] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. "S4L: Self-Supervised Semi-Supervised Learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[52] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. "Pyramid Scene Parsing Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6230–6239. DOI: 10.1109/CVPR.2017.660.

[53] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. "Rethinking Semantic Segmentation From a Sequence-to-Sequence Perspective With Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 6881–6890.

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. *Semantic segmentation on MIT ADE20K dataset in PyTorch*. 2018.

A Error over time

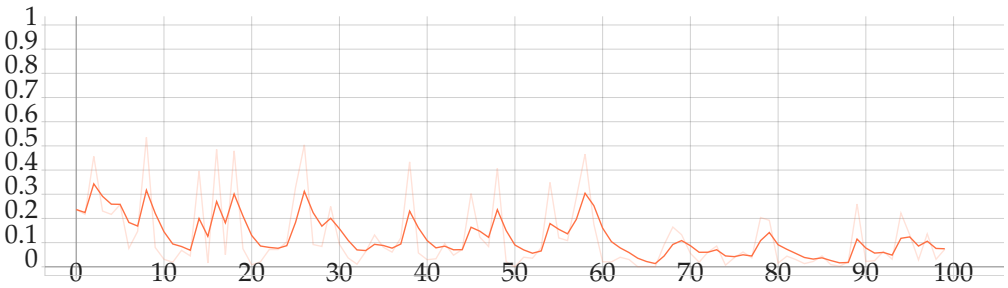

Figure A.1: Error percentage for each downstream epoch for the model pre-trained using SwAV for 3 epochs. The graph is smoothed using an exponential moving average with the value 0.6.

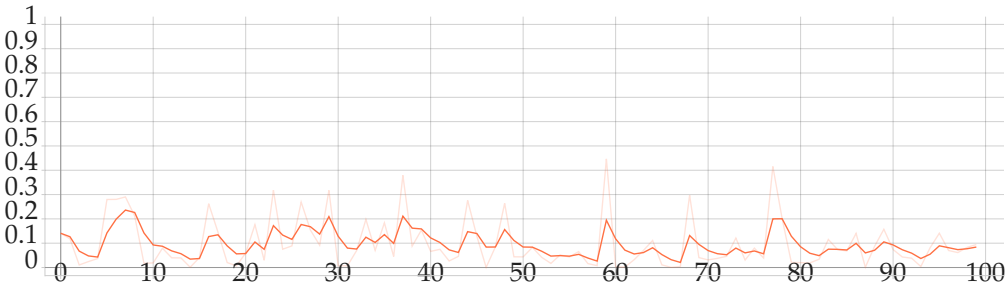

Figure A.2: Error percentage for each downstream epoch for the model pre-trained using Barlow twins for 1 epochs. The graph is smoothed using an exponential moving average with the value 0.6.

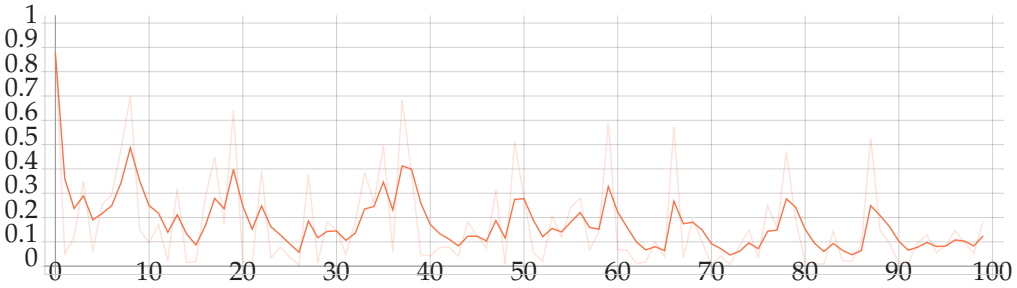

Figure A.3: Error percentage for each downstream epoch for the model pre-trained using DINO for 20 epochs. The graph is smoothed using an exponential moving average with the value 0.6.

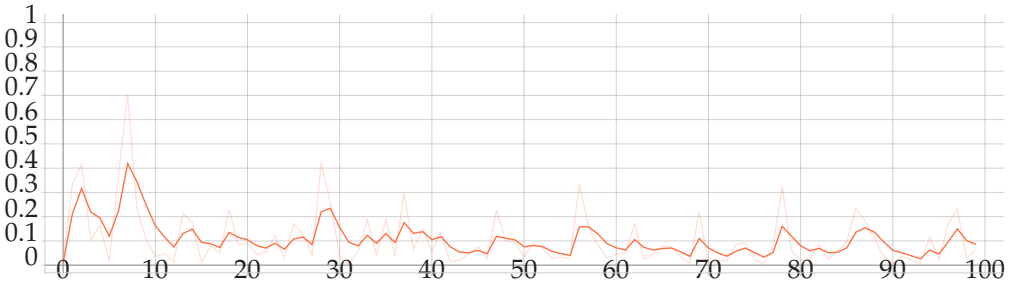

Figure A.4: Error percentage for each downstream epoch for the model not pre-trained. The graph is smoothed using an exponential moving average with the value 0.6.

B IoU

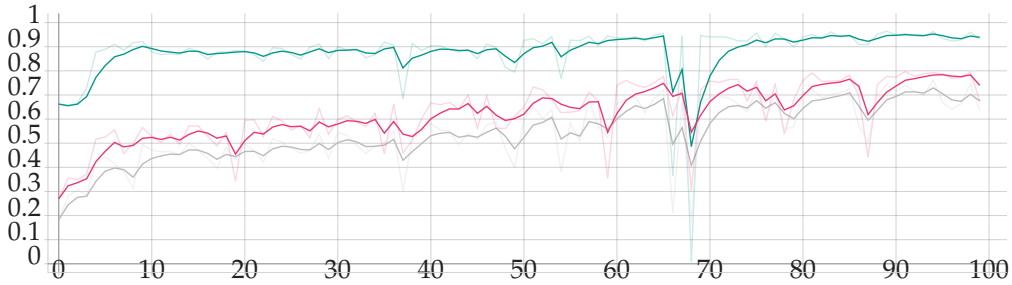

Figure B.1: IoU on the downstream test dataset each epoch for a model pre-trained with DINO for 15 epochs. Green is the IoU for the background, red is for the aluminium and grey is for other material. The graph is smoothed using an exponential moving average with the value 0.6.

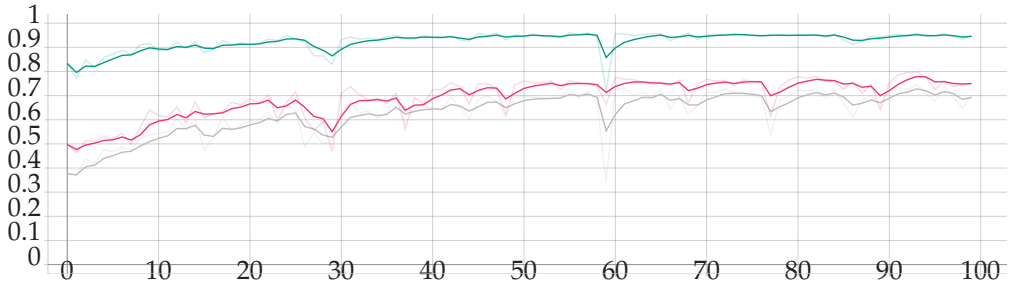

Figure B.2: IoU on the downstream test dataset each epoch for a model pre-trained with Barlow Twins for 10 epochs. Green is the IoU for the background, red is for the aluminium and grey is for other material. The graph is smoothed using an exponential moving average with the value 0.6.

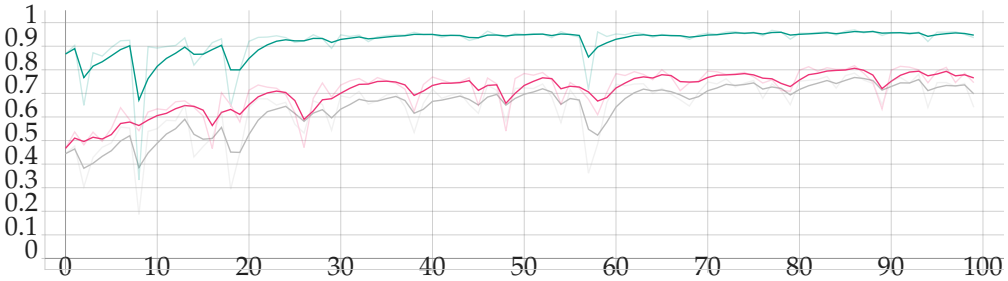

Figure B.3: IoU on the downstream test dataset each epoch for a model pre-trained with SwAV for 20 epochs. Green is the IoU for the background, red is for the aluminium and grey is for other material. The graph is smoothed using an exponential moving average with the value 0.6.

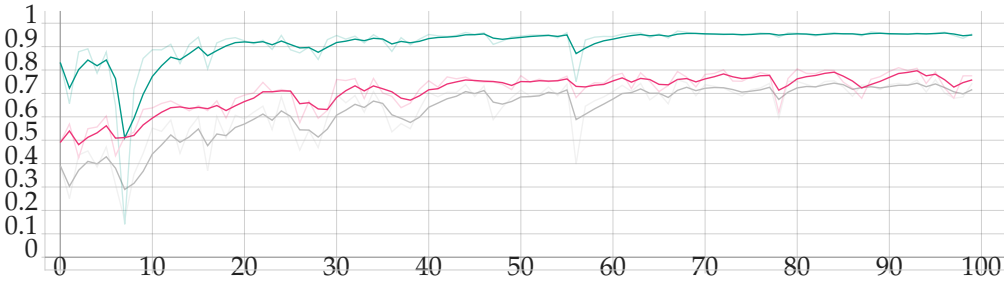

Figure B.4: IoU on the downstream test dataset each epoch for a model with no pre-training. Green is the IoU for the background, red is for the aluminium and grey is for other material. The graph is smoothed using an exponential moving average with the value 0.6.

C Inference

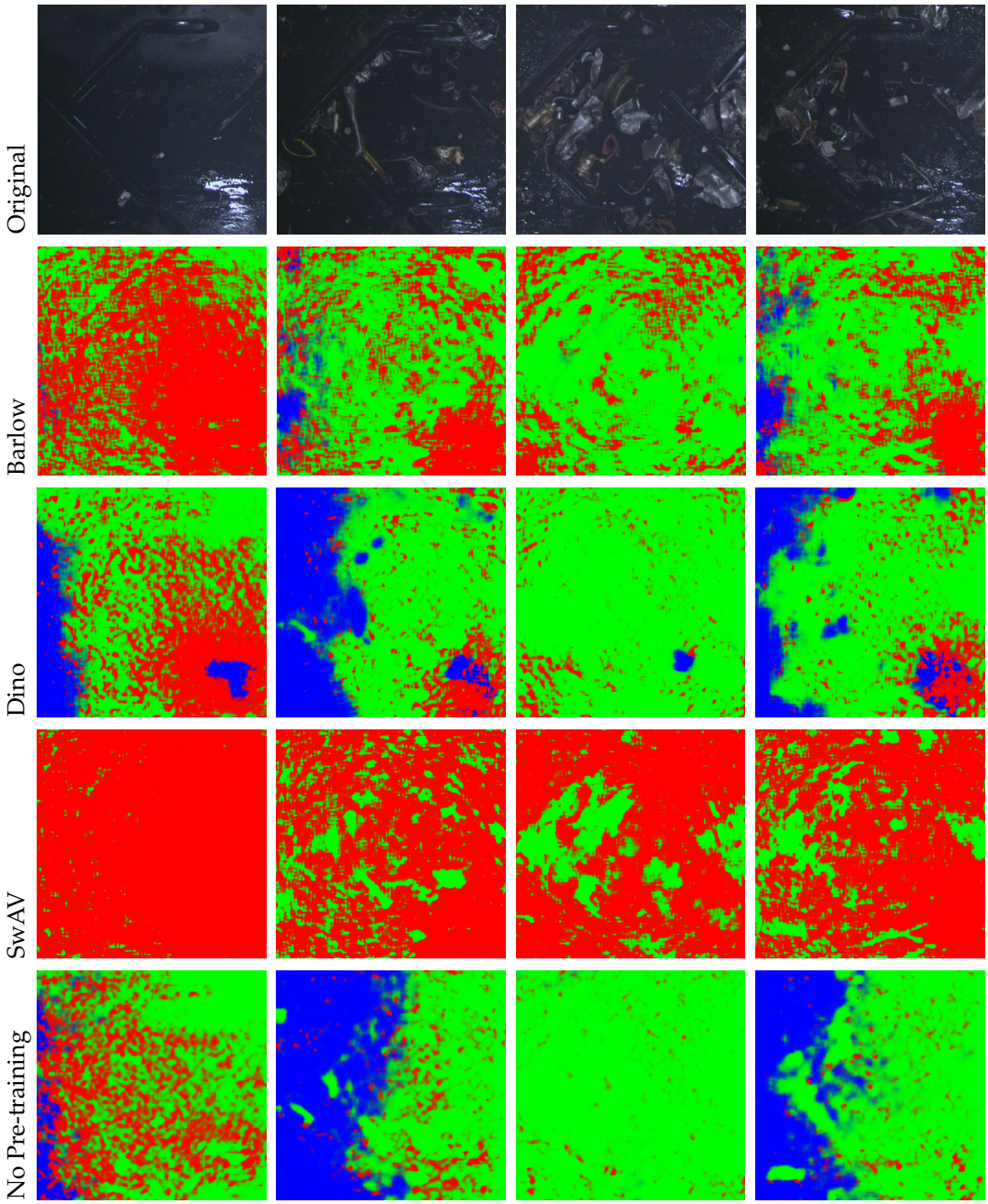

Figure C.1: The first row of images are the 4 selected according to Section 4.7, the first is empty, the second contain a few objects, the third contains large objects and the fourth contains small objects. Each row contain the segmentation produced by a model on those images that have been trained on the downstream tasks with preprocessed images discussed in Section 4.2.2. The following rows contains segmentations of the images in the top row. The second row has the segmentations produced the model pre-trained with the Barlow SSL-framework. The third row is DINO, the fourth is SwAV and the final row contains the predictions of a model that was not pre-trained.