UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

Bachelors Thesis (15 ECTS)

Alexander Mittet (sxn123)

Foundational Model for Endoscopy

Lifting classification accuracy of Ulcerative Colitis mayo endoscopic subscore-score, using self-supervised pre-training on images with Vision Transformers

Date of submission: 10/06-2024

Advisor: Bulat Ibragimov
Co-advisor: Bjørn Leth Møller

Faculty:	Faculty of Science
Department	Department of Computer Science
Author(s):	Alexander Mittet (sxn123)
Title and subtitle:	Foundational Model for Endoscopy: Lifting classification accuracy of Ulcerative Colitis Mayo Endoscopic Subscore, using self-supervised pre-training on images with Vision Transformers
Repository:	https://github.com/alexandermittet/BA

Abstract: Ulcerative Colitis (UC) is a chronic inflammatory bowel disease that significantly impacts quality of life. Endoscopy is the modern standard for diagnosing and monitoring UC using the Mayo Endoscopic Subscore (MES). However, there is high inter-observer variability among doctors in assessing the MES score. This thesis explores using self-supervised pre-training with Vision Transformers and Masked Autoencoders (MAE) on unlabelled endoscopy images to improve accuracy of an MES classification model. A baseline achieved an F1-score 54 on the supervised dataset. An MAE model was then pre-trained on a large unlabelled dataset of 370,000 endoscopy images. The pre-trained model achieved an F1-score of 61-71, depending on hyperparameters. We found a top-1 improvement of 16 ± 0.35 points and an average improvement of 12 ± 3.2 points compared to the baseline, excluding one result lower than baseline. The pre-trained models reduced confusion between the middle MES classes compared to the baseline. Analysis showed the model's mean squared error loss could potentially identify low quality images for removal. Embedding visualization with t-SNE did not reveal clear clustering by MES score. Dimensionality reduction with PCA suggested the model size could be reduced while retaining most variance. The results demonstrate that self-supervised pre-training with MAE can improve MES classification accuracy and efficiency by leveraging unlabelled data. Future directions include exploring alternative pre-training methods, refining the masking strategy, and accounting better for class imbalance.

Advisor:	Bulat Ibragimov
Co-advisor:	Bjørn Leth Møller
Date:	10/06-2024

Table of content

1 Introduction

Ulcerative Colitis (UC) is a chronic inflammatory bowel disease (IBD) that significantly decreases life quality through symptoms such as bloody diarrhoea and abdominal pain. Endoscopy is the modern standard for diagnosing and monitoring UC (Lo et al., 2022). UC IBD is scored with a so-called MES-score of 0 to 3 (Schroeder et al., 1987). Here, one point can be the difference between a simple treatment or amputation of part of the intestine. Therefore, doctors are interested in how other doctors score the same images. Doctors of different seniority unfortunately differ in their analyses of to which extent a patient is ill (Møller et al., 2024). A machine learning model is more consistent in assessing this. This has been done before in Lo et al., 2022. However, the accuracy is insufficient, and raising the image diagnostic success rate using supervised learning, would mean labelling more data, which is expensive. So what about unsupervised learning, where we don't need labelled data? Pre-training using self-supervised learning on endoscopic videos has proven to lift accuracy (Hirsch et al., 2023) and reduce amount of labelled data needed (Zhao Wang et al., 2024). However we want to focus on images instead of video. Therefore this project aims to apply self-supervised learning in the domain of IBD images, thereby aiding doctors in diagnosing IBD.

2 Theory

2.1 IBD and MES-score

Mayo Endoscopic Subscore, MES, is a component of the Mayo score, classifying mucosal inflammation based on a 4-point scale from 0 to 3 according to endoscopic findings (0: normal; 1: erythema, decreased vascular pattern, and mild friability; 2: marked erythema, absent vascular pattern, friability, and erosions; 3: ulceration and spontaneous bleeding)(Schroeder et al., 1987). An example image of each MES-score can be seen in Figure 1 where 4 is the 'Bad Image'-class.

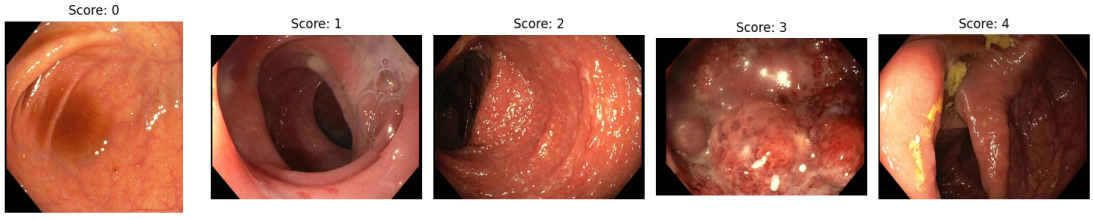

Figure 1: Example images of MES Score 0-3. Score = 4 is used to mark bad images.

2.2 IBD and Machine Learning

We explore the use of machine learning in the IBD domain because the inter-observer variability is very high. Møller et al., 2024 show that doctors score the same images differently. They used Cohen's kappa score of agreement (which ranges from 0 to 1) to quantify how much the doctors agreed on the illness level/MES-score. They found in their research that the kappa score was 0.6 among junior physicians and 0.7 among senior phyisicians, reflecting that there is a moderate level of agreement among physicians. Ideally this kappa score should be close to 1, giving all patients with the same degree of illness the same degree of treatment. This shows the need for a more objective approach, such as a machine learning model.

Using machine learning for IBD has been done before (Lo et al., 2022). However, using supervised learning to increasing accuracy requires more labelled data. Using such a procedure is expensive on resources, since only doctors are qualified to do this. Therefore, we propose a method that doesn't cost expensive doctors hours: Pre-training the machine learning model using self-supervised learning.

Pre-training models on unlabelled data using self-supervised learning is a promising approach because it can leverage the abundant amount of unlabelled data. Research has proven to raise accuracy and reduce the amount of labelled data needed for subsequent supervised learning/fine-tuning (Hirsch et al., 2023; Erhan et al., 2010; Yosinski et al., 2014; Devlin et al., 2018). Simply recording the endoscopic video feed from a doctor's consultation saves a lot of still images for us to use in our pre-training. This makes the data collection almost free. If effective, this approach could not only improve diagnostic accuracy and ensure that the correct patients receive the appropriate procedures, but also reduce unnecessary interventions for those who do not require treatment. Ultimately, this methodology has potential to enhance patient care while reducing costs and patient discomfort.

2.3 Review Existing Pre-Training Methods

This section will review and compare several popular self-supervised pre-training methods. We will cover the key principles, advantages, and trade-offs of approaches like Masked Image Modelling (MIM), contrastive learning, and distillation methods. Understanding the strengths and limitations of these techniques will inform our choice of strategy for the proposed method in this work. We will take a closer look at MAE in Section 2.5 as this ends up being our proposed model.

Recent years of machine learning research have witnessed a good amount of progress in self-supervised visual representation learning. With Transformer-based models pre-trained on

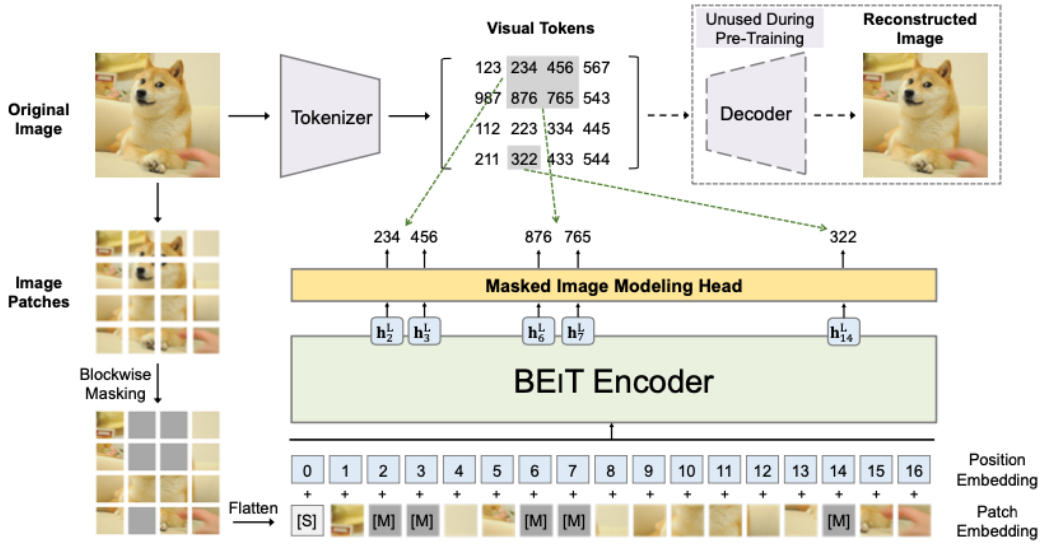

Figure 2: "Overview of BEiT pre-training. Before pre-training, we learn an "image tokenizer" via autoencoding-style reconstruction, where an image is tokenized into discrete visual tokens according to the learned vocabulary. During pre-training, each image has two views, i.e., image patches, and visual tokens. We randomly mask some proportion of image patches (gray patches in the figure) and replace them with a special mask embedding [M]. Then the patches are fed to a backbone vision Transformer. The pre-training task aims at predicting the visual tokens of the *original* image based on the encoding vectors of the *corrupted* image. " from Bao et al., 2021

large unlabelled datasets, these methods learn visual representations that can be transferred to downstream tasks like classification and medical image analysis.

Popular self-supervised methods include Masked Image Modeling (H. Wang et al., 2023) where the model learns to reconstruct masked portions of input images. Methods like BEiT(Bao et al., 2021), SimMIM (Xie et al., 2022) and Masked Autoencoders (MAE) (He, X. Chen, et al., 2021) all do the job, but in different ways.

BEiT uses a pre-trained discrete Variational Auto Encoder. In Figure 2 we see that an image of a dog is run into two pipelines: First to the right through a tokenizer, and also down through a patching and encoding structure. It then compares and minimizes the loss between the encoder's prediction and the visual tokens. In this way, it doesn't use the decoder during pre-training.

SimMIM doesn't use a tokenizer. It focuses on directly predicting the pixel values of masked patches using a very lightweight head/decoder, often a single linear layer. MAE, on the other hand, aims to reconstruct the image using a heavier decoder. It is possible to do SimMIM with CNNs/convnets but you cannot do MAE. This is since MAE works by chopping up

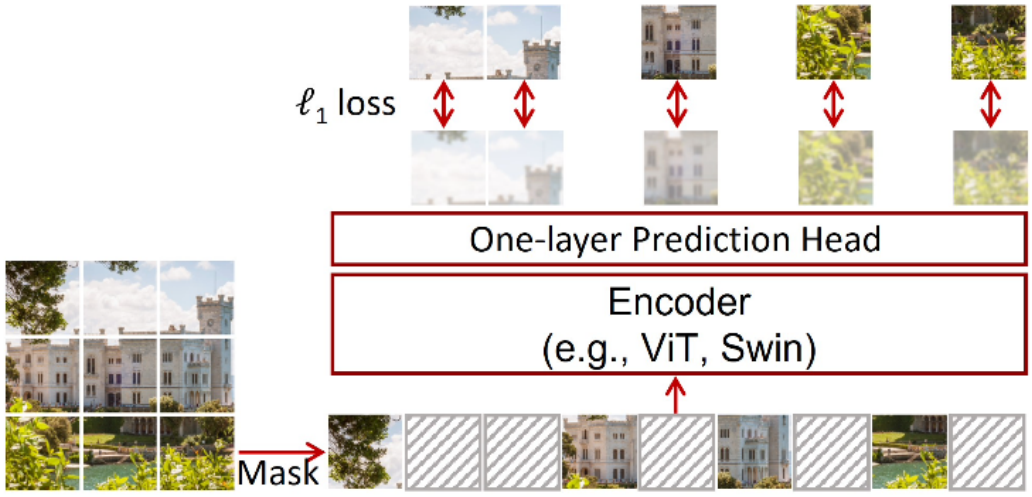

Figure 3: "An illustration of our simple framework for masked language modeling, named *SimMIM*. It predicts raw pixel values of the randomly masked patches by a lightweight one-layer head, and performs learning using a simple ℓ_1 loss " from Xie et al., 2022

parts of the images and generalizing. But you cannot chop up an input for a CNNs sliding kernel. MAE reconstructs the original image pixels for masked patches. This allows us to get a grasp of its quality, simply by looking at the reconstructed images.

Apart from masking, there also exists contrastive methods such as SimCLR (T. Chen et al., 2020), MoCO (He, Fan, et al., 2020), and Bootstrap Your Own Latent (BYOL) (Grill et al., 2020). These learn by bringing positive pairs of augmented images closer together in embedding space, while pushing negative pair apart. Other techniques like self-**di**stillation with **no** labels (DINO) (Caron et al., 2021) are based on distillation principles. However some of these methods have an implicit bias towards uniform class balanced data as discussed in Assran, Balestriero, et al., 2022.

DINO uses a student-teacher framework, where a student Vision Transformer (ViT) is trained to match the output of a teacher ViT, whose parameters are an exponential moving average of the student's. It is trained by feeding different augmented versions of the same image into the networks. The student is then optimized to produce consistent representations of the teacher. This learns transferable visual representations that can be fine-tuned for downstream tasks.

MAEs are a self-supervised learning method for pre-training ViTs on images. It masks (removes) a large amount (e.g. 75%) of patches away the image randomly. This forces the model to learn high-level semantic information to correctly reconstruct the input image rather than relying on low-level details. The encoder only operates on the visible subset of patches. This means we only send a small amount (e.g. 25%) of data through the model,

Figure 4: "Self-attention from a Vision Transformer with 8×8 patches trained with no supervision. We look at the self-attention of the [CLS] token on the heads of the last layer. This token is not attached to any label nor supervision. These maps show that the model automatically learns class-specific features leading to unsupervised object segmentations." from Caron et al., 2021.

reducing computation, enabling using a larger model. The smaller decoder reconstructs the original image from the latent representation plus mask tokens indicating the coordinates of the missing patches. Similar to how language models predict missing words like BERT (Devlin et al., 2018), the MAE decoder predicts missing patches in the pixel space. After pre-training, the decoder is removed, and the encoder is fine-tuned for a downstream task (He, X. Chen, et al., 2021). One key aspect of a MAE, is that we can look at the reconstructed images and qualitatively determine how good it is, as seen in Figure 8.

Self-supervised pre-training methods like MIM, contrastive learning and distillation have advanced visual representation learning from unlabelled data. Among these MAEs are promising for their ability to reconstruct images in a qualitatively interpretable manner, meanwhile being computationally efficient.

It should also be mentioned that most self-supervised methods assume uniform data class distribution. This is most likely not the case in the real world as discussed in Assran, Balestriero, et al., 2022.

2.4 What is a Vision Transformer?

Before diving into the MAE, let's review the architecture used for its encoder: The Vision Transformer.

The Transformer architecture, originally made for Natural Language Processing (NLP) proposed in the infamous paper "Attention is All you Need" (Vaswani et al., 2017), has been adapted for vision giving rise to the Vision Transformer. In a ViT the input image is first divided into patches, often 16x16 pixels. Each patch is flattened and linearly projected into vector space using a linear layer. These are called patch embeddings. To keep position

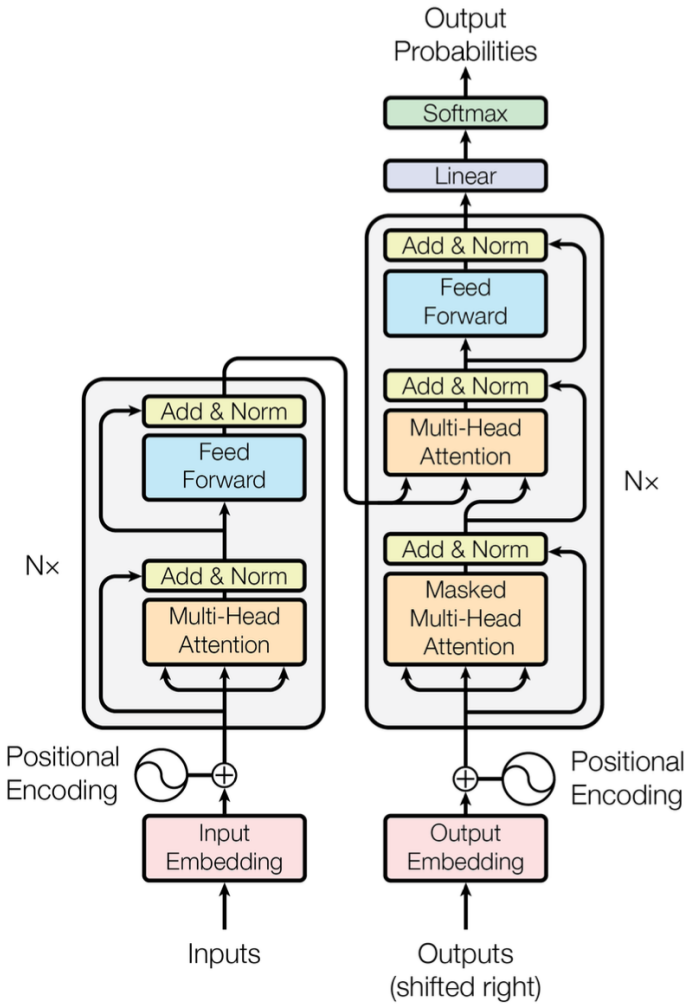

Figure 5: "The Transformer - model architecture" from Vaswani et al., 2017.

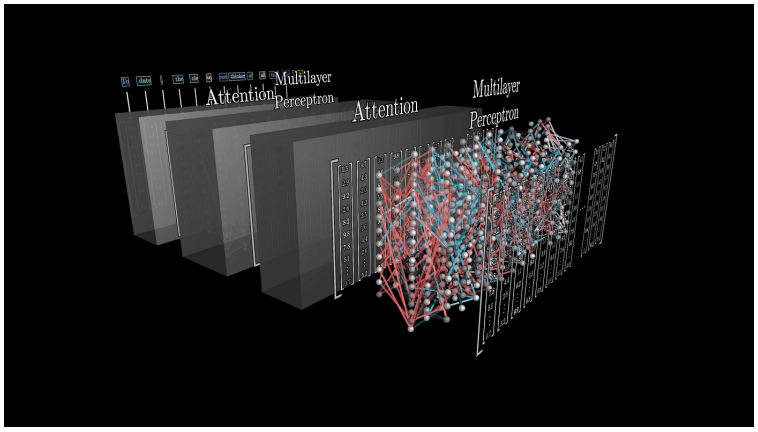

Figure 6: Visualisation of the alternating layers of Attention layers and Linear (Multilayer perceptron) layers (3Blue1Brown, 2022).

information, position embeddings are learned into the patch embeddings. This results in a sequence fed into alternating multi-head attention layers and linear layers as in Figure 6. The output of the Transformer is finally passed to a head. For classification this is usually a linear layer.

The self-attention mechanism is a key part of the architecture. Unlike Recurrent Neural Networks (RNNs) that process word-tokens sequentially, self-attention enables our Transformer to consider relationships between all word-tokens/image-patches at the same time/in parallel. For each token/patch, the Transformer calculates attention scores, meaning how much focus to place on every other token/patch, when encoding the current. This allows the Transformer to capture long-range dependencies and learn context well.

Unlike Convolutional Neural Networks (CNNs) which have been dominant in the space, thus, ViTs treat an image classification task similarly to an NLP problem.

ViTs have achieved a good performance on image classification benchmarks, coming close and often surpassing close to state-of-the-art (SOTA) CNNs while using much less computation to train the model (Dosovitskiy et al., 2020). ViTs are flexible and scalable. Both in standard supervised training, and when pre-trained on massive unlabelled datasets, and fine tuning them afterwards for a downstream task (Bommasani et al., 2021).

2.5 What is a Masked AutoEncoder?

An MAE (He, X. Chen, et al., 2021) is a type of autoencoder (Rumelhart et al., 1986) that reconstructs masked images, visualized in Figure 8. It has shown promising results in learning useful visual representations from unlabelled data. The key idea behind MAEs is to randomly mask out/remove a large portion of the input image (e.g. 75%) and train

the model to reconstruct the image. This encourages the model to learn strong features that capture high-level/holistic information about the image in order to repaint the missing regions.

The architecture of an MAE consists of an assymetric encoder-decoder, meaning that the encoder is often 3 to 4 times larger than the decoder. The encoder takes a small amount (e.g. 25%) of still visible patches. This is the input. It then maps them to latent representations. Mask tokens are then added, to tell the model where those visible patches are in the image, implicitly telling the model what needs to be reconstructed. The lightweight decoder then attempts to reconstruct the original image from the latent representation and mask tokens. However, the decoder is only used for pre-training, and is discarded afterwards. The encoder is then kept and fine-tuned for downstream tasks like classification. One thing that is worth highlighting about MAE, is that pixels are the target of the reconstruction. This means a seperate tokenizer is not needed, like in BEiT (Bao et al., 2021) or I-JEPA (Assran, Duval, et al., 2023). Only the patches are used as tokens.

Balestriero et al., 2024 critisizes standard reconstruction-based learning, where the goal only is to reconstruct the input accurately. This often misaligns with what is wanted for a perception task such as IBD classification. In other words the features that captures the most pixel-level details (and help the most with reconstruction) are usually the parts of the image that are least semantically informative. The paper shows that adding noise can help these issues to some extent. While adding gaussian noise doesn't help, masking seems to be beneficial. This is because the masking forces the model to lear/grasp a more global understanding of the images, which is much better than just focusing on low-level details (Balestriero et al., 2024).

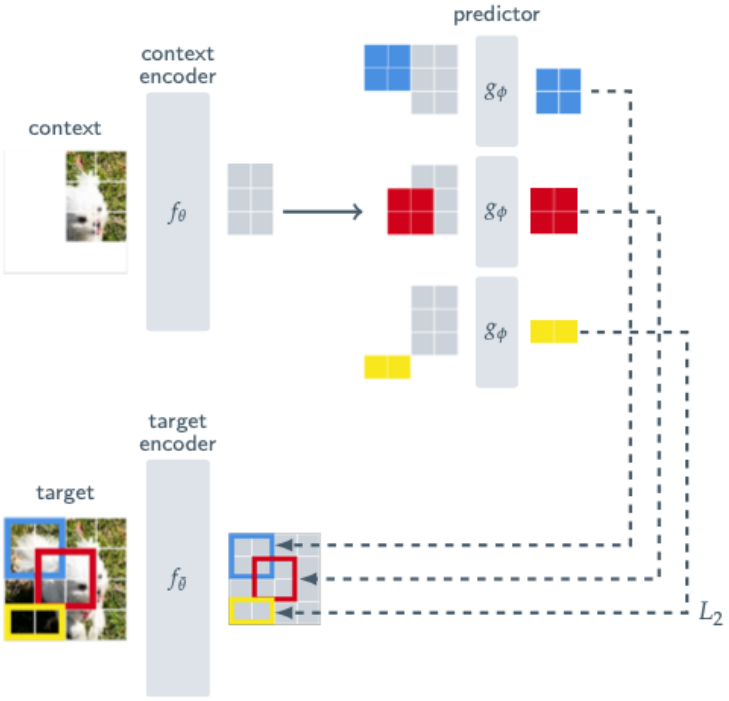

Figure 7: "I-JEPA. The Image-based Joint-Embedding Predictive Architecture uses a single context block to predict the representations of various target blocks originating from the same image. The context encoder is a Vision Transformer (ViT), which only processes the visible context patches. The predictor is a narrow ViT that takes the context encoder output and, conditioned on positional tokens (shown in color), predicts the representations of a target block at a specific location. The target representations correspond to the outputs of the target-encoder, the weights of which are updated at each iteration via an exponential moving average of the context encoder weights." From Assran, Duval, et al., 2023.

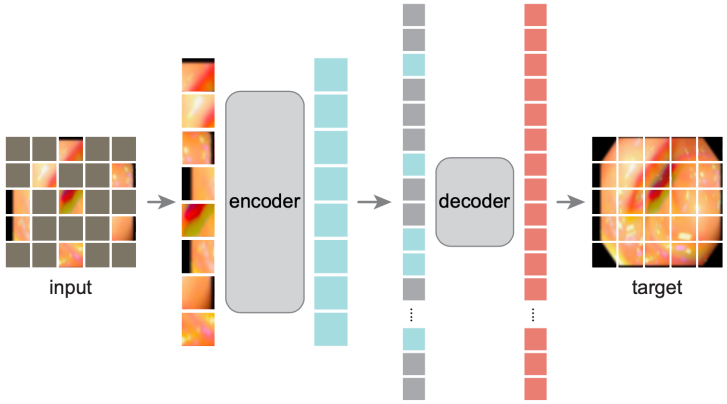

Figure 8: MAE architecture. During pre-training, a large random subset of image patches (e.g., 75%) is masked out. The encoder is applied to the small subset of visible patches. Mask tokens are introduced after the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks (He, X. Chen, et al., 2021).

3 Methodology

This section describes the methodology used for our experiments on developing a ViT classifier for UC IBD MES-score on endoscopy images.

We first establish a baseline by training a ViT classifier on the supervised dataset and evaluate its performance. Next an unsupervised dataset will be created by extracting frames from IBD endoscopy videoes (Zhao Wang et al., 2024) and assessing their sharpness. After that, an MAE will be pre-trained on these extracted frames with balanced representation of sharpness. Next, the decoder is discarded, leaving a ViT classifier. Finally, this will be fine-tuned and evaluated on our supervised dataset.

3.1 Dataset

3.1.1 Supervised endoscopy dataset - MES

We use the Lo et al., 2022 dataset for supervised training and fine-tuning. It consists of labelled images with the MES-score distribution seen in Table 1. Our supervisor gave us this dataset, which was created at Hvidovre Hospital to train a model for MES-scoring UC IBD patients. The train/test split is 80%/20%. The train set is then split again into train/-validation at 80%/20%. This results in a final train/test/validation split of 64%/20%/16%. We stratify these to make sure each set contains the same balance of classes (MES scores).

Table 1: Distribution of Data

MES-score	Count
Bad image	823
0.0	822
1.0	452
2.0	351
3.0	113
Total	**2561**

3.1.2 Unsupervised endoscopy dataset

The unsupervised dataset was created by us. It was created by extracting every 5th frame from approximately 20GB of endoscopy videos (Zhao Wang et al., 2024). We then subsequently calculated the sharpness using a Laplacian score (OpenCV, 2023) of each image, placing them into 21 buckets with their sharpness values. The Laplacian operator is a filter used in image processing for edge detection and sharpness calculation (Gonzalez et al., 2007). Since the sharpness of our images follows a normal distribution with $\mu = 10.000, \sigma \approx 4.000$, we decided sharpness was a good measure to balance our dataset. We use "WeightedRandomSampler" to rebalance the dataset classes. These first 20 buckets with steps of 1000. After this, since there was still some images with a higher sharpness score, we created Bucket20+, with sharpness values above 20.000. These will be used later to balance the different types of images on which the model is pre-trained. In total we have 370.000 images in this unsupervised dataset. To gain an unbiased performance estimate, we have to ensure not to have images from the same videos in both training and validation sets. Although frames from the same video are different, consecutive frames may appear very similar, potentially contaminating the train/validation split and introducing bias.

3.2 Model Hyperparameters

We use a standard ViT architecture for our classifier. We train our baseline classifier with the architecture specified in Table 2 and hyperparameters seen in Table 5. We pre-train our MAE with the architecture seen in Table 2 and 3 and hyperparameters seen in Table 4.

Table 2: Architecture Hyperparameters for the Encoder and Classifier

Hyperparameter	Value
image_size	224x224
patch_size	16x16
emb_dim	192
num_layer (encoder)	12
num_head (encoder)	4

Table 3: Architecture Hyperparameters for the Decoder

Hyperparameter	Value
image_size	224x224
patch_size	16x16
emb_dim	192
num_layer	4
num_head	3

Table 4: Training Hyperparameters

Hyperparameter	Values
Supervised Training/Pre-Training epochs	50/33
Fine-Tuning epochs	5
Optimizer	Prodigy, Adam
Learning rate (Adam)	0.001
Batch size	32, 64, 128, 512
Mask ratio	0.5, 0.75

3.3 Training

3.3.1 Baseline

First we trained a baseline ViT classifier for 50 epochs on only the supervised dataset to get a baseline accuracy. We trained both including and excluding the "Bad image" class. However, we only found reliable results, excluding it from training. Therefore, our dataset was reduced to 1738 labelled images.
We chose the number of embedding dimensions to be 192, resulting in a model size of 5.5M parameter, equivalent to timm_tiny, enabling high accuracies while being runnable on our local machine.

Table 5: Baseline Hyperparameters

Hyperparameter	Values
Training epochs	50
optimizer	prodigy
batch_size	32
mask_ratio	0.5, 0.75

We use Cross Entropy Loss for our supervised training, defined in Equation 1, where y is ground truth, $\hat{y}$ is the predicted from the i-th class over C classes.

$$\mathcal{L}(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{1}$$

However, we are doing machine learning, and we have the label for what class a given image is. This means the probability of the image being any other class than our label, is zero. Therefore, we can simplify the loss as seen in Equation 2

$$\mathcal{L}(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) = -\log(\hat{y}_{label_i}) \tag{2}$$

3.3.2 Pre-Trained model from MAE

We trained our MAE on the 370K images, manually sweeping over the given hyperparameters in Table 4. We used part of the MAE implementation by IcarusWizard, 2023 of He, X. Chen, et al., 2021, incorporating it into our own training system. We tried both masking out 75% as He, X. Chen, et al., 2021 suggests, and 50%, which yielded similar quantitative results, but 50% yielded more consistent qualitative results. This is because the 75% would often

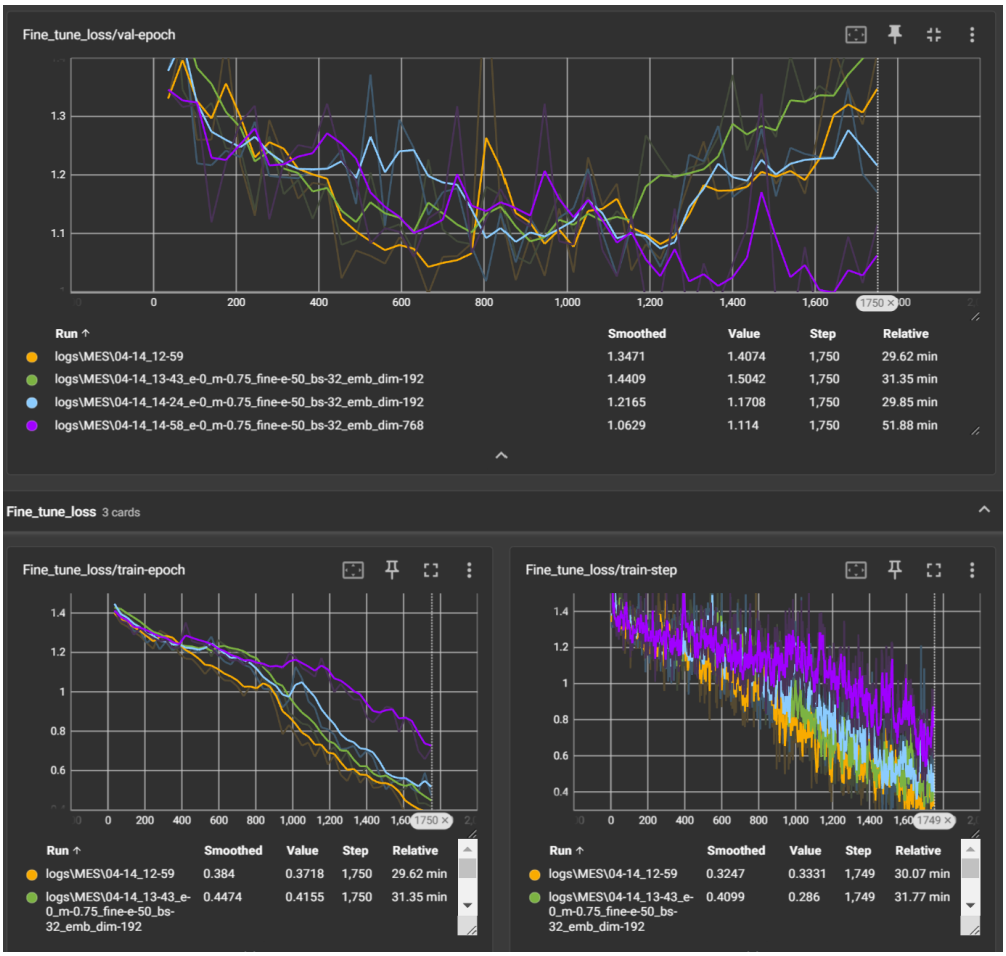

Figure 9: Training loss curves for a bunch of baseline classifier training runs. The top shows validation loss over training epochs. The bottom left shows training loss over epochs, and the bottom right shows training loss over every training step. The first three runs have embedding dimensions = 192, and one run (purple) has embedding dimensions = 768 to match the model size of timm_vit_base. We see that the larger model learns slower and tends to overfit less.

mask out important parts of the image, such as the center, where you can see the most of the bowel. See Figure 10.

We tried both the Adam and Prodigy optimizers, and found Prodigy to perform better since we didn't have time or compute to hand-tune the Adam for every combination of hyperparameters of Table 4. Using Prodigy vs. Adam is discussed in Section 3.3.4. None of the batch sizes in Table 4 really stood out. However, due to computational limitations, we focused on 32 and 64 because of their smaller size and larger number of steps we can train on a GPU in a single day. After the pre-training, we fine-tune the encoder of the MAE to be our classifier for 5 epochs using batch size 32. We use a custom mean squared error (MSE) loss for the MAE, because we have to take the masking into account. The loss is defined in

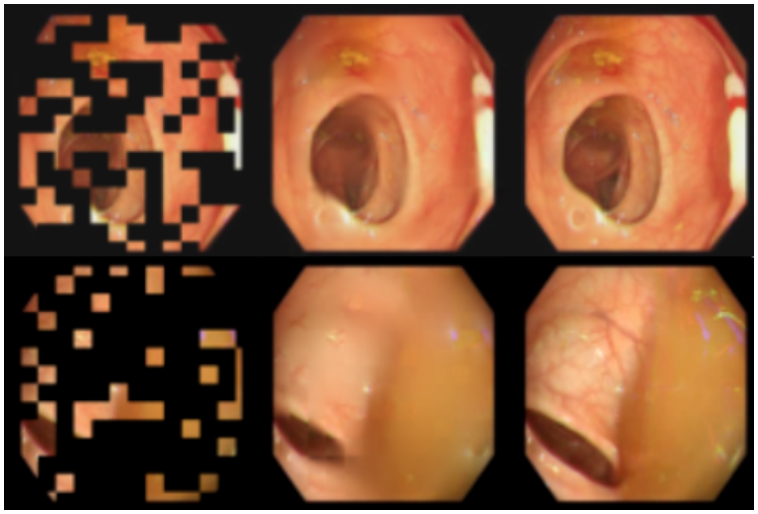

Figure 10: The left is input image (masked), the middle is model prediction, and the right is ground truth. Masking out 50% (top) vs 75% (bottom), we see the model has a harder time reconstructing the bottom middle image and its most detailed parts at 75% masking.

Equation 3

$$\mathcal{L}(y, \hat{y}) = \left(\frac{\mathbb{E}\left[(\hat{\mathbf{y}} - \mathbf{y})^2 \cdot \mathbf{M} \right]}{\text{mask\_ratio}} \right) \tag{3}$$

Here $\mathbf{y}$ is the original image, $\hat{\mathbf{y}}$ is the predicted image, $\mathbf{M}$ is the mask, and mask_ratio is how much we are masking out (from 0 to 1).

3.3.3 Pre-Trained model from timm

Just out of curiosity, we also fine-tuned a bunch of timm (Torch IMage Models) models (Wightman, 2019). These were pre-trained on ImageNet21k (Ridnik et al., 2021) as seen in Dosovitskiy et al., 2020 and after that, is fine-tuned for 5 epochs. We train the models from Table 7.

Table 6: timm Fine-tuned for 5 Epochs with Batch Size 32

Model Name	Model Size (Num params)
vit_tiny_patch16_224	5.5 M
vit_small_patch16_224	21.6 M
vit_base_patch16_224	85.8 M
vit_large_patch16_224	303.3 M

Table 7: List of timm models fine-tuned for 5 epochs. They all contain 12 layers in the Transformer. The difference is the width of the layers.

3.3.4 What is the prodigy optimizer and why is there no learning rate?

The Prodigy optimizer is a deep learning optimizer devised to tackle the challenge of learning rate estimation in adaptive optimization methods like Adagrad (Duchi et al., 2011) and Adam (Kingma et al., 2014). Unlike conventional approaches requiring manual learning rate tuning, Prodigy aims to autonomously determine the appropriate learning rate without explicit configuration.
Experimental findings in (Mishchenko et al., 2023) consistently demonstrate Prodigy's superiority over D-Adaptation (Defazio et al., 2023), achieving test accuracy levels comparable to hand-tuned Adam.

The idea behind Prodigy is to estimate the distance to the optimal solution, denoted as D, which is important for setting the learning rate optimally. Prodigy does this by employing two techniques: Prodigy and Resetting. These techniques are designed to provably estimate D and improve upon the convergence rate of D-Adaptation, a method for learning-rate-free and scheduler-free learning. For a deeper discussion see Mishchenko et al., 2023.

3.4 Evaluation

We evaluate the baseline models by calculating their F1-score and confusion matrices on the test set of 348 test images. We evaluate each pre-trained model, first with SSIM score. Next by fine-tuning them on the supervised dataset for 5 epochs and then calculating their F1-score and confusion matrix on the same test set of 348 test images. We fine-tune each model 3 times to more accurately assess their performance, abiding the central limit theorem.

4 Results

Baseline

Embedding dimensions	Optimizer	Num params	F1-score ($\mu \pm \sigma$)
192	Prodigy	5.5M	54 $\pm$6.6
768	Prodigy	74M	55 $\pm$0.7

Table 8: Baseline models trained on supervised dataset for 50 epochs.

timm

timm model	Optimizer	Num params in model	F1-score ($\mu \pm \sigma$)
timm_tiny_vit_16_224	Prodigy	5.5M	83 ±2.7
timm_small_vit_16_224	Prodigy	21.6M	84 ±1.0
timm_base_vit_16_224	Prodigy	85.8M	83 ±3.1
timm_large_vit_16_224	Prodigy	303M	84 ±1.0

Table 9: timm-models pre-trained on IN21k (number of epochs is unavailable), fine-tuned on supervised dataset for 5 epochs.

Our Method

Batch Size	Optimizer	Masking	Num params	F1-score ($\mu \pm \sigma$)
64	Prodigy	0.5	5.5M	71.0 ± 0.0
32	Prodigy	0.75	5.5M	70.0 ± 4.7
32	Prodigy	0.5	5.5M	67.0 ± 5.5
64	AdamW	0.5	5.5M	66.0 ± 3.5
128	Prodigy	0.5	5.5M	66.0 ± 3.8
512	Prodigy	0.5	5.5M	61.0 ± 0.0
64	Adam	0.5	5.5M	38.0 ± 3.0

Table 10: Unsupervised pre-training for 33 epochs, supervised fine-tuning for 5 epochs. Sorted by mean F1-score.

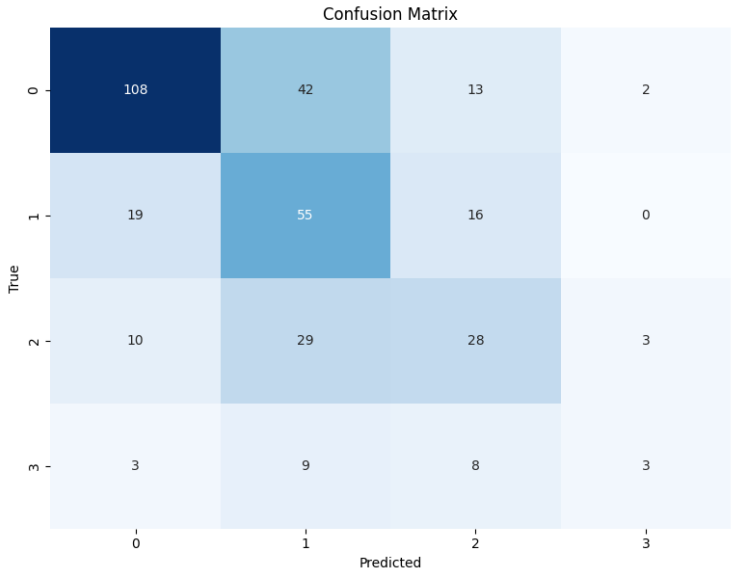

Figure 11: Baseline model test confusion matrix for MES-scores 0 to 3.

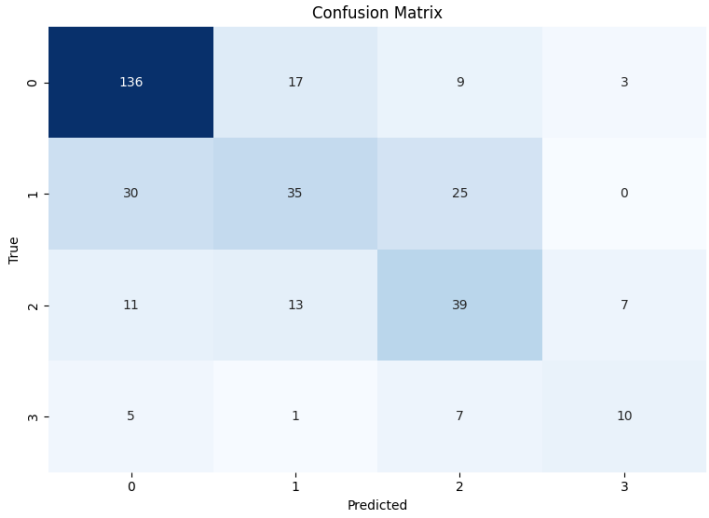

Figure 12: Our method, fine-tuned MAE model confusion matrix for MES-scores 0 to 3. We see reduced confusion around the middle classes (1 and 2).

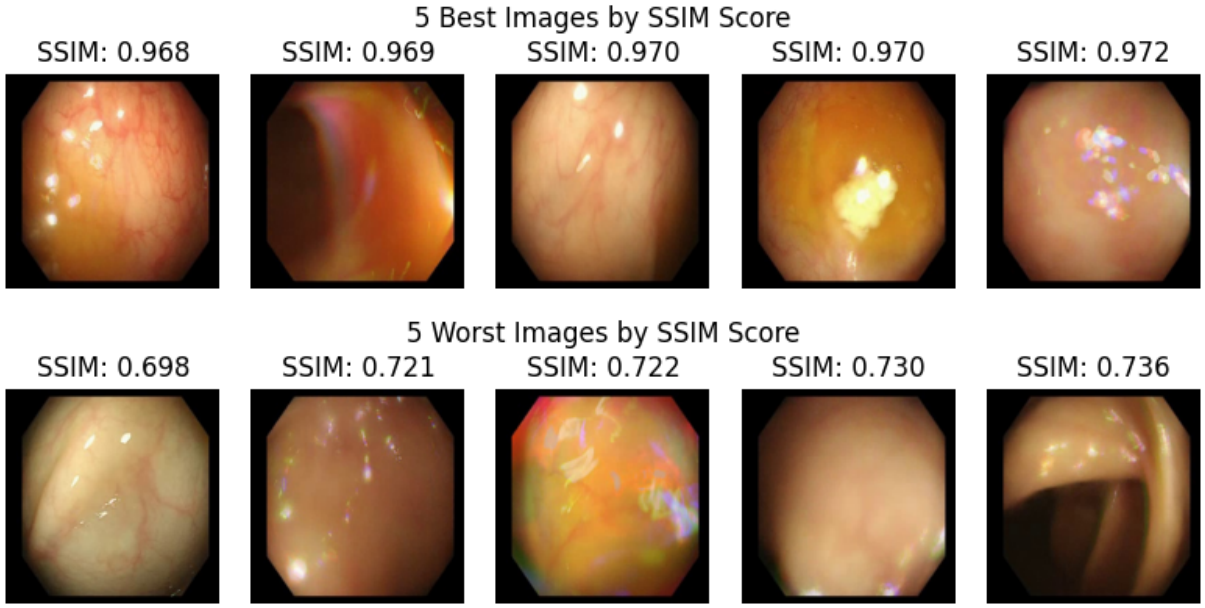

Figure 13: The top 5 best and worst image reconstructions by the MAE model, based on Structural Similarity Index Measure (SSIM) scores. It is unclear what makes the bottom row harder to reconstruct than the top row.

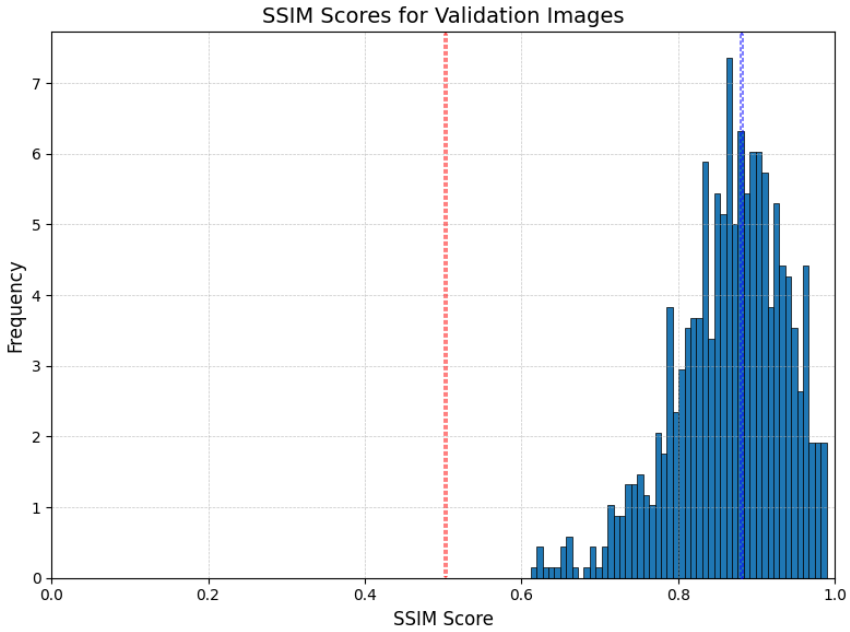

Figure 14: Histogram of MAE reconstruction SSIM scores on a subset of validation images. The red line shows the average SSIM score of images reconstructed using the average pixel value blur i Photoshop, i.e., performs no reconstruction. The blue dotted line represents the MAEs mean SSIM score of 0.88.

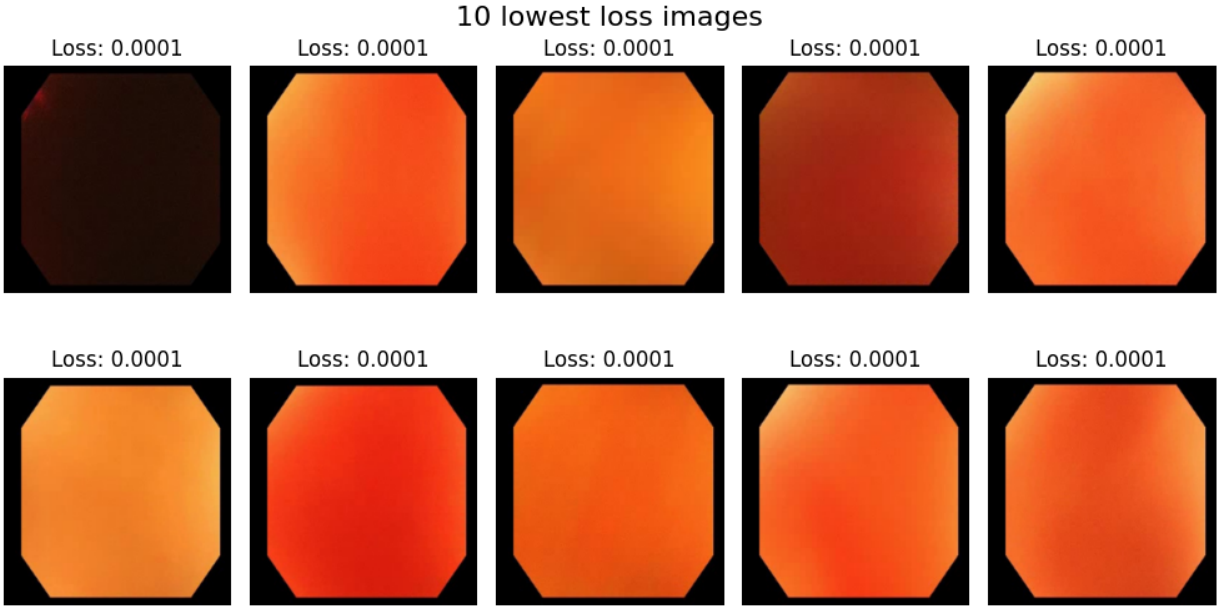

Figure 15: The 10 images with the lowest MSE loss, as reconstructed by the MAE model. We see these are all blurry and unusable for diagnosing and training.

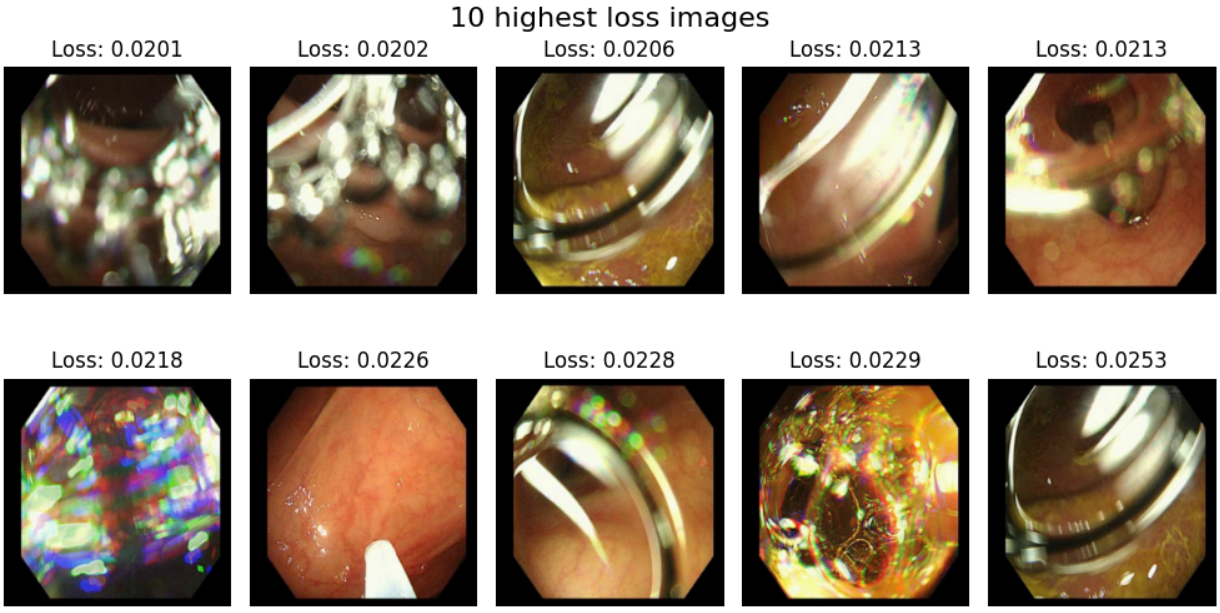

Figure 16: The 10 images with the highest MSE loss, as reconstructed by the MAE model. We see that these all contain reflections or doctors instruments obscuring the bowel. Therefore, these are unusable for diagnosing and training.

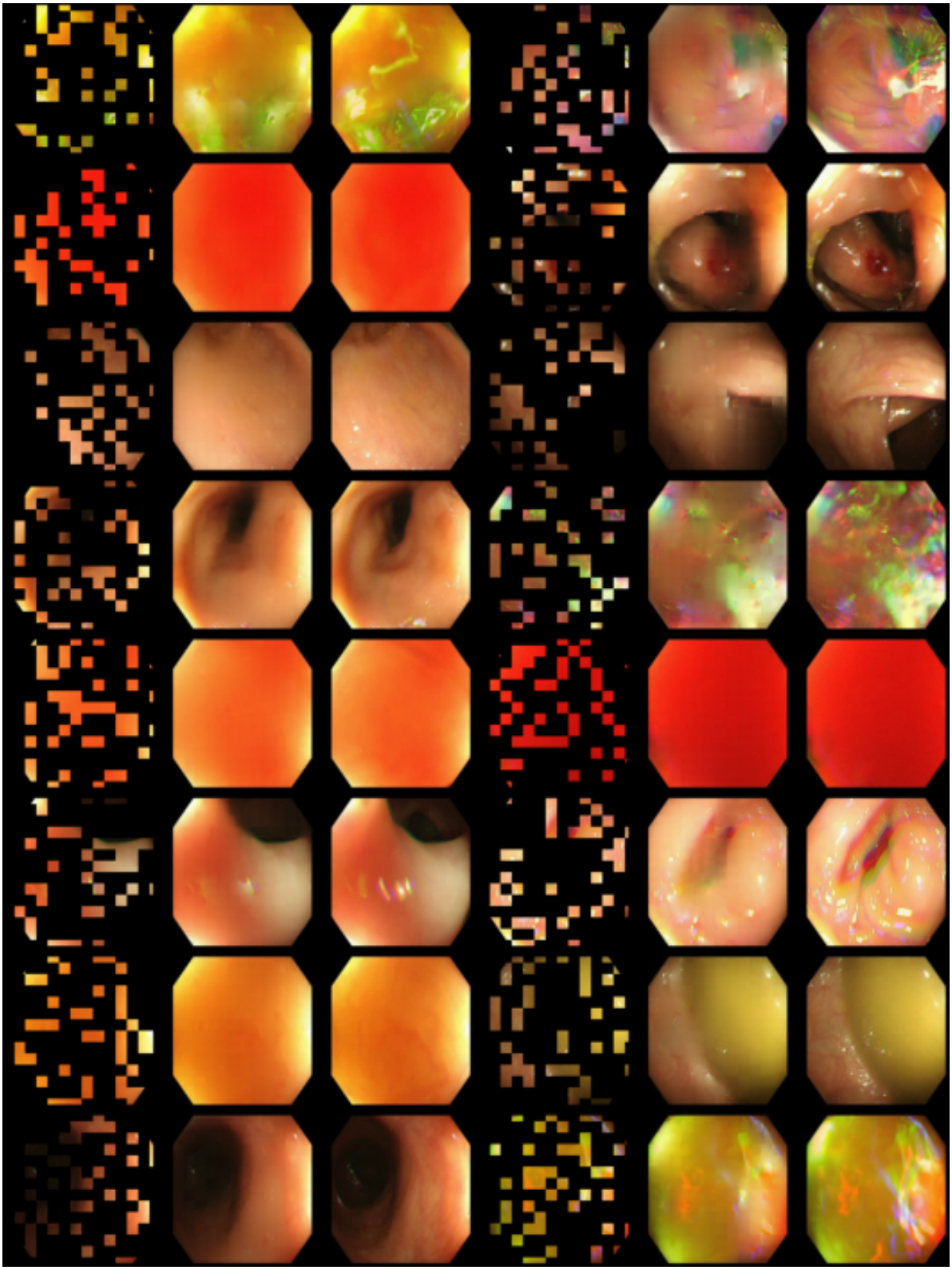

Figure 17: 16 validation image reconstructions by the MAE model after 33 epochs of pre-training. The left is masked input, the middle is prediction, and the right is the ground truth. We see that MAE reconstructs the images well in most scenarios. However, it has a hard time when the middle of the image is heavily masked. E.g. the image to the right, 3 from the bottom.

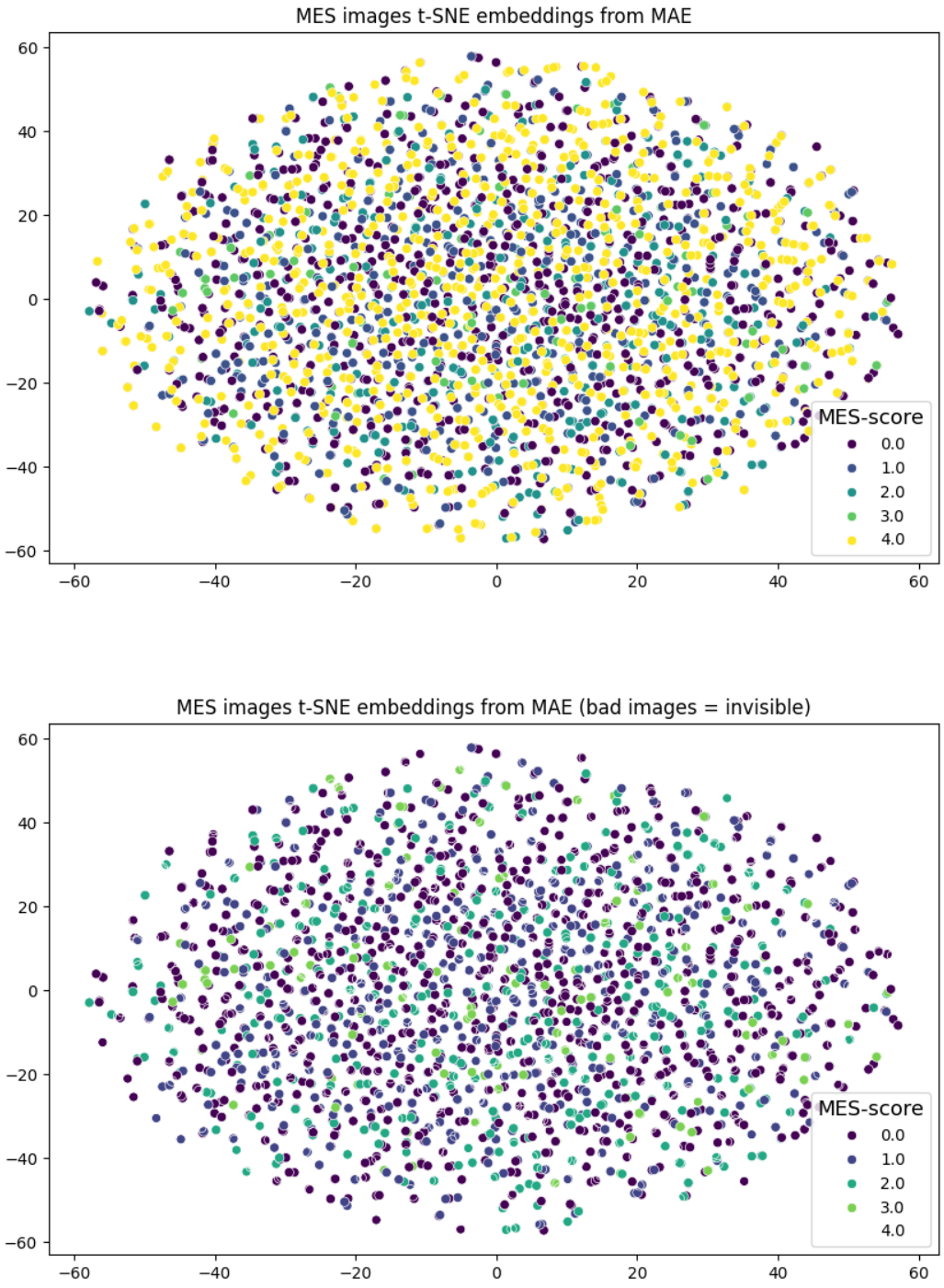

Figure 18: t-SNE on MAE-encoder embeddings of the supervised dataset, with (top) and without (bottom) the 'bad image' class (MES-score = 4). Ideally we would see good separation/clustering for each MES-score class.

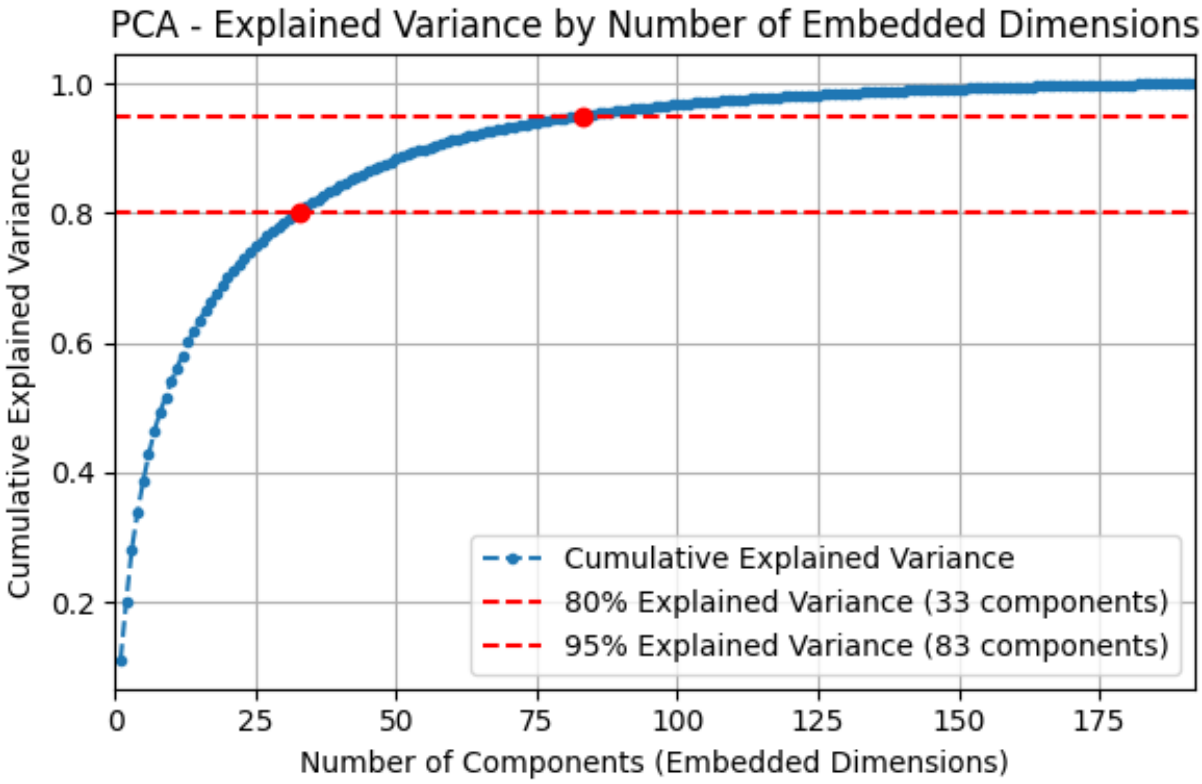

Figure 19: Running PCA on the MAE-encoder embeddings of our supervised dataset. We see that even using 83 out of 192 embedding dimensions retains 95% of the variance. Indicating that our model parameter count could be smaller without much sacrifice.

5 Discussion

5.1 Baseline Results

Our baseline achieves a classification F1-score of 54. As shown in Figure 11, the main source of error is confusion between the middle classes (1 and 2).

5.2 Pre-Training Results

Using our MAE pre-trained models, we get classification F1-scores ranging from 61 to 71, excluding the run using old Adam optimizer, performing worse than baseline. The confusion between the middle classes (1 and 2) is reduced in Figure 12 compared to the baseline confusion matrix in Figure 11. However, due to the high standard deviation in the results 10, the F1-scores overlapped across different hyperparameter combinations, making it challenging to confidently determine the optimal combination. Despite the variability in the results, there is still an indication that pre-training does improve the classification F1-score compared to the baseline. Out of curiosity we also tried some timm pre-trained models. Table 9 shows that even the tiniest timm model far outperforms our own method. This indicates that we are on the right track, using pre-training, but our method is far from state-of-the-art.

5.3 SSIM

We measured the Structural Similarity Index Measure of the MAE model on a subset of the unsupervised validation dataset. We see in Figure 14 that the mean SSIM of our model is 0.88 compared to 0.52 using pixel average blur in Photoshop. This indicates that our model captures more structural information than a simple blurring technique. We do have to criticize that SSIM is known to prefer blurry images (Zhao et al., 2017) (Zhou Wang et al., 2004).

5.4 Using loss for cleaning the dataset

We discovered that the images with the highest and lowest MSE loss proved to be bad images for our task. Figure 15 shows blurry images, and Figure 16 shows images with bubbles, reflections, and doctors' instruments occluding the bowel. This indicates that MSE loss can be used to remove bad images from our dataset.

5.5 t-SNE

In Figure 18 we see a 2D representation of the feature embeddings of our images in the MAE-encoder produced using t-SNE. Ideally, we would like to see the different MES scores cluster and separate from each other. This doesn't appear to be the case, even when making the 'bad image' class (MES score 4) invisible.

5.6 PCA

Another thing we found was many dimensions in our feature embeddings did not contribute much to the variance. In Figure 19, we see that if we cut the width of our networks in our Transformer in half, we would still retain 95% of the variance, indicating that our model could be smaller. This is also backed up by the high F1-score of the 'timm_tiny' model in Table 9.

6 Conclusion

In this study, we demonstrated that pre-training using the MAE method improves our models classification performance. Top-1 comparison shows model F1-score improvement of 16 ± 0.35 points compared to the baseline. Average comparison, excluding the result of Adam optimizer that was lower than baseline, shows an F1-score of 66 ± 2.9, which is 12 ± 3.2 points higher than the baseline's 54 ± 3.6. However, our method still lags behind state-of-the-art pre-trained models like those from the timm library. We found that the loss could potentially be used to identify and remove low quality images from the dataset. SSIM scores of our pretrained model, suggests that it is reconstructing images better than average blur. Furthermore, the t-SNE visualization of feature embeddings did not show clear clustering of MES-score represention in the model. The PCA analysis suggested that the model size could be reduced while retaining most of the variance in the feature embeddings. The results demonstrate that self-supervised pre-training with MAE on unlabelled data, can improve MES classification accuracy and reduce amount of labelled data needed.

7 Future Work

In future studies we would like to compare our MAE to other pre-training methods, such as Masked Siamese Networks, to see if they learn the task better with the same data. Next we would use MSE loss to filter out bad images and run the experiments again. After that, we would like to have enough compute (computational power) to do a proper sweep/grid search

to find the optimal hyperparameters for our method, instead of choosing what configurations are most likeely to give best results. This way we check all possibilities.

It would also be interesting to see how modifications to MAE similar to the ones in Prior Matching for Siamese Networks (PMSN) proposed by Assran, Balestriero, et al., 2022 could account for class imbalance.

As discussed in Balestriero et al., 2024, the masking strategy for an MAE matters a lot. Tailoring the right masking method could be the decisive factor in achieving performance gains.

A Appendix A

Figure 20: Fine tuning to MES score should be short. Pre-trained model overfits after just a few epochs of fine-tuning.

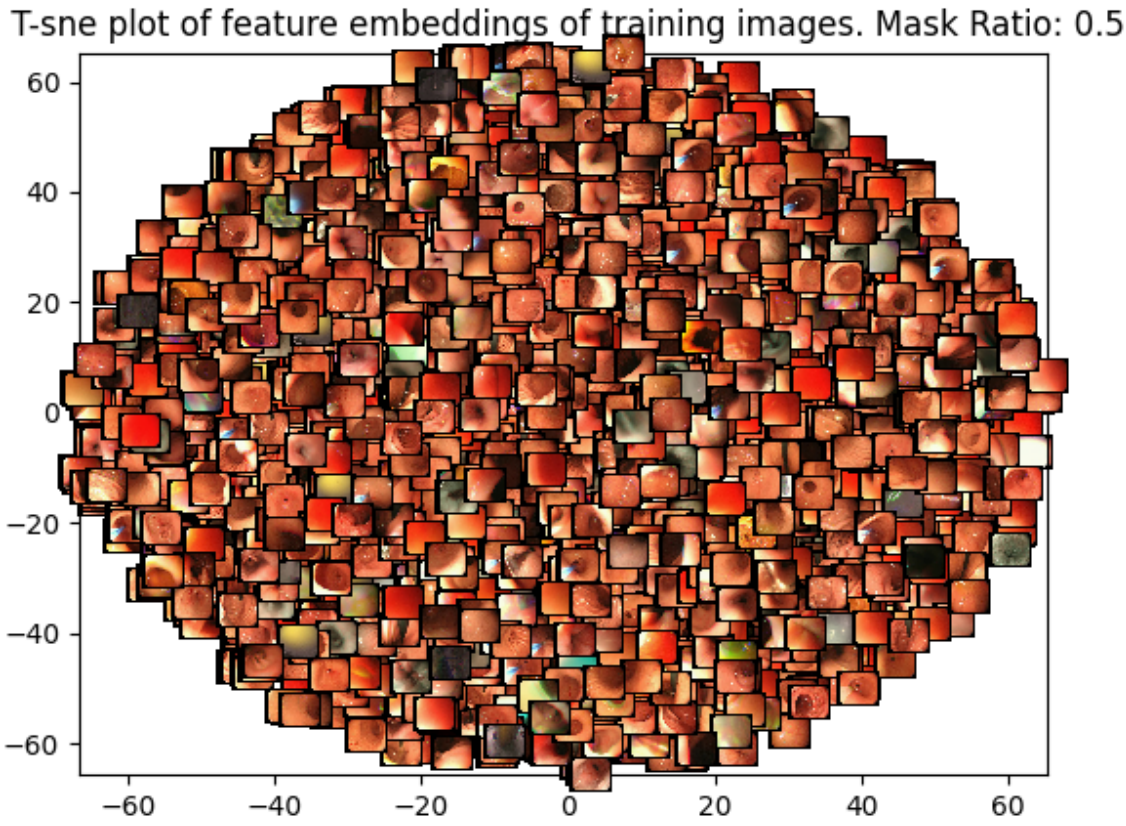

Figure 21: Scatterplot of t-SNE on MAE embedding of unseen supervised dataset, with images as points.

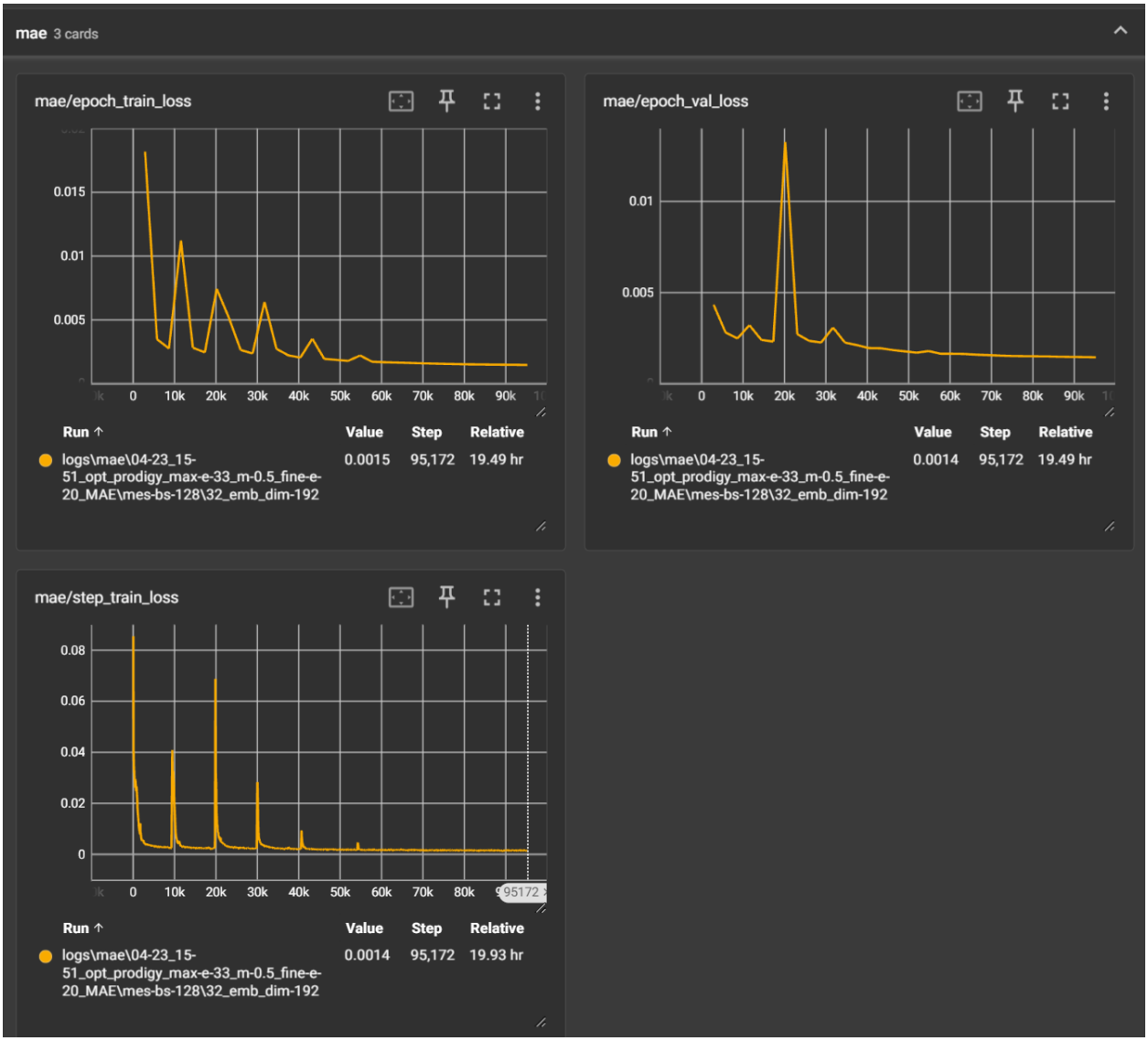

Figure 22: We see spikes in MAE training loss approximately every 10.000 steps.

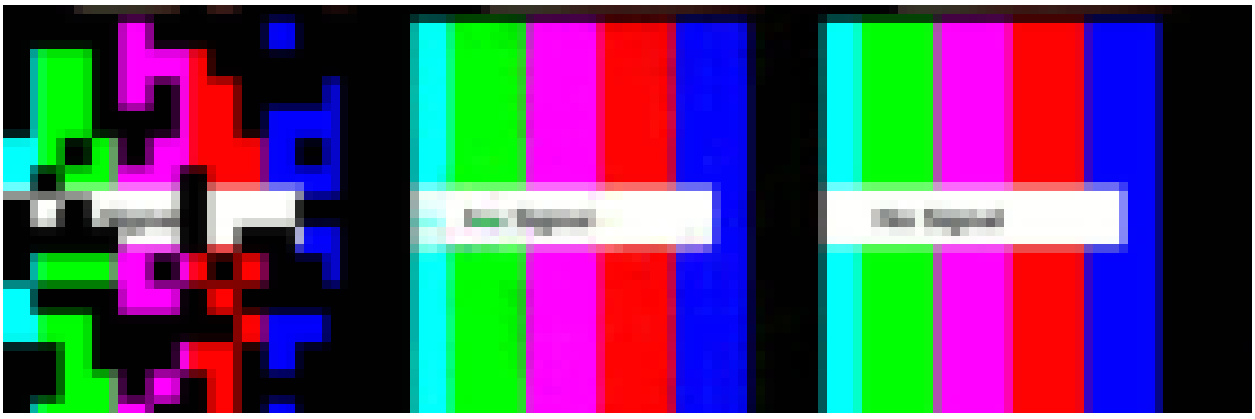

Figure 23: Model reconstructs ”NO SIGNAL” image.

Figure 24: Training loss curves of a full run. From top left: Fine-tune train loss over epochs, fine-tune train loss over steps, fine-tune validatoin loss over epoch. From the bottom left: Pre-train loss over epoch, pre-train validation loss over epoch, pre-train loss over step.

Bibliography

[1] 3Blue1Brown. *But what is a GPT? Visual intro to transformers | Chapter 5, Deep Learning*. YouTube Video. Screenshot from video at 18:30. 2022. URL: `https://www.youtube.com/watch?v=nBVmfxmDXFw` (cit. on p. 11).

[2] Mahmoud Assran, Randall Balestriero, et al. *The Hidden Uniform Cluster Prior in Self-Supervised Learning*. 2022. arXiv: `2210.07277 [cs.LG]` (cit. on pp. 8, 9, 30).

[3] Mahmoud Assran, Quentin Duval, et al. *Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture*. 2023. arXiv: `2301.08243 [cs.CV]` (cit. on pp. 12, 13).

[4] Randall Balestriero et al. *Learning by Reconstruction Produces Uninformative Features For Perception*. 2024. arXiv: `2402.11337 [cs.CV]` (cit. on pp. 12, 30).

[5] Hangbo Bao et al. “BEiT: BERT Pre-Training of Image Transformers”. In: *arXiv preprint arXiv:2106.08254* (2021). Accessed in June 2021 (cit. on pp. 7, 12).

[6] Rishi Bommasani et al. “On the Opportunities and Risks of Foundation Models”. In: *arXiv preprint arXiv:2108.07258* (2021) (cit. on p. 11).

[7] Mathilde Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”. In: *arXiv preprint arXiv:2104.14294* (2021) (cit. on pp. 8, 9).

[8] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607 (cit. on p. 8).

[9] Aaron Defazio et al. “Learning-Rate-Free Learning by D-Adaptation”. In: *arXiv preprint arXiv:2301.07733* (2023). Submitted on 23 Jan 2023 (v1), last revised 5 Feb 2023 (this version, v5). DOI: 10.48550/arXiv.2301.07733. arXiv: 2301.07733 [cs.LG]. URL: https://arxiv.org/abs/2301.07733 (cit. on p. 20).

[10] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018) (cit. on pp. 6, 9).

[11] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *International Conference on Learning Representations*. 2020 (cit. on pp. 11, 19).

[12] John Duchi et al. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research* 12 (July 2011). Submitted 3/10; Revised 3/11; Published 7/11, pp. 2121–2159 (cit. on p. 20).

[13] Dumitru Erhan et al. “Why does unsupervised pre-training help deep learning?” In: *Journal of Machine Learning Research* 11.Feb (2010), pp. 625–660 (cit. on p. 6).

[14] Rafael C. Gonzalez et al. *Digital Image Processing*. Prentice Hall, 2007 (cit. on p. 15).

[15] Jean-Bastien Grill et al. “Bootstrap your own latent: A new approach to self-supervised learning”. In: *arXiv preprint arXiv:2006.07733* (2020) (cit. on p. 8).

[16] Kaiming He, Xinlei Chen, et al. “Masked Autoencoders Are Scalable Vision Learners”. In: *arXiv preprint arXiv:2111.06377* (2021). Tech report. arXiv v2: add more transfer learning results; v3: add robustness evaluation. DOI: 10.48550/arXiv.2111.06377. arXiv: 2111.06377 [cs.CV]. URL: https://arxiv.org/abs/2111.06377 (cit. on pp. 7, 9, 11, 14, 17).

[17] Kaiming He, Haoqi Fan, et al. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9729–9738 (cit. on p. 8).

[18] Roy Hirsch et al. "Self-supervised Learning for Endoscopic Video Analysis". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*. Springer Nature Switzerland, 2023, pp. 569–578. ISBN: 9783031439049. DOI: 10.1007/978-3-031-43904-9_55. URL: http://dx.doi.org/10.1007/978-3-031-43904-9_55 (cit. on pp. 5, 6).

[19] IcarusWizard. *MAE*. https://github.com/IcarusWizard/MAE. 2023. URL: https://github.com/IcarusWizard/MAE (cit. on p. 17).

[20] Diederik P. Kingma et al. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2014). Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015, Submitted on 22 Dec 2014 (v1), last revised 30 Jan 2017 (this version, v9). DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980 (cit. on p. 20).

[21] Bobby Lo et al. "High Accuracy in Classifying Endoscopic Severity in Ulcerative Colitis Using Convolutional Neural Network". In: *The American Journal of Gastroenterology* (2022). DOI: 10.14309/ajg.0000000000001904 (cit. on pp. 5, 6, 14).

[22] Konstantin Mishchenko et al. "Prodigy: An Expeditiously Adaptive Parameter-Free Learner". In: *arXiv preprint arXiv:2306.06101* (Oct. 2023). arXiv: 2306.06101 [cs.LG] (cit. on p. 20).

[23] Bjørn Leth Møller et al. "Building an AI Support Tool for Real-Time Ulcerative Colitis Diagnosis". In: *KI - Künstliche Intelligenz* (Feb. 2024). ISSN: 1610-1987. DOI: 10.1007/s13218-023-00820-x. URL: http://dx.doi.org/10.1007/s13218-023-00820-x (cit. on pp. 5, 6).

[24] OpenCV. *OpenCV Documentation*. Accessed: 2024-06-05. 2023. URL: https://docs.opencv.org/3.4/d5/db5/tutorial_laplace_operator.html (cit. on p. 15).

[25] Tal Ridnik et al. *ImageNet-21K Pretraining for the Masses*. 2021. arXiv: 2104.10972 [cs.CV] (cit. on p. 19).

[26] David E. Rumelhart et al. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536. URL: https://api.semanticscholar.org/CorpusID:205001834 (cit. on p. 11).

[27] K. W. Schroeder et al. “Coated oral 5-aminosalicylic acid therapy for mildly to moderately active ulcerative colitis: A randomized study”. In: *New England Journal of Medicine* 317.25 (1987), pp. 1625–1629. DOI: 10.1056/nejm198712243172603. URL: https://doi.org/10.1056/nejm198712243172603 (cit. on p. 5).

[28] Ashish Vaswani et al. “Attention Is All You Need”. In: *arXiv preprint arXiv:1706.03762* (2017). 15 pages, 5 figures. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762 (cit. on pp. 9, 10).

[29] Haoqing Wang et al. *Masked Image Modeling with Local Multi-Scale Reconstruction*. 2023. arXiv: 2303.05251 [cs.CV] (cit. on p. 7).

[30] Zhao Wang et al. *Foundation Model for Endoscopy Video Analysis via Large-scale Self-supervised Pre-train*. 2024. arXiv: 2306.16741 [cs.CV] (cit. on pp. 5, 14, 15).

[31] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612 (cit. on p. 28).

[32] Ross Wightman. *PyTorch Image Models*. https://github.com/rwightman/pytorch-image-models. Accessed: June 3, 2024. 2019 (cit. on p. 19).

[33] Zhenda Xie et al. *SimMIM: A Simple Framework for Masked Image Modeling*. 2022. arXiv: 2111.09886 [cs.CV] (cit. on pp. 7, 8).

[34] Jason Yosinski et al. “How transferable are features in deep neural networks?” In: *Advances in neural information processing systems*. 2014, pp. 3320–3328 (cit. on p. 6).

[35] Hang Zhao et al. “Loss functions for image restoration with neural networks”. In: *IEEE Transactions on Computational Imaging*. Vol. 3. 1. IEEE, 2017, pp. 47–57 (cit. on p. 28).