

Received 11 January 2024, accepted 14 March 2024, date of publication 25 March 2024, date of current version 2 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381517

RESEARCH ARTICLE

A Study on Self-Supervised Pretraining for Vision Problems in Gastrointestinal Endoscopy

EDWARD SANDERSON^{ID} AND BOGDAN J. MATUSZEWSKI^{ID}, (Member, IEEE)

Computer Vision and Machine Learning (CVML) Group, University of Central Lancashire, PR1 2HE Preston, U.K.

Corresponding author: Edward Sanderson (esanderson4@uclan.ac.uk)

This work was supported by the Science and Technology Facilities Council [grant number ST/S005404/1].

ABSTRACT Solutions to vision tasks in gastrointestinal endoscopy (GIE) conventionally use image encoders pretrained in a supervised manner with ImageNet-1k as backbones. However, the use of modern self-supervised pretraining algorithms and a recent dataset of 100k unlabelled GIE images (Hyperkvasir-unlabelled) may allow for improvements. In this work, we study the fine-tuned performance of models with ResNet50 and ViT-B backbones pretrained in self-supervised and supervised manners with ImageNet-1k and Hyperkvasir-unlabelled (self-supervised only) in a range of GIE vision tasks. In addition to identifying the most suitable pretraining pipeline and backbone architecture for each task, out of those considered, our results suggest three general principles. Firstly, that self-supervised pretraining generally produces more suitable backbones for GIE vision tasks than supervised pretraining. Secondly, that self-supervised pretraining with ImageNet-1k is typically more suitable than pretraining with Hyperkvasir-unlabelled, with the notable exception of monocular depth estimation in colonoscopy. Thirdly, that ViT-Bs are more suitable in polyp segmentation and monocular depth estimation in colonoscopy, ResNet50s are more suitable in polyp detection, and both architectures perform similarly in anatomical landmark recognition and pathological finding characterisation. We hope this work draws attention to the complexity of pretraining for GIE vision tasks, informs this development of more suitable approaches than the convention, and inspires further research on this topic to help advance this development. Code available: github.com/ESandML/SSL4GIE.

INDEX TERMS Gastrointestinal endoscopy, computer vision, self-supervised pretraining, anatomical landmark recognition, pathological finding characterisation, polyp detection, polyp segmentation, monocular depth estimation.

I. INTRODUCTION

Gastrointestinal endoscopy (GIE) is a procedure for screening and treating various digestive disorders that involves the insertion of a thin, flexible tube with a camera and light at the end, known as an endoscope, into either the mouth (gastroscopy) or anus (colonoscopy or sigmoidoscopy) of the patient. The endoscope is then traversed through the gastrointestinal tract as it transmits images of the inner lining to a monitor, where the endoscopist can inspect them for abnormalities and perform any necessary interventions. However, this poses several challenges for the endoscopist, such as the high volume and complexity of visual information, the

variability and subtlety of the lesions, and the need for real-time decision making [1].

To help overcome these challenges, computer vision has been identified as offering a promising set of tools for assisting endoscopists with various aspects of data analysis. Such aspects may be framed as traditional computer vision tasks such as image classification, object detection, semantic segmentation, and monocular depth estimation, among others, where the current state-of-the-art solutions for these tasks use deep learning models trained on large amounts of data.

A. RELATED WORK

While large datasets suitable for training models to perform image classification with *everyday* images exist; most notably

The associate editor coordinating the review of this manuscript and approving it for publication was Binit Lukose^{ID}.

the publicly available ImageNet-1k [2], but also the privately held JFT-300M [3], [4] and JFT-3B [5]; the datasets available for other computer vision tasks and distributions of images, particularly GIE images [6], are notably smaller. It has become clear that the amount of data a model is trained on has a strong influence on its performance [7], and efforts have therefore been taken to identify ways in which the largest available datasets can be leveraged in the training of models for tasks which these large datasets do not include suitable annotations for, and which may involve images of a dissimilar distribution. A now well-established approach [8] is to train (*pretrain*) an image classifier from random initialisation with the ImageNet-1k dataset (1.2M everyday images), remove the classification layer and add any decoder components required for the intended (*downstream*) task to the then pretrained image encoder, and train (*fine-tune*) the resulting model with a dataset which does include suitable annotations for the downstream task. Encoders used in this manner are often referred to as *backbones*.

The approach of pretraining backbones on image classification with ImageNet-1k may however be limiting for two main reasons. Firstly, the model will learn to make high-level abstractions during pretraining, and since this pretraining is task-specific, these abstractions may not generalise well and may need to be *unlearned* during fine-tuning. For example, the ground truth class of many images in ImageNet-1k refers to objects in the foreground and training a model to classify images on this basis may lead to the model learning to pay less attention to the background, which could contain information that is useful for the downstream task. Secondly, image classification datasets require annotations which can be expensive to produce, limiting the degree to which we can leverage more data in pretraining. This is particularly true of GIE images [6], which are especially expensive to annotate, and the use of which in pretraining may be beneficial when the downstream task involves such images.

With the aim of addressing these limitations, a significant amount of research into *self-supervised* pretraining has been undertaken in recent years, leading to a range of popular algorithms [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. Self-supervised pretraining algorithms set *task-agnostic* objectives that require models to predict targets extracted from the input data, which can allow for the learning of *generalisable* high-level feature recognition. Additionally, since this paradigm of learning does not require annotations, it provides the potential for leveraging a much larger amount of data and/or data of a more similar distribution to that involved in the downstream task.

A significant amount of research into self-supervised pretraining with everyday images [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], as well as several modalities of medical images [20], [21], [22], [23], [24], [25], [26], [27], has now been undertaken. However, it is still the convention in GIE to employ backbones that have been pretrained in a supervised manner with ImageNet-1k. A set of 99,417 unlabelled GIE images (Hyperkvasir-unlabelled)

was however included in the recently released Hyperkvasir dataset [28] which, while much smaller than ImageNet-1k, is significantly larger than other datasets of GIE images. This data should allow for the self-supervised pretraining of GIE-specific backbones, which may be better suited to some tasks in GIE than the described convention. Additionally, self-supervised pretraining with datasets of everyday images, e.g. ImageNet-1k, may also provide opportunities for improvements.

B. CONTRIBUTIONS

This paper presents a study on pretraining encoders for use as backbones in solutions to vision tasks in GIE. We consider twelve encoders, each of a ResNet50 [29] or ViT-B [30] architecture and pretrained with one of six pipelines, including two self-supervised pretraining algorithms per architecture, each used separately with both ImageNet-1k and Hyperkvasir-unlabelled, as well as baselines of supervised pretraining with ImageNet-1k and random initialisation (not pretrained). We use state-of-the-art methods for adapting and fine-tuning each encoder for a range of vision tasks in GIE, namely: anatomical landmark recognition, pathological finding characterisation, polyp detection, polyp segmentation, and monocular depth estimation in colonoscopy; and we compare the resulting models on the basis of their fine-tuned performance using well-established metrics. The overall workflow of our experimentation is illustrated in Fig. 1.

In addition to identifying which architecture and pretraining pipeline (algorithm and data) is most suitable for each task, our results suggest that self-supervised pretraining with ImageNet-1k consistently allows for better performance than supervised pretraining with ImageNet-1k, across all considered tasks and architectures. We also demonstrate that self-supervised pretraining with ImageNet-1k is typically more suitable than self-supervised pretraining with Hyperkvasir-unlabelled, with the notable exception of monocular depth estimation in colonoscopy where the similarity of the pretraining data to the downstream data appears to be more critical than the amount of pretraining data. Additionally, we find that ViT-B backbones are typically more suitable for polyp segmentation and monocular depth estimation in colonoscopy, that ResNet50 backbones are more suitable for polyp detection, and that both architectures perform similarly in anatomical landmark recognition and pathological finding characterisation.

While a number of studies have experimented with self-supervised pretraining for certain GIE vision tasks before [31], [32], [33], [34], [35], only two [34], [35] have compared self-supervised pretraining against the convention of supervised pretraining with ImageNet-1k. Additionally, in their experiments with GIE vision tasks, these works either compared self-supervised pretraining against supervised pretraining of a different architecture with the same dataset, or the same architecture with a different dataset. Our work is therefore the first to compare self-supervised pretraining

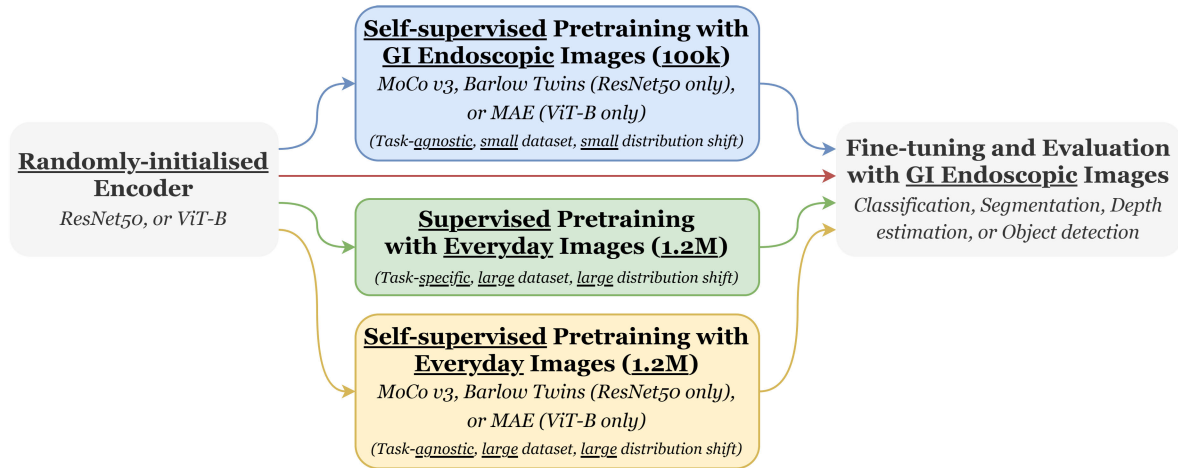


FIGURE 1. The overall workflow of our experimentation.

against supervised pretraining for the same encoder architecture and pretraining data, in terms of fine-tuned performance on GIE vision tasks. Additionally, we consider a much wider scope of self-supervised pretraining algorithms and GIE vision tasks than these previous works, each of which focuses on a single task, and are the first that we know of to experiment with self-supervised pretraining for polyp detection and monocular depth estimation in colonoscopy. Beyond the value of these results in isolation, this wide scope allows us to expose the general principles revealed by our analysis.

II. INVESTIGATED SELF-SUPERVISED PRETRAINING ALGORITHMS

Self-supervised algorithms for pretraining image encoders for use as backbones can be grouped into four families [36]:

- **Deep metric learning (DML)**-based self-supervised pretraining algorithms train an encoder to describe *semantically* similar images with *quantifiably* similar representations, and *semantically* dissimilar images with *quantifiably* dissimilar representations. This is typically achieved by creating *positive* pairs, which are distorted variants of the same image, and *negative* pairs, which are distorted variants of different images, and training the encoder with a *contrastive* loss that is minimised through a reduction in the distance or angle between the representations of positive pairs, and an increase in the distance or angle between the representations of negative pairs.
- **Self-distillation**-based self-supervised pretraining algorithms train an encoder to describe a variant of an image with a representation that allows for a representation of a different variant of the image, produced by another encoder, to be predicted. As a means of avoiding *collapse*, which occurs when both encoders learn to output the same representation for all images, the second encoder is typically an *exponential moving average* of

the encoder being optimised, though collapse can be avoided through a Siamese network with a stop-gradient on one branch [19].

- **Canonical correlation analysis (CCA)**-based self-supervised pretraining algorithms train an encoder to describe an image in such a way that each feature of its representation is informative of a *distinct* attribute of the image. This is typically achieved with a loss function that encourages the encoder to maintain a certain amount of *variance* for each feature in the representation, while establishing *uncorrelatedness* between features.
- **Masked image modelling (MIM)**-based self-supervised pretraining algorithms aim to reproduce the success of masked language modelling (MLM) pretraining algorithms, first introduced for pretraining the transformer-based text encoder BERT [37], in the domain of vision. MIM algorithms are therefore typically used with ViT architectures, which are also inspired by BERT, where the image is split into patches that are treated as a sequence of *visual* tokens akin to the sequence of word tokens used to represent input text for BERT. In both MLM and MIM, input tokens are randomly *masked*, and a model is trained to *reconstruct* these tokens based on the information contained in the remaining tokens.

The rest of this section presents the selection of algorithms considered in our experimentation, which we ensured spanned these four families of self-supervised algorithms. We illustrate and provide a definition of the key details of each algorithm, where we use f_θ to denote the image encoder being optimised for use as a backbone, and explain how we obtained and used encoders pretrained using each algorithm with either ImageNet-1k or Hyperkvasir-unlabelled. Note that any training performed as part of this work was done on an ASUS ESC8000-G4 GPU server with $6 \times$ NVIDIA RTX A6000 48GB GPUs. Due to the number of GPUs and amount of memory, the batch sizes used are in multiples of 6 and, for pretraining, are the maximum that we could allow for.

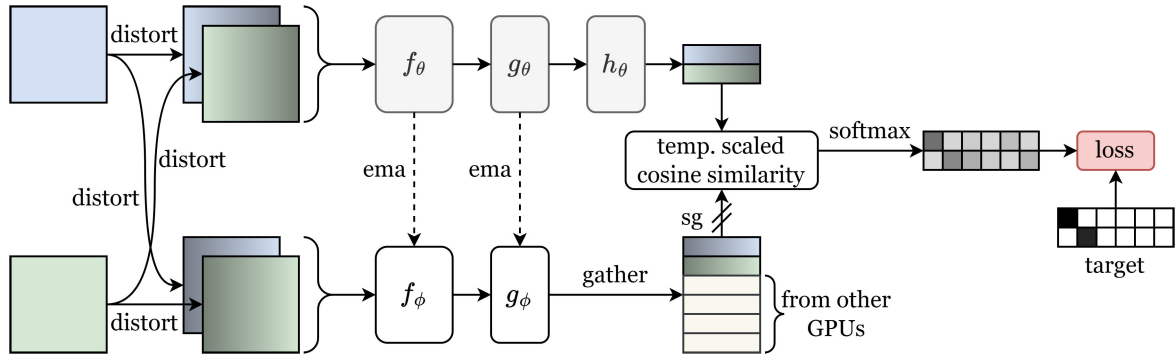


FIGURE 2. Visualisation of the MoCo v3 algorithm. Shown for a per-GPU batch size of 2, and 3 GPUs. We use g_θ to denote the projector, h_θ to denote the predictor, ϕ to denote the momentum parameters that are computed with an exponential moving average (denoted ema) of the online parameters θ , and sg is a stop-gradient.

A. MoCo v3

MoCo v3 [14], illustrated in Fig. 2, is the latest iteration of the *momentum contrast (MoCo)* algorithm, which started as an example of DML. While the distinguishing feature of all iterations of MoCo is the momentum encoder and projector $g_\phi \circ f_\phi$, which is used to compute a representation for one image variant in each pair, rather than using the online encoder and projector $g_\theta \circ f_\theta$ to compute both representations as is more conventional in DML, e.g. SimCLR [9], MoCo v3 incorporates a prediction head h_θ . The resulting algorithm can be framed as either a DML algorithm that incorporates the principle of self-distillation, or a self-distillation algorithm which uses a contrastive loss. As such, we consider MoCo v3 as a representative of both the DML and self-distillation families.

We define a batch of positive pairs of image variants on a single GPU as $\{(\mathbf{x}_{i,1}, \mathbf{x}_{i,2})\}_{i=1}^{N_b}$. We then define the representations used by MoCo v3 as:

$$\mathbf{q}_{i,j} = h_\theta (g_\theta (f_\theta (\mathbf{x}_{i,j}))), \quad i = 1, \dots, N_b \text{ and } j = 1, 2 \quad (1)$$

$$\mathbf{k}_{i,j} = g_\phi (f_\phi (\mathbf{x}_{i,j})), \quad i = 1, \dots, N_{Gb} \text{ and } j = 1, 2 \quad (2)$$

where $N_{Gb} = N_G N_b$, where N_G is the number of GPUs, and the representations $\{\mathbf{k}_{i,j}\}_{i=N_b+1, j=1}^{N_{Gb}, 2}$ are gathered from the other GPUs (see Fig. 2), where they are computed in the same manner as $\{\mathbf{k}_{i,j}\}_{i=1, j=1}^{N_b, 2}$ on different image variants, i.e. $\{(\mathbf{x}_{i,1}, \mathbf{x}_{i,2})\}_{i=N_b+1}^{N_{Gb}}$. The loss function used by MoCo v3 for a batch on a single GPU can then be defined:

$$\begin{aligned} \mathcal{L}_{MC3} & \left(\{\mathbf{q}_{i,1}\}_{i=1}^{N_b}, \{\mathbf{k}_{i,1}\}_{i=1}^{N_{Gb}}, \{\mathbf{q}_{i,2}\}_{i=1}^{N_b}, \{\mathbf{k}_{i,2}\}_{i=1}^{N_{Gb}} \right) \\ & = \frac{2\tau}{N_b} \sum_{i=1}^{N_b} \left[\mathcal{L}_{INCE} \left(\mathbf{q}_{i,1}, \{\mathbf{k}_{j,2}\}_{j=1}^{N_{Gb}} \right) + \mathcal{L}_{INCE} \left(\mathbf{q}_{i,2}, \{\mathbf{k}_{j,1}\}_{j=1}^{N_{Gb}} \right) \right] \end{aligned} \quad (3)$$

where τ is the temperature parameter, a constant positive scalar, and \mathcal{L}_{INCE} is the InfoNCE loss [38], which is defined:

$$\mathcal{L}_{INCE}(\mathbf{q}_i, \{\mathbf{k}_j\}_{j=1}^N) = -\log \left(\frac{e^{\text{CoSim}(\mathbf{q}_i, \mathbf{k}_i)/\tau}}{\sum_{j=1}^N e^{\text{CoSim}(\mathbf{q}_i, \mathbf{k}_j)/\tau}} \right) \quad (4)$$

where CoSim is the cosine similarity:

$$\text{CoSim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (5)$$

Note that Fig. 2 can be seen as illustrating

$$\frac{2\tau}{N_b} \sum_{i=1}^{N_b} \mathcal{L}_{INCE} \left(\mathbf{q}_{i,1}, \{\mathbf{k}_{j,2}\}_{j=1}^{N_G N_b} \right) \quad (6)$$

whereas \mathcal{L}_{MC3} makes this symmetrical.

The algorithm has been designed to work effectively for optimising both ResNet and ViT architectures of f_θ . For ViT architectures, the patch embedding layer is frozen as a random linear projection for stability reasons, and the *class token* [cls] is taken as the output of f_θ . The projectors g_θ and g_ϕ , and the predictor h_θ , are defined as multilayer perceptrons (MLPs) composed of fully connected layers, batch normalisation and ReLU activations.

We consider the use of MoCo v3 for pretraining both ResNet50 and ViT-B architectures, and use the torchvision implementation of ResNet50 and the ViT-B implementation from the official MoCo v3 codebase. For the encoders pretrained using MoCo v3 with ImageNet-1k, we use the weights provided by the authors. We then used the implementation of MoCo v3 in the official codebase to pretrain encoders with Hyperkvasir-unlabelled, modifying the code only for loading Hyperkvasir-unlabelled and to change the batch size from 4096 to 1536/768 (ResNet50/ViT-B). When fine-tuning the ViT-B models, we unfreeze the patch embedding layer.

B. BARLOW TWINS

Barlow Twins [13], illustrated in Fig. 3, is an example of a CCA algorithm. Barlow Twins trains a model to maintain a certain amount of variance for each feature and to establish uncorrelatedness between features with a loss function that encourages an *identity* empirical cross-correlation matrix between representations of two distorted variants of the same image. Other examples of CCA differ mainly in the loss function. For example, the loss function

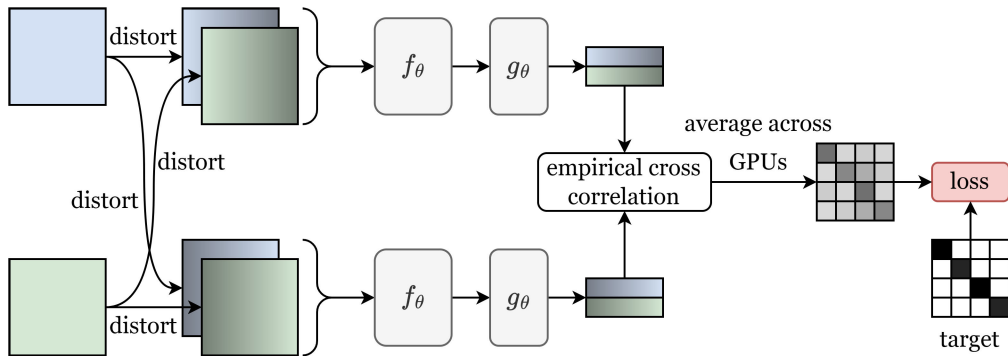


FIGURE 3. Visualisation of the Barlow Twins algorithm. Shown for a per-GPU batch size of 2, and representations of dimensionality 4. We use g_θ to denote the projector.

used by VicReg [16] encourages the variance of features to be maintained, and uncorrelatedness between features to be established, on representations of *individual* variants of an image directly, as well as minimising the Euclidean distance between representations of *two* variants of the same image.

We define a batch of positive pairs of image variants on a single GPU as $\{(\mathbf{x}_{i,1}, \mathbf{x}_{i,2})\}_{i=1}^{N_b}$. We then define the representations used by Barlow Twins as:

$$\mathbf{z}_{i,j} = g_\theta (f_\theta (\mathbf{x}_{i,j})), \quad i = 1, \dots, N_b \text{ and } j = 1, 2 \quad (7)$$

which may also be written as $(z_{i,j,k})_{k=1}^d = \mathbf{z}_{i,j}$. These representations are normalised to give:

$$\hat{z}_{i,j,k} = \frac{z_{i,j,k} - \frac{1}{N_b} \sum_{m=1}^{N_b} z_{m,j,k}}{\sqrt{\frac{1}{N_b} \sum_{n=1}^{N_b} (z_{n,j,k} - \frac{1}{N_b} \sum_{m=1}^{N_b} z_{m,j,k})^2}} \quad (8)$$

The elements of the empirical cross correlation matrix $(c_{k,l})_{k=1,l=1}^{d,d}$ can then be defined:

$$c_{k,l} = \frac{1}{N_b} \sum_{i=1}^{N_b} \hat{z}_{i,1,k} \hat{z}_{i,2,l} \quad (9)$$

which is averaged across GPUs, the result of which we denote $(\bar{c}_{k,l})_{k=1,l=1}^{d,d}$. Finally, the Barlow Twins loss can be defined:

$$\mathcal{L}_{BT} \left((\bar{c}_{k,l})_{k=1,l=1}^{d,d} \right) = \sum_{k=1}^d (1 - \bar{c}_{k,k})^2 + \lambda \sum_{k=1}^d \sum_{l=1}^d \mathbb{1}_{[k \neq l]} \bar{c}_{k,l}^2 \quad (10)$$

where λ is a constant positive scalar and $\mathbb{1}$ is an indicator function.

The algorithm has been designed to work effectively for ResNet architectures of f_θ . The projector g_θ is defined as an MLP composed of fully connected layers, batch normalisation and ReLU activations.

We consider the use of Barlow Twins for pre-training ResNet50 architectures, for which we use the torchvision implementation. For the ResNet50 pre-trained using Barlow Twins with ImageNet-1k, we use

the weights provided by the authors. We then used the implementation of Barlow Twins in the official codebase to pretrain a ResNet50 with Hyperkvasir-unlabelled, modifying the code only for loading Hyperkvasir-unlabelled and to change the batch size from 2048 to 1536.

C. MAE

Masked autoencoders (MAE) [12], illustrated in Fig. 4, is a particularly popular example of the MIM family. It differs from other popular MIM algorithms on two main fronts. First, examples such as BEiT [15], PeCo [39], and SimMIM [40] use an *arbitrary* token in place of the masked tokens in the input to f_θ , where MAE simply omits them. Notably, this is only possible with ViTs due to the use of position embeddings that inform a model of the specific patch of an image that a token corresponds to *explicitly*. For reconstruction, this does however require the insertion of an arbitrary token at each position of a masked token in the output of f_θ , and the processing of the resulting sequence of tokens by a decoder g_θ , which has a smaller ViT architecture. Secondly, BEiT and PeCo use the discrete variational autoencoder introduced as a component of DALL-E [41] to quantise all possible image patches into a finite set of visual tokens akin to a *vocabulary* of words, rather than directly using the patches as visual tokens. This allows the reconstruction to be framed as classifying which token in this finite set the masked token should be, closely following BERT. MAE however takes a more conventional approach to image reconstruction and frames it as a regression problem.

As is typical for a ViT, an image is first divided into a sequence of flattened non-overlapping patches that are projected by a patch embedding layer and translated by a position embedding to produce the sequence of visual tokens $(\mathbf{x}_i)_{i=1}^{N_p}$ that are to be concatenated with the [cls] token and fed into the first block. Before concatenating with the [cls] token however, MAE generates a set of uniformly distributed random values $\{\alpha_i \sim \mathcal{U}(0, 1)\}_{i=1}^{N_p}$ and computes the permutation σ which sorts the set into reverse order, i.e. $\alpha_{\sigma(i)} \geq \alpha_{\sigma(i+1)}$ for $i = 1, \dots, N_p - 1$. For a proportion of masking $\gamma \in [0, 1]$, selected to ensure that

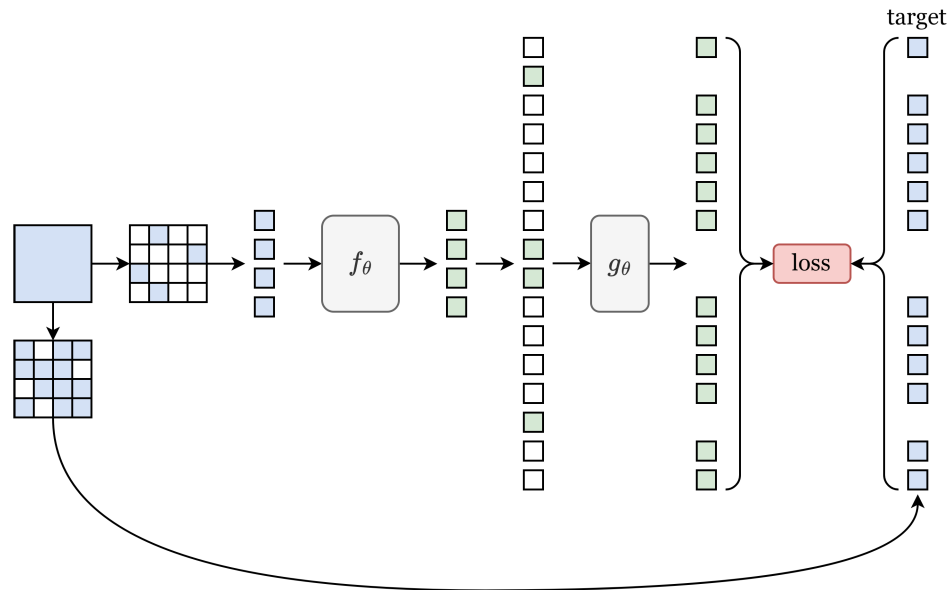


FIGURE 4. Visualisation of the MAE algorithm. Shown for a ViT encoder that treats an image as a 4×4 grid of patch tokens, with 75% masking.

$\gamma N_p - \lfloor \gamma N_p \rfloor = 0$, the sequence passed forward is then $(\tilde{\mathbf{x}}_i)_{i=1}^{(1-\gamma)N_p} = (\mathbf{x}_{\sigma(i)})_{i=1}^{(1-\gamma)N_p}$. In contrast to MIM algorithms that replace rather than omit the masked tokens from the input to f_θ , it is important in MAE that the same number of tokens in each input are masked, i.e. γN_p is constant, to allow for batching. If the sequence of visual tokens, i.e. omitting the [cls] token, in the output of f_θ is denoted $(\tilde{\mathbf{z}}_i)_{i=1}^{(1-\gamma)N_p}$, we then create the sequence $(\mathbf{z}_i)_{i=1}^{N_p}$, where:

$$\mathbf{z}_i = \begin{cases} \tilde{\mathbf{z}}_{\sigma^{-1}(i)} & \text{if } 1 \leq \sigma^{-1}(i) \leq (1-\gamma)N_p \\ \mathbf{m} & \text{if } (1-\gamma)N_p + 1 \leq \sigma^{-1}(i) \leq N_p \end{cases} \quad (11)$$

where \mathbf{m} is a learnt arbitrary token. The tokens in $(\mathbf{z}_i)_{i=1}^{N_p}$ are then translated by another position embedding and fed through the decoder blocks with the [cls] token. The output of the decoder blocks is then fed through a prediction head and the [cls] token is removed, leaving the sequence of reconstructed flattened patches for the entire image $(\hat{\mathbf{y}}_i)_{i=1}^{N_p}$. Denoting the sequence of ground truth flattened patches $(\mathbf{y}_i)_{i=1}^{N_p}$, in which the features have been zero-centred and scaled to unit variance for each patch independently, the loss function is defined:

$$\mathcal{L}_{MAE} \left((\hat{\mathbf{y}}_i)_{i=1}^{N_p} \right) = \frac{1}{\gamma N_p d_p} \sum_{i=1}^{N_p} \mathbb{1}_{[(1-\gamma)N_p + 1 \leq \sigma^{-1}(i) \leq N_p]} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|^2 \quad (12)$$

where d_p is the dimensionality of a patch. The loss is then averaged over all images in the batch on a single GPU, and the update to the model is averaged over GPUs, as is typical in distributed supervised learning.

As mentioned, MAE has been designed for pretraining ViT architectures specifically. A notable distinction between the use of ViT in MAE and in MoCo v3 is that the loss is computed on the processed visual tokens in MAE, whereas it is computed on the processed [cls] token in MoCo v3.

We consider the use of MAE for pretraining ViT-B architectures, for which we use the implementation from the official MAE codebase. For the ViT-B pretrained using MAE with ImageNet-1k, we use the weights provided by the authors. We then used the implementation of MAE in the official codebase to pretrain a ViT-B with Hyperkvasir-unlabelled, modifying the code only for loading Hyperkvasir-unlabelled and to change the batch size from 4096 to 768.

III. BASELINES

For each of the considered encoder architectures, ResNet50 and ViT-B, we consider two baselines to compare the discussed self-supervised pretraining pipelines against. Most importantly, we consider supervised pretraining with ImageNet-1k, representing the conventional approach for pretraining image encoders for use as backbones in solutions to GIE vision tasks. We then consider no pretraining, i.e. finetuning from random initialisation. We use the torchvision implementation and weights for ResNet50, and the timm implementation and weights for ViT-B.

We note that we do not directly compare against the state-of-the-art methods for each task. While our primary aim is to study the relative effectiveness of different pretraining pipelines, which such comparisons would not be suitable for due to the need for consistency in all other details, we believe that this would still be informative. However, we cannot compare against previously reported results due to the lack

of standardisation in the benchmarks, with different works using different splits and different evaluation methodologies, and re-implementing these methods to allow for a direct comparison would be too time-consuming. To the best of our knowledge, the state-of-the-art for each task uses either a convolutional neural network or some derivative of ViT that has been pretrained in a supervised manner with ImageNet-1k as a backbone, and as such we consider models with a ResNet50 or ViT-B backbone that has been pretrained in a supervised manner with ImageNet-1k as representative of the state-of-the-art.

IV. IMAGE CLASSIFICATION

Image classification is the problem of determining which, out of a predefined set of classes, a given image should be assigned to. In the context of GIE, the predefined set of classes may cover, for example, possible anatomical landmarks, pathological findings, or categories of polyps. In this section, we detail and present our evaluation of the fine-tuned performance of backbones in two of these image classification tasks, namely anatomical landmark recognition and pathological finding characterisation.

A. DATA

The data used in our image classification experiments is taken from the Hyperkvasir-labelled dataset [28], which does not share any instances with Hyperkvasir-unlabelled. We specifically used the anatomical landmarks and pathological findings subsets, which we treated the classification of as two separate problems. For each subset, we combined the data for the upper and lower gastrointestinal tract, and applied a random 80%/10%/10% training/validation/test split, where the validation data is used to determine whether to save the weights after each epoch of training on the training data, and the test data is reserved for evaluating the model after fine-tuning. The number of instances of each class, in total and in each split, are given in Table 1.

B. DECODERS

In image classification, it is typical to simply add a linear classifier to the final representation computed by an encoder to allow for prediction. Following convention, we implement this as a fully connected layer that maps the final representation to a vector of logits, one for each possible class, which is softmax normalised prior to computation of the loss. For the ViT-B models, we use the output [cls] token as the final representation.

C. FINE-TUNING PROCEDURE

We separately train each model to perform both anatomical landmark recognition and pathological finding characterisation through the same procedure. We use the common fine-tuning procedure hyperparameters given in Table 3 and pre-process the training images using the pipeline detailed in Table 2. The loss is then computed using a cross entropy loss function which, due to the significant class imbalance in the

data, is weighted with a value of N_D/N_iN_c for the i^{th} class, where N_D is the total number of images in the dataset, N_i is the number of images in a particular class, and N_c is the number of classes. Note that these numbers are for the entire dataset, rather than the training set. This weighting ensures that the total sum of weights across all instances is N_D , for consistency with unweighted cross entropy. We use the macro F1-score (mF1)¹ as the validation metric:

$$\text{mF1} = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{2\text{TP}_i + \epsilon}{2\text{TP}_i + \text{FP}_i + \text{FN}_i + \epsilon} \quad (13)$$

where TP_i is the number of true positives for the i^{th} class, FP_i is the number of false positives, FN_i is the number of false negatives, and $\epsilon = 1e - 8$. The transformations applied to the validation images include the same resizing and normalisation applied to the training images. Finally, the model is trained on this basis for 50 epochs, with the parameters saved after each epoch that leads to an improvement in mF1 on the validation set, with any batch normalisation synchronised across GPUs.

D. EVALUATION

We evaluate the resulting image classification models using the corresponding test data, which is pre-processed in the same manner as the validation data, with four metrics, namely mF1 (as defined in (13)), mPrecision, mRecall, and Accuracy:

$$\text{mPrecision} = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{\text{TP}_i + \epsilon}{\text{TP}_i + \text{FP}_i + \epsilon} \quad (14)$$

$$\text{mRecall} = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{\text{TP}_i + \epsilon}{\text{TP}_i + \text{FN}_i + \epsilon} \quad (15)$$

$$\text{Accuracy} = \frac{\sum_{i=1}^{N_c} \text{TP}_i}{N_D} \quad (16)$$

where $\epsilon = 1e - 8$.

For all metrics, a higher value indicates better performance. The results for anatomical landmark recognition are presented in Table 4 and the results for pathological finding characterisation are presented in Table 5.

V. OBJECT DETECTION

Object detection is the problem of recognising and locating any objects of interest in an image. In the context of GIE, the objects of interest may be polyps, tools, artefacts, or disease. In this section, we detail and present our evaluation of the fine-tuned performance of backbones in polyp detection specifically.

A. DATA

The data used in our object detection experiments is taken from the Kvasir-SEG dataset [45], which does not share

¹Two different formulations of mF1 can be found in the literature — (13) is the more robust [42] arithmetic mean of individual F1 scores.

TABLE 1. Number of instances of each class, in total and in each split.

Classification Task	Tract	Class	Total	Train.	Val.	Test.
Anatomical landmark recognition	Lower	Cecum	1009	794	109	106
		Ileum	9	6	2	1
		Retroflex-rectum	391	308	32	51
	Upper	Pylorus	999	816	87	96
		Retroflex-stomach	764	610	82	72
		Z-line	932	750	98	84
Pathological finding characterisation	Lower	Hemorrhoids	6	3	3	0
		Polyps	1028	825	100	103
		Ulcerative colitis grade 0-1	35	28	4	3
		Ulcerative colitis grade 1	201	160	17	24
		Ulcerative colitis grade 1-2	11	7	2	2
		Ulcerative colitis grade 2	443	347	53	43
	Upper	Ulcerative colitis grade 2-3	28	23	2	3
		Ulcerative colitis grade 3	133	109	13	11
		Barrett's short-segment	53	44	5	4
		Barrett's	41	32	4	5
Esophagitis A	403	333	33	37		
	Esophagitis B-D	260	203	28	29	

TABLE 2. Pre-processing of training images, which is performed online during training. 1) pads to $\max(h, w) \times \max(h, w)$ for original height h and width w . 2) resizes to 224×224 using bicubic interpolation with anti-aliasing [43]. 3) applies colour jitter with brightness factor sampled uniformly from $[0.6, 0.4]$, contrast factor sampled uniformly from $[0.5, 1.5]$, saturation factor sampled uniformly from $[0.75, 1.25]$, and hue factor sampled uniformly from $[0.99, 1.01]$. 4) applies Gaussian blur with a 25×25 kernel with a standard deviation sampled uniformly from $[0.001, 2]$. 5) applies a rotation of 90° with a probability of 0.5. 6) applies a horizontal flip with a probability of 0.5. 7) applies a vertical flip with a probability of 0.5. 8) applies a rotation of an angle sampled uniformly from $[-180^\circ, 180^\circ]$. 9) applies an affine transform with, horizontal translation sampled uniformly from $[-28, 28]$, vertical translation sampled uniformly from $[-28, 28]$, scaling with factor sampled uniformly from $[0.5, 1.5]$, and shearing of an angle sampled uniformly from $[-22.5^\circ, 22.5^\circ]$. 10) applies normalisation using the ImageNet-1k pixel mean and standard deviation, for consistency with all pretraining pipelines.

Operation	Image classification	Object detection	Semantic segmentation	Monocular depth estimation
1) Pad to square	✗	✗	✗	✓
2) Resize	✓	✗	✓	✓
3) Colour jitter	✓	✓	✓	✓
4) Gaussian blur	✓	✓	✓	✗
5) Discrete rotation	✗	✓	✗	✗
6) Horizontal flip	✓	✓	✓	✓
7) Vertical flip	✓	✓	✓	✓
8) Continuous rotation	✓	✗	✓	✗
9) Affine transform	✗	✗	✓	✗
10) Normalisation	✓	✓	✓	✓

TABLE 3. Common fine-tuning procedure hyperparameters.

Hyperparameter	Value
Batch size	48
Optimiser	AdamW [44]
Initial learning rate	$1e-4$
Learning rate schedule	Halve when validation performance does not improve over 10 epochs, until reaching $1e-6$

any instances with Hyperkvasir-unlabelled. The dataset includes 1000 GIE images, each of which shows at least one polyp and is paired with both a set of bounding boxes, specifying the location and the horizontal and vertical dimensions of any polyps in the image, and a binary segmentation map indicating which pixels correspond to a polyp and which don't. While the segmentation maps were used in our semantic segmentation experiments, here we use the sets of bounding boxes. We applied a random 80%/10%/10%

training/validation/test split, where the validation data is used to determine whether to save the weights after each epoch of training on the training data, and the test data is reserved for evaluating the model after fine-tuning.

B. DECODERS

For our object detection experiments, we implemented the listed backbones within a Faster R-CNN pipeline [46] with feature pyramid network (FPN) [47], which we used the `torchvision` implementation of. We used the existing implementation of the pipeline with a ResNet50 backbone, specifying that all layers of the backbone should be trainable. For the ViT-B models, based on previous analyses of using ViT backbones in object detection [48], [49], we first modified the encoders to efficiently process larger image sizes² by bilinearly interpolating the position embeddings

²All considered pretraining was done with images resized to 224×224 , whereas object detection typically involves larger images, e.g. 1024×1024 .

TABLE 4. Performance in anatomical landmark recognition. The best results for each architecture are highlighted as bold, and the best results overall are underlined.

Backbone arch.	Pretraining data	Pretraining algo.	mF1	mPrecision	mRecall	Accuracy
ResNet50	Hyperkvasir-unlabel.	MoCo v3	0.823	0.989	0.823	0.988
		Barlow Twins	0.824	0.989	0.826	0.990
	ImageNet-1k	MoCo v3	0.828	0.993	0.829	0.993
		Barlow Twins	0.826	0.991	0.828	0.990
		Supervised	0.826	0.827	0.825	0.988
	None	None	0.793	0.957	0.795	0.956
ViT-B	Hyperkvasir-unlabel.	MoCo v3	0.818	0.983	0.819	0.983
		MAE	0.823	0.990	0.823	0.988
	ImageNet-1k	MoCo v3	0.828	0.993	0.829	0.993
		MAE	0.823	0.989	0.823	0.988
		Supervised	0.815	0.979	0.817	0.980
	None	None	0.782	0.955	0.778	0.944

TABLE 5. Performance in pathological finding characterisation. The best results for each architecture are highlighted as bold, and the best results overall are underlined.

Backbone arch.	Pretraining data	Pretraining algo.	mF1	mPrecision	mRecall	Accuracy
ResNet50	Hyperkvasir-unlabel.	MoCo v3	0.542	0.531	0.495	0.746
		Barlow Twins	0.613	0.527	0.561	0.765
	ImageNet-1k	MoCo v3	0.595	0.515	0.516	0.758
		Barlow Twins	0.628	0.682	0.545	0.765
		Supervised	0.587	0.645	0.492	0.777
	None	None	0.491	0.584	0.437	0.621
ViT-B	Hyperkvasir-unlabel.	MoCo v3	0.589	0.690	0.526	0.777
		MAE	0.576	0.563	0.526	0.723
	ImageNet-1k	MoCo v3	0.596	0.608	0.527	0.731
		MAE	0.652	0.723	0.596	0.780
		Supervised	0.596	0.605	0.523	0.758
	None	None	0.434	0.679	0.366	0.621

and using *non-overlapping* window self-attention in all but the 3rd, 6th, 9th, and 12th blocks. Window attention, also known as *restricted* attention [50], independently applies attention to subsets of the sequence of visual tokens, where each subset corresponds to the tokens in a square window of the equivalent feature map, with no overlapping windows. We used 256 tokens in each subset, corresponding to a 16×16 window of a feature map. We then modified the Faster R-CNN pipeline to use the resulting encoders as backbones with a ViTDet FPN [49].

C. FINE-TUNING PROCEDURE

In the fine-tuning of both model architectures, we use the common fine-tuning procedure hyperparameters given in Table 3 and pre-process the training images using the pipeline detailed in Table 2. For the ResNet50 models, using the default pre-processing pipeline for the Faster R-CNN implementation, the images in a batch are then each resized with bilinear interpolation to a scale of $\min(800/\min(h, w), 1333/\max(h, w))$ of the original size $h \times w$, and then padded to $H \times W$, where H is the maximum

height of the resized images and W is the maximum width of the resized images across the batch. For the ViT-B models, inspired by a previous analysis [48], the images are padded to 1024×1024 — since several images in the dataset have a height or width larger than 1024, these images are downsampled to half the resolution using bicubic interpolation with anti-aliasing before padding. Transformations are also applied to the bounding boxes in accordance with any spatial transformations applied to the image. The usual multi-task loss function for the Faster R-CNN pipeline is used to compute the loss, and we use $AP@[.5:.95]$ as the validation metric for predicted bounding boxes that have a confidence score ≥ 0.05 :

$$AP@[.5:.95] = \frac{1}{10} \sum_{t \in T} AP@t \quad (17)$$

where $T = \{0.5, 0.55, \dots, 0.95\}$ is the set of intersection over union (IoU) thresholds and $AP@t$ is the average precision at the t^{th} IoU threshold. We compute $AP@t$ by first ranking all predicted bounding boxes with respect to the confidence score, from high to low. We then step through

the predicted bounding boxes in rank order and assign the prediction to the true positives if it has an IoU with a target bounding box for the same image that is greater than the IoU threshold, and otherwise assign it to the false positives. At each rank, we then compute the precision and recall using the cumulative number of true positives and false positives and the total number of false negatives. We then determine a strictly monotonically increasing sequence of recall values $(r_i)_{i=1}^{N_r}$, with $r_1 = 0$, $r_{N_r} = 1$, and $(r_i)_{i=2}^{N_r-1}$ being the recall values (excluding 0 and 1) for ranks where false positives and resulting drops in the precision occur, and $AP@t$ is then:

$$AP@t = \sum_{i=2}^{N_r} (r_i - r_{i-1}) p(r_i) \quad (18)$$

where $p(r_i)$ is the maximum precision value out of those which correspond to r_i , for $i = 2, \dots, N_r$. The transformations applied to the validation images include the same resizing and/or padding and normalisation applied to the training images, and a batch size of 1 is used to ensure the evaluation of a ResNet50 model on a particular instance is not influenced by other images in a batch (through the padding to $H \times W$). Finally, the model is trained on this basis for 200 epochs, with the parameters saved after each epoch that leads to an improvement in $AP@[.5:.95]$ on the validation set, with any batch normalisation synchronised across GPUs.

D. EVALUATION

We evaluate the resulting object detection models using the test data, which is pre-processed in the same manner as the validation data, with $AP@[.5:.95]$ (AP for conciseness), $AP@.5$ (AP_{50}), and $AP@.75$ (AP_{75}) computed for predicted bounded boxes with a confidence score ≥ 0.05 . For all metrics, a higher value indicates better performance. The results are presented in Table 6, and some examples for predicted bounding boxes with a confidence score ≥ 0.5 are shown in Fig. 5.

VI. SEMANTIC SEGMENTATION

Semantic segmentation is the problem of determining which, out of a predefined set of classes, each pixel in an image should be assigned to. In the context of GIE, the predefined set of classes will typically include a background class that accounts for anything that is not of interest, as well as any classes that are of interest, for example, polyps, tools, artefacts, or disease. In this section, we detail and present our evaluation of the fine-tuned performance of backbones in polyp segmentation specifically, which is notably a binary segmentation problem.

A. DATA

We used two datasets in our semantic segmentation experiments, namely Kvasir-SEG [45] and CVC-ClinicDB [51]. Kvasir-SEG has already been discussed in the context of our object detection experiments, and we use the

same training/validation/test split here. CVC-ClinicDB includes 612 GIE images, each of which shows at least one polyp and is paired with a binary segmentation map indicating which pixels correspond to a polyp and which don't. We applied a random 80%/10%/10% training/validation/test split, where the validation data is used to determine whether to save the weights after each epoch of training on the training data, and the test data is reserved for evaluating the model after fine-tuning.

B. DECODERS

For our semantic segmentation experiments, we used the listed ResNet50 backbones with a DeepLabV3+ [52] decoder, using the `segmentation-models-pytorch` implementation. We then used the ViT-B backbones with the segmentation variant of the dense prediction transformer (DPT) [53] decoder, using the implementation provided in the official codebase.

C. FINE-TUNING PROCEDURE

We separately train each model to perform polyp segmentation with each dataset through the same procedure. We use the common fine-tuning procedure hyperparameters given in Table 3 and pre-process the training images using the pipeline detailed in Table 2. Transformations are also applied to the segmentation maps in accordance with any spatial transformations applied to the image. The loss is then computed using the Dice loss function [54], and we use mDice as the validation metric:

$$mDice = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{2TP_i + \epsilon}{2TP_i + FP_i + FN_i + \epsilon} \quad (19)$$

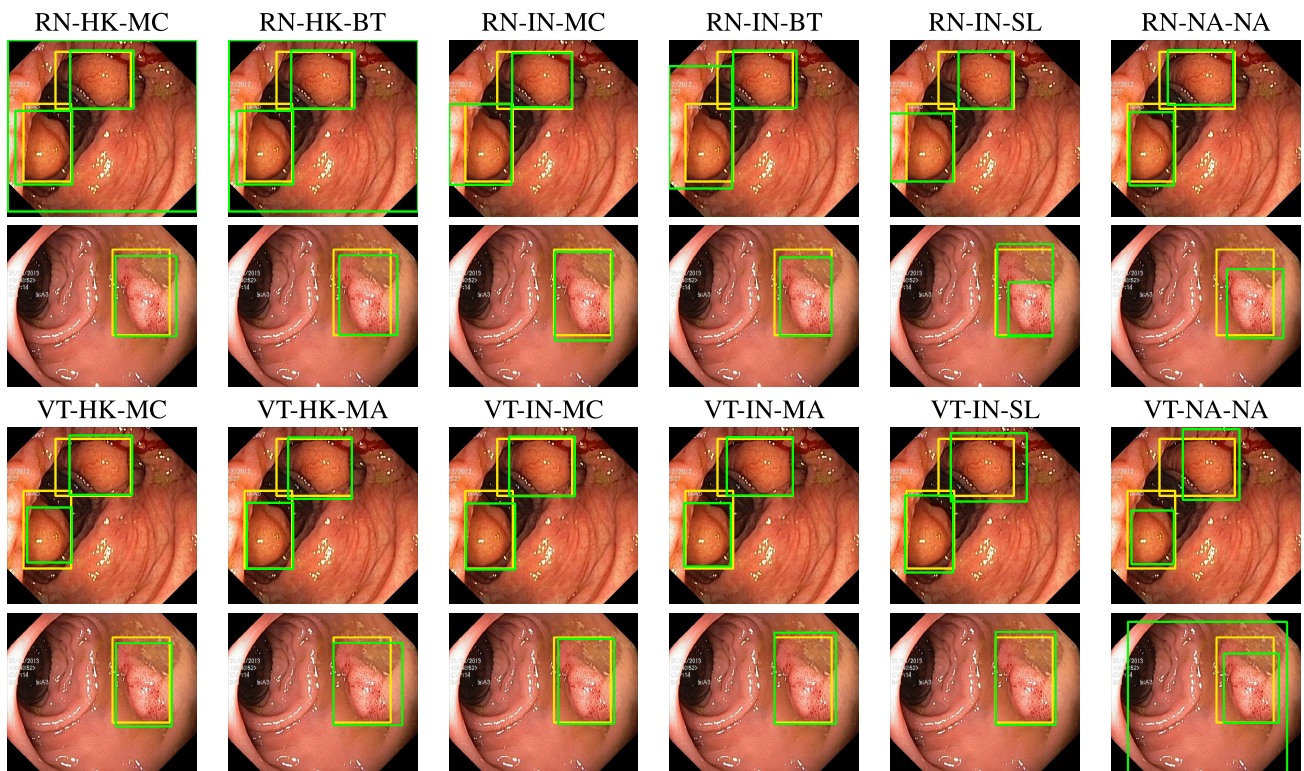
where N_e is the number of instances in the validation/test set, TP_i is the number of true positives for the i^{th} image, FP_i is the number of false positives, FN_i is the number of false negatives, and $\epsilon = 1e - 8$. The transformations applied to the validation images include the same resizing and normalisation applied to the training images, with the validation maps also resized to 224×224 . Finally, the model is trained on this basis for 200 epochs, with the parameters saved after each epoch that leads to an improvement in mDice on the validation set, with any batch normalisation synchronised across GPUs.

D. EVALUATION

We evaluate the resulting semantic segmentation models using the corresponding test data, where the images are pre-processed in the same manner as the validation images, but the segmentation maps are left at their original size. The predictions are therefore resized to this original size using bilinear interpolation prior to binarisation. We then use four metrics, namely the mDice (as defined in (19)), mIoU,

TABLE 6. Performance in polyp detection. The best results for each architecture are highlighted as bold, and the best results overall are underlined.

Backbone arch.	Pretraining data	Pretraining algo.	AP	AP ₅₀	AP ₇₅
ResNet50	Hyperkvasir-unlabel.	MoCo v3	0.604	0.895	0.702
		Barlow Twins	0.647	0.895	0.717
	ImageNet-1k	MoCo v3	0.640	0.905	0.731
		Barlow Twins	0.653	0.906	0.772
	None	Supervised	0.633	0.889	0.716
None	None	0.492	0.795	0.614	
ViT-B	Hyperkvasir-unlabel.	MoCo v3	0.549	0.867	0.578
		MAE	0.563	0.839	0.666
	ImageNet-1k	MoCo v3	0.572	0.873	0.652
		MAE	0.643	0.921	0.748
	None	Supervised	0.577	0.832	0.685
None	None	0.281	0.609	0.197	

**FIGURE 5.** Targets (yellow bounding boxes) and predictions (green bounding boxes) for two randomly selected instances of the Kvasir-SEG test set. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, Barlow Twins with *BT*, MAE with *MA*, supervised pretraining with *SL*, and no pretraining with *NA-NA*.

mPrecision, and mRecall:

$$mIoU = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{TP_i + \epsilon}{TP_i + FP_i + FN_i + \epsilon} \quad (20)$$

$$mPrecision = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{TP_i + \epsilon}{TP_i + FP_i + \epsilon} \quad (21)$$

$$mRecall = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{TP_i + \epsilon}{TP_i + FN_i + \epsilon} \quad (22)$$

For all metrics, a higher value indicates better performance. The results for Kvasir-SEG are presented in Table 7 and the results for CVC-ClinicDB are presented in Table 8. Examples for Kvasir-SEG are shown in Fig. 6.

VII. MONOCULAR DEPTH ESTIMATION

Monocular depth estimation is the problem of predicting the length of the ray of light, that a particular pixel in an image corresponds to, between the camera and the object that the ray of light has come from, for every pixel in the image. Since the

TABLE 7. Performance in polyp segmentation with Kvasir-SEG. The best results for each architecture are highlighted as bold, and the best results overall are underlined.

Backbone arch.	Pretraining data	Pretraining algo.	mDice	mIoU	mPrecision	mRecall
ResNet50	Hyperkvasir-unlabel.	MoCo v3	0.841	0.753	0.857	0.878
		Barlow Twins	0.852	0.772	0.859	0.900
	ImageNet-1k	MoCo v3	0.883	0.812	0.866	0.936
		Barlow Twins	0.873	0.795	0.879	0.899
		Supervised	0.871	0.800	0.882	0.893
None	None		0.632	0.506	0.639	0.780
ViT-B	Hyperkvasir-unlabel.	MoCo v3	0.861	0.788	0.867	0.898
		MAE	0.885	0.816	0.899	0.906
	ImageNet-1k	MoCo v3	0.889	0.824	0.900	0.907
		MAE	0.896	0.834	0.921	0.902
		Supervised	0.871	0.795	0.894	0.883
None	None		0.755	0.650	0.785	0.815

TABLE 8. Performance in polyp segmentation with CVC-ClinicDB. The best results for each architecture are highlighted as bold, and the best results overall are underlined.

Backbone arch.	Pretraining data	Pretraining algo.	mDice	mIoU	mPrecision	mRecall
ResNet50	Hyperkvasir-unlabel.	MoCo v3	0.909	0.839	0.906	0.921
		Barlow Twins	0.880	0.799	0.916	0.863
	ImageNet-1k	MoCo v3	0.920	0.856	0.938	0.909
		Barlow Twins	0.901	0.826	0.893	0.920
		Supervised	0.879	0.805	0.933	0.861
None	None		0.595	0.462	0.584	0.678
ViT-B	Hyperkvasir-unlabel.	MoCo v3	0.909	0.848	0.916	0.926
		MAE	0.901	0.838	0.920	0.903
	ImageNet-1k	MoCo v3	0.911	0.848	0.940	0.901
		MAE	0.927	0.867	0.926	0.933
		Supervised	0.907	0.841	0.912	0.910
None	None		0.813	0.708	0.839	0.826

absolute scale of the scene can only be determined from the parallax observed with a second view, the problem is however inherently ill-posed and only relative scale can be determined. In this section, we detail and present our evaluation of the fine-tuned performance of backbones in monocular depth estimation in colonoscopy.

A. DATA

The data used for our depth estimation experiments is taken from the C3VD dataset [55], the only dataset that we know of which includes images captured with a clinical GIE camera (colonoscope, specifically) with paired ground truth depth maps. The dataset was collected by recording segments (sigmoid, descending, transcending, ascending, and cecum) of a high-fidelity 3D silicone phantom colon model with varying textures, emulating different patient-specific tissue features and vasculature patterns at varying optical depths, and varying illumination modes with a clinical colonoscope. Views of an equivalent 3D virtual colon model were then registered with key frames of the resulting videos, allowing

for the rendering of a ground truth depth map for each frame, as well as a surface normal, optical flow, and occlusion map. Each video is also paired with ground truth camera pose, surface model, and coverage map. 22 videos were recorded, with variation in the segment, camera pose, textures, and illumination, amounting to 10015 frames in total. We selected 18 videos (8610 frames) for training, 2 videos (977 frames) for validation, and 2 videos (528 frames) for testing, where the validation and test sets each include one randomly sampled video of the cecum and one randomly sampled video of the transcending segment, since the majority of videos were of one of these segments (8 of cecum and 9 of transcending).

B. DECODERS

For our monocular depth estimation experiments, we used the listed ViT-B backbones with the depth estimation variant of the dense prediction transformer (DPT) [53] decoder, using the implementation provided in the official codebase. Since there is no clear precedent for a decoder architecture

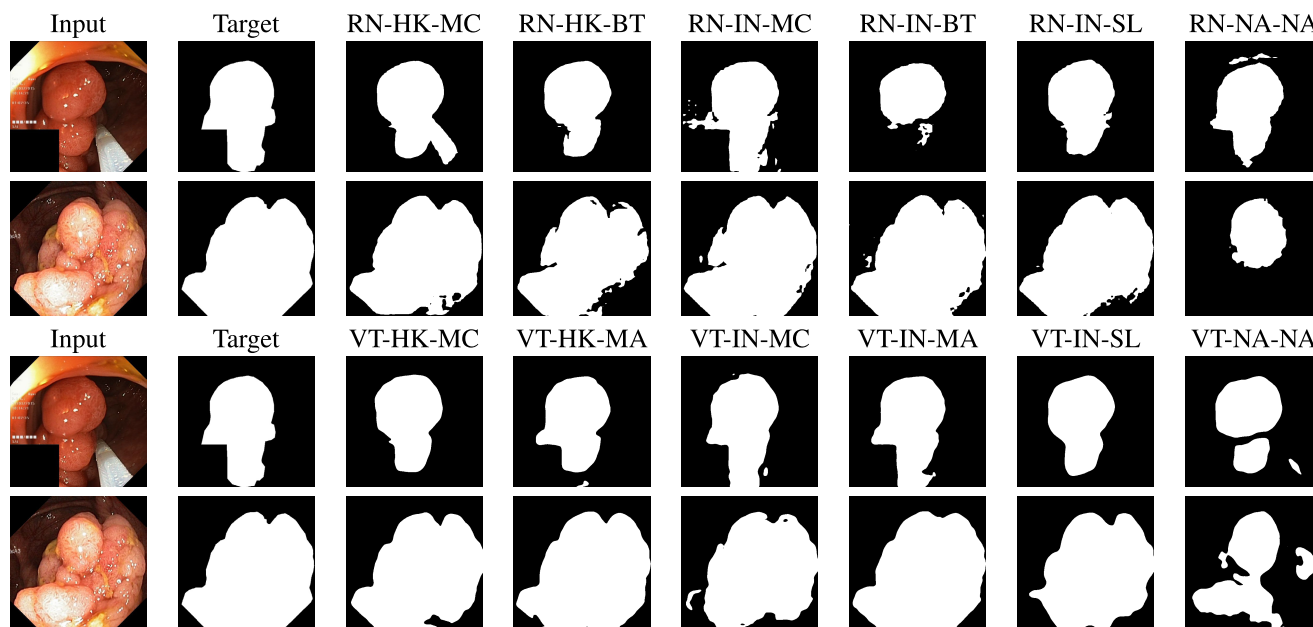


FIGURE 6. Targets and predictions for two randomly selected instances of the Kvasir-SEG test set. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, Barlow Twins with *BT*, MAE with *MA*, supervised pretraining with *SL*, and no pretraining with *NA-NA*.

TABLE 9. Performance in monocular depth estimation in colonoscopy. The best results for each architecture are highlighted as bold, and the best results overall are underlined.

Backbone arch.	Pretraining data	Pretraining algo.	mRMSE (cm)	mMRAE	mMAE (cm)
ResNet50	Hyperkvasir-unlabel.	MoCo v3	0.177	0.0345	0.131
		Barlow Twins	0.178	0.0362	0.134
	ImageNet-1k	MoCo v3	0.207	0.0428	0.159
		Barlow Twins	0.203	0.0428	0.156
		Supervised	0.208	0.0452	0.159
	None	None	0.241	0.0515	0.188
ViT-B	Hyperkvasir-unlabel.	MoCo v3	0.151	0.0259	0.104
		MAE	0.143	0.0246	0.098
	ImageNet-1k	MoCo v3	0.166	0.0274	0.116
		MAE	0.160	0.0298	0.115
		Supervised	0.168	0.0322	0.121
	None	None	0.226	0.0436	0.167

for ResNet50-based depth estimation,³ we designed our own. This decoder, designed to mirror the architecture of ResNet50, has three fusion levels. The first starts with the final feature maps output by a ResNet50 and halves the number of channels with a 1×1 convolutional layer followed by batch normalisation, before upsampling the resulting feature maps to twice the resolution with bilinear interpolation and concatenating it with the feature maps output by the previous level of the ResNet50. The concatenated features are then processed by three blocks that have the same design as the blocks used in each level of ResNet50. The second and third

³Popular dense prediction architectures that adopt certain details of ResNets in their design and which may be suitable for depth estimation, such as ResUNet [56] or ResUNet++ [57], do not actually use a ResNet encoder.

levels of the decoder follow the same logic as the first, except that they start with the output of the previous level of the decoder. A prediction head, which has the same design as the prediction head used in the depth estimation variant of the DPT decoder, is then used to predict a depth map from the output of the third level.

C. FINE-TUNING PROCEDURE

In the fine-tuning of both model architectures, we use the common fine-tuning procedure hyperparameters given in Table 3 and pre-process the training images using the pipeline detailed in Table 2. Transformations are also applied to the depth maps in accordance with any spatial transformations applied to the image, with absolute depth values scaled to

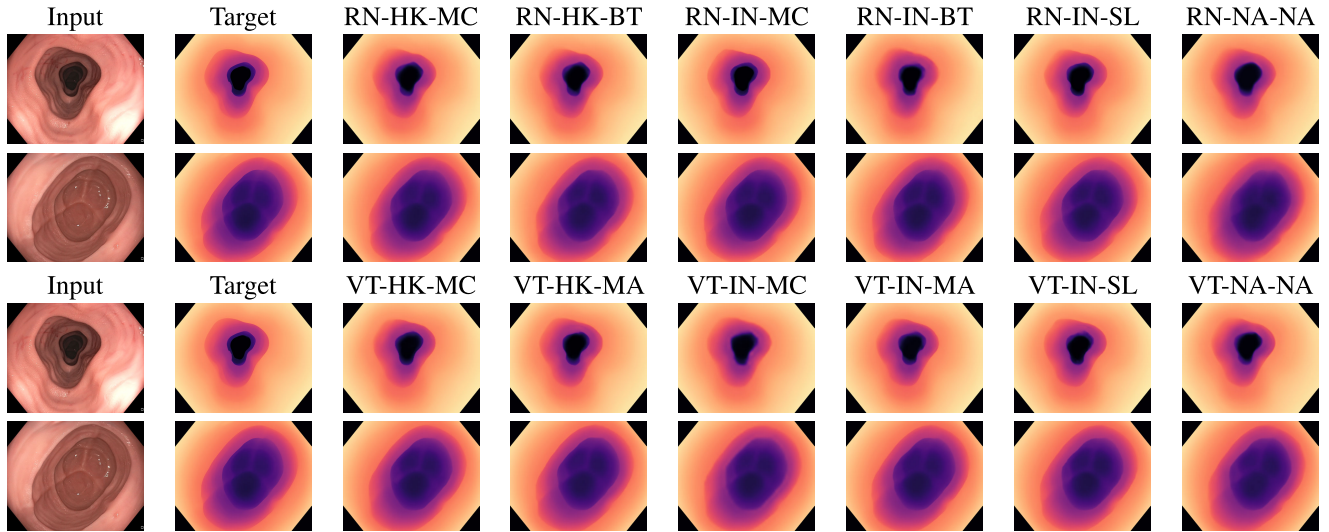


FIGURE 7. Targets and post-processed predictions for a randomly selected instance from each of the test videos for C3VD. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, Barlow Twins with *BT*, MAE with *MA*, supervised pretraining with *SL*, and no pretraining with *NA-NA*.

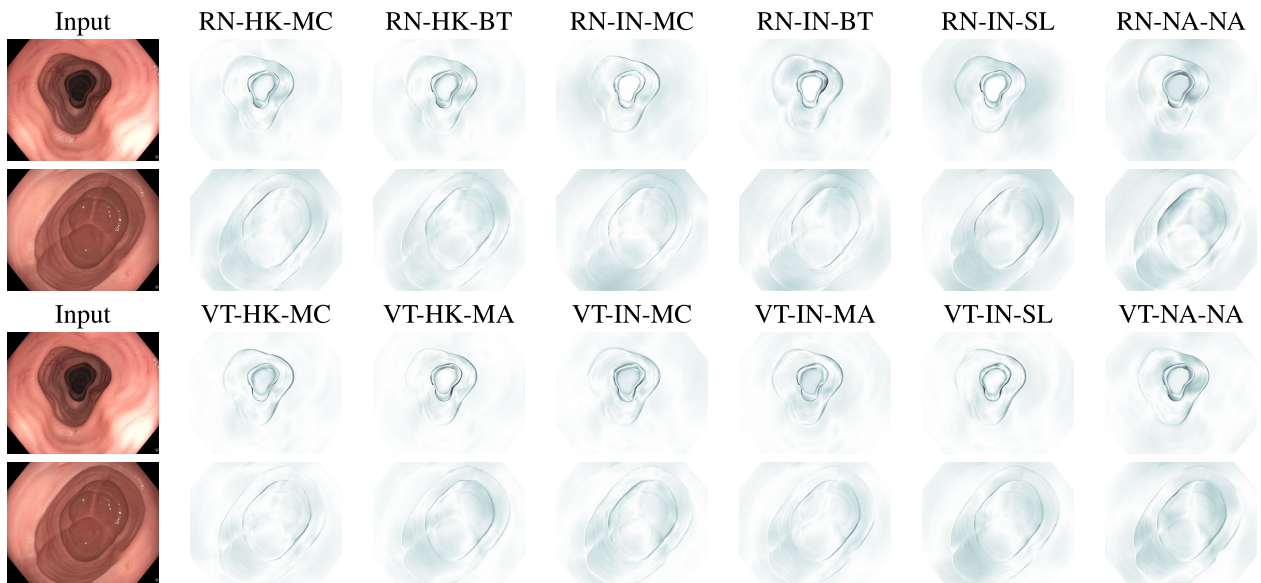


FIGURE 8. Error maps for the post-processed predictions shown in Fig. 7, illustrating the absolute error with a larger value represented by a darker shade. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, Barlow Twins with *BT*, MAE with *MA*, supervised pretraining with *SL*, and no pretraining with *NA-NA*.

[0, 1]. The loss is then computed using the scale- and shift-invariant (SSI) mean squared error (MSE) [58] with a multi-scale shift-invariant gradient matching term [59], which is computed only on the pixels that are covered by the lens (corners are not covered — see examples in Fig. 7), and we use the mSSI-MSE for pixels covered by the lens as the validation metric:

$$\text{mSSI-MSE} = \frac{1}{N_e N_v} \sum_{i=1}^{N_e} \sum_{j=1}^{N_v} (s_i \hat{y}_{i,j} + t_i - y_{i,j})^2 \quad (23)$$

where N_v is the number of pixels covered by the lens in an image, $\hat{y}_{i,j}$ is the output value for the j^{th} pixel covered

by the lens in the i^{th} image, $y_{i,j}$ is the corresponding target value, and s_i and t_i are the scale and shift computed using the closed form solution to the standard least squares problem:

$$\mathbf{h}_i^* = \arg \min_{\mathbf{h}_i} \sum_{j=1}^{N_v} (\hat{\mathbf{y}}_{i,j}^\top \mathbf{h}_i - y_{i,j})^2 \quad (24)$$

where $\mathbf{h}_i = (s_i, t_i)^\top$ and $\hat{\mathbf{y}}_{i,j} = (\hat{y}_{i,j}, 1)^\top$. The transformations applied to the validation images include the same padding, resizing, and normalisation applied to the training images, with the validation maps also padded and resized to

224 × 224 and depth values scaled to [0, 1]. Finally, the model is trained on this basis for 50 epochs, with the parameters saved after each epoch that leads to an improvement in SSI MSE on the validation set, with any batch normalisation synchronised across GPUs.

D. EVALUATION

We evaluate the resulting monocular depth estimation models using the test data, where the images are pre-processed in the same manner as the validation images. We load two target depth maps for each image, one which is pre-processed in the same manner as the validation depth maps, for computing the scale and shift for pixels covered by the lens, and one left at the original size and scale ([0cm, 10cm]), for computing the performance. We compute and apply the scale and shift for the prediction, then resize the result to $\max(h, w) \times \max(h, w)$, where h and w are the height and width of the original image, crop to $h \times w$ to remove values for padded pixels, clip values to [0, 1], set any values for pixels not covered by the lens to 0, and scale the resulting values to [0cm, 10cm]. We then use the four metrics used the SimCol3D challenge [60], namely the arithmetic mean across the test set of: the root MSE (mRMSE), the median relative absolute error (mMRAE), and the mean absolute error (mMAE), which are only applied to pixels covered by the lens:

$$\text{mRMSE} = \frac{1}{N_e} \sum_{i=1}^{N_e} \sqrt{\frac{1}{N_V} \sum_{j=1}^{N_V} (\hat{y}_{i,j} - y_{i,j})^2} \quad (25)$$

$$\text{mMRAE} = \frac{1}{N_e} \sum_{i=1}^{N_e} \text{median}_{j=1, \dots, N_V} \left(\left| \frac{\hat{y}_{i,j} - y_{i,j}}{y_{i,j}} \right| \right) \quad (26)$$

$$\text{mMAE} = \frac{1}{N_e N_V} \sum_{i=1}^{N_e} \sum_{j=1}^{N_V} |\hat{y}_{i,j} - y_{i,j}| \quad (27)$$

where N_V is the number of pixels covered by the lens in an image at its original size, $\hat{y}_{i,j}$ is the value in the post-processed prediction for the j^{th} pixel covered by the lens in the i^{th} image at its original size, and $y_{i,j}$ is the corresponding target value. For all metrics, a lower value indicates better performance. The results are presented in Table 9, and some examples are shown in Fig. 7 with corresponding error maps shown in Fig. 8 to help visualise the differences.

VIII. ANALYSIS

The results presented in the previous sections primarily provide an indication of the ranking of the pretraining pipelines for each considered GIE vision task. Notably, there is some variation in this ranking, as illustrated in Fig. 9, however the ViT-B encoder pretrained with MAE and ImageNet-1k most consistently allows for either the best, or highly competitive, downstream performance. Beyond this identification, however, these results provide evidence for more general principles regarding the pretraining of encoders

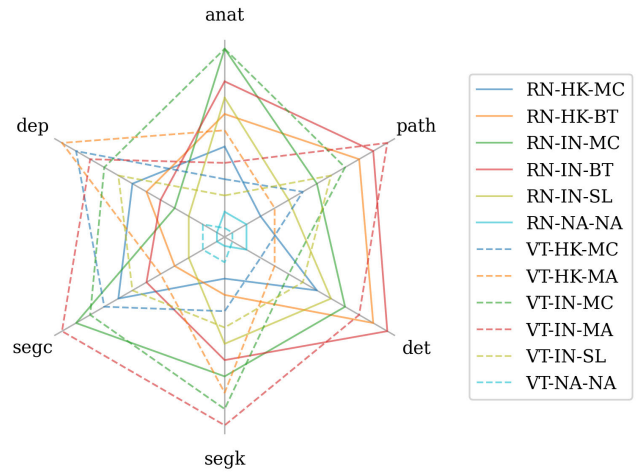


FIGURE 9. Ranking of the performance of each model on each task, as measured by mF1 (anatomical landmark recognition and pathological finding characterisation), AP (polyp detection), mDice (polyp segmentation), and mRMSE (monocular depth estimation in colonoscopy), where a better rank is represented by a greater distance from the centre. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, Barlow Twins with *BT*, MAE with *MA*, supervised pretraining with *SL*, and no pretraining with *NA-NA*. Additionally, we refer to anatomical landmark recognition as *anat*, pathological finding characterisation as *path*, polyp detection as *det*, polyp segmentation with Kvasir-SEG as *segk*, polyp segmentation with CVC-ClinicDB as *segc*, and monocular depth estimation in colonoscopy as *dep*.

for use as backbones in solutions to GIE vision tasks, which we reveal through an analysis presented in this section.

First, we demonstrate that self-supervised pretraining is generally more suitable than supervised pretraining. To assess this, we evaluate the *relative* improvement of each model that uses a backbone pretrained in a self-supervised manner with ImageNet-1k vs. the equivalent model (same architecture and task) that uses a backbone pretrained in a supervised manner with ImageNet-1k. To compute the relative improvement, we consider the primary metric for each task as mF1 (image classification), AP (object detection), mDice (semantic segmentation), and mRMSE (depth estimation), as defined in the discussion of each task. Then, for all but mRMSE, we take the absolute difference between the result and a perfect score of 1, in order to convert each *score* (higher is better) to a measure of *error* (lower is better). We do not do this for mRMSE since it is already a measure of error. We then compute the relative improvement using:

$$\% \text{Improvement}_{SL \rightarrow SSL} = 100 \frac{\delta_{SL} - \delta_{SSL}}{\delta_{SL}} \quad (28)$$

where δ_{SSL} is the error for a model with a backbone pretrained in a self-supervised manner and δ_{SL} is the error for an equivalent model (same architecture, pretraining data, and task) with a backbone pretrained in a supervised manner. Note that this analysis omits any results for pretraining with Hyperkvasir-unlabelled or no pretraining. We visualise the results of this analysis in Fig. 10, where it can be seen that self-supervised pretraining overwhelmingly provides improvements over

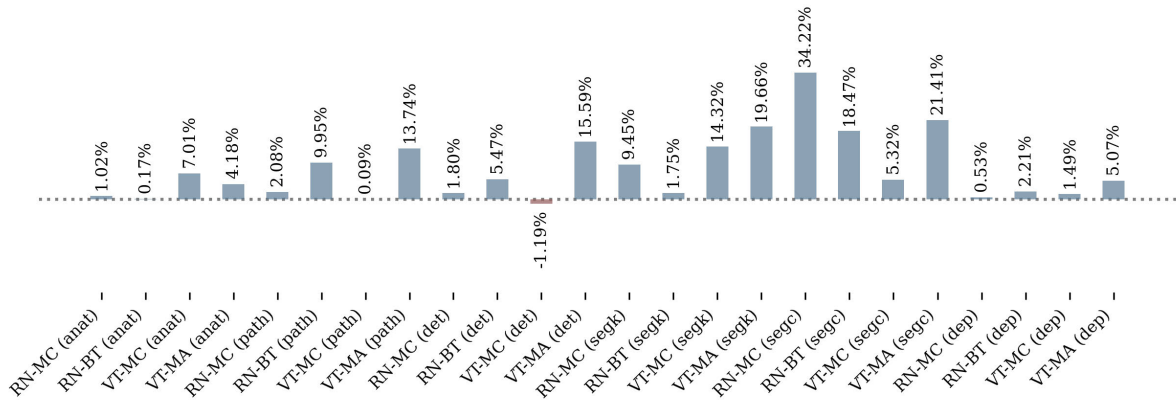


FIGURE 10. Improvement of self-supervised pretraining vs supervised pretraining for same architecture and pretraining data (ImageNet-1k). For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, MoCo v3 with *MC*, Barlow Twins with *BT*, and MAE with *MA*. Additionally, we refer to anatomical landmark recognition as *anat*, pathological finding characterisation as *path*, polyp detection as *det*, polyp segmentation with Kvasir-SEG as *segk*, polyp segmentation with CVC-ClinicDB as *segc*, and monocular depth estimation in colonoscopy as *dep*.



FIGURE 11. Improvement of pretraining with Hyperkvasir-unlabelled vs pretraining with ImageNet-1k for same architecture and self-supervised pretraining algorithm. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, MoCo v3 with *MC*, Barlow Twins with *BT*, and MAE with *MA*. Additionally, we refer to anatomical landmark recognition as *anat*, pathological finding characterisation as *path*, polyp detection as *det*, polyp segmentation with Kvasir-SEG as *segk*, polyp segmentation with CVC-ClinicDB as *segc*, and monocular depth estimation in colonoscopy as *dep*.

supervised pretraining in our experiments, with only a single marginal exception to this observed across all self-supervised pretraining algorithms, architectures, and tasks. We can therefore confidently conclude that self-supervised pretraining with ImageNet-1k generally provides better backbones than supervised pretraining with ImageNet-1k. Since supervised pretraining with ImageNet-1k is still the conventional pretraining pipeline for backbones used in solutions to vision tasks in GIE, including the state-of-the-art, this is a crucial finding.

We also demonstrate that self-supervised pretraining with ImageNet-1k is generally more suitable than self-supervised pretraining with Hyperkvasir-unlabelled in the considered downstream tasks, with the notable exception of monocular depth estimation in colonoscopy. To assess this, we use the same measures of error used in the previous analysis and evaluate the relative improvement from pretraining with

Hyperkvasir-unlabelled vs. ImageNet-1k using:

$$\%Improvement_{IN \rightarrow HK} = 100 \frac{\delta_{IN} - \delta_{HK}}{\delta_{IN}} \quad (29)$$

where δ_{HK} is the error for a model with a backbone pretrained with Hyperkvasir-unlabelled and δ_{IN} is the error for an equivalent model (same architecture, pretraining algorithm, and task) with a backbone pretrained with ImageNet-1k.

Note that this analysis omits any results for supervised pretraining or no pretraining. We visualise the results of this analysis in Fig. 11, where it can be seen that self-supervised pretraining with ImageNet-1k generally provides better performance than self-supervised pretraining with Hyperkvasir-unlabelled, with exceptions including the anatomical landmark recognition models with MAE pre-trained backbones, as well as all monocular depth estimation models. While the result for the anatomical landmark

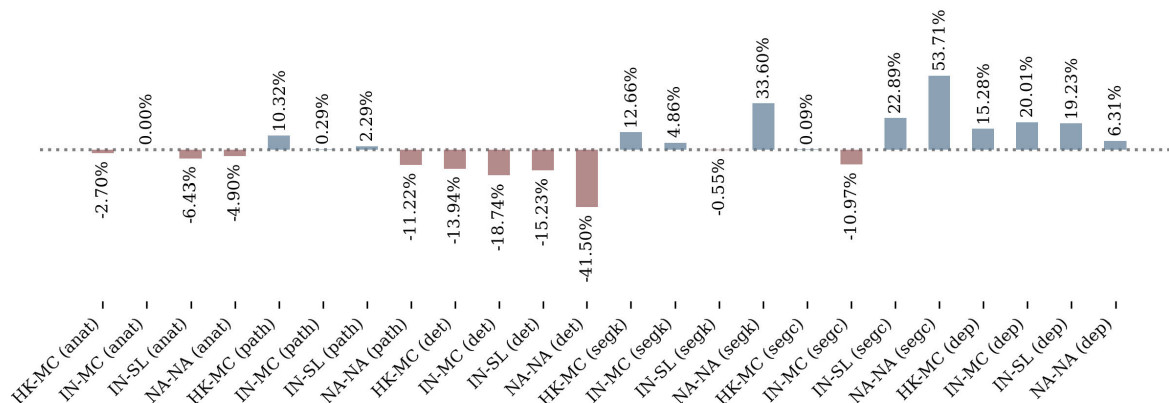


FIGURE 12. Improvement of ViT-B over ResNet50 for same pretraining pipeline (data and algorithm). For conciseness, we denote Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, supervised pretraining with *SL*, and no pretraining with *NA-NA*. Additionally, we refer to anatomical landmark recognition as *anat*, pathological finding characterisation as *path*, polyp detection as *det*, polyp segmentation with Kvasir-SEG as *segk*, polyp segmentation with CVC-ClinicDB as *segc*, and monocular depth estimation in colonoscopy as *dep*.

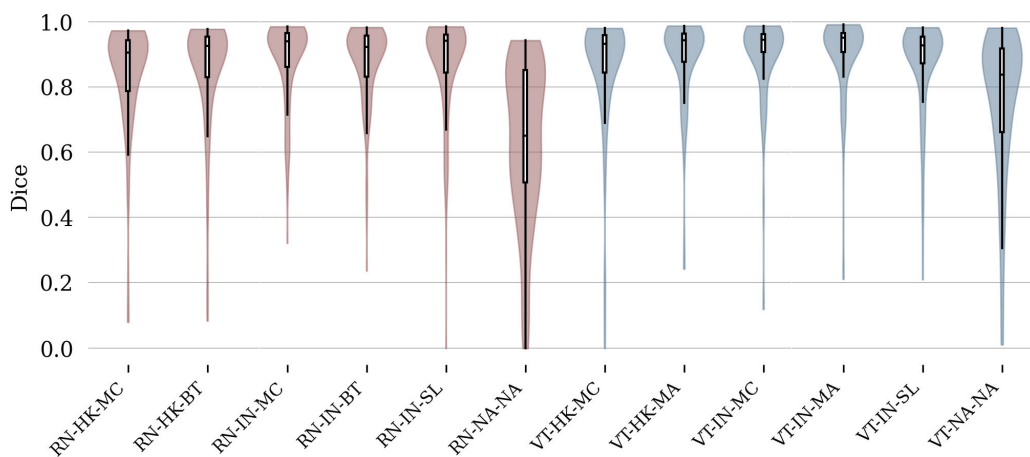


FIGURE 13. Distribution of Dice score (higher is better) across the test set for each Kvasir-SEG polyp segmentation model, visualised as box and violin plots. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, Barlow Twins with *BT*, MAE with *MA*, supervised pretraining with *SL*, and no pretraining with *NA-NA*. For clarity, the violin plots for ResNet50 models are coloured red and the violin plots for ViT-B models are coloured blue.

recognition models with MAE pretrained backbones shows only a marginal improvement for pretraining with Hyperkvasir-unlabelled vs. ImageNet-1k, the results for the depth estimation models are more significant. This implies that the similarity of the pretraining data to the data used in the depth estimation experiments is much more critical than the amount of pretraining data, in comparison to other tasks. While this finding is significant for the development of solutions to vision tasks in GIE, it may have broader implications and further work may find this to be true for monocular depth estimation in general.

Finally, we demonstrate that models with a ViT-B backbone are generally better than models with a ResNet50 backbone in polyp segmentation and monocular depth estimation in colonoscopy, generally worse in polyp detection, and generally similar in image classification. To assess this,

we use the same measures of error used in the previous analyses and evaluate the relative improvement from using a ViT-B vs. a ResNet50 using:

$$\%Improvement_{RN \rightarrow VT} = 100 \frac{\delta_{RN} - \delta_{VT}}{\delta_{RN}} \quad (30)$$

where δ_{VT} is the error for a model with a ViT-B backbone and δ_{RN} is the error for an equivalent model (same pretraining pipeline and task) with a ResNet50 backbone.

Note that this analysis omits any results for pretraining with Barlow Twins or MAE. We visualise the results of this analysis in Fig. 12, where it can be seen that the ResNet50 and ViT-B models perform similarly in anatomical landmark recognition and pathological finding characterisation, that the ResNet50 models perform better than the ViT-B models perform in polyp detection, and that the ViT-B models

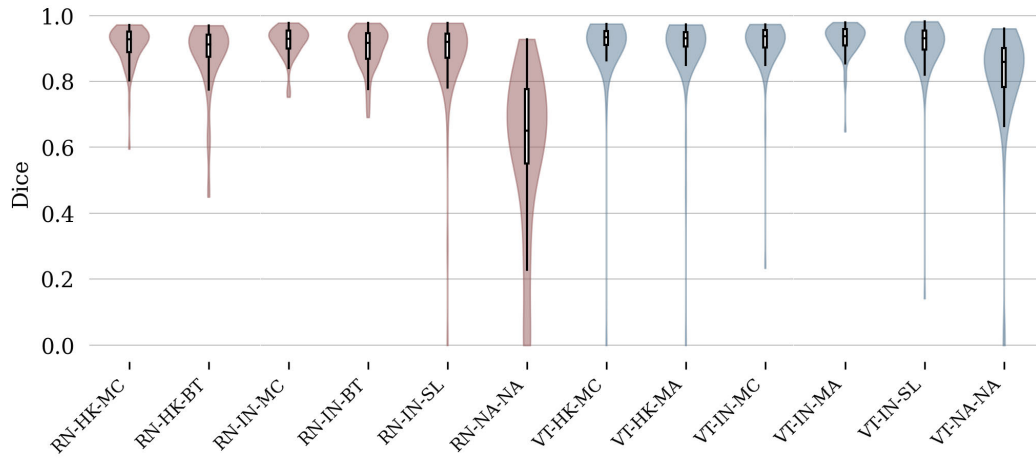


FIGURE 14. Distribution of Dice score (higher is better) across the test set for each CVC-ClinicDB polyp segmentation model, visualised as box and violin plots. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, Barlow Twins with *BT*, MAE with *MA*, supervised pretraining with *SL*, and no pretraining with *NA-NA*. For clarity, the violin plots for ResNet50 models are coloured red and the violin plots for ViT-B models are coloured blue.

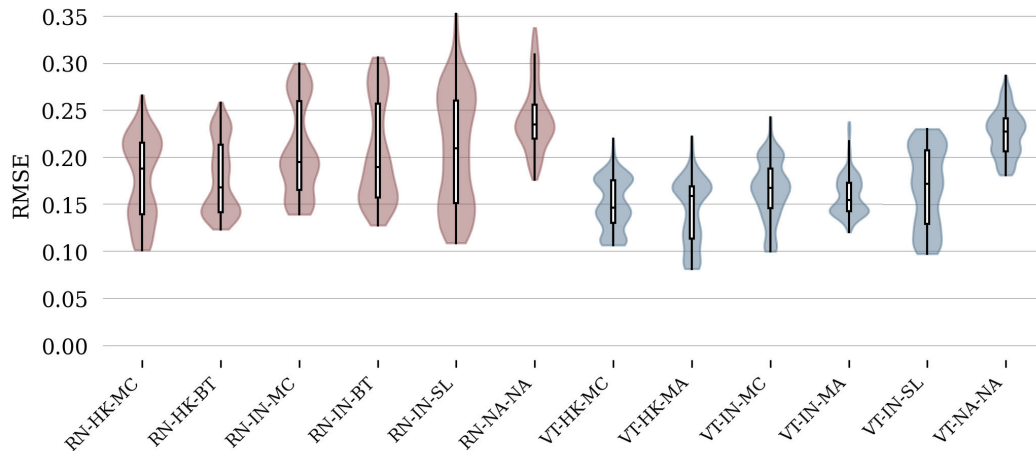


FIGURE 15. Distribution of RMSE (lower is better) across the test set for each C3VD monocular depth estimation model, visualised as box and violin plots. For conciseness, we denote ResNet50s with *RN*, ViT-Bs with *VT*, Hyperkvasir-unlabelled with *HK*, ImageNet-1k with *IN*, MoCo v3 with *MC*, Barlow Twins with *BT*, MAE with *MA*, supervised pretraining with *SL*, and no pretraining with *NA-NA*. For clarity, the violin plots for ResNet50 models are coloured red and the violin plots for ViT-B models are coloured blue.

generally perform better in the dense prediction tasks of polyp segmentation and monocular depth estimation colonoscopy. We further demonstrate the advantage of the ViT-B models over the ResNet50 models in dense prediction by visualising the distribution of performance across the Kvasir-SEG, CVC-ClinicDB, and C3VD test sets in Fig. 13, Fig. 14, and Fig. 15, respectively. Such visualisations are only suitable for these experiments since the metrics measure the performance on each instance prior to averaging, which is not the case for our image classification or object detection experiments. While we observe that ResNet50 models are typically better on polyp detection, we note that the polyp detection model with an MAE pretrained backbone with ImageNet-1k performs better than all but two models with ResNet50 backbones with respect to AP, and performs best with respect to AP₅₀, further

emphasising the particular robustness of this pretraining pipeline. There is still much to understand about the relative strengths and weaknesses of these architectures, particularly in the context of domains where the availability of data is much lower than that of everyday images, such as GIE. However, these results provide useful insights into which architecture may be better suited to each considered task.

One final note we make is that, as expected, pretraining with any of the considered pipelines consistently leads to better fine-tuned performance than training on the downstream task from random initialisation.

IX. CONCLUSION

In this work, we studied the pretraining of image encoders for use as backbones in solutions to vision tasks in GIE,

considering variation in encoder architecture, pretraining pipeline (data and algorithm), and downstream task. This was motivated by recent opportunities to improve on the convention of supervised pretraining backbones on image classification with ImageNet-1k, namely modern self-supervised pretraining algorithms and Hyperkvasir-unlabelled — a relatively large dataset of unlabelled GIE images. We primarily identified the best pretraining pipeline and architecture, out of those considered, for each considered task by adapting the encoders to the tasks with state-of-the-art decoders, fine-tuning the resulting models on datasets that include suitable annotations for the tasks, and evaluating the performance on test sets with well-established metrics. Overall, we found that a ViT-B backbone pretrained using the MAE algorithm and ImageNet-1k was most robust. Additionally, our findings suggest three general principles regarding the pretraining of encoders for use as backbones in solutions to vision tasks in GIE, which we revealed through an analysis of the downstream performance. These include:

- Self-supervised pretraining generally produces more suitable backbones than supervised pretraining. This result is significant as it is still the convention to use backbones that have been pretrained on ImageNet-1k in a supervised manner — this implies that the current state-of-the-art could be improved upon through self-supervised pretraining. Additionally, this result contrasts with the results observed for tasks involving everyday images, where supervised pretraining typically leads to better performance.
- Self-supervised pretraining with ImageNet-1k generally produces more suitable backbones than self-supervised pretraining with Hyperkvasir-unlabelled, with the notable exception of monocular depth estimation in colonoscopy where the similarity of the pretraining data to the downstream data appears to be more critical than the amount of pretraining data. While this is a useful insight for the development of monocular depth estimation models for GIE, this finding may also be true for monocular depth estimation solutions in other domains.
- That ResNet50 backbones are generally better for polyp detection, whereas ViT-B backbones are generally better for polyp segmentation and monocular depth estimation in colonoscopy, and both architectures perform similarly in anatomical landmark recognition and pathological finding characterisation.

We hope that this paper encourages further work on the topic of pretraining image encoders for use as backbones in solutions to vision tasks in GIE. Firstly, the scope of this work could be extended to more tasks and datasets, as well as decoder architectures and fine-tuning procedures. For example, we considered the Faster R-CNN object detection pipeline, which is a 2-stage detector, and it is worth investigating whether our findings are also true for 1-stage detectors. Additionally, we considered supervised fine-tuning

for monocular depth estimation in colonoscopy, while self-supervised fine-tuning for monocular depth estimation is also a promising research avenue and may benefit from an investigation into pretraining. Also, the impact of existing pretraining pipelines on the hybrid architectures that combine both convolutional and transformer components and that have found success in polyp segmentation can be investigated. We believe that such research should lay the groundwork for the development of backbones that are better suited to tasks in GIE, which should allow for significant advancement in the state-of-the-art. Beyond extending the scope of this study and the further investigation of existing pretraining algorithms, we suggest that future work also studies the development of pretraining algorithms specifically for this domain, as well as for other encoder architectures.

ACKNOWLEDGMENT

Data Access Statement: this publication is supported by multiple datasets which are openly available as cited in the ‘References’ section of this paper.

REFERENCES

- [1] F. Renna, M. Martins, A. Neto, A. Cunha, D. Libório, M. Dinis-Ribeiro, and M. Coimbra, “Artificial intelligence for upper gastrointestinal endoscopy: A roadmap from technology development to clinical practice,” *Diagnostics*, vol. 12, no. 5, p. 1278, May 2022.
- [2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [3] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, *arXiv:1503.02531*.
- [4] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.
- [5] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, “Scaling vision transformers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1204–1213.
- [6] S. Zhu, J. Gao, L. Liu, M. Yin, J. Lin, C. Xu, C. Xu, and J. Zhu, “Public imaging datasets of gastrointestinal endoscopy for artificial intelligence: A review,” *J. Digit. Imag.*, vol. 36, no. 6, pp. 2578–2601, Dec. 2023.
- [7] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 843–852.
- [8] M. Huh, P. Agrawal, and A. A. Efros, “What makes ImageNet good for transfer learning?” 2016, *arXiv:1608.08614*.
- [9] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1597–1607.
- [10] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9630–9640.
- [11] M. Oquab et al., “DINOv2: Learning robust visual features without supervision,” 2023, *arXiv:2304.07193*.
- [12] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15979–15988.
- [13] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 139, 2021, pp. 12310–12320.
- [14] X. Chen, S. Xie, and K. He, “An empirical study of training self-supervised vision transformers,” 2021, *arXiv:2104.02057*.
- [15] H. Bao, L. Dong, S. Piao, and F. Wei, “BEiT: BERT pre-training of image transformers,” 2021, *arXiv:2106.08254*.
- [16] A. Bardes, J. Ponce, and Y. LeCun, “VICReg: Variance-invariance-covariance regularization for self-supervised learning,” 2021, *arXiv:2105.04906*.

- [17] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 9912–9924.
- [18] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 21271–21284.
- [19] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15745–15753.
- [20] S.-C. Huang, A. Pareek, M. Jensen, M. P. Lungren, S. Yeung, and A. S. Chaudhari, "Self-supervised learning for medical image classification: A systematic review and implementation guidelines," *npj Digit. Med.*, vol. 6, no. 1, p. 74, Apr. 2023.
- [21] F. C. Ghesu, B. Georgescu, A. Mansoor, Y. Yoo, D. Neumann, P. Patel, R. S. Vishwanath, J. M. Balter, Y. Cao, S. Grbic, and D. Comaniciu, "Self-supervised learning from 100 million medical images," 2022, [arXiv:2201.01283](https://arxiv.org/abs/2201.01283).
- [22] S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, V. Natarajan, and M. Norouzi, "Big self-supervised models advance medical image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3458–3468.
- [23] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "Self-supervised learning for medical image analysis using image context restoration," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101539.
- [24] A. Kalapos and B. Gyires-Tóth, "Self-supervised pretraining for 2D medical image segmentation," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2022, pp. 472–484.
- [25] J. Xu, "A review of self-supervised learning methods in the field of medical image analysis," *Int. J. Image, Graph. Signal Process.*, vol. 13, no. 4, pp. 33–46, Aug. 2021.
- [26] S. Shurrab and R. Duwairi, "Self-supervised learning methods and applications in medical imaging analysis: A survey," *PeerJ Comput. Sci.*, vol. 8, p. e1045, Jul. 2022.
- [27] S. Ramesh, V. Srivastav, D. Alapatt, T. Yu, A. Murali, L. Sestini, C. I. Nwoye, I. Hamoud, S. Sharma, A. Fleurentin, G. Exarchakis, A. Karargyris, and N. Padoy, "Dissecting self-supervised learning methods for surgical computer vision," *Med. Image Anal.*, vol. 88, Aug. 2023, Art. no. 102844.
- [28] H. Borgli, V. Thambawita, P. H. Smedsrud, S. Hicks, D. Jha, S. L. Eskeland, K. R. Randel, K. Pogorelov, M. Lux, D. T. D. Nguyen, D. Johansen, C. Griwodz, H. K. Stensland, E. Garcia-Ceja, P. T. Schmidt, H. L. Hammer, M. A. Riegler, P. Halvorsen, and T. De Lange, "HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy," *Scientific Data*, vol. 7, no. 1, p. 283, Aug. 2020.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, [arXiv:2010.11929](https://arxiv.org/abs/2010.11929).
- [31] J. Y. Kim, G. Tangriberganov, W. Jung, D. S. Kim, H. S. Koo, S. Lee, and S. M. Kim, "Effective representation learning via the integrated self-supervised pre-training models of StyleGAN2-ADA and DINO for colonoscopy images," *BioRxiv*, pp. 2022–2026, Jun. 2022, doi: [10.1101/2022.06.15.496360](https://doi.org/10.1101/2022.06.15.496360).
- [32] L. T. T. Hong, N. C. Thanh, and T. Q. Long, "Self-supervised visual feature learning for polyp segmentation in colonoscopy images using image reconstruction as pretext task," in *Proc. 8th NAFOSTED Conf. Inf. Comput. Sci. (NICS)*, Dec. 2021, pp. 254–259.
- [33] N. Chi Thanh, "Colonoscopy image classification using self-supervised visual feature learning," *J. Mil. Sci. Technol.*, vol. CSC5E, no. 5, pp. 3–13, Dec. 2021.
- [34] R. Hirsch, M. Caron, R. Cohen, A. Livne, R. Shapiro, T. Golany, R. Goldenberg, D. Freedman, and E. Rivlin, "Self-supervised learning for endoscopic video analysis," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. Cham, Switzerland: Springer*, 2023, pp. 569–578.
- [35] H. Guo, S. A. Somayajula, R. Hosseini, and P. Xie, "Improving image classification of gastrointestinal endoscopy using curriculum self-supervised learning," *Sci. Rep.*, vol. 14, no. 1, p. 6100, Mar. 2024.
- [36] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, "A cookbook of self-supervised learning," 2023, [arXiv:2304.12210](https://arxiv.org/abs/2304.12210).
- [37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, [arXiv:1810.04805](https://arxiv.org/abs/1810.04805).
- [38] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, [arXiv:1807.03748](https://arxiv.org/abs/1807.03748).
- [39] X. Dong, J. Bao, T. Zhang, D. Chen, W. Zhang, L. Yuan, D. Chen, F. Wen, N. Yu, and B. Guo, "PeCo: Perceptual codebook for BERT pre-training of vision transformers," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 552–560.
- [40] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "SimMIM: A simple framework for masked image modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9643–9653.
- [41] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 8821–8831.
- [42] J. Opitz and S. Burst, "Macro F1 and macro F1," 2019, [arXiv:1911.03347](https://arxiv.org/abs/1911.03347).
- [43] G. Parmar, R. Zhang, and J.-Y. Zhu, "On aliased resizing and surprising subtleties in GAN evaluation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11410–11420.
- [44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, [arXiv:1711.05101](https://arxiv.org/abs/1711.05101).
- [45] D. Jha, P. H. Smedsrud, M. A. Riegler, P. Halvorsen, T. De Lange, D. Johansen, and H. D. Johansen, "Kvasir-SEG: A segmented polyp dataset," in *Proc. 26th Int. Conf. Multimedia Modeling*, Springer, 2020, pp. 451–462.
- [46] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [47] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
- [48] Y. Li, S. Xie, X. Chen, P. Dollár, K. He, and R. Girshick, "Benchmarking detection transfer learning with vision transformers," 2021, [arXiv:2111.11429](https://arxiv.org/abs/2111.11429).
- [49] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, Oct. 2022, pp. 280–296.
- [50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
- [51] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, and F. Vilariño, "WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Computerized Med. Imag. Graph.*, vol. 43, pp. 99–111, Jul. 2015.
- [52] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.
- [53] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12159–12168.
- [54] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Proc. Int. Workshop Deep Learn. Med. Image Anal., 7th Int. Workshop Multimodal Learn. Clin. Decis. Support. Cham, Switzerland: Springer*, Sep. 2017, pp. 240–248.
- [55] T. L. Bobrow, M. Golhar, R. Vijayan, V. S. Akshintala, J. R. Garcia, and N. J. Durr, "Colonoscopy 3D video dataset with paired depth from 2D-3D registration," *Med. Image Anal.*, vol. 90, Dec. 2023, Art. no. 102956.
- [56] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual U-Net," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 749–753, May 2018.
- [57] D. Jha, P. H. Smedsrud, M. A. Riegler, D. Johansen, T. D. Lange, P. Halvorsen, and H. D. Johansen, "ResUNet++: An advanced architecture for medical image segmentation," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2019, p. 225.

- [58] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, Mar. 2022.
- [59] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from Internet photos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2041–2050.
- [60] A. Rau, S. Bano, Y. Jin, P. Azagra, J. Morlana, E. Sanderson, B. J. Matuszewski, J. Young Lee, D.-J. Lee, E. Posner, N. Frank, V. Elangovan, S. Raviteja, Z. Li, J. Liu, S. Lalithkumar, M. Islam, H. Ren, J. M. M. Montiel, and D. Stoyanov, "SimCol3D - 3D reconstruction during colonoscopy challenge," 2023, *arXiv:2307.11261*.



EDWARD SANDERSON received the Ph.D. degree in data science from the University of Central Lancashire, Preston, U.K., in 2022. Since, he has been a Postdoctoral Researcher with the Computer Vision and Machine Learning (CVML) Group, University of Central Lancashire, on the Science and Technology Facilities Council (STFC) CDN+ funded "Machine Learning System for Decision Support and Computational Automation of Early Cancer Detection and Categorization in Colonoscopy (AIdDeCo)" Project. His research interests include machine learning, computer vision, and medical image processing. He has won several awards for his research, including the first place in the Depth Estimation Task of the SimCol3D EndoVis Challenge at the 2022 MICCAI, the second place in the Poster Competition at the 2023 STFC CDN+ Celebration Event, and the third place in the Best Paper Award at the 2022 MIUA.



BOGDAN J. MATUSZEWSKI (Member, IEEE) received the Ph.D. degree in electronic engineering from Wrocław University of Science and Technology, Poland, in 1996. He is currently a Professor of computer vision, the Deputy Director of the Research Centre in Engineering, and the Head of the Computer Vision and Machine Learning (CVML) Group, University of Central Lancashire, Preston, U.K. He has participated in 24 research projects, leading 11 of them, with funding secured from the U.K. Research Councils, EU, and industry. Most recently, he has been the Principal Investigator of the Science and Technology Facilities Council CDN+ funded "Machine Learning System for Decision Support and Computational Automation of Early Cancer Detection and Categorization in Colonoscopy (AIdDeCo)" Project. He has published over 150 research articles, won several awards for his research, and supervised 18 Ph.D. students to successful completion. His research interests include computer vision, data science, and AI (machine learning) within areas of imaging, healthcare technologies, digital engineering, deformation modeling, segmentation, registration, object characterization, and 3D scene reconstruction and understanding

...