Enhancing Vision Transformers on Small Datasets Using Self-Supervised Auxiliary Tasks

Alexandros Floros

Master of Science

School of Informatics

University of Edinburgh

2024

Abstract

Vision Transformers (ViTs) have emerged as a strong alternative to Convolutional Neural Networks (CNNs), which have been dominant in the field of computer vision over the past decade. Whilst ViTs excel across various tasks and perform well on medium to large datasets, they tend to underperform on smaller datasets. This is due to their lack of locality-focused inductive biases, which are inherent in CNNs, requiring ViTs to learn local features from excess data. Motivated by this limitation, this study explores the enhancement of ViTs, during supervised training on small datasets, through self-supervised techniques. Specifically, it experiments with the Dense Relative Localisation (DRLoc) task and introduces a Masked Embedding (ME) task, inspired by DRLoc and Masked Autoencoders (MAEs). Results demonstrate that ME consistently improves model performance on CIFAR, outperforming DRLoc on most baselines.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Alexandros Floros*)

Table of Contents

1 Introduction **1**

1.1 Vision Transformers and Limited Data 1

1.2 Aims and Objectives 2

2 Background **4**

2.1 Vision Transformers 4

2.2 Self-Supervised Learning 5

3 Methodology **8**

3.1 Experimental Setup 8

3.2 Dense Relative Localisation (DRLoc) Task 9

3.3 Masked Embedding (ME) Task 10

4 Results **13**

4.1 Overview 13

4.2 DRLoc Performance 13

4.3 ME Performance 14

4.4 Mixed Loss Performance 15

4.5 Cross-Comparison 16

4.6 Masking Ratio Ablation 17

4.7 Robustness to Adversarial Attacks 18

5 Conclusion **20**

5.1 Summary 20

5.2 Limitations and Future Work 20

Bibliography **22**

Chapter 1

Introduction

1.1 Vision Transformers and Limited Data

Vision Transformers (ViTs) have garnered significant interest as a rival to Convolutional Neural Networks (CNNs), which have been dominant in the field of computer vision since the success of AlexNet [1], being applied on tasks such as image classification [2], image segmentation [3] and object detection [4]. ViTs have already demonstrated their potential in such tasks [5, 6, 7] as well as more sophisticated ones like image generation [8]. They draw their inspiration from the original Transformer architecture [9], responsible for major breakthroughs of Natural Language Processing (NLP) in the following years such as GPT-4 [10]. One attractive aspect of ViTs is their potential to create a unified framework for processing both visual and textual information, driving multi-modal applications such as image captioning [11]. The original ViT architecture [5] divides an image into a grid of non-overlapping patches, each linearly projected into the input embedding space to create a patch token. These tokens are processed through a sequence of multi-head attention and feed-forward layers, in a similar way to how word tokens are handled in NLP Transformers.

One key advantage of ViTs is their ability to utilise the attention mechanism to capture global relationships between patch tokens, which contrasts with CNNs, where the receptive field of convolutional kernels limits the relationships that can be learned to local contexts. However, this greater representational capacity comes with the drawback of lacking the inherent inductive biases found in CNNs, such as locality, translation invariance, and the hierarchical structure of visual information [12, 13, 14]. ViTs

therefore tend to require significantly more training data as they learn local visual properties from samples. These properties, in CNNs, are instead modelled in their architecture [15].

To address this issue, various methods have been proposed through the development of newer generations of the ViT as well as different training strategies. A common approach is to combine convolutional layers with attention layers, thereby introducing priors for locality into the ViT [12, 13, 14]. These hybrid architectures offer the best of both worlds: attention layers capture long-range dependencies, whilst convolutional layers emphasise the local properties of the image content. Although these architectures have been proven capable of matching Residual Network (ResNet) [2] performance on medium-sized datasets such as ImageNet-1K [16], they are yet not able to reach the same performance on smaller datasets like CIFAR-10 [17] [18]. This dependency on large training sets presents a significant challenge in fields like medical imaging, environmental monitoring and surveillance, where available labeled data is often limited. Other approaches, such as self-supervised learning [19], have demonstrated great potential, enabling the acquisition of visual representations in unlabelled data via pretext tasks. Whilst these methods have often been used for pre-training large models, they may also be combined with the supervised learning paradigm to facilitate learning from small datasets [20]. This is especially important for domains in which fine-tuning may not be optimal due to substantial differences in samples to large-scale, general-purpose datasets.

1.2 Aims and Objectives

Focusing on the auxiliary role of self-supervised learning when training with labelled data, the aim of this thesis was to study how these methods perform on different architectures for classification under limited data regimes, specifically using the (relatively small) CIFAR-10 [17] and CIFAR-100 [21] datasets. The objectives were the following:

- Implementing the Dense Relative Localisation (DRLoc) [22] task for 32×32 images and investigating whether its distance-based learning objective on the final embedding grid remains beneficial when the spatial resolution of the input image is low.
- Developing a novel self-supervised task, inspired by the Masked Autoencoder (MAE) [23], that operates on the same embedding grid as DRLoc to assist in

capturing both local and global features.

- Assessing how these self-supervised losses impact classification accuracy when added to the standard cross-entropy, both individually and collectively, including an ablation study for different weight values and a brief investigation on robustness.

Chapter 2

Background

2.1 Vision Transformers

Although the attention mechanism had been previously applied on CNNs [24, 25], the first entirely transformer-based architectures for vision were iGPT [26] and ViT [5]. iGPT is trained with a self-supervised method involving masking pixels in an image and training a model to predict them, similarly to the masked-word task used in NLP architectures such as BERT [27] and GPT [28]. In contrast, ViT follows a supervised training approach, employing a special class token in the input, aggregating information from all image patches during the self-attention process, and a classification head that is connected to the final embedding of this token, shown in Figure 2.1. Both approaches are computationally intensive and, even though they deliver impressive results on large datasets, they fall short of CNN-based architectures when trained from scratch on medium-sized ones such as ImageNet-1K [18]. To minimise the dependence of ViTs on extensive training, DeiT [29] implemented comprehensive data augmentation, regularisation techniques and the use of distillation tokens, derived from CNNs.

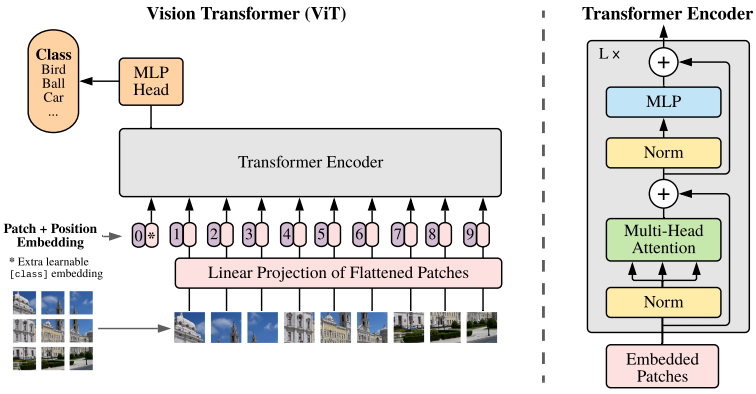

Figure 2.1: The ViT architecture [5].

ViT's success has gained significant attention in the computer vision scene, leading to the emergence of various architectural variants for a wide range of tasks. Despite that, the absence of inductive biases for locality in ViTs makes it challenging to train effectively without requring large datasets. Consequently, recent efforts have focused on developing newer generations of ViTs that combine convolutional operations with long-range attention layers. The central idea behind these hybrid architectures is to organise the sequence of token embeddings into a grid, where each embedding vector aligns with a specific location in the input image. This geometric arrangement allows convolutional layers to operate on neighbouring embeddings, thereby encouraging the network to capture local image features. The primary differences amongst these approaches lie in where the convolutional operations take place. This, for instance, can be in the initial representations [13], across all layers [12, 14] or in the query/key/value projections [14]. Similar to the first Transformer design proposed for NLP [9], the original ViT includes (absolute) positional embeddings to encode the order of input tokens. In some architectures, relative positional embedding is used, representing the position of each token in relation to the others. As mentioned in the previous chapter, these hybrid Transformer-CNN structures can perform similarly to strong CNNs when trained from scratch on ImageNet-1K, yet a performance gap persists on small datasets such as CIFAR-10 [18].

2.2 Self-Supervised Learning

Self-supervised learning initially gained traction in NLP, where it provided a way to replace expensive manual annotations by creating pretext tasks that allow the model

to learn from text itself [19]. A common example is BERT [27], which involves masking a word in a sentence and training the model to predict the missing word. In computer vision, contrastive methods like SimCLR [30] and MoCo [31] have been used to minimise the distance between augmented versions of the same image (positive pairs) whilst maximising it for different images (negative pairs). Non-contrastive approaches such as BYOL [32] and DINO [33] focus solely on minimising the distance between positive pairs, without requiring negative pairs. Reconstruction-based methods have also been proven effective for self-supervised learning in computer vision [23, 34]. These methods typically involve an encoder that processes a portion of an image to generate a latent representation, which a decoder then uses to reconstruct the original image from the latent representation. A notable example is Masked Autoencoders (MAEs) [23], which work by masking parts of an input image, encoding the remaining visible patches and then reconstructing the masked regions. An illustration of this is shown is Figure 2.2, in which the encoder and decoder are jointly trained. During inference, only the encoder is used, which has learned to efficiently encode images for downstream tasks. Another approach is to predict the correct arrangement of a scrambled grid of 3×3 image patches, a task known as Jigsaw [34], which was inspired by deshuffling in NLP [35].

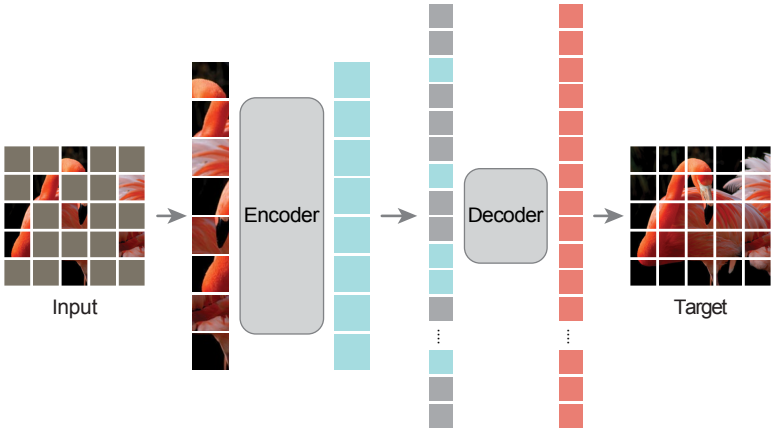

Figure 2.2: The Masked Autoencoder (MAE) task [23].

Although self-supervised learning is often used to pre-train ViTs on a large scale, boosting their performance on various downstream tasks, these pretext tasks can also be combined with supervised training to regularise the training process, a form of multi-task learning [20]. This strategy can help improve a model's performance without altering its architecture. Dense Relative Localisation (DRLoc) [22], for instance, is an auxiliary

self-supervised task that enhances the robustness of ViTs on smaller datasets. It uses a classifier predicting the relative distances amongst token embeddings, with a loss function derived from the offset predictions that complements the standard cross-entropy computed from the image classification task. This multi-task approach eliminates the need for extensive pre-training and helps ViTs learn meaningful representations with limited data. A diagram of this is shown in Figure 2.3. The MAE and jigsaw tasks have also been incorporated into the multi-task framework [36, 37], demonstrating the potential of these self-supervised auxiliary methods to play a vital role in enhancing ViTs in situations where annotated data for a specialised domain is limited and large-scale fine-tuning ineffective in reaching ResNet performance.

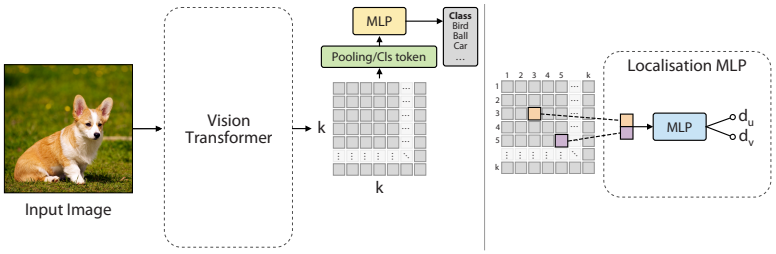

Figure 2.3: The Dense Relative Localisation (DRLoc) task [22].

Chapter 3

Methodology

3.1 Experimental Setup

This study explores the use of self-supervised auxiliary tasks to enhance different models for image classification. Specifically, the experiments utilise Dense Relative Localisation (DRLoc) [22] and a task operating on the former's embedding space, inspired by Masked Autoencoders (MAEs) [23]. This is being referred to here as the Masked Embedding (ME) task. Four baseline architectures were employed, each adjusted to have approximately 15 million trainable parameters. To ensure a fair comparison, the supplementary modules used for self-supervised training were allocated an equivalent number of parameters for each baseline. The architectures include a custom ResNet [2] composed entirely of residual blocks with convolutional layers, the original Vision Transformer (ViT) [5] and two second-generation hybrid models: Tokens-to-Token (T2T) [13] and Convolutional Vision Transformer (CvT) [14]. Since these methods are used as part of a multi-task learning framework, acting as regularisation for supervised training, the underlying architectures remained unchanged. The models were trained and evaluated on the CIFAR-10 [17] and CIFAR-100 [21] datasets, each of which contains 50,000 training images and 10,000 test images, with 10 and 100 balanced classes, respectively. Both datasets consist of images with a resolution of 32×32. The training process utilised the Adam optimizer with a learning rate of 10^{-4}, a batch size of 128 and 100 epochs. All experiments were carried on a single NVIDIA A100 GPU, with 80GB VRAM, on the university's Eddie cluster.

3.2 Dense Relative Localisation (DRLoc) Task

The DRLoc task works by densely sampling multiple pairs of embeddings from each image and having a simple network predict their relative distances. This is used jointly with supervised learning to train the model to generate features that are more informative, capturing both local and global information.

More specifically, let $\mathbf{X}$ be an image batch, where $\mathbf{X} \in \mathbb{R}^{b \times c \times h \times w}$, with b, c, h and w being the batch size, number of channels, height and width, respectively. The baseline model used is f, where f_{emb} is the sequence of layers up until the last feature map (before the classification head). This model encodes the image to generate embeddings $\mathbf{E}$:

$$\mathbf{E} = f_{emb}(\mathbf{X}), \quad \mathbf{E} \in \mathbb{R}^{b \times c_{emb} \times h_{emb} \times w_{emb}} \tag{3.1}$$

During training, multiple pairs of embeddings from $\mathbf{E}$ are randomly sampled. For every pair $(\mathbf{e}_{i,j}, \mathbf{e}_{l,m})$, where $\mathbf{e}_{i,j} \in \mathbb{R}^{b \times c_{emb}}$, the normalised target distance vector $(t_u, t_v)^T$ is as follows:

$$t_u = \frac{|i-l|}{h_{emb}}, \quad t_v = \frac{|j-m|}{w_{emb}}, \quad 1 \leq i, j \leq h_{emb}, w_{emb}, \quad (t_u, t_v)^T \in [0,1]^2 \tag{3.2}$$

The two embedding vectors $\mathbf{e}_{i,j}$ and $\mathbf{e}_{l,m}$ are concatenated and fed into a small Multi-Layer Perceptron (MLP) g, which has two hidden layers and two output neurons (one for each spatial dimension). This MLP predicts the relative offset between points (i, j) and (l, m) on the matrix as shown below:

$$(d_u, d_v)^T = g(\mathbf{e}_{i,j}, \mathbf{e}_{l,m})^T \tag{3.3}$$

The dense relative localisation loss $\mathcal{L}_{drloc}$ is defined as follows:

$$\mathcal{L}_{drloc} = \mathbb{E}_{(\mathbf{e}_{i,j}, \mathbf{e}_{l,m}) \sim \mathbf{E}}[|(t_u, t_v)^T - (d_u, d_v)^T|_1] \tag{3.4}$$

In the above equation, for each image batch $\mathbf{X}$, the expectation is calculated by uniformly sampling n pairs $(\mathbf{e}_{i,j}, \mathbf{e}_{l,m})$ from $\mathbf{E}$ and averaging the L_1 loss between the respective $(t_u, t_v)^T$ and $(d_u, d_v)^T$. Throughout this project, the sample size n is fixed at 32. The

dense relative localisation loss $\mathcal{L}_{drloc}$ is weighted by a hyperparameter λ_{drloc} and added to the cross-entropy loss $\mathcal{L}_{ce}$ of each model. The overall loss is thus given by:

$$\mathcal{L}_{tot} = \mathcal{L}_{ce} + \lambda_{drloc}\mathcal{L}_{drloc} \tag{3.5}$$

Experiments used a range of values between 0.025 and 2 for λ_{drloc}.

In the paper that introduced DRLoc [22], experiments assumed a resolution of 224×224 ($h = w = 224$), and all datasets had been scaled to accommodate this, leading to a 7×7 embedding grid ($h_{emb} = w_{emb} = 7$). In this work, since the 32×32 images from CIFAR are used unmodified, some minimal adjustments to architectural hyperparameters were made to obtain an 8×8 embedding grid. The grid's embedding dimension c_{emb} was set to 312 for all models. Having the exact same grid dimensions across experiments ensures that both the DRLoc task and the ME task discussed below behave consistently and that the number of parameters associated with them remains the same.

3.3 Masked Embedding (ME) Task

The recent success of MAEs as an auxiliary task for enhancing ViTs [36], along with DRLoc's focus on the latent space, has inspired the development of a Masked Embedding (ME) task as part of this project. In the original MAE [23], patches of an image are masked during training before being fed into the model (encoder), and the unmasked image is reconstructed by a decoder to calculate the Mean Squared Error (MSE) loss between the original and reconstructed image. However, the modified ME task does not apply masking on the input image but directly on its latent representation. The decoder does not reconstruct the image but instead attempts to restore the missing embeddings, with the loss function being the MSE between the original and reconstructed latent grids. An illustration of this task is shown in Figure 3.1. This approach was selected primarily for the following reasons:

- In tasks such as image classification, not all patches of an image may be useful for correctly identifying the class. For example, in an image of a cat, patches containing background elements like sky or grass offer little information about its presence. Masking embeddings, which represent higher-level features extracted from the image patches, allows the model to focus on reconstructing and learning from the most informative parts of the image, such as the cat's fur or facial

features, as opposed to redundant features. This gives the model a task that is consistently challenging.

- Masking embeddings is computationally more efficient than masking patches as the encoder only needs to be used once for every training batch instead of twice.
- Especially in smaller datasets such as CIFAR-10, which are prone to overfitting, masking embeddings can improve generalisation performance by encouraging the model learn more robust and meaningful features rather than relying on specific, possibly noisy details in the data.

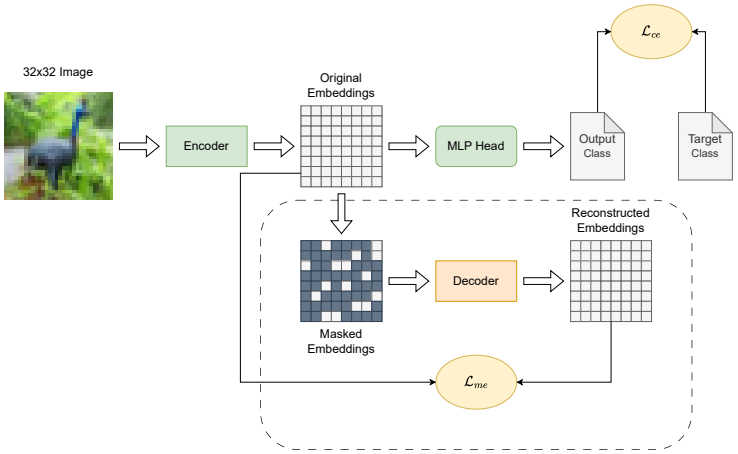

Figure 3.1: The Masked Embedding (ME) task.

Just like the traditional MAE, the ME variant discussed consists of an encoder, belonging to the host architecture, and a decoder, which is trained alongside the encoder and is ignored during inference. Encoding is done as discussed for DRLoc, assuming the previous definitions, with the same image batch $\mathbf{X}$ (where $\mathbf{X} \in \mathbb{R}^{b \times c \times h \times w}$) being encoded by f_{emb} to obtain the embedding grid $\mathbf{E}$ (where $\mathbf{E} \in \mathbb{R}^{b \times c_{emb} \times h_{emb} \times w_{emb}}$). A masking function $\mathcal{M}_r$ is applied on the embeddings, which masks a ratio r from $\mathbf{E}$, giving the compressed tensor $\mathbf{Z}$. The value of r in the main experiments was set to 0.75, as done in the paper proposing MAE:

$$\mathbf{Z} = \mathcal{M}_r(\mathbf{E}), \quad \mathbf{Z} \in \mathbb{R}^{s \times b \times c_{emb}}, \quad s = h_{emb} w_{emb} r, \quad r \in [0,1] \tag{3.6}$$

The decoder $\mathcal{D}$ attempts to reconstruct the embeddings $\mathbf{E}$ by giving a prediction $\hat{\mathbf{E}}$:

$$\hat{\mathbf{E}} = \mathcal{D}(\mathbf{Z}), \quad \hat{\mathbf{E}} \in \mathbb{R}^{b \times c_{emb} \times h_{emb} \times w_{emb}} \tag{3.7}$$

Within the decoder, trainable parameters serving as mask tokens are introduced in the masked positions, padding $\mathbf{Z}$ to a tensor $\mathbf{Z}'$, where $\mathbf{Z}' \in \mathbb{R}^{h_{emb} w_{emb} \times b \times c_{emb}}$. Learnable positional embeddings are then added to $\mathbf{Z}'$ to encode information about order. $\mathbf{Z}'$ is then reshaped to be $b \times h_{emb} w_{emb} \times c_{emb}$. The latter is fed to a sequence of 4 transformer blocks, each with 4 attention heads, and the output is further reshaped to give $\hat{\mathbf{E}}$.

Let $\mathbf{M}$ be a tensor of zeros, where $\mathbf{M} \in \mathbb{R}^{b \times c_{emb} \times h_{emb} \times w_{emb}}$, for which entries corresponding to masked tokens are set to one. The masked embedding loss $\mathcal{L}_{me}$ is defined as follows:

$$\mathcal{L}_{me} = \frac{\mathbb{E}[(\mathbf{E} - \hat{\mathbf{E}})^2 \odot \mathbf{M}]}{r} \tag{3.8}$$

The above is similar to the MAE loss, with the original and reconstructed embedding grids $\mathbf{E}$ and $\hat{\mathbf{E}}$ being used as opposed to image batches $\mathbf{X}$ and $\hat{\mathbf{X}}$. The element-wise multiplication with $\mathbf{M}$ ensures that the loss is evaluated only at the embeddings being masked, and the masking ratio r is used in the denominator to normalise its value. Just like with DRLoc, the masked embedding loss $\mathcal{L}_{me}$ is weighted by a hyperparameter λ_{me} and added to the cross-entropy loss $\mathcal{L}_{ce}$ of each model. The overall loss is hence given by:

$$\mathcal{L}_{tot} = \mathcal{L}_{ce} + \lambda_{me} \mathcal{L}_{me} \tag{3.9}$$

Similarly to DRLoc, experiments for ME used a λ_{me} ranging between 0.025 and 2.

Chapter 4

Results

4.1 Overview

This chapter presents the results obtained by applying the DRLoc and ME self-supervised tasks on the four baselines. The models were evaluated based on classification accuracy on the CIFAR-10 and CIFAR-100 test sets. The following tables compare accuracy obtained solely from supervised training (i.e. using only cross-entropy loss) with the total self-supervised loss, dictated by parameters λ_{drloc} and λ_{me}. The results primarily focus on experiments keeping the masking ratio r fixed at 0.75 and varying the lambda values, as well as combining them. However, an ablation study is also included at the end of the chapter, experimenting with different masking ratios.

4.2 DRLoc Performance

Table 4.1 shows the effect of DRLoc on the baseline models. The best accuracy values are highlighted in bold. For ResNet there is an evident increase in accuracy, which is highest at $\lambda_{drloc} = 0.1$. Results show an increase of 4% and 3% on CIFAR-10 and CIFAR-100, respectively, hinting that the DRLoc task provides the model with a better capacity to capture global features. Despite that, it is proven ineffective on the original ViT on both datasets and on T2T, for CIFAR-10. The reason the task appears to confuse the model in these cases could be due to nonoptimal selection of λ_{drloc}. On the contrary, T2T's performance on CIFAR-100 with $\lambda_{drloc} = 0.2$, and CvT's on both datasets, with $\lambda_{drloc} = 0.1$ and $\lambda_{drloc} = 0.025$, respectively, improve, albeit not by more than 2%. The last two, being hybrid transformer models with built-in locality, are not expected to

benefit as much as the first two, which have weaker inductive biases.

Dataset		CIFAR-10	CIFAR-100
Model	Loss	Classification Accuracy	
ResNet	$\mathcal{L}_{ce}$	78.65%	35.08%
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{drloc}$	80.28%	35.56%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	81.68%	36.72%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	**82.90%**	**38.60%**
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{drloc}$	79.24%	33.96%
ViT	$\mathcal{L}_{ce}$	**62.45%**	**33.44%**
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{drloc}$	60.57%	33.21%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	61.80%	32.81%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	61.08%	32.71%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{drloc}$	61.10%	32.97%
T2T	$\mathcal{L}_{ce}$	**74.12%**	41.53%
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{drloc}$	73.01%	42.42%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	72.45%	41.04%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	72.72%	42.49%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{drloc}$	73.46%	**43.09%**
CvT	$\mathcal{L}_{ce}$	71.28%	39.30%
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{drloc}$	71.69%	**40.21%**
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	71.31%	39.46%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	**71.90%**	38.73%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{drloc}$	71.26%	39.43%

Table 4.1: Comparison of models using DRLoc, with different values of λ_{drloc}.

4.3 ME Performance

Table 4.2 illustrates how ME impacts model performance, using the same setup as before. As in Table 4.1, the best accuracies are highlighted in bold. ResNet appears to benefit substantially from the information acquired through demasking embeddings, with a 5% and 12% increase, on CIFAR-10 and CIFAR-100, respectively, using $\lambda_{me} = 0.2$. Using the same λ_{me}, the task improves accuracy in the vanilla ViT by 1% on CIFAR-10 and 2% on CIFAR-100. Just like in the DRLoc case, performance increase is even smaller in the 2nd generation transformers, with a trivial increase for T2T on CIFAR-10 and less than 2% on CIFAR-100. The task appears to hinder feature learning on CvT, something that could be attributed to nonoptimal choice of hyperparameters, such as

λ_{me} or the masking ratio.

Dataset		CIFAR-10	CIFAR-100
Model	Loss	Classification Accuracy	
ResNet	$\mathcal{L}_{ce}$	78.65%	35.08%
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{me}$	82.58%	40.58%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	83.17%	43.28%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	83.62%	44.32%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{me}$	**83.83%**	**47.39%**
ViT	$\mathcal{L}_{ce}$	62.45%	33.44%
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{me}$	62.36%	33.51%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	62.45%	34.02%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	**63.82%**	34.52%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{me}$	**63.82%**	**35.56%**
T2T	$\mathcal{L}_{ce}$	74.12%	41.53%
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{me}$	**74.13%**	**43.14%**
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	74.10%	42.76%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	72.71%	42.22%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{me}$	73.37%	42.65%
CvT	$\mathcal{L}_{ce}$	**71.28%**	**39.30%**
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{me}$	70.28%	37.80%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	70.34%	38.40%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	70.06%	37.79%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{me}$	70.76%	37.23%

Table 4.2: Comparison of models using ME, with different values of λ_{me}.

4.4 Mixed Loss Performance

Table 4.3 compares the accuracies from Tables 4.1 and 4.2 for λ_{drloc} and λ_{me} values of 0.05 and 0.1, respectively, with the losses obtained by combining the two self-supervised tasks. The best performances are denoted in bold. ResNet's accuracy is increased by 6% on CIFAR-10 and 9% on CIFAR-100. In both cases, the mixed loss with λ_{drloc} and λ_{me} set to 0.1 yields better performance than each term on its own, suggesting that they complement one another in the features they teach the model. On the other hand, ViT, T2T and CvT do not seem to benefit from the hybrid self-supervised loss even further. The introduction of a third task likely confuses these models, which are already sensitive to just a second term being added. Nonetheless, the limited data points

obtained for this ternary loss may be insufficient to draw robust conclusions, and a more in-depth investigation would be needed, acquiring results for different combinations of λ_{drloc} and λ_{me}.

Dataset		CIFAR-10	CIFAR-100
Model	Loss	Classification Accuracy	
ResNet	$\mathcal{L}_{ce}$	78.65%	35.08%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	81.68%	36.72%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	82.90%	38.60%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	83.17%	43.28%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	83.62%	44.32%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc} + 0.05 \times \mathcal{L}_{me}$	83.58%	44.27%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	**84.71%**	**44.99%**
ViT	$\mathcal{L}_{ce}$	62.45%	33.44%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	61.80%	32.81%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	61.08%	32.71%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	62.45%	34.02%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	**63.82%**	**34.52%**
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc} + 0.05 \times \mathcal{L}_{me}$	63.21%	33.40%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	63.68%	34.15%
T2T	$\mathcal{L}_{ce}$	**74.12%**	41.53%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	72.45%	41.04%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	72.72%	42.49%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	74.10%	**42.76%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	72.71%	42.22%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc} + 0.05 \times \mathcal{L}_{me}$	72.17%	42.61%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	72.42%	42.64%
CvT	$\mathcal{L}_{ce}$	71.28%	39.30%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	71.31%	**39.46%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	**71.90%**	38.73%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	70.34%	38.40%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	70.06%	37.79%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc} + 0.05 \times \mathcal{L}_{me}$	70.32%	38.69%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	70.67%	38.11%

Table 4.3: Comparison of models using DRLoc, ME and both, with different values of λ_{drloc} and λ_{me}.

4.5 Cross-Comparison

The best accuracies from Tables 4.1, 4.2 and 4.3, with non-zero λ_{drloc} and λ_{me}, have been placed in Table 4.4. Bold is used to highlight the overall highest values on each baseline. With the exception of CvT, in which the ME task was unsuccessful at enhancing the model, ME appears to outperform DRLoc, especially on ResNet, which

is weaker at capturing global information on its own.

Dataset		CIFAR-10	Dataset		CIFAR-100
Model	Loss	Classification Accuracy	Model	Loss	Classification Accuracy
ResNet	$\mathcal{L}_{ce}$	78.65%	ResNet	$\mathcal{L}_{ce}$	35.08%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	82.90%		$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	38.60%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{me}$	83.83%		$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{me}$	**47.39%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	**84.71%**		$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	44.99%
ViT	$\mathcal{L}_{ce}$	62.45%	ViT	$\mathcal{L}_{ce}$	33.44%
	$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc}$	61.80%		$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{drloc}$	33.21%
	$\mathcal{L}_{ce} + \{0.1, 0.2\} \times \mathcal{L}_{me}$	**63.82%**		$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{me}$	**35.56%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	63.68%		$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	34.15%
T2T	$\mathcal{L}_{ce}$	74.12%	T2T	$\mathcal{L}_{ce}$	41.53%
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{drloc}$	73.46%		$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{drloc}$	43.09%
	$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{me}$	**74.13%**		$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{me}$	**43.14%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	72.42%		$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	42.64%
CvT	$\mathcal{L}_{ce}$	71.28%	CvT	$\mathcal{L}_{ce}$	39.30%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	**71.90%**		$\mathcal{L}_{ce} + 0.025 \times \mathcal{L}_{drloc}$	**40.21%**
	$\mathcal{L}_{ce} + 0.2 \times \mathcal{L}_{me}$	70.76%		$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{me}$	38.40%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	70.67%		$\mathcal{L}_{ce} + 0.05 \times \mathcal{L}_{drloc} + 0.05 \times \mathcal{L}_{me}$	38.69%

Table 4.4: Comparison of the best models using DRLoc, ME and both, with different values of λ_{drloc} and λ_{me}.

4.6 Masking Ratio Ablation

Table 4.5 provides accuracies obtained on CIFAR-10 for the previous baselines with the ME task when λ_{me} is set to 0.1 and the masking ratio r is changed. The highest values appear in bold. ResNet and and T2T perform better when 85% of the embeddings are masked, whereas the best value reimains at 0.75 for ViT. CvT accuracy is consistently below the standard cross-entropy. Varying the masking ratio in the range 0.65 to 0.90 does not seem to significantly influence learning.

Dataset		CIFAR-10
Model	Loss	Classification Accuracy
ResNet	$\mathcal{L}_{ce}$	78.65%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.65$	83.57%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.70$	83.06%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.75$	83.62%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.80$	81.90%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.85$	**83.80%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.90$	83.37%
ViT	$\mathcal{L}_{ce}$	62.45%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.65$	63.23%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.70$	63.72%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.75$	**63.82%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.80$	62.97%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.85$	62.86%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.90$	63.33%
T2T	$\mathcal{L}_{ce}$	74.12%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.65$	73.85%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.70$	73.43%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.75$	72.71%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.80$	72.96%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.85$	**74.32%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.90$	73.90%
CvT	$\mathcal{L}_{ce}$	**71.28%**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.65$	70.27%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.70$	70.36%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.75$	70.06%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.80$	70.06%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.85$	69.95%
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}, \quad r = 0.90$	70.48%

Table 4.5: Comparison of models using ME, with different values of r.

4.7 Robustness to Adversarial Attacks

To assess the robustness of features between the two self-supervised methods, the images in the test set were altered using adversarial noise [38] to artificially worsen classification accuracy. More specifically, let $\mathbf{X}$ be a batch of images, with $\mathbf{X} \in \mathbb{R}^{b \times c \times h \times w}$, where b is the batch size, c the number of channels, h the image height and w the image width. Furthermore, let $\mathbf{Y}$ be a batch of class labels, with $\mathbf{Y} \in \mathbb{R}^{b \times k}$, and $\hat{\mathbf{Y}}$ is a batch of predictions, with $\hat{\mathbf{Y}} \in \mathbb{R}^{b \times k}$, where k the number of classes. The gradient of the

cross-entropy loss $\mathcal{L}_{ce}(\hat{\mathbf{Y}}, \mathbf{Y})$ with respect to the image batch $\mathbf{X}$ is computed, indicating the direction in which the input should be altered to increase the loss:

$$\mathbf{g} = \nabla_{\mathbf{X}} \mathcal{L}_{ce}(\hat{\mathbf{Y}}, \mathbf{Y}), \quad \mathbf{g} \in \mathbb{R}^{b \times c \times h \times w} \tag{4.1}$$

The image batch is then updated as follows:

$$\mathbf{X}_{\text{adv}} = \mathbf{X} + \varepsilon \cdot \text{sign}(\mathbf{g}), \quad \mathbf{X}_{\text{adv}} \in \mathbb{R}^{b \times c \times h \times w} \tag{4.2}$$

where ε is a small value controlling the impact of the adversarial attack, set to $\frac{1}{255}$ here. This corresponds to at most a single pixel change in the images and results in adversarial perturbations that are imperceptible.

Table 4.6 demonstrates how this alteration of the CIFAR-10 test set impacts model performance when the DRLoc, ME and mixed losses have been employed with λ_{drloc} and λ_{me} set to 0.1. Bold is used where the reduction in accuracy is lowest. The results hint that DRLoc produces embeddings that are more robust to such attacks in all four models.

Dataset		CIFAR-10	Adversarial CIFAR-10
Model	Loss	Classification Accuracy	
ResNet	$\mathcal{L}_{ce}$	78.65%	50.93% (-27.72)
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	82.90%	61.04% **(-21.86)**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	83.62%	55.48% (-28.14)
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	84.71%	59.09% (-25.62)
ViT	$\mathcal{L}_{ce}$	62.45%	34.13% (-28.32)
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	61.08%	35.48% **(-25.60)**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	63.82%	36.69% (-27.13)
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	63.68%	37.78% (-25.90)
T2T	$\mathcal{L}_{ce}$	74.12%	35.65% (-38.47)
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	72.72%	34.43% **(-38.29)**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	72.71%	34.11% (-38.60)
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	72.42%	32.19% (-40.23)
CvT	$\mathcal{L}_{ce}$	71.28%	38.95% (-32.33)
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc}$	71.90%	40.11% **(-31.79)**
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{me}$	70.06%	37.45% (-32.61)
	$\mathcal{L}_{ce} + 0.1 \times \mathcal{L}_{drloc} + 0.1 \times \mathcal{L}_{me}$	70.67%	36.95% (-33.72)

Table 4.6: Comparison of models using DRLoc, ME and both, with different values of λ_{drloc} and λ_{me}, in terms of robustness to adversarial inputs.

Chapter 5

Conclusion

5.1 Summary

Vision Transformers (ViTs) have proven to be a powerful alternative to Convolutional Neural Networks (CNNs), particularly on medium to large datasets. However, their lack of locality priors, a strength of CNNs, limits their effectiveness on smaller datasets. This study attempts to address this challenge by incorporating self-supervised learning techniques into supervised training, specifically through the Dense Relative Localisation (DRLoc) task and a novel Masked Embedding (ME) task drawing inspiration from DRLoc and Masked Autoencoders (MAEs). The findings reveal that the ME task consistently enhances ViT performance across the baselines, offering a promising approach to improving ViTs' capability on small datasets. Given the above results, there are numerous refinements that could be applied on the methodology presented here as well as directions for further work. These are outlined in this chapter.

5.2 Limitations and Future Work

Due to time and resource constraints, certain potential improvements and topics for further exploration were not addressed in this research. The following recommendations are presented for consideration in future work:

- **In-Depth Hyperparameter Tuning:** A more extensive investigation could be carried regarding the hyperparameters used in the DRLoc and ME losses. A wider range of coefficients λ_{drloc} and λ_{me}, and masking ratio r can be used. This would

be especially beneficial in the ternary loss previously studied where insight was limited. Other hyperparameters that could be tweaked are the spatial and channel dimensions of the embedding grid $\mathbf{E}$, as well as the ME decoder's number of transformer blocks and attention heads.

- **More Datasets:** All the experiments presented here were based on the CIFAR-10 and CIFAR-100 datasets, which are both limited to 32×32 images. A wider range of spatial resolutions could be used to acquire a more holistic view of how the ME loss performs. Furthermore, having larger images would make the distinction between local and global features more meaningful.
- **More Baselines:** Additional baseline models could be used to perform enhancements, such as the Shifted window (Swin) ViT [12], CCT [39], SL-ViT [40] and DHVT [18]. Moreover, the auxiliary ME task introduced could be compared and combined with MAE [36] and Jigsaw [37].
- **Other Learning Tasks:** New self-supervised tasks could be introduced, potentially inspired from existing ones in computer vision or NLP. Similarly to this work's direction, existing tasks previously applied on images could be adapted to operate on the embedding space.

Bibliography

[1] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. V. Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," 2018.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[5] 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 16x16 words: Transformers for image recognition at scale," 2021.

[6] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," 2021.

[7] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.

[8] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two pure transformers can make one strong gan, and that can scale up," 2021.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[10] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin,

S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder,

J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph, "Gpt-4 technical report," 2024.

[11] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," 2019.

[12] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[13] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," 2021.

[14] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," 2021.

[15] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?," 2022.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2015.

[17] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research),"

[18] Z. Lu, H. Xie, C. Liu, and Y. Zhang, "Bridging the gap between vision transformers and convolutional neural networks on small datasets," 2022.

[19] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, "A survey on self-supervised learning: Algorithms, applications, and future trends," 2024.

[20] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," 2020.

[21] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-100 (canadian institute for advanced research),"

[22] Y. Liu, E. Sangineto, W. Bi, N. Sebe, B. Lepri, and M. D. Nadai, "Efficient training of visual transformers with small datasets," 2021.

[23] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021.

[24] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, "A survey on self-supervised learning: Algorithms, applications, and future trends," 2024.

[25] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[26] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 1691–1703, PMLR, 13–18 Jul 2020.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[28] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018.

[29] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers distillation through attention," 2021.

[30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[31] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2020.

[32] 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," 2020.

[33] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021.

[34] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," 2017.

[35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[36] S. Das, T. Jain, D. Reilly, P. Balaji, S. Karmakar, S. Marjit, X. Li, A. Das, and M. S. Ryoo, "Limited data, unlimited potential: A study on vits augmented by masked autoencoders," 2023.

[37] Y. Chen, X. Shen, Y. Liu, Q. Tao, and J. A. K. Suykens, "Jigsaw-vit: Learning jigsaw puzzles in vision transformer," 2023.

[38] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[39] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, "Escaping the big data paradigm with compact transformers," 2022.

[40] S. H. Lee, S. Lee, and B. C. Song, "Vision transformer for small-size datasets," 2021.