SiT: Self-supervised vIsion Transformer

Sara Atito, *Member IEEE,* Muhammad Awais, and Josef Kittler, *Life Member, IEEE*

Abstract—
Self-supervised learning methods are gaining increasing traction in computer vision due to their recent success in reducing the gap with supervised learning. In natural language processing (NLP) self-supervised learning and transformers are already the methods of choice. The recent literature suggests that the transformers are becoming increasingly popular also in computer vision. So far, the vision transformers have been shown to work well when pretrained either using a large scale supervised data [1] or with some kind of co-supervision, e.g. in terms of teacher network. These supervised pretrained vision transformers achieve very good results in downstream tasks with minimal changes [1], [2], [3]. In this work we investigate the merits of **self-supervised learning** for pretraining image/vision transformers and then using them for downstream classification tasks. We propose **S**elf-supervised v**i**sion **T**ransformers (**SiT**) and discuss several self-supervised training mechanisms to obtain a pretext model. The architectural flexibility of SiT allows us to use it as an autoencoder and work with multiple self-supervised tasks seamlessly. We show that a pretrained SiT can be finetuned for a downstream classification task on small scale datasets, consisting of a few thousand images rather than several millions. The proposed approach is evaluated on standard datasets using common protocols. The results demonstrate the strength of the transformers and their suitability for self-supervised learning. We outperformed existing self-supervised learning methods by large margin. We also observed that SiT is good for few shot learning and also showed that it is learning useful representation by simply training a linear classifier on top of the learned features from SiT. Pretraining, finetuning, and evaluation codes will be available under: https://github.com/Sara-Ahmed/SiT.

Index Terms—Vision Transformer, Self-supervised Learning, Discriminative Learning, Image Classification, transformer based autoencoders.

1 Introduction

Recent trend particularly in NLP showed that self-supervised pretraining can improve the performance of downstream task significantly [4], [5]. Similar trends have been observed in speech recognition [6] and computer vision applications [7], [8], [9], [10]. The self-supervised pretraining particularly in conjunction with transformers [11] as shown for BERT [4], [5] are the models of choice for natural language processing (NLP). The success of self-supervised learning comes at the cost of massive datasets and huge capacity models, e.g., NLP based transformers are trained on hundreds of billions of words consisting of models with several billions parameters [5]. The recent success of Transformers in image classification [1] generated a lot of interest in the computer vision community. However, the pretraining of vision transformer is mainly studied for very large scale supervised learning datasets, e.g., datasets consisting of hundred of millions of labelled samples [1]. Very recently vision transformer have been shown to perform well on imagenet without external data [2], however, they need distillation approaches and guidance from CNNs counterparts. In short, a pretraining using large scale supervised datasets is a norm in computer vision to train deep neural networks in order to obtain better performance. However, manual annotation of training data is quite expensive, despite the advances in the crowd engineering innovations. To address this limitation, self-supervised learning methods [7], [9], [10], [12], [13], [14] have been proposed to construct image representations that are semantically meaningful from unlabelled data.

Self-supervised methods can roughly be categorised in to generative and discriminative approaches. Generative approaches [15], [16], [17] learn to model the distribution of the data. However, data modelling generally is computationally expensive and may not be necessary for representation learning in all scenarios. On the other hand, discriminative approaches, typically implemented in a contrastive learning framework [8], [18], [19], [20] or using pre-text tasks [21], [22], [23], demonstrate the ability to obtain better generalised representations with modest computational requirements.

The primary focus of contrastive learning is to learn image embeddings that are invariant to different augmented views of the same image while being discriminative among different images. Despite the impressive results achieved by contrastive learning methods, they often disregard the learning of contextual representations, for which alternative pretext tasks, such as reconstruction-based approaches, might be better suited. In recent years, a stream of novel pretext tasks have been proposed in the literature, including inpainting patches [24], colourisation [21], [25], [26], relative patch location [15], solving jigsaw puzzles [27], [28], cross-channel prediction [29], predicting noise [30], predicting image rotations [22], spotting artefacts [23], etc.

In this work, we introduce a simple framework for self-supervise learning that leverages the advantage of both contrastive learning and pre-text approaches. The main contributions and findings of this study are summarised as follows:

- We propose Self-supervised vision Transformer (SiT), a novel method for self-supervised learning of visual representations.

- *Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford, United Kingdom*
- *{s.a.ahmed,m.a.rana,j.kittler}@surrey.ac.uk*

- We endow the SiT architecture with a decoder and demonstrate that it can be implemented by essentially using one linear layer, thanks to the intrinsic characteristics of the transformer. This transformer based autoencoder avoids the need for a whole decoder block which is typically present in CNNs based encoder-decoder architectures.
- Drawing on the natural ability of the autoencoding transformer to support multi-task learning, we develop a strong self-supervised framework which jointly optimises the reconstruction (image inpainting), rotation classification and constrastive losses.
- We demonstrate the effectiveness of the proposed framework on standard benchmarks following different evaluation protocols including linear evaluation, domain transfer, and finetuning.
- We outperform the concurrent state-of-the-art results in different datasets with a large margin reaching +13.53% improvements.

2 Related Works

2.1 Handcrafted Pretext Tasks

The basic pretraining meachanisms is autoencoding [31], which forces a network to find a representation that allows the reconstruction of the input image, even if corrupted by perturbations or noise. Many self-supervised pretext tasks manipulate the input data to obtain better image representations. For example, Pathak *et al.* [24] trained a convolutional network to predict the content of arbitrary missing regions in an image based on the rest of the image.

Zhang *et al.* [25] proposed image colourisation task by predicting a coloured version of the given grey-scale input image and used class re-balancing to increase the diversity of the predicted colours.

Doersch *et al.* [15] presented one of the pioneer work of using spatial context information for feature learning by training a convolutional network to recognise the relative positions of random pairs of image patches. Following this idea, several methods were proposed to learn image features by solving even more difficult spatial puzzles (e.g. jigsaw puzzles [27], [28]).

Gidaris *et al.* [22] proposed RotNet, a convolutional network that learns image features by training the network to recognise a pre-defined 2d rotation that is applied to the input image.

2.2 Clustering and Contrastive Learning

Contrastive approaches train the network by bringing representation of different augmented views of the same image closer and spreading representations of views from different images apart.

DeepCluster [18] clusters data points using the prior representation, by bootstrapping on previous versions of its representation to produce targets for the next representation. The cluster index of each sample is then used as a classification target for the new representation. This approach is computationally expensive as it requires a clustering phase with precautions to avoid collapsing to trivial solutions.

Hjelm *et al.* [19] investigated the use of mutual information for unsupervised representation learning through Deep InfoMax by maximising the mutual information in global and local scales on single views following the InfoMax principle [32].

SimCLR [8] proposed contrastive self-supervised learning algorithms without requiring specialised architectures or a memory bank.

Patacchiola and Storkey [20] proposed a self-supervised formulation of relational reasoning that allows a learner to bootstrap a signal from the information implicit in unlabelled data.

3 Methodology

Supervised learning, as demonstrated in [1], allows the transformer to learn a bottleneck representation where the mixing of content and context is centred primarily about the class token. This creates a rather superficial model of the data, and its linking to labels requires a huge number of samples for training.

In contrast, unsupervised learning exploits information redundancy and complementarity in the image data by learning to reconstruct local content by integrating it with context. In this paper, this is achieved by three principles: i) *learning to reconstruct the input stimulus by a mechanism akin to autoencoding, and denoising, implemented by means of random data perturbation using masking, etc.* ii) *a perception-action mechanism* [33], *which learns to recognise an action from its impact on perception*, and iii) *learning the notion of similarity of content from the preservation of content identity in the data impacted by a geometric transformation.* The proposed self-supervised learning approach is instrumental in extracting an intrinsic data model, that is robust to perturbations and is admirably able to adapt to downstream tasks by fine tuning. The proposed approach offers remarkable advantages:

- The self-supervised transformer can be trained with unlabelled data.
- The amount of labelled training data required for finetuning to learn a downstream task is two orders of magnitude lower than the counterpart needed for direct training.
- The total amount of training data (labelled and unlabelled) is also several orders of magnitude lower.
- The performance achieved is significantly better than state-of-the-art self-supervised methods. !!!

The proposed methodology of transformer pretraining by self-supervision is expected to have a significant impact on the advancement of science by enabling the wider research community starved of resources to contribute to deep learning.

Thus the main goal of this work is to learn a representation of the data in an unsupervised fashion. This is achieved by completing partially masked or transformed local parts of the image. The underlying hypothesis is that, by recovering the corrupted part of an image from the uncorrupted part based on the context from the whole visual field, the network will implicitly learn the notion of visual integrity. This notion of visual integrity is further enhanced by using pseudo labels that can be generated automatically based on some attributes of the data. Learning from recovery of the transformed parts and learning from pseudo label may

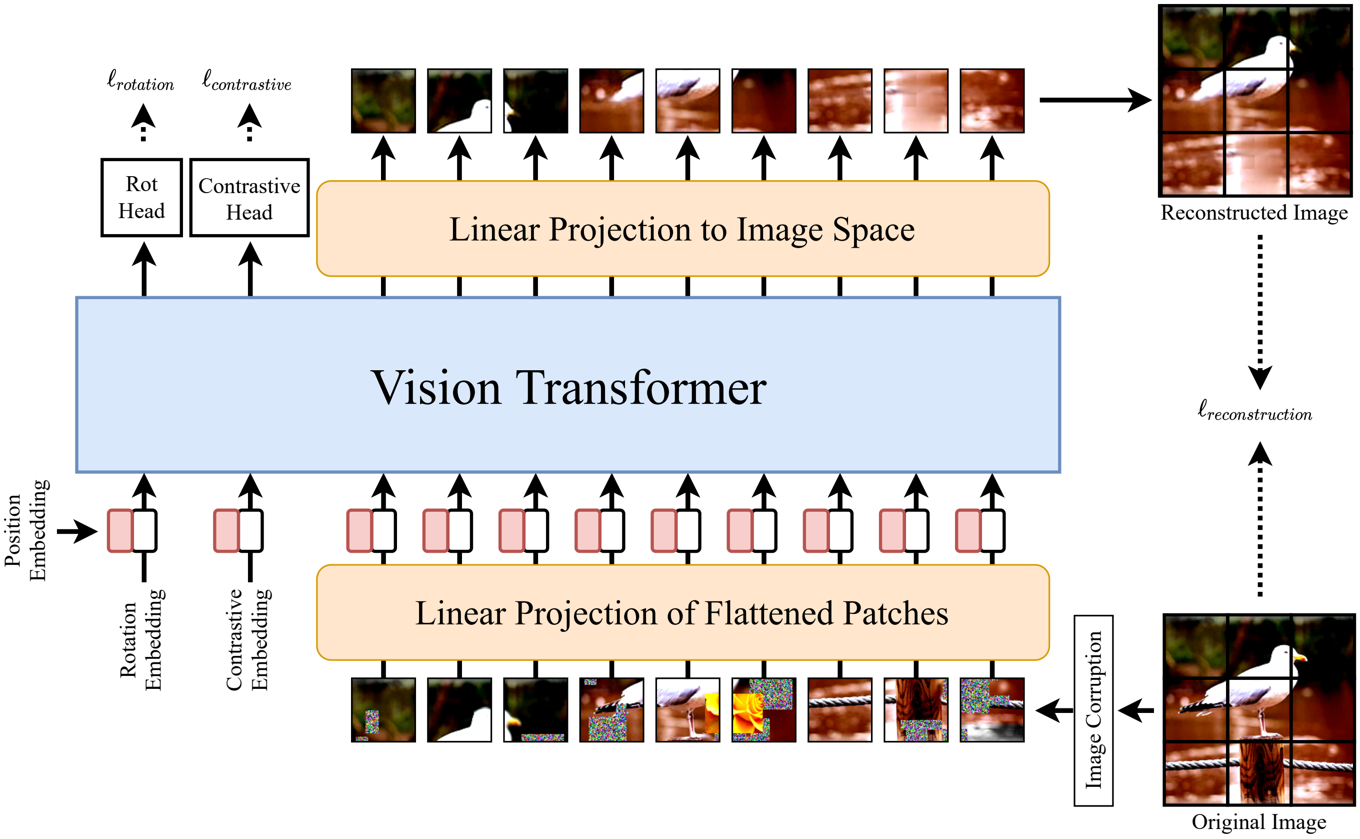


Fig. 1: Self-supervised vision Transformer (SiT)

seem different but the underlying motivation behind both kinds of self-supervised learning mechanisms is the same, i.e., learning visual integrity. For example, intuitively the network will only be able to recover the pseudo labels if it learns the characteristic properties of visual stimuli corresponding to specific actions impacting on the visual input. The weights of the learned model can then be employed as an initialisation point for any downstream task like image classification, object detection, segmentation, etc. To achieve this goal, we propose a Self-supervised image Transformer (SiT) in which the model is trained to estimate different geometric transformations applied to the input image and hence, better image representation can be obtained.

3.1 Self-Supervised Vision Transformer

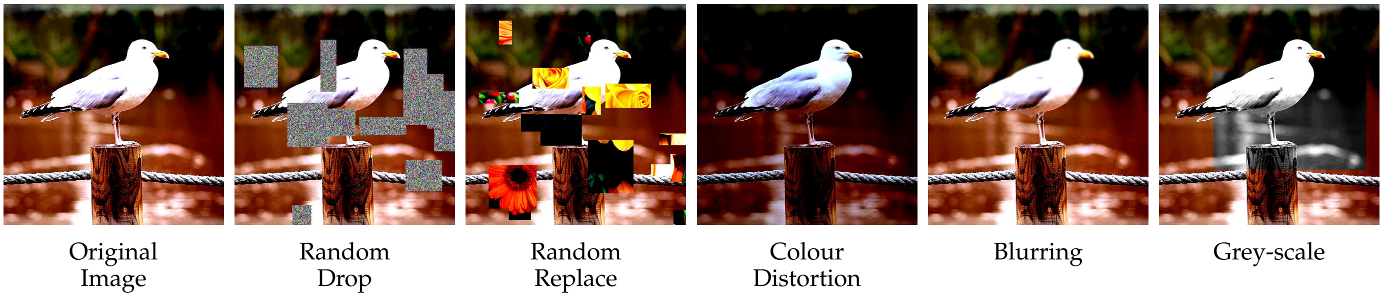
Transformer [11] has shown great success in various natural language processing tasks [4], [5], [34]. Recently, many researchers attempted to explore the benefits of transformer-based models in computer vision tasks [1], [2], [35]. In this work, we introduce Self-supervised image Transformer (SiT) adapted from Vision Transformer (ViT) proposed by [1] with some modifications. Unlike [1], our work is based on unsupervised learning/pretraining, hence, classification token is not required. Instead we have two additional tokens, beside data tokens (image patches token) used for image reconstruction, to serve the proposed self-supervised pretraining tasks. The first of these tokens, the rotation token adopted from RotNet [22], is employed to predict the rotation transformation of the given image. The second token, the contrastive token adopted from SimCLR [8], is employed to serve the contrastive prediction task. The rotation and constrastive tokens are presented in Section 3.2.2 and Section 3.2.3, respectively. The main architecture of our proposed network is shown in Figure 1.

3.2 Self-Supervised Tasks

As noted earlier, the transformer architecture allows seamless integration of multiple task learning simultaneously. We leverage this strength of the transformers to train SiT with three different objectives: (1) Image reconstruction, (2) Rotation prediction, and (3) Contrastive learning. In the rest of the section, we describe the different types of self-supervised tasks employed in this work.

3.2.1 Task #1: Image Reconstruction

For image reconstruction we propose to use the transformer as an autoencoder, i.e., visual transformer autoencoder. Unlike CNNs based autoencoders which require encoder and expensive decoders consisting of convolutional and transposed convolution layers, the autoencoder in transformers, more specifically the decoder can be implemented using a simple linear layers. One limitation of CNN based encoder is the step of information summarisation in which the information is essentially discarded using strided convolutions or max pool operations. This information is then recovered by series of upsampling and convolution operations (or transposed convolutions) with skip connection between encoder and decoder. By analogy to auto-encoders, our network is trained to reconstruct the input



Original Image

Random Drop

Random Replace

Colour Distortion

Blurring

Grey-scale

Fig. 2: Examples of transformed local parts of the image.

image through the output tokens of the transformer. To learn better semantic representations of the input images, we apply several transformations to local patches of the image. Unlike standard masked tokens in BERT, we apply these local transformations to a block of neighbouring tokens arranged spatially (in 2D rather than in sequence only). In BERT and other NLP based transformers, it makes sense to mask only one token, as a single token can represent semantic meaning. However, for visual signals it is critical to transform neighbouring tokens consisting of either one or more semantic concepts. The aim is to recover these transformed local parts at the output of SiT. In doing so, SiT implicitly learns the semantic concepts in the image. It should be noted that these transformed tokens can be either on the foreground object or on the background and recovering these token is equally valid for both scenarios. Indeed while modelling the visual signal in this way we are are moving away from the notion of foreground and background and every part of the content is consider as a semantic concept whether it is a horse grazing in a meadow or the meadow itself. The intuition is that by modelling all semantic concepts SiT will generalise better for the unseen tasks whether they are related to an object, a distributed object or to the whole visual signal. In this work, several local transformation operations are applied to connected patches including random drop and random replace, colour distortions, recolouring etc. An example subset is shown in Figure 2.

Image inpainting is a simple but effective pre-text task for self-supervision, which proceeds by training a network to predict arbitrary transformed regions based on the context. This context can be from the same object on which the transformed region is applied or from the surrounding objects/concepts. With CNNs this context is defined by the, so called, receptive field, while with transformers the context consists of the whole image. The motivation behind image inpainting is that the network is required to learn the knowledge including the colour, texture and structure of the objects/concepts to infer the missing areas. In this work, we employed two types of image inpainting, random dropping, by randomly replacing neighbouring patches from the image with random noise, and random replacement, by randomly replacing patches from the image with patches from another image.

As for the colour transformations, it involves basic adjustments of colour levels in an image including colour distortions, blurring, and converting to grey-scale. Colour distortions are applied to the whole image to enable the network to recognise similar images invariant to their colours. Unlike colour distortions, blurring and conversion to grey-scale transformations are applied to the local neighbourhood of arbitrary patches of the image rather than the full image to enable the network to learn the texture and colour transformations from the surrounding pixels. Blurring is performed by applying a Gaussian filter to the selected patches and converting to grey-scale is performed by a linear transformation of the coloured (RGB) patches. Figure 2 shows the employed geometric transformations applied separately on a sample image. During training, all the mentioned transformations are applied to the input image simultaneously.

The objective of the image reconstruction is to restore the original image from the corrupted image. For this task, we used the $\ell 1$-loss between the original and the reconstructed image as shown in Equation 1:

$$\mathcal{L}_{\text{recons}}(\mathbf{W}) = \frac{1}{N}\sum_{i}^{N} ||\mathbf{x_i} - \text{SiT}_{\text{recons}}(\bar{\mathbf{x}}_\mathbf{i})|| \tag{1}$$

where, $||.||$ is the $\ell 1$ norm, $\mathbf{x_i}$ is the input image, $\bar{\mathbf{x}}_\mathbf{i}$ is the corrupted image, $\text{SiT}_{\text{recons}}(.)$ returns the reconstructed image and N is the batch size. $\mathbf{W}$ denotes the parameters of the transformer to be learned during training.

3.2.2 Task #2: Rotation Prediction

As noted earlier the flexibility of the transformer architecture allows us to combine the reconstruction losses with other complementary losses. For the second objective, the network is trained to predict the rotation of the input image. To obtain the pseudo labels for this task, the input image is randomly rotated by $0°$, $90°$, $180°$, or $270°$ degree and then fed to the network where the network is required to classify the rotation of the input image to one of the aforementioned degree categories. The motivation behind this task is that the network is expected to first learn the notion of the objects in the image in order to be able to predict their orientations. For the objective function, we employed the cross entropy loss as follows:

$$\begin{gathered}\hat{y} = \text{softmax}(\text{SiT}_{\text{rotation}}(R(\mathbf{x_i}, \theta))) \\ \mathcal{L}_{\text{rotation}}(\mathbf{W}) = -\frac{1}{N}\sum_{i}^{N} \log \hat{y}^{\theta}\end{gathered} \tag{2}$$

where $R(., \theta)$ is an operator that rotates the given input by θ degrees, $\text{SiT}_{\text{rotation}}(.)$ is the output from the rotation head, and $\hat{y}^{\theta}$ is the predicted probability from the index corresponding to the rotation θ.

3.2.3 *Task #3: Contrastive learning*

In self-supervised learning we do not have any concept labels for the training data. However, by applying geometric and perturbation transformations to a training sample we do not change the perceptual identity of the content, and the transformer should produce for all such synthetically-generated content-matching pairs a similar output. We adopt the cosine similarity as the uderlying measure of similarity of representation. Inspired by recent contrastive learning algorithms [36], we incorporated a contrastive loss to the objective function where the network is trained to minimise the distance between positive pairs, i.e. augmented images coming from the same input image, and maximise the distance between negative pairs, i.e. samples coming from different input images. In particular, we employed the normalised temperature-scaled softmax similarity [8], [19], [37] between a sample $\mathbf{x_i}$ and any other point $\mathbf{x_j}$ defined as follows:

$$\ell^{x_i,x_j}_{\text{contr}}(\mathbf{W}) = \frac{\mathrm{e}^{\text{sim}(\text{SiT}_{\text{contr}}(x_i),\ \text{SiT}_{\text{contr}}(x_j))/\tau}}{\sum_{k=1,k\neq i}^{2N} \mathrm{e}^{\text{sim}(\text{SiT}_{\text{contr}}(x_i),\ \text{SiT}_{\text{contr}}(x_k))/\tau}} \tag{3}$$

where $\text{SiT}_{\text{contr}}(.)$ denotes the image embedding coming from the contrastive head, $\text{sim}(.,\ .)$ is the dot product of the ℓ_2 normalised inputs, which is the cosine similarity, and τ denotes a constant temperature parameter which we set to 0.5 throughout the experiments as suggested by [8]. Rather than using the cosine similarity measure directly, the normalised softmax in (3) has the advantage that its optimisation enhances the similarity of matching pairs, as well as the dissimilarity of negative pairs. Now, let $x_{\bar{j}}$ be a sample matching in content the sample, x_j. The contrastive loss, $\mathcal{L}_{\text{contr}}(\mathbf{W})$, is then defined as the arithmetic mean over all positive pairs in the batch of the cross entropy of their normalised similarities, i.e.

$$\mathcal{L}_{\text{contr}}(\mathbf{W}) = -\frac{1}{N}\sum_{j=1}^{N} \log \ell^{x_j,x_{\bar{j}}}(\mathbf{W}) \tag{4}$$

3.3 End-to-End Self-Supervised Training

For a given mini-batch consisting of N samples, two different random augmentations are applied to each sample. Augmented images created from the same samples are considered to constitute positive pairs and images coming from different samples are considered as negative pairs. The $2N$ samples are then randomly rotated with one of the pre-defined degrees (0°, 90°, 180°, or 270°) to obtain a pseudo-label for the rotation prediction task. Next, random image corruption including image inpainting and colour transformation are applied on top of the randomly rotated image and fed to the SiT model. The network is then trained to (1) Reconstruct the images after the applied corruption, (2) Predict the random rotation applied to the image, and (3) maximise the cosine similarity between the positive pairs. The overall loss function is shown in Equation 5.

$$\begin{aligned}\mathcal{L}_{\text{total}}(\mathbf{W}) = \alpha_1 \times \mathcal{L}_{\text{recons}}(\mathbf{W}) + \alpha_2 \times \mathcal{L}_{\text{rotation}}(\mathbf{W})\\ + \alpha_3 \times \mathcal{L}_{\text{contr}}(\mathbf{W})\end{aligned} \tag{5}$$

Note that $\alpha_{1,\ldots,3}$ are the scaling factors of our multi-task objective function.

Training the model with several objectives simultaneously is more efficient and often leads to better performance [38]. A simple way to combine multiple losses into a one single loss is by computing a weighted sum of the losses for the different tasks. The weights (i.e. $\alpha_{1,\ldots,3}$) could be optimised by a grid search but in our case, it would be a very expensive process. To mitigate this issue, we incorporated the uncertainty weighting approach proposed by Kendall *et al.* [39]. Each task is weighted by a function of its homoscedastic aleatoric uncertainty rather than by a fixed weight. In particular, the objective function is extended to learn the parameters of the model, as well as the relative weights of the losses for each individual task. The overall loss function is then calculated as follows:

$$\begin{aligned}\mathcal{L}_{\text{total}}(\mathbf{W}, \alpha_1, \alpha_2, \alpha_3) = \frac{1}{\alpha_1} &\times \mathcal{L}_{\text{recons}}(\mathbf{W})\\ + \frac{1}{\alpha_2^2} &\times \mathcal{L}_{\text{rotation}}(\mathbf{W})\\ + \frac{1}{\alpha_3^2} &\times \mathcal{L}_{\text{contr}}(\mathbf{W})\\ + \log(\alpha_1) &+ \log(\alpha_2) + \log(\alpha_3)\end{aligned} \tag{6}$$

where $\alpha_{1,\ldots,3}$ are learnable parameters. Large scale values of α decrease the contribution of the corresponding loss, whereas small scale values of α increase its contribution. In practice, to avoid any division by zero and to improve the numerical stability, the network is trained to predict the log variance, $s = \log\alpha^2$ instead of the variance $s = \log\alpha^2$. The rotation and contrastive tasks can be modelled with softmax likelihood (refer to [39]). Unlike [39], for the regression task we use ℓ_1 norm and thus, we define the likelihood as a Laplace instead of a Gaussian. Hence, the scalar parameter of the reconstruction loss is $\frac{1}{\alpha_1}$ instead of $\frac{1}{\alpha_1^2}$.

4 Experimental Results

The common evaluation to demonstrate the generalisation of the learnt features by self-supervised methods is to pretrain the model in unsupervised fashion, followed by finetuning the model on a downstream task like image classification, object detection, segmentation, etc. In this work, we conduct several experiments on four well-known multi-class data sets (Section 4.1) to show the effectiveness of our proposed self-supervised image transformer.

4.1 Data sets

We conduct an extensive experimental analysis on four standard multi-class classification problems based on object detection and recognition in an unconstrained background.

CIFAR-10 [40]: This data set consists of $60,000$ (32×32) images belonging to 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck), and is divided into $50,000$ images for training and $10,000$ for testing.

CIFAR-100 [40]: Similar to CIFAR-10 data set, CIFAR-100 consists of $50,000$ training images and $10,000$ test images. There are 100 classes in this data set, grouped into 20 super-classes. Each image comes with a "fine label" which is the class label and a "coarse label" which is the super-class to which it belongs. In our study, we make use of the fine labels.

Tiny-ImageNet [41]: Derived from the ImageNet Large Scale Visual Recognition Challenge [42]. The dataset consists of $100,000$ training images (64×64) and $10,000$ test images. There are 200 classes in this dataset.

STL-10 [43]: STL-10 is an image recognition dataset inspired by CIFAR-10 with some improvements. There are 10 classes in this dataset, 500 training images per category, and 800 test images per category. Additionally, there are $100,000$ unlabeled images for unsupervised learning. Unlike CIFAR-10, the dataset has a higher resolution (96×96) which makes it a challenging benchmark for developing more scalable unsupervised learning methods.

4.2 Implementation Details

In our experiments, we implement the self-supervised architecture using ViT transformer [1]. We employed the Base variant of ViT (ViT-B) with 16×16 input patch size, 768 hidden dimension, 12 layers, and 12 heads on each layer. There are 86 million parameters in total in this architecture.

The rotation prediction task is implemented by a linear layer with 4 output nodes that represent the $0°$, $90°$, $180°$, and $270°$ different rotations. In our preliminary experiments, we used 8 output nodes including the rotations of the horizontal flipping of the given image. We found that the network struggles to distinguish between the rotated image and the rotation of the flipped image as two different classes. Instead, we included the horizontal flipping to the data augmentation step before applying rotation, and hence, the network is trained to classify the image and the flipped image to the same class.

For the contrastive learning, we implemented the constrastive head by a linear layer with 512 output nodes that represent the image embeddings.

For the optimisation of self-supervised models, we trained all models using the Adam optimiser with batch size 72, momentum 0.9, weight decay 0.05 and learning rate of $5e^{-4}$ for 400 epochs in total. In fact, we mostly rely on the vision transformer developer's default hyper-parameters. We believe that further improvements can be obtained by tuning the hyper-parameters for the self-supervised model.

Simple data augmentation techniques are applied during the self-supervised training. We found that to learn low-level features as well as high-level semantic information, aggressive data augmentation like MixUp [44] and Auto-Augment [45] hurts the training, specially with the objective functions in hand. Therefore, we used only cropping, colour jittering and horizontal flipping by selecting a random patch from the image and resizing it to 224×224 with a random horizontal flip.

After data augmentation, image distortion techniques described in Section 3 are applied to the perturbed image and the network is optimised together with the rotation prediction and contrastive learning to reconstruct the image after distortion. In order to get a feel for the reconstruction capability of SiT, in Figure 3 we show the reconstruction of randomly selected images from the training data, testing data, and from internet.

For the finetuning step on the downstream tasks, the rotation and contrastive heads are replaced with an output layer with n nodes corresponding to the number of classes in the downstream task. The final prediction is based on the average predictions from both heads. The model is optimised in the exactly same way of the pretraining. For the data augmentation, random cropping, random horizontal flipping, MixUp and Auto-Augment are applied.

4.3 Linear Evaluation and Domain Transfer

In order to assess the quality of the learnt representations, we follow the linear evaluation protocol defined by [46]. First, we train the SiT model employing the unlabelled training set, and then a linear classifier corresponding to the number of classes is trained on top of the learnt features. Beside the linear evaluation, we adopted the domain transfer protocol defined by [20] by pretraining on the unlabelled CIFAR-10 dataset, followed by a linear evaluation on the labeled CIFAR-100 dataset (and vice versa). Table 1 shows that our method outperforms the state-of-the-art with a large margin. Our model achieved 81.2%, 55.97%, and 40.67% on CIFAR-10, CIFAR-100, and Tiny-ImageNet on the linear evaluation which is an improvement of +3.69%, +8.07%, and 10.13%, respectively. Furthermore, in the domain transfer from CIFAR-100 to CIFAR-10 and vice versa, we achieved 73.79% and 55.72% which is an improvement of +5.13% and +13.53%, respectively.

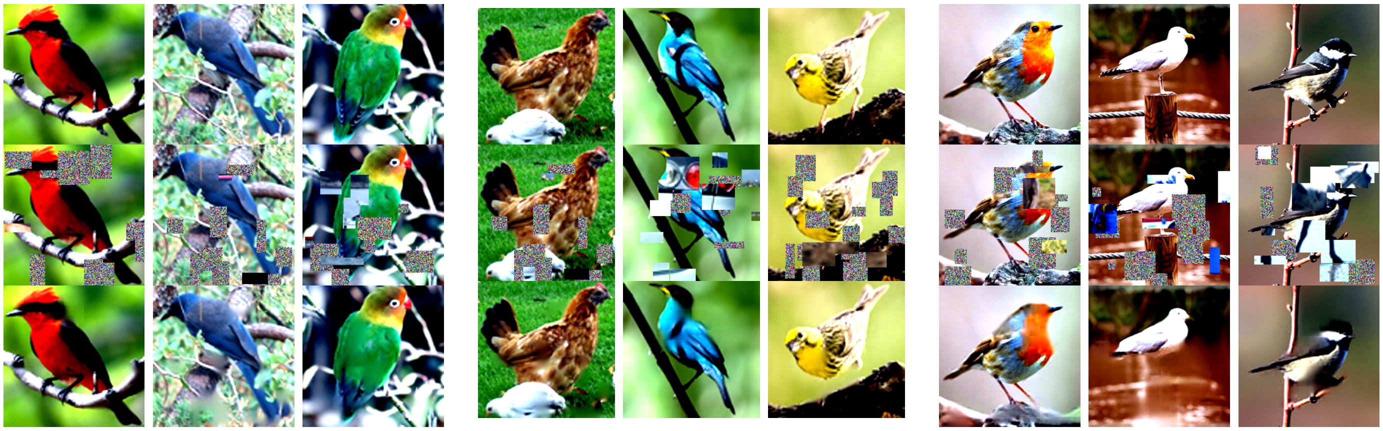
Beside the linear evaluation, in Table 2, we show the test accuracy on CIFAR-10 and CIFAR-100 datasets as a function of the percentage of the available labeled data. [20] incorporated the available labeled data in the pretraining stage, followed by a linear evaluation on the entire training labeled datasets. For a fair comparison with [20], we first finetuned the self-supervised network on the available percentage of the labeled dataset, followed by a linear evaluation on the entire labeled datasets. We report both, the accuracy after finetuning, which is categorised under few-shot learning, and the accuracy after linear evaluation on the entire labeled datasets. The results show that the quality of the representations improves with the increase of the available labeled data. Besides, our finetuned model is able to outperform the state-of-the-art in most cases, even without a further linear evaluation on the full training dataset.

4.4 Finetuning

In this set of experiments, we investigate the effectiveness of our model on STL-10 dataset. The SiT model is pretrained using the unlabelled data provided, followed by finetuning the model using the training portion of the dataset. Our proposed model outperforms the state-of-the-art with an improvements of +3.40%.

4.5 Ablation Study

The aim of the ablation study was to investigate the effect of individual elements of the pretext learning. We used the image recognition task defined on the STL-10 dataset as a vehicle for measuring the impact of the different types of pretext self-supervised pretraining. We used the protocol defined for the dataset to perform the finetuning and linear



(a) Samples from training set

(b) Samples from testing set

(c) Images from internet

Fig. 3: Reconstructed images from our trained SiT model. The images are randomly obtained from (a) Training data, (b) Test data, and (c) From the internet. Each row refers to the original images, corrupted images, and the reconstructed images, respectively.

Method	**Backbone**	**Linear Evaluation**			**Domain Transfer**	
		CIFAR10	**CIFAR100**	**Tiny-ImageNet**	**C100→C10**	**C10 →C100**
DeepCluster [18]	ResNet-32	$43.31\% \pm 0.62$	$20.44\% \pm 0.80$	$11.64\%0.21\pm$	$43.39\% \pm 1.84$	$18.37\% \pm 0.41$
RotationNet [22]	ResNet-32	$62.00\% \pm 0.79$	$29.02\% \pm 0.18$	$14.73\%0.48\pm$	$52.22\% \pm 0.70$	$27.02\% \pm 0.20$
Deep InfoMax [19]	ResNet-32	$47.13\% \pm 0.45$	$24.07\% \pm 0.05$	$17.51\%0.15\pm$	$45.05\% \pm 0.24$	$23.73\% \pm 0.04$
SimCLR [8]	ResNet-32	$77.02\% \pm 0.64$	$42.13\% \pm 0.35$	$25.79\%0.4\pm$	$65.59\% \pm 0.76$	$36.21\% \pm 0.16$
Relational Reasoning [20]	ResNet-32	$74.99\% \pm 0.07$	$46.17\% \pm 0.16$	$30.54\%0.42\pm$	$67.81\% \pm 0.42$	$41.50\% \pm 0.35$
[20]	ResNet-56	$77.51\% \pm 0.00$	$47.90\% \pm 0.27$	n/a	$68.66\% \pm 0.21$	$42.19\% \pm 0.28$
SiT (ours)	Transformer	$\mathbf{81.20}\%$	$\mathbf{55.97}\%$	$\mathbf{40.67}\%$	$\mathbf{73.79}\%$	$\mathbf{55.72}\%$

TABLE 1: Linear evaluation after self-supervised pretraining. Mean accuracy (percentage) and standard deviation over three runs are reported on CIFAR-10 (C10) and CIFAR-100 (C100) datasets. $X \rightarrow Y$ implies that the pretraining is performed on unlabelled dataset X and the linear evaluation is performed on labeled dataset Y. The best results are highlighted in bold.

Method	**0%**	**1%**	**10%**	**25%**	**50%**	**100%**
				CIFAR-10		
[20]	$74.99\% \pm 0.07$	$76.55\% \pm 0.27$	$80.14\% \pm 0.35$	$85.30\% \pm 0.28$	$89.35\% \pm 0.11$	$90.66\% \pm 0.23$
SiT (ours) - fewshot	n/a	74.78%	87.16%	92.90%	94.84%	97.70%
SiT (ours)	81.20%	81.72%	87.90%	93.12%	95.14%	97.53%
				CIFAR-100		
[20]	$46.17\% \pm 0.17$	$46.10\% \pm 0.29$	$49.55\% \pm 0.36$	$54.44\% \pm 0.58$	$58.52\% \pm 0.70$	$58.96\% \pm 0.28$
SiT (ours) - fewshot	n/a	27.50%	53.72%	67.58%	74.46%	80.30%
SiT (ours)	55.97%	56.81%	61.35%	70.58%	75.97%	80.20%

TABLE 2: Test accuracy on CIFAR10 and CIFAR100 datasets with respect to the available percentage of the labeled data. SiT is finetuned with the available labeled data (fewshot), followed by linear evaluation on the entire labeled dataset.

evaluation experiments, starting with training from scratch. From the results reported in Table 4 it is evident that the training from random initialisation has produced accuracies for both scenarios of less than 40%, as the amount of data available is insufficient to train the transformer. We then pretrained SiT using just the autoencoder version of the transformer working with the reconstruction error loss function. The performance jumped to 90%. Testing the other self-supervised training mechanisms on their own we found that the least effective was the rotation classification learning, reaching only circa 65% in finetuned performance. There was not much difference in the results of linear evaluation between reconstruction error and rotation, both being rather poor. Interestingly, the contrastive loss pretraining lifted the finetuned performance to 2nd rank position, and achieved significantly better performance in the linear evaluation. The pairing of self-supervision training mechanisms had an egalitarian effect, pushing the finetuning test performance to around 90%. For the linear evaluation the significantly best combination was achieved by the rotation-contrastive loss pair. This was further improved by combining all three types of self-learning. Finally, weighting the contribution of the different self-learning terms by learnt weights pushed the finetuned test result to 93%, with the linear evaluation remaining at 78.5%.

In summary, using the reconstruction loss on its own as a means of self-supervision provided an effective starting point for efficient downstream task finetuning. Further marginal improvements can be made by extending the range of mechanisms for self-supervised pretraining. However, when we measure the effectivness of the pretrained weights per se using the linear evaluation, the importance

Method	Backbone	Accuracy
Exemplars [47]	Conv-3	72.80%
Artifacts [23]	Custom	80.10%
ADC [48]	ResNet-34	56.70%
Invariant Info Clustering [49]	ResNet-34	88.80%
DeepCluster [18]	ResNet-34	73.37%
RotationNet [22]	ResNet-34	83.22%
Deep InfoMax [19]	AlexNet	77.00%
Deep InfoMax [19]	ResNet-34	76.03%
SimCLR [8]	ResNet-34	89.31%
Relational Reasoning [20]	ResNet-34	89.67%
SiT (our)	Transformer	**93.02%**

TABLE 3: A comparison with the state-of-the-art methods based on an unsupervised training and finetuning experiment involving the STL-10 dataset

Self-supervision tasks	Finetuned	Linear Evaluation
Trained from scratch	59.38%	37.75%
Reconstruction	90.03%	45.49%
Rotation	65.48%	46.80%
Contrastive	84.50%	69.90%
Reconstruction+Rotation	91.80%	70.38%
Reconstruction+Contrastive	89.46%	73.90%
Rotation+Contrastive	90.44%	77.10%
Reconstruction+Rotation+Contrastive	91.49%	**78.58%**
Reconstruction+Rotation+Contrastive (uncertainty weighting)	**93.02%**	78.51%

TABLE 4: Performance of the individual elements of the pretext learning.

of using a diversity of self-supervised techniques jointly becomes apparent.

5 Conclusion

In this work we present a self-supervised image transformer, trained with unlabelled data to perform pretext tasks, and used the pretrained model as initialisation for finetuning for a downstream classification task. We proposed to use transformers as an autoencoder, which is realisable by using a single linear layer at the output (thanks to the transformer architecture). We leveraged the attractive property of the transformer architecture of being particularly suited for combining different loss functions along with reconstruction loss. We added a token per loss and combined rotation and contrastive losses along with reconstruction loss. The proposed SiT outperformed state-of-the-art self-supervised methods with wide margins. This work focused on image classification as a downstream task. We believe that the SiT is admirably suitable for many other downstream tasks like segmentation and detection, however, this conjecture is left for future investigation.

Acknowledgments

This work was supported in part by the EPSRC Programme Grant (FACER2VM) EP/N007743/1 and the EPSRC/dstl/MURI project EP/R018456/1.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[2] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[3] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, "Training vision transformers for image retrieval," *arXiv preprint arXiv:2102.05644*, 2021.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pretraining of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[6] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7414–7418.

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[9] 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 *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *arXiv preprint arXiv:2006.07733*, 2020.

[10] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *arXiv preprint arXiv:2006.09882*, 2020.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[12] J. Schmidhuber, "Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook," Ph.D. dissertation, Technische Universität München, 1987.

[13] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2794–2802.

[14] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta, "The curious robot: Learning visual representations via physical interactions," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–18.

[15] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1422–1430.

[16] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[17] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[18] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.

[19] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *International Conference on Learning Representations (ICLR)*, 2019.

[20] M. Patacchiola and A. J. Storkey, "Self-supervised relational reasoning for representation learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 4003–4014.

[21] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6874–6883.

[22] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[23] S. Jenni and P. Favaro, "Self-supervised feature learning by learning to spot artifacts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2733–2742.

[24] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[25] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.

[26] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European conference on computer vision*. Springer, 2016, pp. 577–593.

[27] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*. Springer, 2016, pp. 69–84.

[28] D. Kim, D. Cho, D. Yoo, and I. S. Kweon, "Learning image representations by completing damaged jigsaw puzzles," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 793–802.

[29] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1058–1067.

[30] P. Bojanowski and A. Joulin, "Unsupervised learning by predicting noise," in *International Conference on Machine Learning*. PMLR, 2017, pp. 517–526.

[31] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[32] R. Linsker, "Self-organization in a perceptual network," *Computer*, vol. 21, no. 3, pp. 105–117, 1988.

[33] G. H. Granlund, "Special issue on perception, action and learning," *Image Vis. Comput.*, vol. 27, no. 11, pp. 1639–1640, 2009.

[34] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[35] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *arXiv preprint arXiv:2101.01169*, 2021.

[36] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," 2020.

[37] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[38] J. Baxter, "A model of inductive bias learning," *Journal of artificial intelligence research*, vol. 12, pp. 149–198, 2000.

[39] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[40] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[41] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, p. 7, 2015.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[43] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[44] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[45] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

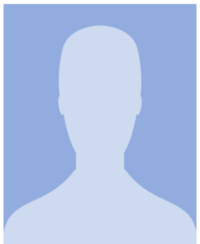
[46] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1920–1929.

[47] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks." Citeseer, 2014.

[48] P. Haeusser, J. Plapp, V. Golkov, E. Aljalbout, and D. Cremers, "Associative deep clustering: Training a classification network with no labels," in *German Conference on Pattern Recognition*. Springer, 2018, pp. 18–32.

[49] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9865–9874.

Sara Atito Ali Ahmed received her Bsc. in computer science from Ain Shams University, Egypt, in 2011. Her Msc. degree was a collaboration between Nile University, Egypt and TU Berlin, Germany in 2014, working on vehicles detection and tracking in crowded scenes and sensitivity analysis of deep neural networks. In 2013, she was an intern in Speech & Sound Group in Sony Deutschland GmbH - Stuttgart, Germany, working on character recognition in natural images. She is a PhD student at Sabanci University, with a thesis on deep learning ensembles for image understanding. Currently, she is a fellow researcher in CVSSP group in University of Surrey, UK working on detection and generation of face morphing.



Muhammad Awais received the B.Sc. degree in Mathematics and Physics from the AJK University in 2001, B.Sc. degree in computer engineering from UET Taxila in 2005, M.Sc in signal processing and machine intelligence and PhD in machine learning from the University of Surrey in 2008 and 2011. He is currently a senior research fellow at the Centre for Vision, Speech and Signal Processing (CVSSP) at the University of Surrey. His research interests include machine learning, deep learning, self(un,semi)-supervised learning, NLP, audio-visual analysis, medical image analysis and computer vision.

Josef Kittler (M'74-LM'12) received the B.A., Ph.D., and D.Sc. degrees from the University of Cambridge, in 1971, 1974, and 1991, respectively. He is a distinguished Professor of Machine Intelligence at the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, U.K. He conducts research in biometrics, video and image dataset retrieval, medical image analysis, and cognitive vision. He published the textbook Pattern Recognition: A Statistical Approach and over 700 scientific papers. His publications have been cited more than 68,000 times (Google Scholar).

He is series editor of Springer Lecture Notes on Computer Science. He currently serves on the Editorial Boards of Pattern Recognition Letters, Pattern Recognition and Artificial Intelligence, Pattern Analysis and Applications. He also served as a member of the Editorial Board of IEEE Transactions on Pattern Analysis and Machine Intelligence during 1982-1985. He served on the Governing Board of the International Association for Pattern Recognition (IAPR) as one of the two British representatives during the period 1982-2005, President of the IAPR during 1994-1996. Currently he is a member of the KS Fu Prize Committee of IAPR.