university of groningen
faculty of mathematics and natural sciences
artificial intelligence

Understanding Vision Transformers Through Transfer Learning

Bachelor's Project Thesis

Lucas Monné, s3420108, l.monne.1@student.rug.nl,
Supervisor: Matthia Sabatelli, m.sabatelli@rug.nl

Abstract: Recently, transformers, originally purposed for natural language processing, have begun demonstrating a strong potential to not only compete, but outperform convolutional neural networks (CNN) in machine vision tasks. This paper investigates the transfer learning potential of vision transformers (ViT) in differing contexts, such as with small sample sizes and low- and high-degree differences between the source and target domains. Ultimately, when compared to state of the art CNNs, the ViT significantly outperforms the former on the grand majority of the carried experiments. Particularly, ViTs transfer with ease to depth-prediction tasks regardless of sample size. Results align with previous research, exposing new questions regarding the structure and a possible trade-off of performance versus training time and suggest real-life use cases in biomedical, material and processing industries where the conditions fit the experimental environment used.

1 Introduction

Architectures based on self-attention currently dominate as the standard method by which natural language processing (NLP) tasks are tackled. Attention is a mechanism introduced by Bahdanau et al. (2016) in an effort to improve on the manipulation of long-term semantic relations. More specifically, transformer models have demonstrated exemplary performance in the aforementioned domain (Vaswani et al., 2017)). For instance, models such as BERT (Devlin et al., 2019) or the GPT line of work (Radford et al., 2019; Brown et al., 2020) convey an ability to learn the latent fundamental syntactic relationships in text. In turn, this permits the acquisition of representations capable of generalizing across tasks. Furthermore, some newer transformer models also demonstrate a significant potential in scaling to larger models (Khan et al., 2021). For instance, this is shown by Fedus et al. (2021), with the Switch transformer capable of scaling to 1.6 trillion parameters, and Lepikhin et al. (2020) with the Gshard implementation scaling up to 600 billion parameters. Transfer learning is a method whereby a model trained on a specific task is reused on an other problem (Bozinovski and Fulgosi, 1976; Gao and Mosalam, 2018). At present, the most prevalent approach to NLP tasks with transformers lies in transfer learning: pre-training on sizeable (more than 1000 samples) text corpora and fine-tuning on smaller datasets tailored to the task at hand (Devlin et al., 2019). The considerable advantages exhibited by transformer models have inspired many in the computer vision (CV) community to explore them under the scrutiny of their performance in CV tasks. This includes: image recognition (Touvron et al., 2021; Dosovitskiy et al., 2021), object detection (Carion et al., 2020), segmentation (Ye et al., 2019), image-super resolution (Yang et al., 2020), video understanding (Sun et al., 2019), image generation (Chen et al., 2021), text-image synthesis (Ramesh et al., 2021) and visual question answering (Su et al., 2020). However, little literature exists regarding the transferability of transformers to such tasks. At present, convolutional neural networks (CNN) represent the main architecture used in machine vision models (LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2015). CNNs have been extensively examined with regards to their transferability. Thus, this implies a limited understanding of the transfer learning potential of transformers, justyfing this paper's empirical exploration of the overarching question: *How do vision transformers perform compared to convolutional neural networks in transfer learning regarding visual task adaptation?*. This question is primarily answered by comparing the transferability of visual transformers, namely ViT/L32, to that of CNNs, mainly Resnet152. Either model is initially pre-trained on image classification tasks such as ImageNet and is then transferred, in a supervised manner, to contexts where the target tasks vary in degree of dissimilarity or datasets contain few samples. This includes SmallNorb and its object azimuth/elevation prediction tasks, a grayscaled CIFAR-10, and Kitti with object depth prediction. Overall, findings establish the transformer's superiority in previously unexplored areas including their ability to perform well with little data, and their transferability to the aforementioned target domains. Moreover, this paper also paves the way for further research into the learning quality of transformers. Namely, the ViT/L32 with a majority of frozen parameters achieves 97.33% on Kitti, 96.33% on SmallNorb and 96.00% on the grayscaled CIFAR-10, where it either outperforms or remains competitive to unfrozen ViT and CNN alike. The rest of the paper is divided in the following manner: Section 2 summarizes the related works to this paper and Section 3 describes the ViT and CNN architectures. Section 4 presents the methodology used in this paper, including the datasets, pre-processing, pre-training and fine-tuning steps taken to train and test the models. Section 5 depicts the results obtained from the transferring of models, Section 6 discusses these results and lastly, the conclusion is established in Section 7.

2 Related Work

Amongst the numerous attempts at incorporating self-attention in CNNs, relevant instances include the introduction of non-local relationships to capture long-range dependencies (Wang et al., 2018), and channel-based attention (Hu et al., 2019). However, in these earlier applications, pixels are required to attend to every other pixel on an image. Consequently, naive implementations remain inefficient when scaling to realistic input sizes due to these models' cost growing quadratically with image size. As a result, numerous approximations of self-attention have been attempted. Particularly, one prominent method involves applying self-attention only locally rather than globally. This marks the introduction of local multi-head dot-product blocks similar to those seen in channel-based attention (Hu et al., 2019). There is strong evidence that these self-attention layers are competent in replacing convolutions entirely (Ramachandran et al., 2019; Zhao et al., 2020). Alternative approaches focusing on scalable, global self-attention exist. Weissenborn et al. (2020) achieves this by applying attention to differently sized blocks; Sparse Transformers in Child et al. (2019) utilize approximations that are scalable by nature. These models yield favorable results on CV tasks. In spite of these benefits, efficient implementations in more function-specific contexts, such as hardware acceleration, demand intricate engineering for a large part of these models.

In newer solutions, Cordonnier et al. (2020) apply self-attention to 2×2 dimensional patches that are extracted from the original input. However, as the patches are relatively small in size, this prevents the application of this model to larger images. Building on the aforementioned, recent experiments by Dosovitskiy et al. (2021) used larger sized patches in their ViT line of work for vision transformers. In such tasks, this resulted in precision scores inferior by a few percentage points relative to traditional residual networks (ResNet) of comparable size. Yet, these results are sensible: some of the inductive bias achieved by CNNs, such as translation equivariance and locality, do not apply to transformers. Consequently, in this context, a trend observed in training transformer architectures is their difficulty in generalizing successfully when this is performed using sparse amounts of data. Following these findings, further testing by Dosovitskiy et al. (2021) found that upon pre-training on larger datasets, such as 14M-300M images, contrasting results are obtained when transferring to tasks with less data points. This suggests that large scale training may overcome inductive bias, with the aforementioned model approaching and beating state-of-the art on multiple image recognition benchmarks.

On the other hand, vision transformer experiments of similar essence have focused on comparisons which only fine-tune their models on target domains that directly overlap with the source domain. For instance, Dosovitskiy et al. (2021) train and fine-tune only on datasets containing natural images, acknowledging the necessity for further exploration of transferring to different computer vision tasks and domains. Furthermore, all fine-tuning is performed using a constant 1000 examples per task. Hence, the relationship between the size of the fine-tuning dataset and the model's performance is not analyzed either. Thus, there is little knowledge regarding the extent to which vision transformers succeed in transfer learning. Visual information plays a vital role in the decision process of image classification; degradation of image features such as resolution can drastically alter complex visual information (Kannojia and Jaiswal, 2018). Zhai et al. (2020) presented the Visual Task Adaptation Benchmark (VTAB), designed for evaluating models on their ability to adapt to diverse unseen tasks with little examples. VTAB holds 19 tasks grouped in three sets that indicate the degree of dissimilarity relative to natural image classification: natural, representing standard vision problems; specialized, containing domain-specific images such as medical images; and structured, including synthetically generated images with a higher degree of domain-specificity. There is a lack of vision transformer literature analyzing transfer learning, namely in specialized and structured domains. In contrast, the same cannot be said for literature regarding CNN transfer learning, which is plentiful (Gu et al., 2017; Li et al., 2017; Shaha and Pawar, 2018; Kolesnikov et al., 2020). Transfer learning across tasks by usage of deep convolutional neural networks in unsupervised learning settings has been extensively analyzed (Raina et al., 2007; Mesnil et al., 2011). Similarly, several attempts at domain adaptation in a supervised fashion have been carried (Donahue et al., 2013). Srivastava and Salakhutdinov (2013) proposes discriminative transfer learning with tree-based priors using a multi-layer CNN, at their current time achieving state-of-the-art results on CIFAR100 and the MIR Flickr image-text dataset. Oquab et al. (2014) outperforms their year's top models, taking deep CNNs as seen in Krizhevsky et al. (2012) pre-trained on ImageNet and transferring to object detection tasks on PASCAL VOC. Numerous others have followed suit and found success in transferring from deep CNNs pre-trained in generalized image classification to differing target domains (Akçay et al., 2016; Redmon and Farhadi, 2016; Sun et al., 2017; Zhou et al., 2018). Hong et al. (2015) uses the pre-trained R-CNN model from Girshick et al. (2013) with help from a support vector machine to achieve performances competitive with state-of-the-art when transferring to online image tracking tasks. Rahman et al. (2020) pre-trained on colored images and evaluated on grayscale chest x-rays using a combination of AlexNet (Krizhevsky et al., 2012), ResNet18 (Lecun et al., 2010), DenseNet201 (Huang et al., 2018) and SqueezeNet (Iandola et al., 2016) for pneumonia detection, achieving performances higher than all other available literature on pathology detection. Last, Djolonga et al. (2020); Kolesnikov et al. (2020) explored the impacts of multiple factors, such as dataset size and visual information-related changes in pre-preprocessing, on the transferability and robustness of CNN models, primarily Resnets. This paper hence extends Djolonga et al. (2020); Dosovitskiy et al. (2021), not only contributing to the overall sparse literature regarding visual adaptation of transformers, but also exploring the benefits of parameter-freezing which can allow a model to outperform its regular variant. This suggests a significant enhancement to the transferability potential of such models, in turn raising new questions regarding additive pathways for research and a likely trade-off of performance versus training time.

3 Methods

The vision transformer model's design follows Dosovitskiy et al. (2021), which in turn follows the original transformer (Vaswani et al., 2017). This entails a simplistic model, justified by the idea that the implementation of a scalable NLP transformer architecture enables near off-the-shelf usage. Moreover, maintaining similar models facilitates a meaningful aggregation of the results obtained in Dosovitskiy et al. (2021). The rationale behind the choice of CNN architecture follows a similar reasoning, with Djolonga et al. (2020) instead serving as the point of comparison for CNN transferability.

3.1 ViT: Overall Architecture

The overall architecture of the vision transformer can be explained in a stepwise manner, beginning with an input image as seen on Figure 3.1. As the transformer is originally an NLP architecture, it can only process 1D token embedding sequences as input. In order to allow inputs of higher dimensions, in this case 2D images, the input se-

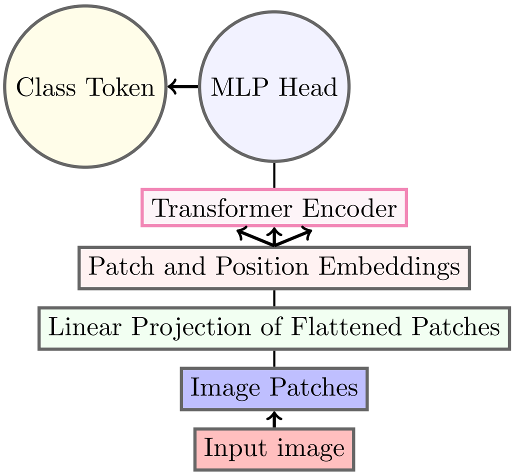
Class Token
MLP Head
Transformer Encoder
Patch and Position Embeddings
Linear Projection of Flattened Patches
Image Patches
Input image

Figure 3.1: Vision Transformer (ViT) architecture (Dosovitskiy et al., 2021)

quence (see Input Image in Figure 3.1) is reshaped from $I_s \in \mathbb{R}^{H \times W \times C}$ to flattened two-dimensional patches $I_p \in \mathbb{R}^{N \times (S^2 C)}$ (see Image Patches in Figure 3.1), wherein the product of H and W is the resolution of the source image, the square of S the resolution of each patch, C the amount of channels and N the per-input count of patches (Dosovitskiy et al., 2021). A linear projection is applied to the flattened patches and maps them to a hidden vector, constant through all of the transformer's layers (see Linear Projection of Flattened Patches in Figure 3.1). 1D Position embeddings are added to these patches in order to extract positional information (see Patch and Position embeddings in Figure 3.1). Following this, a trainable 1D embedding is added to the beginning of the input sequence containing embedded patches. This embedding serves in acquiring an overall representation of the input image. The embedded patches are then introduced as inputs to the transformer encoder (see Figure 3.1, explained in section 3.2) (Vaswani et al., 2017), which is responsible for mapping the input sequence I_p of discrete token representations (in this case pixels) to a sequence of continuous representations. A classification head (see MLP Head in Figure 3.1) is attached to the output of the transformer encoder. It is implemented by a multi-layer perceptron (MLP) with one linear layer (Dosovitskiy et al., 2021). In BERT, (Devlin et al., 2019), each input sequence begins with a special ([CLS]) classification token (see Class Token on Figure 3.1). The final hidden state of this token holds an overall representation of the sequence used in classification. The 24-layer, "Large" 32 × 32 (ViT/L32) input-patch variant of the vision transformer introduced by Dosovitskiy et al. (2021) was used for all experiments. In turn, this variant is heavily based off of BERT's structure (Devlin et al., 2019).

3.2 Transformer Encoder

As depicted by Figure 3.2, the transformer encoder is composed of repeating layers of multi-head self-attention and MLP blocks. The general attention function maps a set of queries and a set of keys both in dimensions d_k, which are mapped to value pairs in dimension d_v obtained from the input sequence I_p (see Section 3.1), with the query, Q, keys, K, and values, V, being matrices. The output, which is also a matrix, is then computed as the weighted sum over all values, wherein each weight depends on a compatibility function of the query and its related key. In the case of scaled dot-product attention (Vaswani et al., 2017) the self-attention implemented in the ViT, the compatibility function is a softmax function, which transforms the output into values between 0 and 1 so as to interpret it as probabilities. Additionally, The dot products of the queries and their related keys are computed, and divided each by the square root of d_k. Hence, this yields the equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^\top}{\sqrt{d_k}} V\Big) \tag{3.1}$$

The aforementioned linear projection maps I_p and its queries, keys and values to a trainable hidden vector in d_k, d_k and d_v, respectively. In multi-head self-attention, this is done h times, corresponding to the number of heads (in our case $h = 8$), and with the attention function computed simultaneously in each head. Ultimately, this yields an output matrix, W^O in dimension d_v. The values obtained from W^O are concatenated and once again projected, resulting in the final values used as input for the MLP blocks. This is best described by the equation:

$$\begin{gathered}\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1 ... \text{head}_h) W^O \\ \text{with any head}_i = \text{Attention}(Q{W_i}^Q, K{W_i}^K, V{W_i}^V)\end{gathered} \tag{3.2}$$

The MLP includes two layers and a Gaussian Error Linear Unit (GELU) is used as an activation function (Hendrycks and Gimpel, 2020). Layer normalization (Ba et al., 2016) is applied before every block by computing the mean and variance utilized in batch normalization (Ioffe and Szegedy, 2015), from all of the summed inputs to the neurons in a layer, on a single training case. Conversely, residual connections (He et al., 2015) are applied after every block.

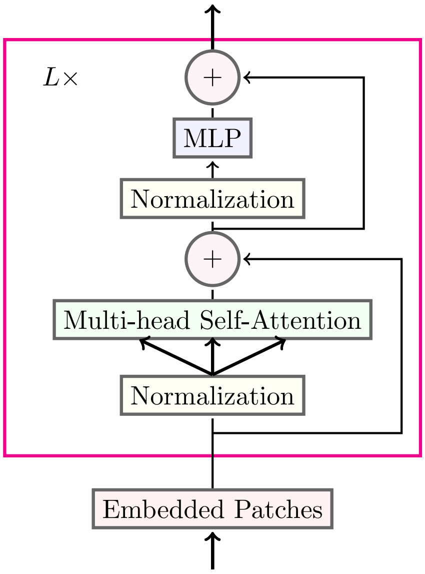

Figure 3.2: Transformer encoder architecture (Vaswani et al., 2017)

3.3 Convolutional Neural Network

As described in Sabatelli (2021), the primary mathematical operation underlying CNNs is the convolution. Convolutions require an input vector $u \in \mathbb{R}^M$, and a kernel $v \in \mathbb{R}^N$, which output a second vector of size $M - N + 1$ such that:

$$(u * v[i]) = \sum_{d=0}^{n-1} u_{d+i} v_d \tag{3.3}$$

Where the $*$ symbol indicates a convolution operation which does not flip the kernel. Djolonga et al. (2020); Dosovitskiy et al. (2021) utilized ResNets (He et al., 2015) and EfficientNets (Tan and Le, 2019) for transferability evaluation and comparison to transformers. ResNets, specifically, Big Transfer (BiT) (Kolesnikov et al., 2020) and ResNet-101x3 offered the best results in image classification, superior to the EfficientNets' performances. BiT requires a specific upstream pre-training procedure. ResNet-101x3 was not utilized in both studies. On the other hand, ResNet-152, the model BiT is based on, is therefore indirectly utilized for BiT and is also evaluated in Djolonga et al. (2020). Therefore, as it follows the relevant studies, ResNet-152 was opted for.

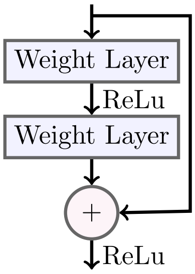

Figure 3.3: Residual network block unit (He et al., 2015).

3.4 ResNet: Overall Architecture

In residual learning, some, but not necessarily all, layers of the network attempt to approximate a function $H(x) - x$, where $H(x)$ is taken to be the original underlying mapping which must be fitted by the aforementioned layers. A plain residual network (ResNet), one without residual connections, consists of an initial 7×7 convolutional layer, which applies a convolution operation to the input, followed by a max pooling layer. Max pooling calculates the maximum value for each patch of the feature map obtained from the convolutional layer (LeCun et al., 1989; Gholamalinejad and Khosravi, 2020). The main body of any ResNet is composed of repeatedly alternating 3×3 convolutional layers (weighted layers) and rectified linear units (ReLu) (Nair and Hinton, 2010). Last, a global average pooling layer averages the output for each patch of the feature map obtained before sending it as input to a 1000-way fully-connected layer with softmax. Actual ResNets, which contain residual connections, are built using repeating residual block units, which replicate the aforementioned plain ResNet architecture with the addition of shortcut connections capable of skipping the outputs of a number of layers (see Figure 3.3). One residual block is defined by the equation:

$$y = F(x, \{W_i\}) + x \tag{3.4}$$

Where x and y are the input and output vectors of the layer where such residual connection exists. $F(x, \{W_i\})$ represents the residual mapping (i.e, $H(x)$) to be learned. Therefore, when factoring in the ReLu present between layers, noted as σ, the function for the repeating residual block in Figure 3.3 is $F = W_2\sigma(W_1 x)$. The ResNet152 used in this paper only contains more weighted layers, totaling 152. It still maintains the over-arching previously mentioned architecture.

4 Experimental Setup

Three individual empirical experiments were carried in order to investigate the transferability of transformers and compare it to that of CNNs. Particularly, the influence of fine-tuning dataset size and differing degrees of domain dissimilarity on transferability are explored. All experiments were performed on 32GB NVIDIA Tesla K-40 and V100 GPUs.

4.1 Datasets

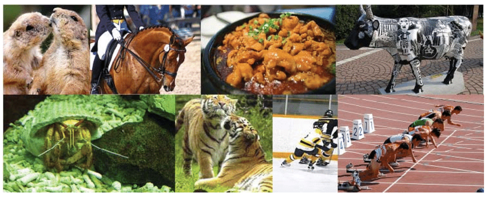

Figure 4.1: Example of ImageNet samples.

Overall, four datasets were involved in the performed experiments. Two variants of ImageNet (Deng et al., 2009), a database of natural human-annotated images used for image classification, were specifically used for pre-training in all experiments. This includes ImageNet-21k, with 21000 classes and 14 million images and ImageNet-1k, with 1000 classes and 1.3 million images. The other three datasets were used in fine-tuning. The Kitti dataset (Geiger et al., 2012), depth prediction and object detection tasks, with 10 classes and 7481 images, was used in both experiment 1 and 3. However, in experiment 1, a large, medium and small version of Kitti is used. The large dataset represents the entire dataset, with the two others containing 3741, and 1871 images respectively, while still maintaining the same number of classes. A grayscale version of CIFAR-10 (Krizhevsky et al.) and its collection of natural photos was used in experiment 2. CIFAR-10 has 60000 images and 10 classes. Lastly, SmallNorb (LeCun et al., 2004), containing artificial object azimuth and elevation prediction tasks, was used in experiment 3. SmallNorb holds 48600 images and five classes. Datasets which did not come with class-balanced or uneven phase splits were shuffled randomly and assigned the same number of classes manually. This is particularly relevant to the fine-tuning datasets. For such datasets, the same random seed was used for each model in a single experiment run and was changed in between runs. The training/validation/testing split followed a ratio of 20:4:1, which also translates into an 80%/16%/4% split, respectively. The dataset splits were renewed at each run. Lastly, image resolution is left intact to prevent any introduction of inductive bias (see Appendix A).

4.2 Experimental Protocol

In NLP, fine-tuning on downstream tasks is parameter inefficient (Houlsby et al., 2019). In fact, Kovaleva et al. (2019) finds that for several tasks in this domain, only the last few layers of the transformer change post-fine-tuning. Similarly, Michel et al. (2019) states that as little as one attention head per layer is required to be retained to maintain a sufficiently functioning model. In both transformers and CNNs, freezing parameters not only leads to an improvement of results (Lee et al., 2019; Eberhard and Zesch, 2021) but also in training times (Brock et al., 2017).Thus, for all experiments, two variants of each models were used. One wherein all classification layers were frozen except for the final classifying layer, and another with no frozen layers. These are referred to as "Frozen" and "Regular" models, respectively. Transfer learning is considered successful relative to a model trained from scratch when either asymptotic, jumpstart or learning-speed improvements are exhibited (Lazaric, 2012), as depicted in Figure 4.1 on the next page. In

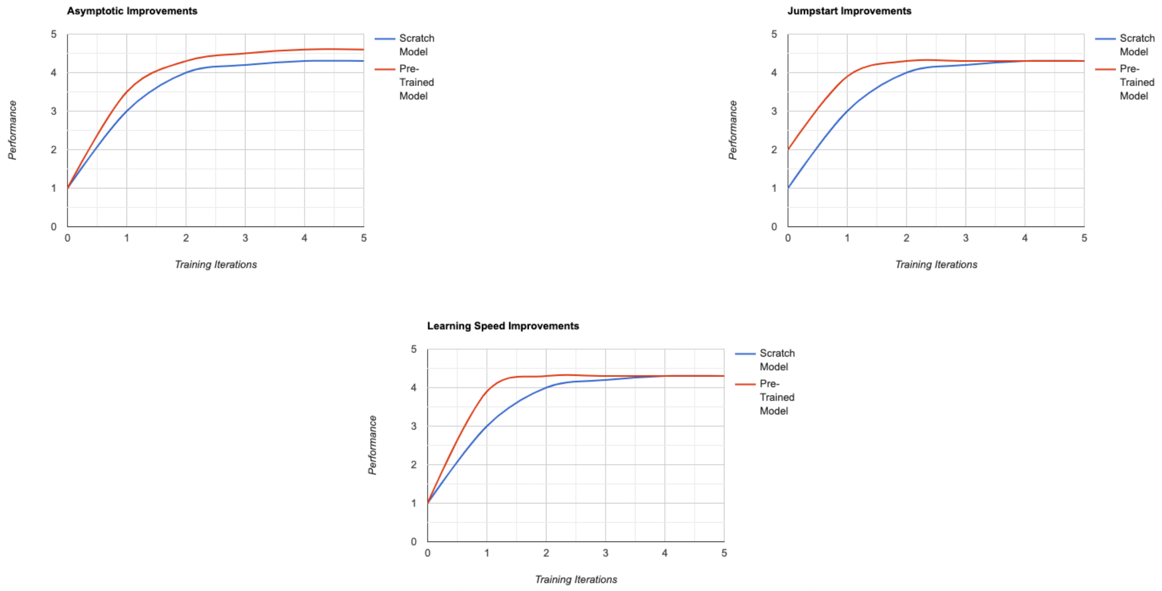

Figure 4.2: Visualization of the three possible desired outcomes of transfer learning as seen in (Lazaric, 2012)

asymptotic improvements, the performance of the pre-trained model is significantly greater than that of the scratch model. Jumpstart improvements are distinguishable by the pre-trained model's performance being closer to its final performance since its first training iteration. Finally, learning-speed improvements occur when the pre-trained model converges faster than its scratch variant. In order to identify any of the three improvements, a third variant of the ResNet152 was trained from scratch. It is referred to as the "Scratch" model. As a ViT/L32 model contains 307M parameters, it requires a significantly greater amount of time to train from scratch. Hence, for temporal reasons, no third variant is trained for the ViT. All experiments are repeated five times to reduce any possibility of result volatility consequent from random initialization. In each of these five runs, the training/validation/testing split remains the same between models, but different between runs. This design choice complements the random initialization of datasets, further reducing result volatility.

Experiment 1 Djolonga et al. (2020); Dosovitskiy et al. (2021) explore pre-training with different sized datasets. Yet, fine-tuning with different sized datasets is overlooked. Intuitively, a wider range of examples in the fine-tuning dataset is likely to lead to better performance as witnessed with pre-training. However, the relationship between performance and fine-tuning dataset size in this context has yet to be established. Thus, the relationship between the size of the data set used for fine-tuning and model performance is considered in this experiment. Source and target domains are maintained different to a high degree, where pre-training uses ImageNet's natural images in classification and fine-tuning uses domain-specific, depth prediction tasks obtained from Kitti. In order to investigate the relationship of different sized datasets (see Datasets), the fine-tuning protocol (see Experi-

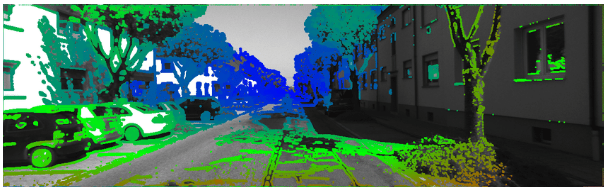

Figure 4.3: Example of a Kitti sample.

mental Protocol, Paragraph 1) was repeated for the small, medium and large variants of Kitti.

Figure 4.4: Example of grayscaled CIFAR-10 samples.

Experiment 2 Djolonga et al. (2020); Dosovitskiy et al. (2021) do not explore transfer learning with domains which differ only by simple features. Though visual information plays an important role in model performance, this matters only if complex visual information is unavailable (Kannojia and Jaiswal, 2018). Tebano et al. (2015) suggests color is not a complex visual feature. Thus, the ViT's potential in visual adaptation with regards to simple domain dissimilarities, in this case color, is examined. This is done through the comparison of the performance of a ViT and CNN where the source domain uses natural photos with three channels (RGB), and the target domain uses only single channelled (grayscale) natural images. If color is not a complex visual feature, then the transformer architecture is less likely to struggle in its ability to generalize. Hence, it is expected that the performance of the vision transformer will remain unaffected by this reduction of color channels. Pre-training uses ImageNet in RGB. All fine-tuning uses the grayscaled CIFAR-10 dataset. As part of pre-processing, image vectors are repeated three times on a new dimension to allow grayscale images to be interpreted as RGB.

Experiment 3 Elaborating on the same rationale as experiment 2, the transferability of ViTs using significantly different domains is also evaluated. As no previous literature has investigated this realm, it remains rather difficult to form an educated guess on the performance of ViTs to this regard. This is further confirmed by the findings of Djolonga et al. (2020) which reported that with high-difference domains, no metric predicts transferability well. Hence, experiment 3 appeals more to exploratory research. Nonetheless, most variants of BiT (see Section 3.3) introduced in Djolonga et al. (2020) transferred to moderate accuracies on datasets of similar nature. Therefore, it is expected that the ResNet152 demonstrates similar results in this case. Pre-training uses ImageNet and both ViT and CNN are fine-tuned on datasets containing tasks defined by VTAB as "structured" (see Introduction). As VTAB is a visual adaptation benchmark, it is better suited to evaluate the transfer learning potential of transformers, offering a wider variety of tests. The datasets in question consist of Kitti and SmallNorb.

Figure 4.5: Example of SmallNorb samples.

Metrics As all datasets were chosen/manipulated in accordance with the criterion of having balanced classes, performance is measured using accuracy. We define accuracy as the percentage of predictions which the model achieves correctly, following the equation:

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}} \times 100 \tag{4.1}$$

The total training time for each individual run in seconds is also recorded so as to obtain an idea of the efficiency of model training. These metrics are then calculated using an average of the five best models found in each run, where the training/validation accuracy and loss are computed at each epoch.

4.3 Pre-Training/Fine-Tuning

In all experiments, both ViTs and CNNs were pre-trained on ImageNet. The ImageNet-21k dataset was used in the pre-training of the ViT. In contrast, the ImageNet-1k dataset was used for the pre-training of the CNN. Moreover, a cross entropy (CE) (Cox, 1958) loss function was used for all training and testing. For N class labels i, the predicted probability p that x is the right observation and an indicator y_i conveying whether the observation is a correct classification, this is defined by the equation (as seen in Sabatelli (2021)):

$$-\mathbb{E}_{x,y\ P(X,Y))} \sum_{i=1}^{N} y_i log\, p_{\text{model}} f(x;\theta) \qquad (4.2)$$

Where the separate loss per label is calculated and the result summed up. CE is proven to be a calibrated loss function (Tewari and Bartlett, 2005), in turn promising well-behaved probability estimates. Consequently, due to its ability to minimize distances between two probability distributions, CE has historically offered consistent top-1/top-5 performance in image classification in both binary and multi-class classification tasks (Cao et al., 2018; Ruby and Yendapalli, 2020) as well as relatively high convergence speeds (Martinez and Stiefelhagen, 2018). As only multi-class classification is present across all experiments, this justifies its usage. In all runs, stochastic gradient descent (Robbins and Monro, 1951) with a learning rate of 0.001 and momentum of 0.9 (Rumelhart et al., 1986) is used in order to keep parameters as similar to Dosovitskiy et al. (2021) as possible. Stopped training (Sjöberg and Ljung, 1992; Finnoff et al., 1993) with a patience of seven epochs and an exponential learning rate decay (An et al., 2017) with a patience of five epochs and a factor of 0.5 are also utilized during fine-tuning.

5 Results

For each dataset, the significance in mean accuracy difference of six model pairs was investigated: frozen CNN versus frozen ViT, frozen CNN versus regular ViT, regular CNN versus frozen ViT, regular CNN versus regular ViT, scratch CNN versus frozen ViT and scratch CNN versus regular ViT. As K-fold cross validation is not utilized during fine-tuning of the models, the assumption of independence is not violated (Demšar, 2006). A Shapiro-Wilk test was run for each investigated dataset pair so as to detect any violations of normality. For all pairs, results lead to acceptance of the null hypothesis that the aforementioned were not different from normally distributed sets ($p > 0.05$ for all). Thus, this justifies the usage of a paired Student's t-test to determine any significant difference in mean accuracies between models (Dietterich, 1998). Moreover, the validation accuracies/losses throughout their epoch counts were both plotted for each individual dataset/variant. Validation losses may be visualized in Appendix D. As such, the y-axes of these graphs constitute the loss/accuracy of any given model in decimal values, and the x-axes, the epoch count in numerical values. The shaded areas around each plotted curve represent their standard uncertainties. Uncertainties for all graphs were calculated as standard errors of means (SEM), where the standard deviation of each sample was utilized in the computation of each SEM. Lastly, tables containing the average percentage test accuracy of each model over their five runs are displayed in this section. Each model's average training time in minutes can be found in Appendix B.

5.1 Kitti

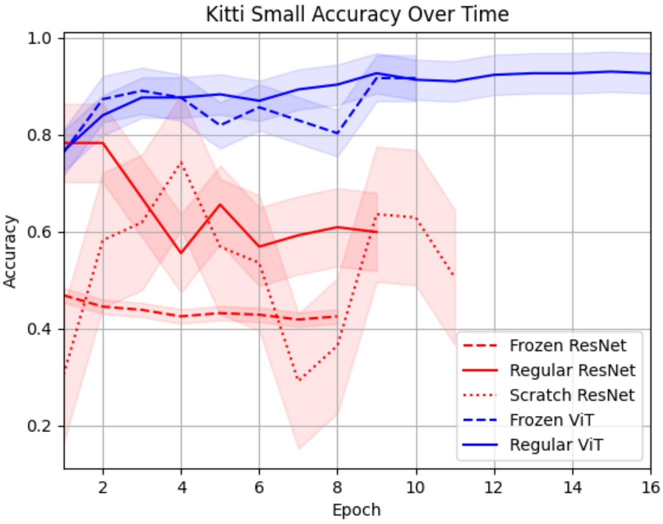

Figure 5.1: Validation accuracy for models fine-tuned on the small variant of the Kitti dataset

Table 5.1: Average Test Performances (in percentages) on Kitti Small

Model	CNN	ViT
Regular	73.23 ± 0.033	87.93 ± 0.05
Frozen	46.52 ± 0.031	89.99 ± 0.025
Scratch	77.14 ± 0.011	

Kitti – Small As seen on Table 3.1, the regular and frozen ViT produce the best average results compared to all other models. Significant differences in mean performance score were found between the regular/frozen ViT and the regular, frozen and scratch CNN ($p = 0.002/p < 0.001; p < 0.001$ for both; $p = 0.02/p < 0.002$, respectively). Table B.1 (see Appendix B) conveys that all CNN have faster average training times than the ViT models. However, no significant differences were found between the mean of the regular/frozen ViT training times and that of the CNN variants for any given pair ($p > 0.05$). Looking at the curves on Figures 5.1 and D.1, the frozen CNN depicts a high validation loss and a validation accuracy below 50 percent. This suggests that the frozen CNN model cannot generalize beyond the training set. Furthermore, taking into account that the regular CNN model generalizes to a sufficient extent, it also suggests that training only the lower layers of a CNN alone are not enough for a CNN model to take on object depth prediction tasks, even when pretrained on ImageNet. Last, the scratch and regular CNN depict validation accuracies which decrease over time after the first two epochs, possibly indicative of the models overfitting. The regular CNN begins at higher accuracies on epoch one, and not only converges faster (in two epochs) but to a higher accuracy than its scratch variant. Thus, a visible mix of asymptotic, learning speed and jumpstart improvements are present for the regular CNN model when compared to its scratch variant's validation accuracy plots.

Table 5.2: Average Test Performances (in percentages) on Kitti Medium

Model	CNN	ViT
Regular	75.51 ± 0.025	81.94 ± 0.076
Frozen	46.63 ± 0.043	94.59 ± 0.096
Scratch	78.74 ± 0.026	

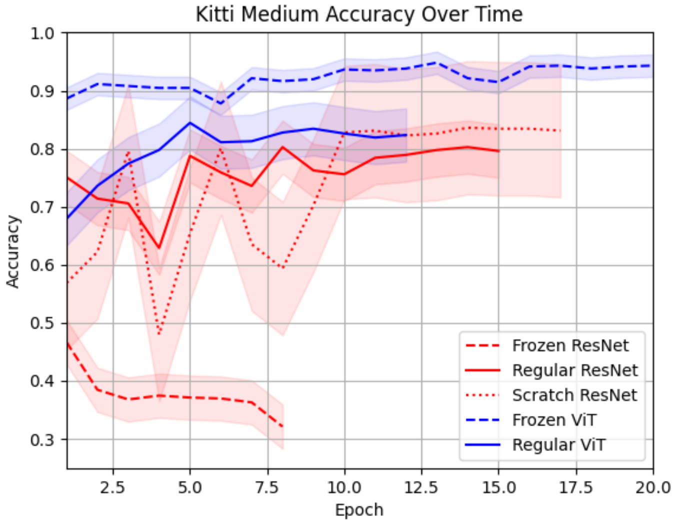

Figure 5.2: Validation accuracy for models fine-tuned on the medium variant of the Kitti dataset

Kitti – Medium As seen on Table 3.2, the regular and frozen ViT produce the best average results compared to all other models. Significant differences in mean performance score between the regular ViT and the frozen CNN ($p < 0.001$) were found, but not when compared to the regular CNN and the scratch CNN ($p > 0.05$ for both). Additionally significant differences in mean performance score were found between the frozen ViT and the aforementioned ($p < 0.001$ for all). Table B.2 (See Appendix B) also conveys that all CNN models have faster average training times than the ViT models. Though a significant difference in mean training times was found between the regular and frozen ViTs and the frozen CNN ($p = 0.04; p = 0.01$, respectively), no significant differences in time were found for any other given pair ($p > 0.05$). Looking at the graphs on figures 5.2 and D.2, the frozen CNN once again depicts an increasing high validation loss and a validation accuracy below 50 percent which continues decreasing. As once again the regular CNN model generalizes successfully, this reinforces the idea that the frozen CNN's final layer, by itself, cannot solely make use of the feature maps acquired through pre-training on ImageNet in order to solve tasks from the Kitti dataset. The regular CNN begins at higher accuracies on epoch one, and converges faster (in eight epochs) than its scratch variant. Thus, a visible mix of learning speed and jumpstart improvements are present for the regular CNN model when compared to its scratch variant's validation accuracy plots.

Table 5.3: Average Test Performances (in percentages) on Kitti Large

Model	CNN	ViT
Regular	84.25 ± 0.011	95.32 ± 0.022
Frozen	38.77 ± 0.016	97.33 ± 0.035
Scratch	85.98 ± 0.042	

Kitti Large As seen on Table 3.3, the regular and frozen ViT produce the best average results compared to all other models. Significant differences in mean performance score between the regular/frozen ViT and the regular, frozen and scratch CNN variants ($p = 0.01/p < 0.001$ for the regular, $p < 0.001$ for both, regarding the frozen and scratch). Table B.3 (See Appendix B) also conveys that all CNN models have faster average training times than the ViT models. However, no significant differences in time were found between the mean of the regular/frozen ViT training times and that of the CNN variants for any given pair ($p > 0.05$). Similarly to Kitti Small and Medium, looking at the graphs on figures 5.3 and D.3, the frozen CNN conveys a high, decreasing validation loss and a validation accuracy below 50 percent. The regular and scratch CNNs quickly converge at epoch four, after which their accuracies begin decreasing and the models display signs of overfitting. Nonetheless, until epoch four the models display no suspicious behaviour. Taking these two observations into account, this further supplements the initial explanation that for object depth prediction, a ResNet pre-trained on ImageNet, whose layers are all frozen except for the final classifying layer, does not have sufficient training and information to appropriately classify elements from the Kitti dataset. No jumpstart, learning speed or asymptotic improvements are visible for the regular CNN. Hence, this suggests the CNN's transfer from ImageNet to the entire Kitti dataset is not successful.

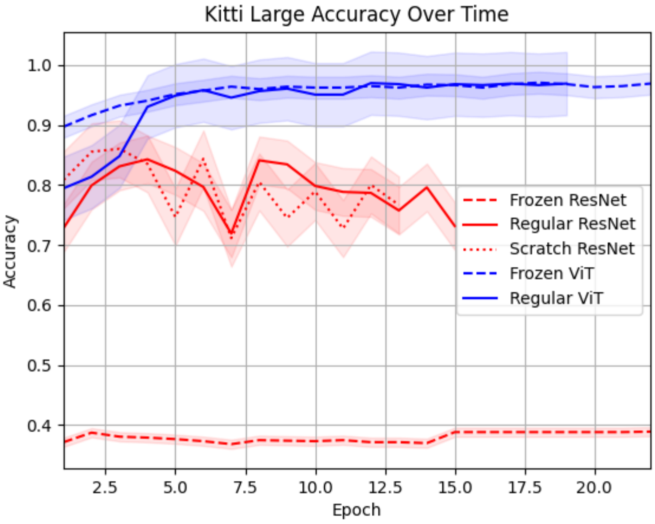

Figure 5.3: Validation accuracy for models fine-tuned on the large variant of the Kitti dataset

Kitti Overview One can see that the performance of all models on the Small dataset, except the frozen CNN, is inferior to that on the Medium dataset, which is in turn inferior to performances witnessed on the Large dataset, where: the regular CNN performance increases from 73.23% to 84.25%; the regular ViT increases from 87.93% to 95.32%; the scratch CNN increases from 77.14% to 85.98%, and the frozen ViT increases from 89.99% to 97.33%.

5.2 Grayscale CIFAR-10

Table 5.4: Average Test Performances (in percentages) on Grayscale CIFAR-10

Model	CNN	ViT
Regular	94.23 ± 0.0077	96.26 ± 0.0036
Frozen	76.66 ± 0.0049	96 ± 0.0044
Scratch	93.98 ± 0.0089	

As seen on Table 3.4, the regular and frozen ViT produce the best average results compared to all other models. Significant differences in mean performance score were found between the regular/frozen ViT and the regular, frozen and scratch CNN variants ($p = 0.02/p = 0.01; p < 0.001/p =$

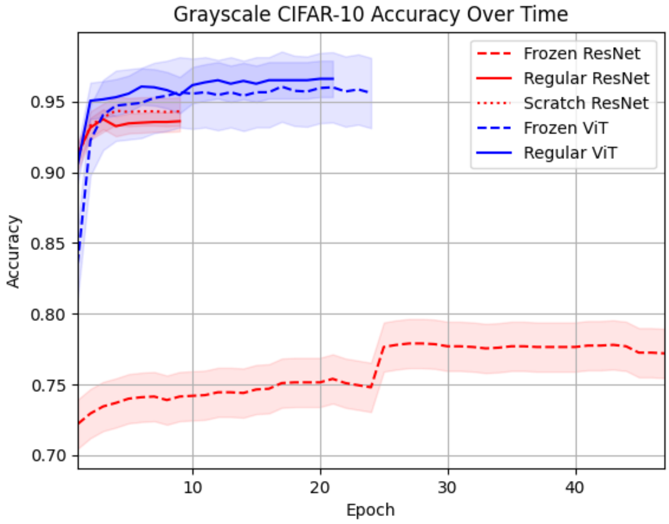

Figure 5.4: Validation accuracy for models fine-tuned on the grayscale variant of the CIFAR-10 dataset

$0.01; p < 0.001/p = 0.004$, respectively). Table B.4 (See Appendix B) also conveys that all CNN models have faster average training times than the ViT models. A significant difference in mean training time was found between the regular/frozen ViT and the scratch CNN ($p = 0.005/p < 0.001$). No significant differences in time were found for any other given pair ($p > 0.05$). Looking at the graphs on figures 5.4 and D.4, learning speed improvements for the regular CNN can be extracted from the validation accuracy plots when compared to the scratch CNN's curves, where the regular CNN converges at epoch eight and the frozen CNN at epoch nine.

5.3 SmallNorb

As seen on Table 3.5, the regular ViT and CNN produce the best average results compared to all other models, with that of the CNN being slightly greater in performance. Significant differences in mean performance score between the regular/frozen ViT and the frozen CNN variant ($p < 0.001$) were found. On the other hand, the opposite is true regarding performance comparison between the regular/frozen ViT and regular and scratch CNN ($p > 0.05$). Table B.5 (See Appendix B) also conveys that all CNN models have faster average training times than the ViT models. However, no significant differences in time were found for any given pair ($p > 0.05$). Looking at the graphs on figures 5.5 and D.5, visible learning speed, jumpstart and asymptotic improvements are visible for both the frozen and regular CNN as they converge faster and to greater accuracies than their scratch variant.

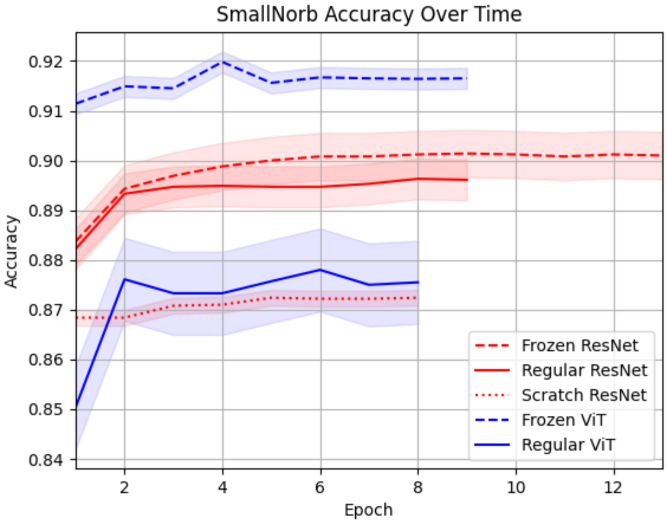

Figure 5.5: Validation accuracy for models fine-tuned on the SmallNorb dataset

Table 5.5: Average Test Performances (in percentages) on SmallNorb

Model	CNN	ViT
Regular	96.85 ± 0.0078	96.82 ± 0.0042
Frozen	90.80± 0.0037	96.14 ± 0.0014
Scratch	96.08 ± 0.017	

6 Discussion

6.1 Experiment 1

Extending the research carried by (Dosovitskiy et al., 2021), the first experiment aimed to investigate the transfer learning potential of vision transformers with fine-tuning datasets of not only highly different target domains, but also containing sparse amounts of data. Irrespective of ViT or CNN, one can see that for all regular and scratch variants the average performance increases as the size of the fine-tuning dataset increases. Hence, this reinforces the initial hypothesis that larger amounts of data lead to better performance. This follows the findings of (D'souza et al., 2020), which investigated the transfer learning optimization of CNN structures using smaller sample sizes. Building on this and factoring in the overall performance of the ViT compared to that of the CNN, a second notable trend is the ViT's significant positive difference in performance. Additionally, it was noted for all three size dataset variants that the frozen CNN had poor prediction accuracy and could not generalize beyond its training set. Both frozen models of the ViT and CNN had all layers frozen except for the final classifying layer. Therefore this suggests that the feature maps acquired through the pre-training of a ResNet on ImageNet is not sufficient to solve object depth prediction tasks. On the other hand, as the frozen ViT performed better than all other models, it also supports the idea that the information acquired by a ViT pre-trained on ImageNet is enough for it to solve the aforementioned tasks. Thus, this all contributes to the suggestion that the transferability of transformers to depth task predictions with smaller dataset sizes is significantly superior to that of a CNN. (D'souza et al., 2020) also suggests that as sample size decreases, the importance of network structure increases. One likely explanation for the ViT's greater performance in such conditions may then span from the structural differences visible in transformers. Notably, there are two substantial structural difference in ViTs: their incorporation of attention-based learning caked in each layer and their relative lack of inductive bias. As such, this suggests that either appropriate incorporation of self-attention in neural structures or building architectures which do not rely on inductive bias may serve in improving their transfer learning potential, as the information learned by such architectures appears to hold greater generalized value. However, one must take into account that the ViT/L32 model is pre-trained on Imagenet-21k, while Imagenet-1k is used for the ResNet152. Therefore, another possible explanation for the significant difference in performance may span from the substantially larger dataset the ViT is pre-trained on. Last, one can see that in both the small and large variants of the per-epoch accuracy plots the regular and scratch ResNet152's validation accuracies decrease over time. This may imply an issue specific to CNNs in the parameters used for experiment 1. In this case, it is likely the learning rate that may have been too high for as the regular CNN model converges in as little as two epochs. Further experimenting with lower learning rates in Appendix C reveals that indeed, with a lower learning rate the CNN's validation accuracy does not decrease over time. However, the average accuracies obtained do not exceed that which were obtained with the higher learning rates. Hence, it is unlikely that this explains the CNN's relatively poorer performance.

6.2 Experiment 2

Further extending (Djolonga et al., 2020; Dosovitskiy et al., 2021), the second experiment aimed to investigate the transferability of vision transformers with regards to low-degree differences in source/target domains. This was approached by changing non-complex visual information in the fine-tuning dataset's images through the manipulation of color. Irrespective of ViT or CNN, one can see that target domains which only differ slightly to their source domain by means of changes in simple visual information are not an obstacle to transfer learning as either structure achieves higher performances than the scratch CNN. This supports the initial hypothesis that changes in simple visual information does not significantly affect model performance. These results also corroborate the similar works of (Kannojia and Jaiswal, 2018) which were performed using changes in resolution, as well as that of (Tebano et al., 2015) which suggested that color does not hold complex visual information. Hence, the absence of color, does not affect a model's transfer learning potential. Building on this and factoring in overall ViT performance compared to CNN performance, the regular and frozen ViT convey a positive significant difference in performance. These combined findings contribute to the idea that ViT models may hold greater transferability than CNNs in low-difference transfer learning. In spite of this, the average training times remain lower in CNN models than for ViTs. Although a significant difference was found regarding any ViT versus the scratch CNN model, all other models revealed insignificant differences. Regardless, this suggests a possible trade-off of training time versus performance for the ViT models. However, recall that training was performed on single GPUs which varied in type (either Tesla K-40/Tesla V100) and thus varied in processing speed. Moreover, the cluster used to train such models was also charged with a variety of different other tasks. Errors in the cluster and latency times were not available for recording. Therefore, it is possible that the training times may not represent the true training time if this experiment was to be carried under ideal controlled conditions.

6.3 Experiment 3

Extending experiment 2 and its associated studies, experiment 3 aimed to investigate transfer learning in the context of high-degree differences between source and target domains. Irrespective of ViT or CNN, one can see that target domains which greatly differ from their source domain are not necessarily an obstacle to transfer learning. In object prediction tasks, the regular CNN achieves similar performances to its scratch variant. All ViT variants significantly outperform the aforementioned. In object elevation/azimuth prediction, the validation accuracy plots depict both frozen and regular CNNs achieving all three types of improvements when compared to the scratch CNN, suggesting a success in transfer learning. Additionally, all variants achieve higher performances than the scratch CNN. In spite of that, differences in performance observed with all variants except the frozen CNN were found to be insignificant. Hence, this suggests that transfer learning to object elevation/azimuth prediction may not automatically be the best approach to achieving top performance. Looking at the comparative performance of ViTs to CNNs in the aforementioned, both regular models achieve nearly identical performances, with the CNN outperforming by a small degree. Additionally, this difference in performance was also found to be insignificant, further reinforcing the idea that these two models perform on a highly similar level. Due to the novelty of transformers, little literature involving ViTs in a transfer learning context exists. However, putting these findings together with that of (Djolonga et al., 2020; Dosovitskiy et al., 2021), the suggestion that ViT models transfer to domains of high dissimilarity with more ease than CNNs is only reinforced. Yet, average training times for the CNN models are faster than that of the ViT for both datasets. However, in the SmallNorb dataset these differences were found to be insignificant. Conversely, on the Kitti dataset a significant difference in training times was found when comparing any ViT to the frozen CNN. Still, this model performed poorly with an accuracy well below 50%. Thus, this only suggests a likely insignificant trade-off in training time versus performance for ViT models in these contexts.

7 Conclusion

This paper aimed to investigate the transferability of vision transformers and compare it to that of convolutional neural networks. Through the three sub-experiments which were carried, ViTs were seen and confirmed to transfer significantly better than CNNs when constrained to small sample sizes or with a low-degree difference between the source and target domains. Furthermore, compared to CNNs, ViTs also demonstrated superior transferability regarding object depth prediction tasks, and near identical transferability to object azimuth/elevation prediction tasks. These aforementioned tasks represent transfer with regards to high-degree differences in target and source domain. In conclusion, as the ViT performed better than CNNs in a majority of tasks, this suggests that ViTs have a significant advantage in transfer learning potential when compared to CNNs. The most likely explanation for such transcendence in transfer learning potential lies in a ViT's structural differences compared to a CNN (D'souza et al., 2020). Additional experiments are necessary in order to understand whether it is the lack of inductive bias, the addition of self-attention or a completely different force altogether responsible for the ViT's observed edge in transfer learning. However, this may be explained when noting that the frozen ViT model consistently offers the best results in some datasets and similar results as that of its regular variant, while the frozen ResNet model pales in comparison. Mainly, the ViT's edge may lie in its ability to learn longer-term dependencies, which spans from its incorporation of self-attention. In some cases, such as object-depth prediction, this ability may result in the quality of the level of knowledge attained by a ViT to be of higher value compared to that of a CNN. Thus, this not only poses a possible explanation for such edge in transferability, it also suggests that longer-term dependencies may hold a higher degree of complexity or generalization, introducing a possible relationship between the transferability of an architecture and the quality of the knowledge it crystallizes. A following examination of the ViT's off-the-shelf performance on Kitti and other object depth prediction tasks can further develop the aforementioned explanation. The ViT's transferability to other different target domains such as semantic segmentation or image retrieval (Sinha et al., 2018) are also still required to ascertain its predominant performance. However, the ViT model requires more finetuning time than a CNN. This particular latency suggests a trade-off between its performance and the time taken for the model to generalize. As the differences in training time between ViT and CNN were found to be insignificant, further studies are required to confirm whether this trade-off can put the ViT at a disadvantage. Nevertheless, the frozen ViT model's prowess builds on (Lee et al., 2019), supporting the findings that even with a fourth of the layers unfrozen, a ViT performs similarly to its regular counterpart. Moreover this seems particularly relevant to small sample sizes, where the frozen ViT outperformed all other models. All in all, with time, one might find that the usage of transfer learning with ViTs or other self-attention based architectures may find their place in real-life cases where samples do not exist in high numbers, such as in biomedical engineering, the process industry or material science (Zhu et al., 2020).

References

Samet Akçay, Mikolaj E. Kundegorski, Michael Devereux, and Toby P. Breckon. Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1057–1061, 2016.

Wangpeng An, Haoqian Wang, Yulun Zhang, and Qionghai Dai. Exponential decay sine wave learning rate for fast deep neural network training. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018.

S. Bozinovski and A. Fulgosi. The influence of similarity and the transfer of training on the perceptron training. In *Proceedings of Symposium Informatica*, volume 3, pages 1–5, 1976.

Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. Freezeout: Accelerate training by progressively freezing layers, 2017.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Jie Cao, Zhe Su, Liyun Yu, Dongliang Chang, Xiaoxu Li, and Zhanyu Ma. Softmax cross entropy loss with unbiased decision boundary for image classification. In *2018 Chinese Automation Congress (CAC)*, pages 2028–2032, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and

Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer, 2021.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019.

Nadav Cohen and Amnon Shashua. Inductive bias of deep convolutional networks through pooling geometry, 2017.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers, 2020.

D.R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society*, 20(2):215–242, 1958.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: a large-scale hierarchical image database. pages 248–255, 06 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7): 1895–1923, 1998.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. *CoRR*, abs/2007.08558, 2020.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition, 2013.

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.

Rhett N. D'souza, Po Yao Huang, and Fang Cheng Yeh. Structural analysis and optimization of convolutional neural networks with a small sample size. *Scientific Reports*, 10, 12 2020. ISSN 20452322.

Onno Eberhard and Torsten Zesch. Effects of layer freezing when transferring deepspeech to new languages, 2021.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021.

William Finnoff, Ferdinand Hergert, and Hans Georg Zimmermann. Improving model selection by nonconvergent methods. *Neural Networks*, 6(6):771–783, 1993. ISSN 0893-6080.

Yuqing Gao and Khalid M. Mosalam. Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):748–768, 2018.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Hossein Gholamalinejad and Hossein Khosravi. Pooling methods in deep neural networks, a review, 09 2020.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 11 2013. doi: 10.1109/CVPR.2014.81.

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes, 2020.

Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Li Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2020.

Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *Proceedings of the 32nd International Conference on Machine Learning, 2015, Lille, France, 6-11 July 2015*, 2015.

Max Horn, Kumar Shridhar, Elrich Groenewald, and Philipp F. M. Baumann. Translational equivariance in kernelizable attention, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

Eyke Hüllermeier, Thomas Fober, and Marco Mernberger. *Inductive Bias*, pages 1018–1018. Springer New York, New York, NY, 2013. ISBN 978-1-4419-9863-7.

Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

Suresh Kannojia and Gaurav Jaiswal. Effects of varying resolution on performance of cnn based image classification an experimental study. *International Journal of Computer Sciences and Engineering*, 6:451–456, 09 2018.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey, 2021.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning, 2020.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert, 2019.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/ kriz/cifar.html`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

Alessandro Lazaric. Transfer in Reinforcement Learning: a Framework and a Survey. In Martijn van Otterlo Marco Wiering, editor, *Reinforcement Learning - State of the art*, volume 12, pages 143–173. Springer, 2012. URL `https://hal.inria.fr/hal-00772626`.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4): 541–551, 1989.

Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2: II97–II104, 2004. ISSN 1063-6919.

Yann Lecun, Koray Kavukcuoglu, and Clement Farabet. Convolutional networks and applications in vision. pages 253–256, 05 2010.

Jaejun Lee, Raphael Tang, and Jimmy Lin. What would elsa do? freezing layers during transformer fine-tuning. *CoRR*, abs/1911.03090, 2019.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. 6 2020.

Xiaogang Li, Tiantian Pang, Biao Xiong, Weixiang Liu, Ping Liang, and Tianfu Wang. Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–11, 2017.

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers, 2021.

Manuel Martinez and Rainer Stiefelhagen. Taming the cross entropy loss, 2018.

Grégoire Mesnil, Yann Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley, Pascal Vincent, Aaron Courville, and James Bergstra. Unsupervised and transfer learning challenge: A deep learning approach. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, UTLW'11, page 97–111. JMLR.org, 2011.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one?, 2019.

Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines vinod nair. volume 27, pages 807–814, 06 2010.

Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 06 2014.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. URL `https://gluebenchmark.com/leaderboard`.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Tawsifur Rahman, Muhammad E. H. Chowdhury, Amith Khandakar, Khandaker R. Islam, Khandaker F. Islam, Zaid B. Mahbub, Muhammad A. Kadir, and Saad Kashem. Transfer learning with deep convolutional neural network (cnn) for pneumonia detection using chest x-ray. *Applied Sciences*, 10(9):3233, May 2020. ISSN 2076-3417.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 759–766, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. URL `https://doi.org/10.1145/1273496.1273592`.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models, 2019.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016.

Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586. URL `https://doi.org/10.1214/aoms/1177729586`.

U Ruby and V Yendapalli. Binary cross entropy with deep learning technique for image classification. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(10), 2020.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

Matthia Sabatelli. *Contributions to Deep Transfer Learning: From Supervised to Reinforcement Learning.* PhD thesis, University of Liège, 2021.

Manali Shaha and Meenakshi Pawar. Transfer learning for image classification. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 656–660, 2018. doi: 10.1109/ICECA.2018.8474802.

Rajat Kumar Sinha, Ruchi Pandey, and Rohan Pattnaik. Deep learning for computer vision tasks: A review. *CoRR*, abs/1804.03928, 2018.

J Sjöberg and L Ljung. Overtraining, regularization, and searching for minimum in neural networks. *IFAC Proceedings Volumes*, 25(14):73–78, 1992. ISSN 1474-6670. 4th IFAC Symposium on Adaptive Systems in Control and Signal Processing 1992, Grenoble, France, 1-3 July.

N. Srivastava and R. Salakhutdinov. Discriminative transfer learning with tree-based priors. *Advances in Neural Information Processing Systems*, 01 2013.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations, 2020.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era, 2017.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning, 2019.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

Riccardo Tebano, Gianluigi Ciocca, Silvia Corchs, Francesca Gasparini, and Emanuela Bricolo. Does color influence image complexity perception? 03 2015.

Ambuj Tewari and Peter L. Bartlett. On the consistency of multiclass classification methods. In Peter Auer and Ron Meir, editors, *Learning Theory*, pages 143–157, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31892-7.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks, 2018.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models, 2020.

Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution, 2020.

Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2020.

Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition, 2020.

Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, and Yi Xu. Scale-transferrable object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 528–537, 2018.

Qun-Xiong Zhu, Zhong-Sheng Chen, Xiao-Han Zhang, Abbas Rajabifard, Yuan Xu, and Yi-Qun Chen. Dealing with small sample size problems in process industry using virtual sample generation: a kriging-based approach. *Soft Computing*, 24:6889–6902, 2020. ISSN 1433-7479.

A Inductive Bias

One substantial difference between transformers and CNNs consists of inductive bias. Inductive bias is defined by a set of assumptions a learning algorithm adopts in order to practice the generalization of observed data into a model (also known as induction) (Hüllermeier et al., 2013). As stated initially, transformers do not hold inductive biases to the same extent as CNNs, where CNNs have better induction potential Cohen and Shashua (2017). This is explained by two characteristics present in the latter but not the former: translational equivariance and locality. Battaglia et al. (2018) define locality as the idea that in the input space, points relatively close to each other share some relational meaning. Following Gordon et al. (2020), a function $f : Z \rightarrow Z$ is translation equivariant to a translation operation $T_t : X \times Z \rightarrow Z$ if $f(T_t(Z_f)) = T_t(f(Z_f))$ for all $t \in X$ and $Z \in Z_f$. For instance, take Z to be a set of M measurements m at locations l, such that $Z = ((l_1, m_1), ..., (l_M, m_M))$. If the translation operation is defined as $T_t Z = ((l_1{-}t, m_1), ..., (l_M{-}t, m_M))$ and initially shifting the measurements and then applying the function leads to the same results as the performing these same operations in reverse order then f is translation equivariant with respect to shifts of measurement locations (Horn et al., 2021). In CNNs, the aforementioned two are present in each convolutional layer, thus throughout the whole model. Conversely, in transformers, locality and translational equivariance are solely witnessed in the MLP layers (Li et al., 2021; Horn et al., 2021). Dosovitskiy et al. (2021) increase the resolution of their images for fine-tuning to ensure optimal results (Touvron et al., 2021; Kolesnikov et al., 2020). However, this introduces new inductive bias to the ViT. In order to minimize introduction of other inductive biases and thus ensure the transformer learns relevant representations, this fine-tuning procedure was discarded.

B Extra Results: Training Time

Below you may find all experimental results regarding the average training times in minutes for all models and their respective variants.

Table B.1: Average Training Times (in minutes) on Kitti Small

Model	CNN	ViT
Regular	37.25 ± 3.61	83.35 ± 5.53
Frozen	15.77 ± 8.25	75.88 ± 10.73
Scratch	31.73 ± 11.70	

Table B.2: Average Training Times (in minutes) on Kitti Medium

Model	CNN	ViT
Regular	62.050 ± 2.95	99.10 ± 15.63
Frozen	26.68 ± 2.01	128.91 ± 5.70
Scratch	84.87 ± 15.23	

Table B.3: Average Training Times (in minutes) on Kitti Large

Model	CNN	ViT
Regular	116.86 ± 27.76	208.23 ± 54.18
Frozen	134.03 ± 20.65	241.50 ± 67.11
Scratch	99.20 ± 24.40	

Table B.4: Average Training Times (in minutes) on Grayscale CIFAR-10

Model	CNN	ViT
Regular	133.88 ± 16.35	149.76 ± 22.62
Frozen	122.61± 26.85	158.35 ± 38.86
Scratch	57.03 ± 24.8	

Table B.5: Average Training Times (in minutes) on SmallNorb

Model	CNN	ViT
Regular	28.87 ± 2.45	39.28 ± 3.33
Frozen	36.16 ± 9.4	61.08 ± 19.7
Scratch	32.87 ± 4.47	

C Overfitting CNN: Learning Rate Experiment

In experiment 1, the regular and scratch CNN converge rather fast (< 5 epochs), before their validation accuracies begin decreasing. As the frozen variant is incapable transferring at all, in the context of object depth prediction, it was omitted. Further experiments were carried by fine-tuning the

CNN with half of the learning rate used in experiment 1. This entails a learning rate of 0.0005. These experiments ensure that the data in obtained in experiment 1 truly captures the accuracies which should be exhibited from the regular and scratch ResNet variants. It is also carried in order to verify whether the CNN models were overfitting from start. This additional experiment was also run five times, where the averaged test accuracy and validation plot accuracies of each variant is displayed in the tables below.

C.1 Kitti: Small

Looking at the graphs on figures C.1 and C.2, visible asymptotic, jumpstart and learning speed improvements are observable when comparing the regular CNN to its scratch variant. It is noted that the regular ResNet still converges at a relatively fast rate of less than three epochs. Furthermore, comparing Table 5.1 to Table C.1 demonstrates that in spite of the results in Table C.1 being relatively lower, they remain nonetheless similar to those obtained in experiment 1.

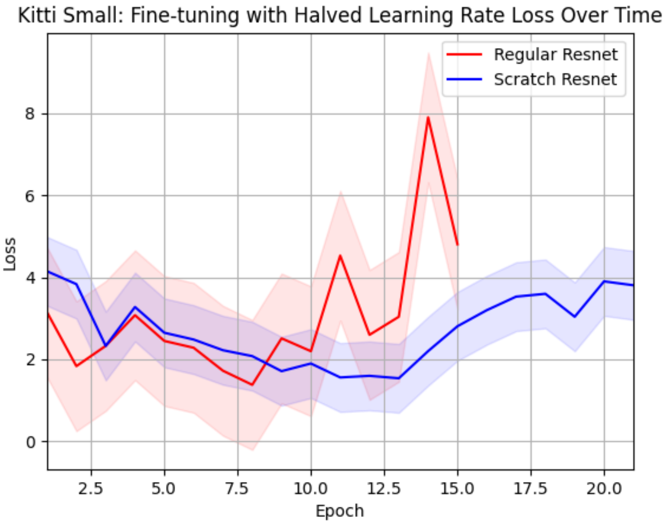

Figure C.1: Validation loss for models fine-tuned on the small variant of the Kitti dataset

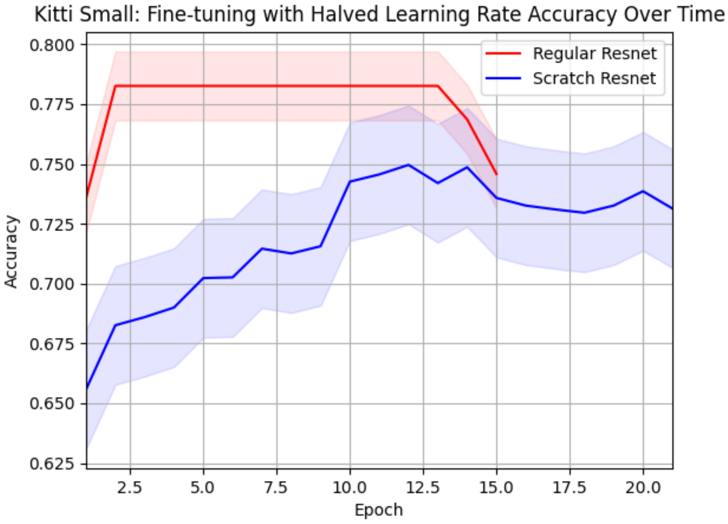

Figure C.2: Validation accuracy for models fine-tuned on the small variant of the Kitti dataset

Table C.1: Average Test Performances (in percentages) on Kitti Small

Model	CNN
Regular	71.91 ± 0.024
Scratch	76.52 ± 0.018

C.2 Kitti: Medium

Looking at the graphs on figures C.3 and C.4, visible asymptotic and jumpstart improvements are observable when comparing the regular CNN to its scratch variant. It is noted that both scratch and regular ResNets still converge at a relatively fast rate of less than four epochs. Furthermore, comparing Table 5.2 to Table C.2 conveys that the results for the regular CNN are marginally higher, whereas those for the scratch CNN are marginally lower. Nonetheless, these averages remain similar to those obtained in experiment 1.

Table C.2: Average Test Performances (in percentages) on Kitti Medium

Model	CNN
Regular	75.80 ± 0.034
Scratch	78.12 ± 0.029

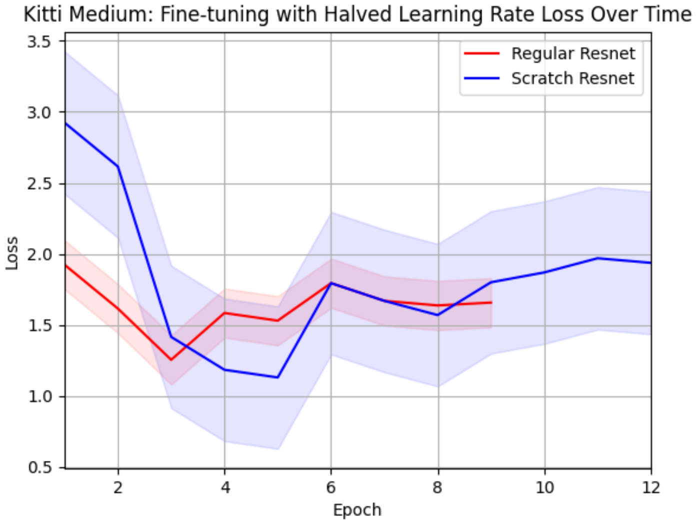

Figure C.3: Validation accuracy for models fine-tuned on the medium variant of the Kitti dataset

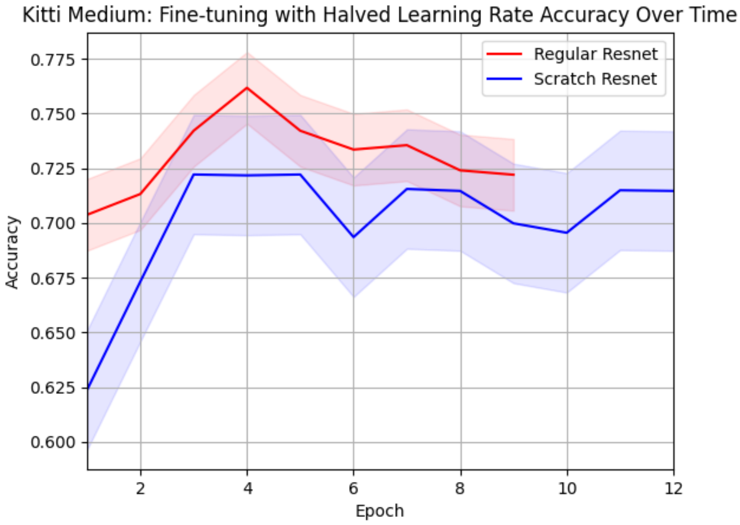

Figure C.4: Validation loss for models fine-tuned on the medium variant of the Kitti dataset

C.3 Kitti: Large

Looking at the graphs on figures C.5 and C.6, visible asymptotic and jumpstart improvements are observable when comparing the regular CNN to its scratch variant. It is noted that both scratch and regular ResNets converge at a rate of less than seven epochs. Furthermore, comparing Table 5.3 to Table C.3, it is noted that the results for the regular CNN are lower, whereas those for the scratch CNN are also marginally greater. Once again, these results remain similar to those obtained in experiment 1.

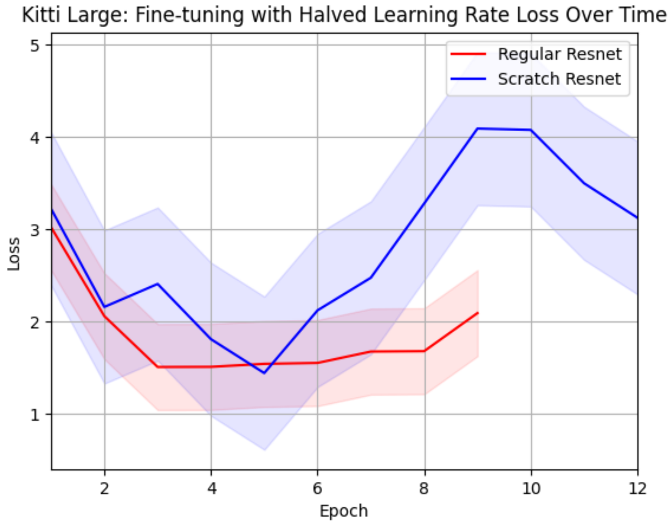

Figure C.5: Validation loss for models fine-tuned on the large variant of the Kitti dataset

Table C.3: Average Test Performances (in percentages) on Kitti Large

Model	CNN
Regular	81.19 ± 0.039
Scratch	85.92 ± 0.023

C.4 Kitti: Overview

All in all, it is made clear that the accuracies and behaviour observed in experiment 1 are not in fact suspicious. This is explained by the idea that in experiment 1, the learning rate is at a magnitude sufficiently high that the CNN converges in less than four epochs and then quickly begins to overfit. However, as noted on each graph analysis, most validation accuracy plots convey the idea that both regular and scratch ResNets tend to converge in a

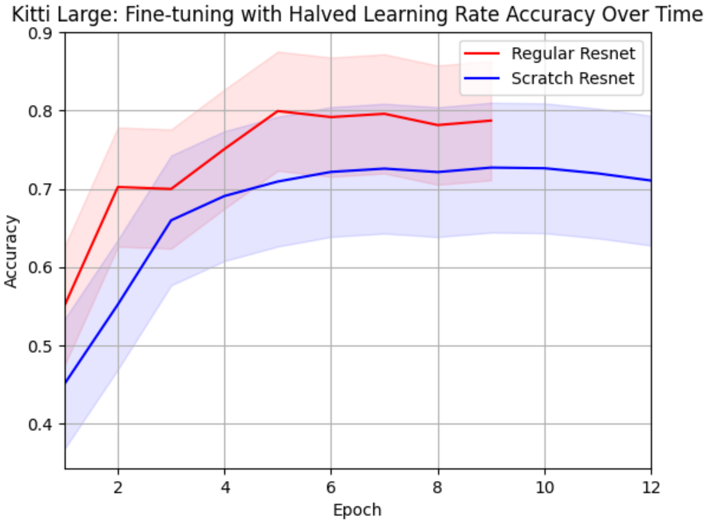

Figure C.6: Validation accuracy for models fine-tuned on the large variant of the Kitti dataset

relatively low amount of training iterations on the Kitti dataset, regardless of the utilized size variant. Hence, this alternative experiment demonstrates that this is not a model-related issue and that the models are not overfitting from the beginning of their training. Consequently, it is unlikely that this would affect or contradict either end-results or further analyses which takes the plots in experiment 1 into account. Additionally, this is further reinforced by the fact that average test performances do not deviate from those observed in experiment 1.

D Validation Loss Plots: Initial Experiments

Below are the initial validation losses plotted for all model variants on a single run of each experiment.

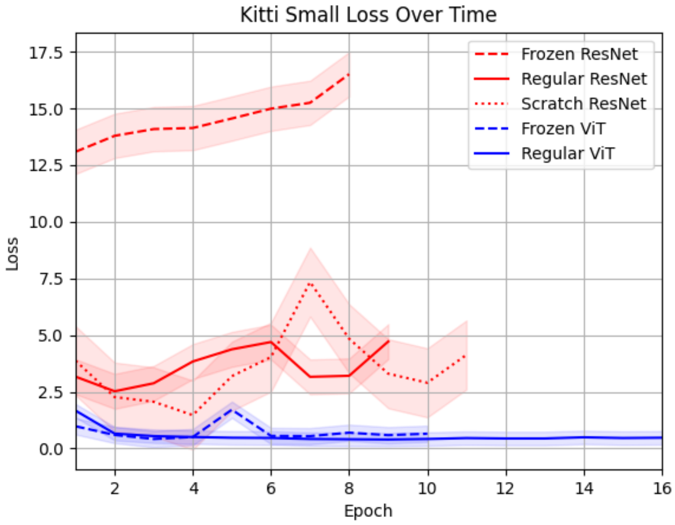

Figure D.1: Validation loss for models fine-tuned on the small variant of the Kitti dataset

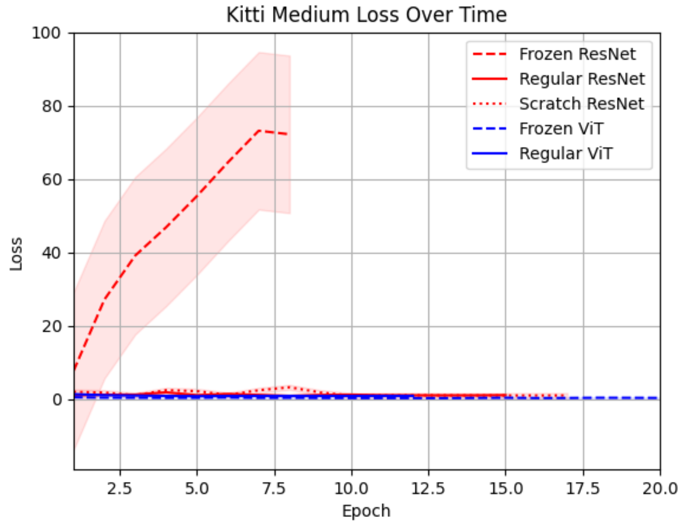

Figure D.2: Validation loss for models fine-tuned on the medium variant of the Kitti dataset

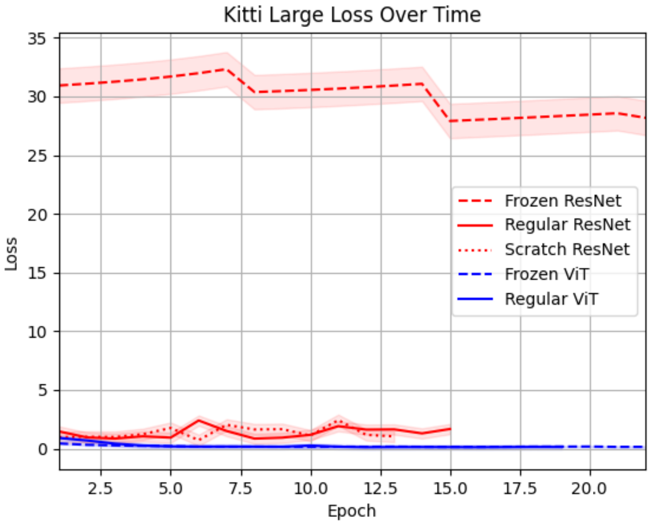

Figure D.3: Validation loss for models fine-tuned on the large variant of the Kitti dataset

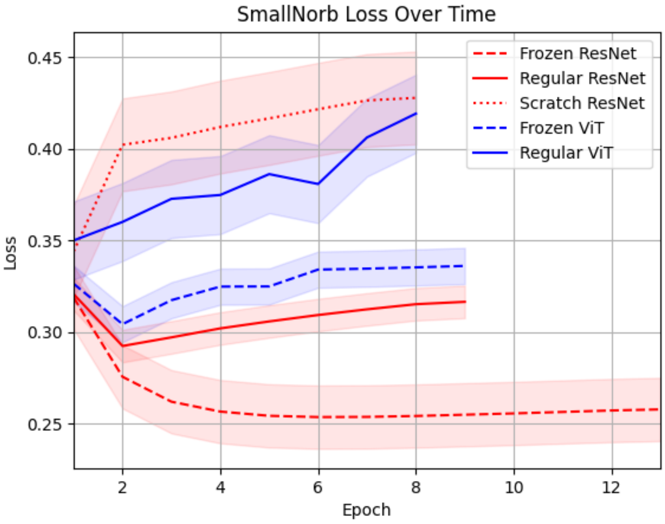

Figure D.5: Validation loss for models fine-tuned on the SmallNorb dataset

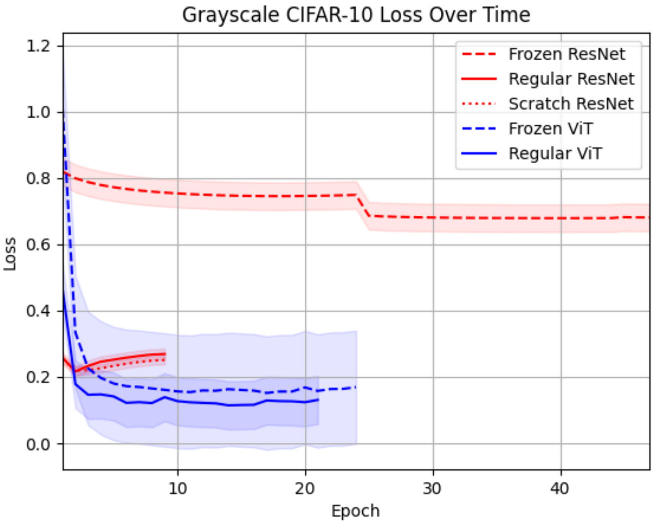

Figure D.4: Validation loss for models fine-tuned on the CIFAR-10 dataset