Investigations of Vision Transformer and its Recent Improvements(VIST)

Jiawei Lu
jl5999

Zihui Ouyang
zo2151

Jianfei Pan
jp4201

***Abstract*—Convolutional neural network (CNN) architectures have thrived and dominated the field of computer vision in the past decade, due to many of its advantages in dealing with image data such as translation equivalence, locality, inductive bias, and so on. However, in recent years, pure transformer architectures such as the Vision Transformer [1], adapted from the field of natural language processing (NLP), have demonstrated excellent ability and results in tackling image processing tasks, which does not rely on CNNs at all. In this report, the team will dive deep into understanding the recent year advances and improvements in the application of pure transformer architectures in the field of computer vision. Specifically, we will briefly introduce the Vision Transformer (ViT) [1] and discuss its problems as well as its two improved successors, the Data efficient image Transformer (DeiT) [6] and Swin Transformer (SwinT)[4]. We will analyze their performances in different computer vision tasks and back up our analysis with our own experiment data and that from the original papers. Lastly, we will talk about the difficulties we faced in trying to reproduce paper results and have fair comparisons, along with our innovations and insights gained from this project.**

I. Introduction and Description of Problem

CNN has dominated the field of computer vision for about a decade since the initial success in image classification by AlexNet [3] in 2012. There are reasons to the success of CNNs in computer vision, ranging from available large dataset, much faster computing power, and more importantly the inherent advantages of CNNs in processing image data. Unlike language data that are related by their word embedding, image data exits spatial invariance at small scales, because there often exist similar objects of different scales in different places in the image. Thus, the limited field of view and shared weights provided by CNNs can help identify similar local features efficiently.

Motivated by the success of Transformer in NLP, people have tried to combine the self-attention layers from Transformer architecture with CNNs for image processing tasks. For example, Parmer et al. attempted to apply self-attention layers in local neighbourhood of each pixel to replace convolutions. [5] However, similar to other various innovations on combining or replacing CNNs with attention operations, they require very complex implementation procedures to work efficiently on hardware accelerators.[1] Therefore, in an attempt to address those issues from previous attempts and in the hope to utilize the full potential of Transformer-like architecture in computer vision, Vision Transformer (ViT) was introduced by Dosovitskiy et al. in 2021, as a direct application of standard Transformer backbone from NLP to images with the fewest possible modifications.[1]

ViT has achieved several impressive results. Large model achieves 88.62% average accuracy when it is training on ImageNet-21k, and achieves 90.54% accuracy when it is training on JFT-300M dataset, which is comparable to ResNet152's 90.54% baseline accuracy but requires significantly less time to train.[1] However, ViT paper demonstrates that in order for vision transformer to match or exceed the performance of the state-of-art CNN models, it has to be trained on a massive image dataset such as JFT-300M, privately owned by Google. This is because in contrast to the advantages of convolution operations on images, self-attention mechanism is global and does not inherently encode locality and thus needs to learn the benefits that CNNs have all from the training data. Thus, despite the promise for ViT to challenge the long standing status of CNNs in computer vision, it is practically not feasible to be put in public use, due to its extremely data inefficient.

In order to reduce its reliance on huge dataset, one successful improvement to ViT is the DeiT [6], which allows ViT architecture to train on standard ImageNet and still achieves state-of-art results. DeiT uses the same architecture as the ViT, but improved the training method by introducing knowledge distillation. Specifically, the ViT acts as a student that learns from a pre-trained powerful CNN teacher via hard-label distillation. To do so, a distillation token is added to the output of the transformer, and the new objective function to optimize becomes the cross-entropy loss between class token and class label, as well as between the distillation token and the prediction by the teacher. The author has shown that the use of hard-label distillation greatly boosts the ViT's training speed and performance when training on medium size dataset. Thus, DeiT reduces the model's data hungriness and makes it possible for users to effectively and efficiently train ViT without the need for huge and private datasets.

In addition to the dataset requirement, the original ViT also suffers from intractable computations for high resolution images because the self-attention in ViT is global and thus quadratic to image size. This makes ViT unsuitable as a general-purpose backbone for computer vision, especially for segmentation tasks which requires pixel level predictions. The Swin Transformer provides a solution to this problem, as it introduces the shifted window multi-head self-attention layers and patch merging layers to create hierarchical feature maps for dense predictions[4]. In addition, Swin Transformer com-

putes self-attentions only locally and within non-overlapping partitioned windows. With these features added, the Swin Transformer beats the state-of-art CNN models in many computer vision benchmarks and can serve as a general backbone for computer vision tasks.

In summary, our project will focus on these three models, ViT, and two of its famous improved successors, DeiT and Swin Transformer. Our project's goals are to firstly reproduce some results from the ViT paper on ImageNet-1k with the pre-trained ViT model and compare it with results that we reproduced with DeiT and Swin Transformer. Doing so is to learn how effective the improved models are compared to ViT. Secondly, in order to learn how transferable these transformer models are to different vision tasks, we will analyze and compare their performances on fine-tuning the pre-trained models on smaller datasets. Thirdly, since the original paper of DeiT suggests a possible way for a common user to train a transformer and achieve promising results, the team will pre-train a DeiT transformer on ImageNet-1k from scratch. Lastly, we will show some innovations and fine-tune the ViT on a subset of data from ImageNet but are labeled incorrectly, from which we hope to gain insights into how robust and transferable ViT is on a small dataset with substantial noise and inaccurate labeling, and compare to some baseline CNN models.

II. Literature Review and Overview of Models

In this section, we are going to talk about the three models, ViT, DeiT and SwinT in details. We will illustrate and explain their model architectures with their key contributions and unique innovations concisely but in necessary details. Understanding the models provides us insights in to the advantages and disadvantages of the models and help us interpret our subsequent results more meaningfully in the subsequent section.

A. Vision Transformer (ViT)

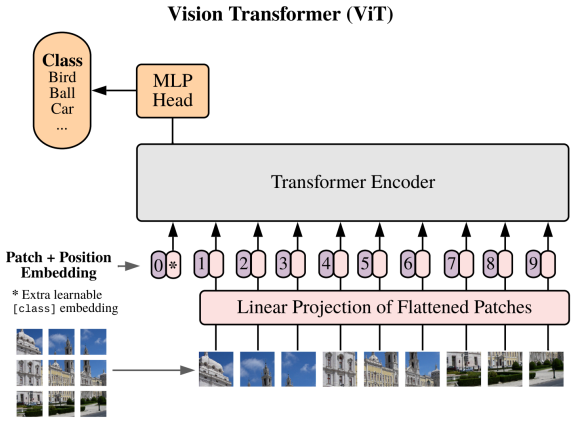

Fig. 1. Architecture of Vision Transformer (ViT) [1]

ViT is one of the earliest attempts to apply pure Transformer architectures to solve computer vision task and has shown ViT can outperform state-of-art CNN models when trained on JFT-300M, a private data by Google of over 300 million labeled images. [1] Figure 1 overviews the ViT model and shows that for a forward pass, the input image is first split into fixed size image patches that are then linearly transformed into patch embedding. Then, positional encoding is added to the patch embedding since we know transformer does not encode positional information by construct. Then, resulting embedding vectors are fed through stacked standard Transformer Encoders. Multi-layer perceptron head is attached at the output class token to perform classification.

B. DeiT: Data-efficient Image Transformers (DeiT)

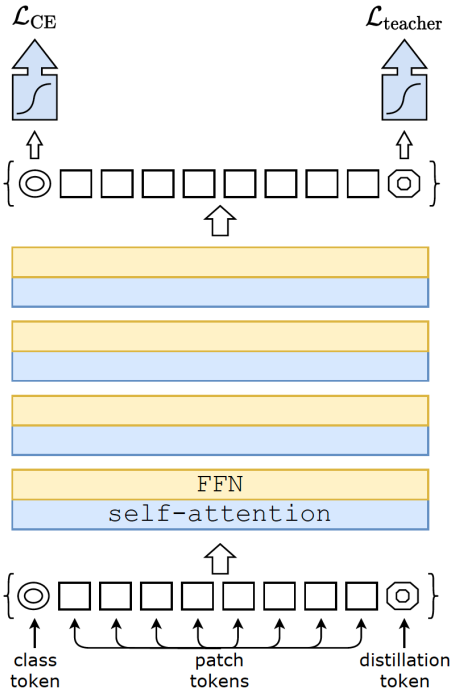

Fig. 2. Architecture of Data Efficient Image Transformer (DeiT) [6]

DeiT greatly improves the data efficiency of ViT by introducing a knowledge distillation process, making it possible for people without huge dataset able to train a ViT-like model. The idea of knowledge distillation is to use a pre-trained teacher network, usually state-of-art CNNs, to supervise the training of DeiT. DeiT shares the same architecture as the ViT except that it has an additional distillation token and is trained with the additional teacher loss. Figure 2 shows the addition of distillation token and output. The output of the teacher token will then be transformed into class probabilities and compared with the teacher model. The author experimented with soft distillation by computing Kullback-Leibler loss (KL) between the softmax of the distillation token and that of the teach model, and hard-label distillation by computing the cross-entropy loss of only the predicted class. The author found

that hard-label distillation works better and thus we will be using this in our experiments. The paper has shown that with a proper teacher model, we are able to train a DeiT on a small dataset such as CIFAR-10. Or we can train on a medium dataset such as ImageNet-1k, and also achieve better performance than ViT. [6]

C. Swin Transformer (SwinT)

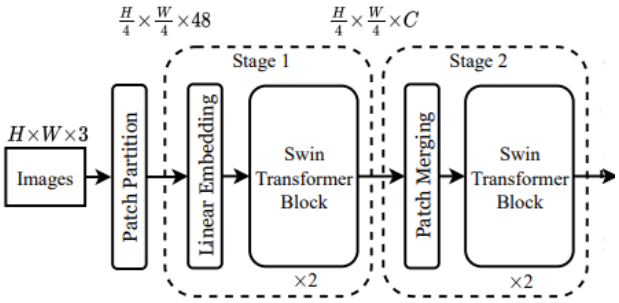

Fig. 3. Architecture of Swin Transformer (SwinT) [4]

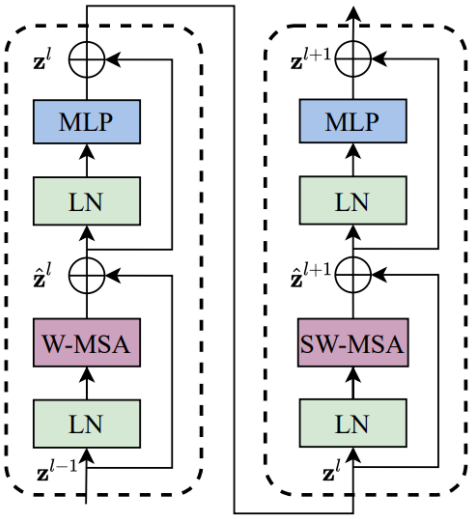

Fig. 4. Two successive Swin Transformer Blocks [4]

SwinT architecture is shown in Figure 3. Just like ViT, SwinT firstly splits images into non-overlapping patches that then pass through linear transformation to acquire the linear embedding. Unlike the positional encoding in ViT, SwinT learns the connections between patches via the shifted window mechanism and adding relative position bias in computing the self-attention in the Swin Transformer block. Thus, the linear embedding is then passed into stacked Swin Transformer Block to get C dimensional embedding vectors. The output patches will be merged to double the dimension of the embedding vectors and reducing the number of tokens by four folds (two in height and two in width), which then enters another series of Swin Transformer Blocks. This hierarchical structure allows SwinT to learn dense information in the images and produce feature maps of different resolutions, which can be used as the general backbones for other computer vision tasks. Figure 4 illustrates the architectures of two consecutive Swin Transformer Blocks. The first block is the regular Transformer block used in ViT but window based, whereas the second block contains the shifted windowing multi-head self-attention layer. In both blocks, attentions are calculated only in the non-overlapping windows, which greatly improves the computation efficient from quadratic to linear with respect to input image dimension. Additionally, the shifted window mechanism in the second transformer block allows cross-window connections by partitioning inputs in two alternative configurations, so that self-attentions can learn connections between neighbouring patches that were previously non-overlapping. SwinT proves to outperform ViT and DeiT in almost all image recognition datasets with significantly higher inference speed, while also serves as a general purpose backbone for other computer vision tasks because its ability to model at various scales with linear computational complexity with respect to the input image size.[4]

TABLE I
SUMMARY OF DATASETS USED FOR OUR DIFFERENT TASKS

Datasets	Train size	Test size	No. of classes
ImageNet-1k	1,281,167	50,000	1000
CIFAR-10	50,000	10,000	10
CIFAR-100	50,000	10,000	100

III. Implementation Details

A. Data Preparation and Processing

Table I summarizes all datasets that are used in the project. We evaluated DeiT and SwinT on ImageNet-1k, and compared their performance with those from their original papers. We fine-tuned DeiT, SwinT and ViT on CIFAR-10 and CIFAR-100 to compare their transfer-ability to a smaller and simpler dataset. Moreover, we pre-train DeiT-S and DeiT-Ti on CIFAR-100 from scratch to investigate the conclusion from the original paper that suggested we could train the networks with reasonable performance without massive dataset. Besides, we train DeiT-S on ImageNet-1k from scratch, transfer it to CIFAR-10 and CIFAR-100, and compare the results to the performance of official pretrained model. In addition to the datasets in Table I, we also used two customized datasets, imagenette and imagewoof [2], to test the robustness of ViT on transferring to small and noisy dataset. Imagenette and imagewoof are subsets of the ImageNet dataset, each with only ten classes and a portion of wrong labels. All data used in this project are directly downloaded from the internet.

Before we feed the data into training the neural nets, we followed the papers and firstly pass them through a variety of different transformations to augment our data. For example, repeated augmentation is used to boost the performance and is showed to be a key component of the original DeiT training procedure. [6]

TABLE II
MODEL SUMMARIES FOR DEIT AND VIT

Model	No. of Params
ViT-B	86M
DeiT-Ti	6M
DeiT-S	22M
DeiT-B	87M

B. Training Details and Hyperparamter Settings

The team has worked on many models from three different papers in this project. Thus, in order to try to reproduce results from the original papers, we stick to the hyperparameter settings in the original papers. Therefore, we have documented all training details and hyperparameter settings in our GitHub repository before experiment results. Therefore, we will not talk about experiment settings and training details in the project report since they vary from one model to another.

IV. RESULTS AND DISCUSSION

Since we have gathered a lot of results in this project as documented in the results section. Discussing our results on every single table and plot would be very verbose. Thus, in this section, we decided to concentrate our analysis selectively on the most inspirational results.

A. Fine tuning DeiT Transformer on CIFAR-10 and CIFAR-100

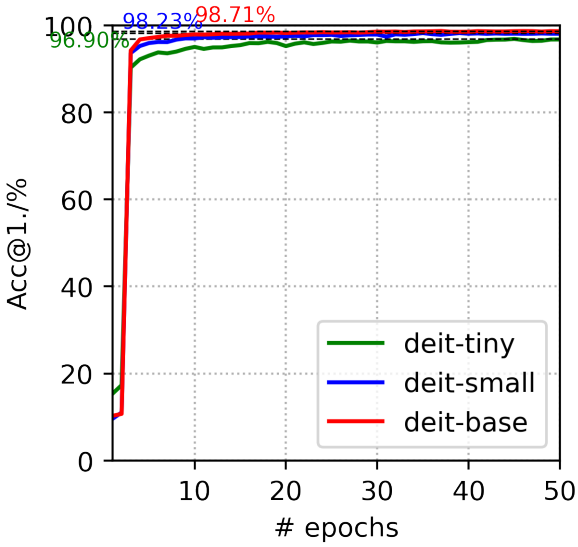

Fig. 5. Fine-tuning pretrained DeiT Transformer on CIFAR-10, Top 1 Accuracy

Figure 5 shows the top 1 accuracy for transferring pretrained DeiT models of different sizes on CIFAR-10 dataset and Figure 6 show the results for transferring them on a more challenging CIFAR-100 dataset. Table II summarizes all three DeiT models that we evaluated. Since CIFAR-10 dataset only contains 10 classes and thus is relatively easier to classify than

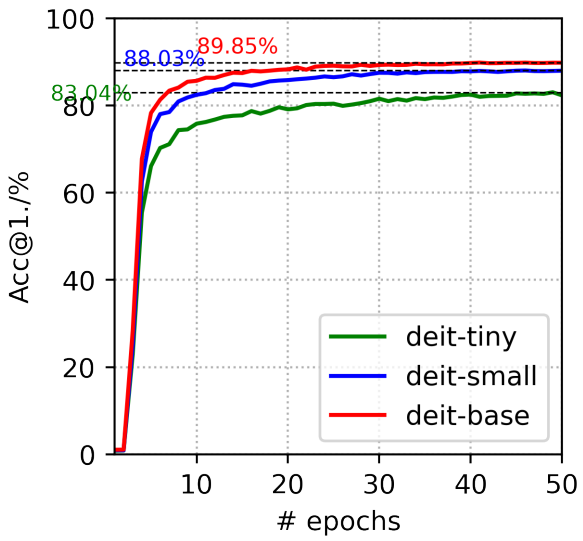

Fig. 6. Fine-tuning pretrained DeiT Transformer on CIFAR-100, Top 1 Accuracy

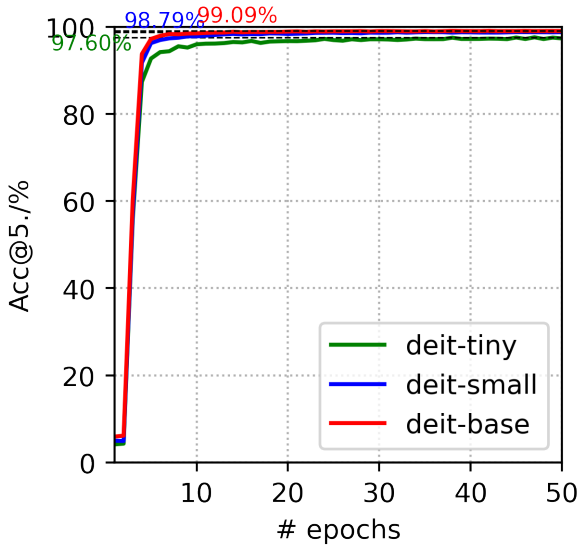

Fig. 7. Fine-tuning pretrained DeiT Transformer on CIFAR-100, Top 5 Accuracy

CIFAR-100, we can see that all three DeiT models achieve very high accuracy in CIFAR-10. However, the difference in performance differs a lot in CIFAR-100, as DeiT-B yields more one percent and six percent higher accuracy than the DeiT-Ti and Deit-S at convergence. Moreover, the larger the model, the faster the convergence speed is. However, if we look at the top 5 accuracy comparison in Figure 7, the difference in performance in the three models becomes less significant and all models have a very high percentage in containing the answers in their top 5 choices. This results shows even though top 1 accuracy is not amazing and differs significantly among different models, DeiT in general can quickly learn important

features and locate answers in its top choices when transfer learning on a small dataset.

B. Fine tuning Swin Transformer on CIFAR-10 and CIFAR-100

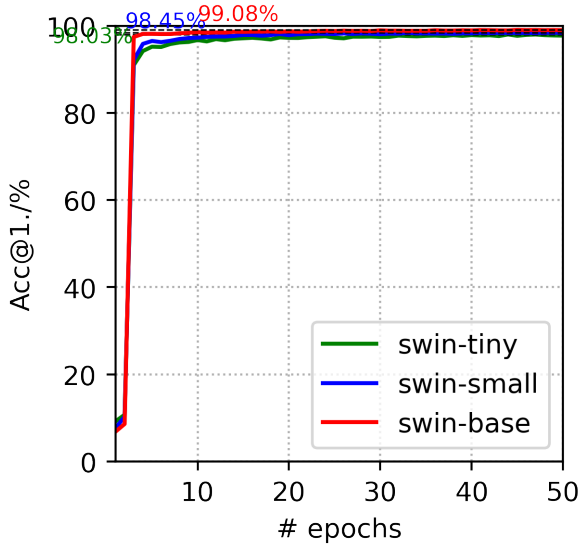

Fig. 8. Fine-tuning pretrained Swin Transformer on CIFAR-10, Top 1 Accuracy

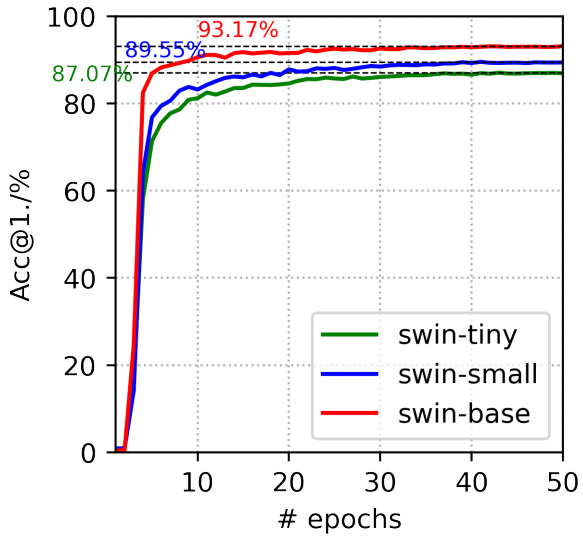

Fig. 9. Fine-tuning pretrained Swin Transformer on CIFAR-100, Top 1 Accuracy

Figure 8 shows the top 1 accuracy for transferring pretrained SwinT models of different sizes on CIFAR-10 dataset outperforms their DeiT counterparts. However, Table III shows that SwinT models generally have a lot more parameters than their DeiT counterparts, which indicates higher capacity for SwinT and thus a reasonable better performance. Figure 9 show the top 1 accuracy results for transferring them on

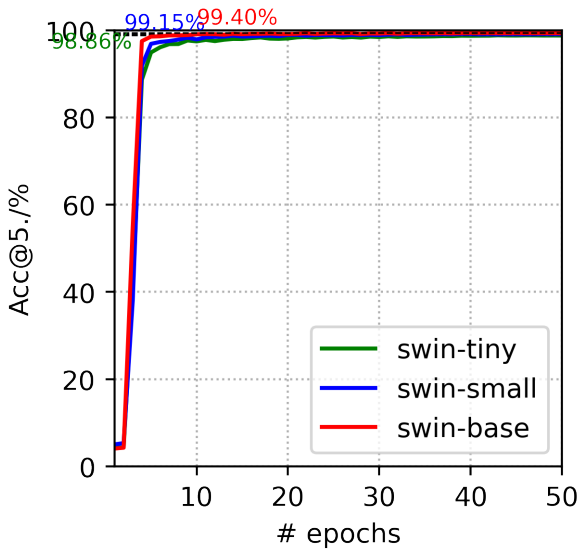

Fig. 10. Fine-tuning pretrained Swin Transformer on CIFAR-100, Top 5 Accuracy

TABLE III
MODEL SUMMARIES FOR SWINT

Model	No. of Params
SwinT-T	29M
SwinT-S	50M
SwinT-B	88M

a more challenging CIFAR-100 dataset, which significantly outperforms their DeiT counterparts by over four percent for their base models. The top 5 accuracy from Figure 10 also demonstrates SwinT outperforms DeiT in every of its counterparts. The results give us the conclusion that is similar to the original paper that SwinT is a more powerful model and better suited for a general purpose backbone for computer vision tasks. However, despite the higher transfer learning performance of SwinT, we learned that SwinT is too large to train from scratch in reality without rich computing power, whereas DeiT is a more realistic choice if we were to train a transformer model for image classification from scratch on a small customized dataset.

C. Comparison of Models Fine tuning on CIFAR-10

Figure 11 summarizes the transfer learning results, from which we fine tune the pretrained models on the smaller CIFAR-10 dataset. Figure 11 shows a general trade-off between accuracy and speed, as higher accuracy usually means slower speed, which is reasonable since we know more complex models having more parameters often have higher model capacity but lower speed. Additionally, this plot also compares performance of different models. Since CIFAR-10 is a very simple dataset, thus the difference in accuracy is not significant with less than three percent difference between the highest accuracy model SwinT-B and the lowest accuracy

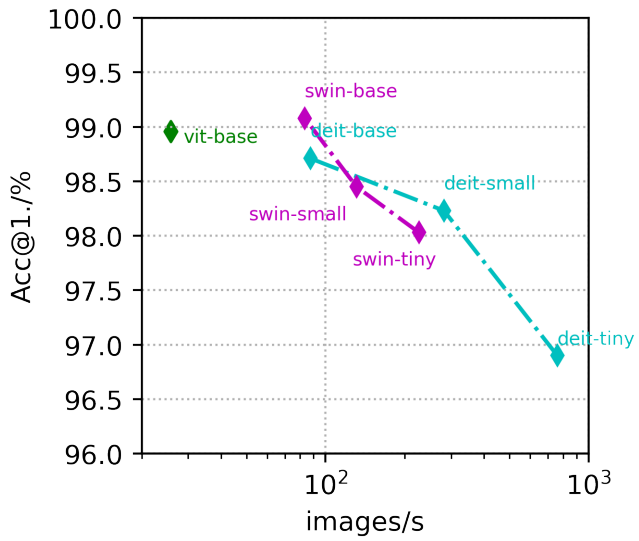

Fig. 11. Comparison of models fine-tuning on CIFAR-10, Validation Accuracy vs Inference Speed

DeiT-Ti. However, the speed difference is very significant, as DeiT-Ti is almost ten times faster than the SwinT-B. Thus, we need to balance the trade-off between accuracy and speed in our model selection. However, both DeiT and SwinT manifest their significant improvement in speed than the original ViT, and SwinT-B also shows it is both faster and more accurate than the original ViT model.

D. Comparison of Models Fine tuning on CIFAR-100

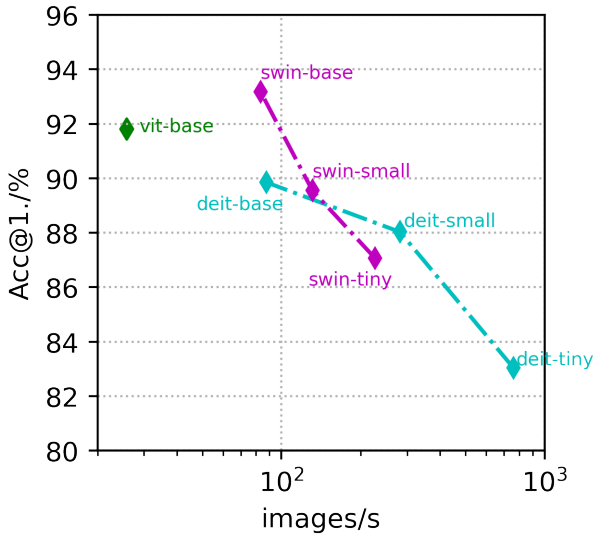

Fig. 12. Comparison of models fine-tuning on CIFAR-100, Validation Accuracy vs Inference Speed

Figure 12 summarizes the transfer learning results, from which we fine tune the pretrained models on the smaller CIFAR-100 dataset. Compared to Figure 11, transferring to CIFAR-100 is more challenging since there are more classes and as expected yield overall poorer results than on CIFAR-10. However, the overall trend still maintains, as we can see a clear trade-off between accuracy and speed, where faster speed usually means poorer accuracy. SwinT has a more significant advantage in accuracy over other models when transferring on CIFAR-100, with over ten percent different from the DeiT-Ti model despite the a ten fold slower inference speed. Overall, DeiT accuracy performance degrades the most when switching to CIFAR-100 dataset. One potential reason may be due to the distillation token and the ineffective choice of teacher network. One future investigation is to use a pre-trained model with state-of-art results on CIFAR-100 as the teacher to train DeiT. Then, the team believe DeiT performance would be improved.

E. Pre-train Deit on CIFAR-100 from scratch

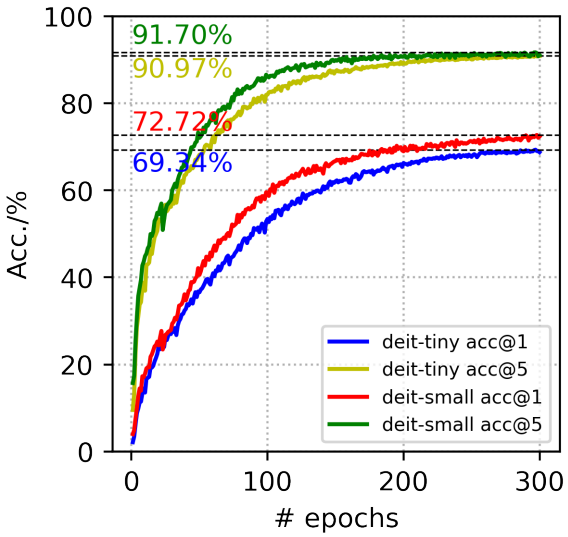

Fig. 13. Accuracy comparison of pre-training DeiT from scratch on CIFAR-100

One of the most important breakthroughs that DeiT brought was introducing the knowledge distillation method so that people without access to massive datasets can also train a vision transformer model. In our project, we attempted to pre-train DeiT-S of 22M parameters and DeiT-Ti of 6M parameters from scratch on a small dataset of CIFAR-100 on a single Tesla T4 GPU. It took 11 hours to train a tiny model DeiT-Ti for 300 epochs and took 25 hours to train a small model DeiT-S for 300 epochs. The accuracy results are shown in Figure 13 and loss results are shown in 14. These two results both show that convergence at around 300 epochs, which showcase that we can indeed pre-train DeiT from scratch on a decent size dataset with 100 classes relatively fast, which could not have been achieved by the original ViT. Moreover, Figure 13 also shows that although DeiT is shown to be relatively efficient in training, the results were not impressive. There exists a huge gap between the top 5 accuracy and top 1 accuracy by almost twenty percent. Although DeiT-S demonstrates

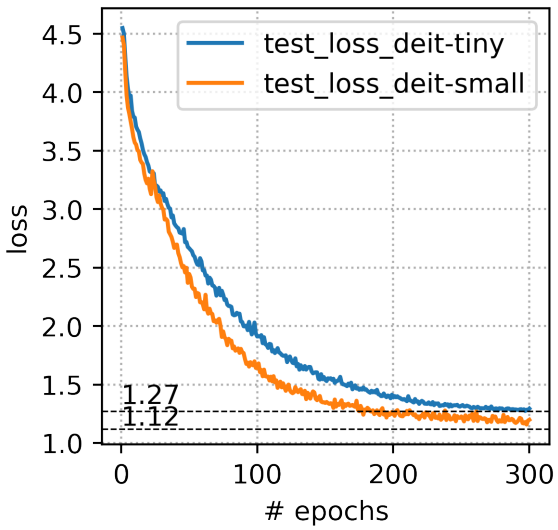

Fig. 14. Loss comparison of pre-training DeiT from scratch on CIFAR-100

slight improvement over DeiT-Ti, both models yield poor performance compared to CNN based models, which shows that DeiT still needs a lot more training data and epochs to elevate its performance. The poor performance is expected, because we know transformer based models need to learn locality and other advantages inherent in CNNs completely from training, and thus would be more demanding than CNN based models. Regardless of performance, our experiments show that it is practically possible to train a DeiT from scratch and get reasonable performance on a small dataset such as CIFAR-100 only.

F. Robustness Analysis of Vision Transformer on imagenette and imagewoof

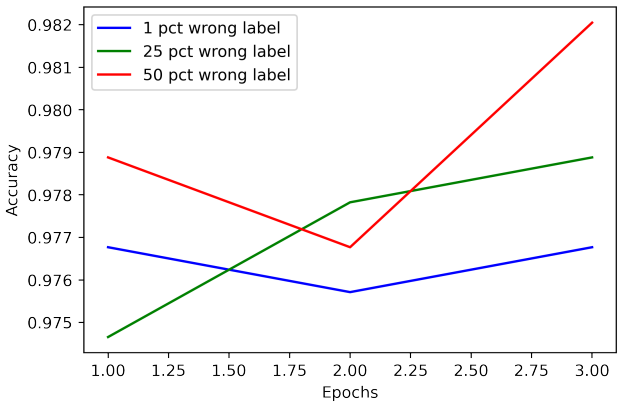

Fig. 15. Fine tuning ViT on imagenette

As discussed in the previous sections, since ViT is trained extensively on a substantial amount of images to achieve state-of-art results, we nevertheless doubts if the performance

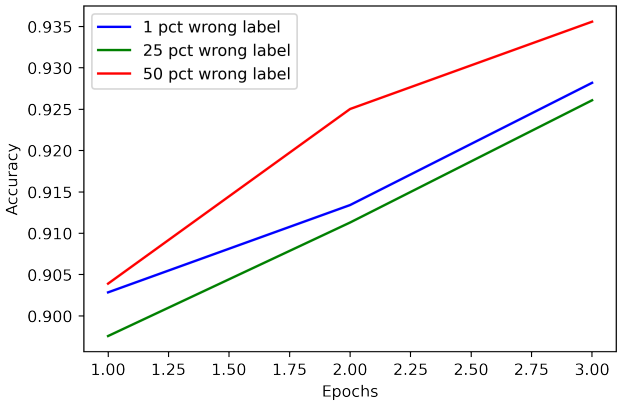

Fig. 16. Fine tuning ViT on imagewoof

is achieved by ViT memorizing all the data or generalizing well beyond the training dataset. To answer this question, the team fine tuned pre-trained ViT on two wrongly labeled datasets Imagenette and Imagewoof. The results are presented in Figure15 and Figure16. Imagenette and Imagewoofs are both subset of Imagenet, except both only contain 10 classes. Classes in Imagenette is more distinguishable because Imagewoof contains only dogs. In each figure, we changes the percentage of wrong labels from 1 percent to 50 percent and plot the fine-tuning learning curves of validation accuracy over five epochs. Both results have surprisingly show us that learning with fifty percent wrong labels outperforms ones with more correct labels. The team suspects a few reasons of this unexpected trend. Firstly, the ViT was pre-trained on ImageNet and thus may have already learned these data before. Thus, transferring the ViT to focus on learning a subset of the training data may easily cause it to overfit to the data. Therefore, adding noise in terms of wrong labels can be an effective data-augmentation technique and improved validation accuracy and the generalizability of ViT. Secondly, since the team is limited with resources, we only generated these very limited results over a small epoch, the tiny difference in performance boost such as in Figure15 may not reflect the entire picture if we would have trained for longer epochs. This is also shown in Figure16 where increasing wrong label rates from one percent to twenty five percents slightly drops the accuracy. However, both these two results show that fifty percent wrong label rates increases the validation accuracy for ViT when transfer learning on a small and simpler dataset. Thus, we do believe ViT is robust to noise and in fact needs some data augmentations such as adding wrong labels to improve generalization during fine-tuning stage.

G. Effort in optimizing results with extensive pre-training

Our previous results have shown that pre-training DeiT-Ti and DeiT-S on small datasets such as CIFAR-100 is possible, although the results were not impressive. Since then, the team has tried to optimize our results to match with the paper's

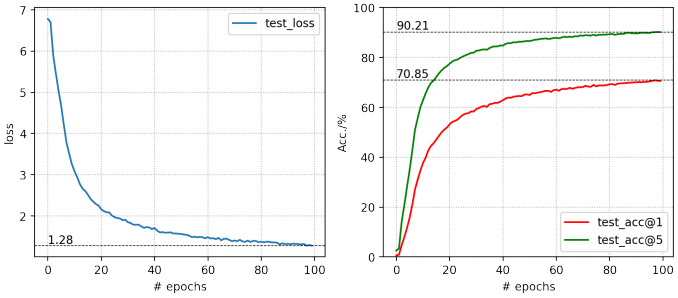

Fig. 17. Training curve for pre-training DeiT-S from scratch on ImageNet-1k

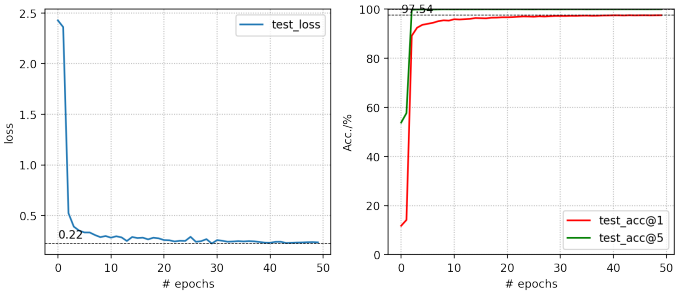

Fig. 18. Fine-tuning our pre-trained DeiT-S on CIFAR-10

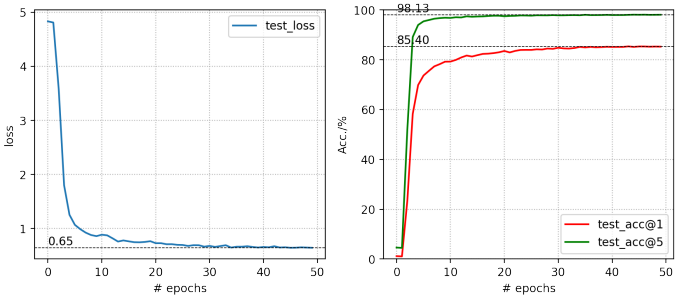

Fig. 19. Fine-tuning our pre-trained DeiT-S on CIFAR-100

results as consistently as possible. Therefore, besides testing with more model variants, we decided to pre-train a DeiT-S on ImageNet-1k in the same fashion of the original paper. Figure 17 shows that the model roughly converges at around 100 epochs, which took around ten days to train on a single Tesla T4 GPU on Google Cloud Platform. Since the original paper pretrained DeiT-S on ImageNet for 300 epochs on 8 V100 GPUs, our DeiT-S is trained with much fewer epochs due to our limited time and resource. Regardless, the sign of convergence shows the preliminary success in our attempt.

Our final pre-training accuracy on ImageNet-1k after 100 epochs only differ about 10% from the paper that trained for 300 epochs. Additionally, we evaluated our pre-trained models by transferring on CIFAR-10 and CIFAR-100. We achieved 97.54% top-1 accuracy on CIFAR-10 for 50 epochs as shown in Figure 18, and 98.13% top-1 accuracy on CIFAR-100 for 50 epochs as shown in Figure 19. On CIFAR-100, our pre-trained model achieves the 85.40% accuracy comparing to the 88.03% using the model from the paper as shown in Figure 12, which less than 4 percent difference but with only 100 epochs. On CIFAR-10, comparing Figure 11 to Figure 18, our pre-trained model achieves less than 1 percent difference from the model in the paper. The impressive transfer learning results on small datasets have demonstrated that DeiT can learn to effectively represent image features and possibly have discovered correlations among images and features within the first 100 epochs. This makes DeiT more practically valuable, because we now know that we can efficiently learn a DeiT that is good enough to work on small dataset in just one third of the training time proposed in the paper.

The results from the pre-training and transferring DeiT-S all by ourselves are extremely valuable, since it proves that, with reasonable and acceptable efforts, DeiT-S can be trained to achieve impressive results only on ImageNet-1k, which otherwise is not possible with ViT or other massive models. Additionally, the most surprising fact is it only took us 100 epochs, in contrast to 300 in the paper, to achieve competitive performance in transferring to small dataset. This demonstrates DeiT is very data efficient to train and can achieve competitive results within a small number of epochs, whose performance is practically much easier to reproduce than the vanilla ViT.

TABLE IV
PRE-TRAINING TIME SUMMARY, * MEANS PRE-TRAINING FROM SCRATCH, ** MEANS TRANSFERRING OUR PRETRAINED MODEL, ELSE MEANING STANDARD TRANSFERRING LEARNED MODELS IN THE REPOSITORY

Model	Training Time	Top-1 Acc	Top-5 Acc
DeiT-Ti on CIFAR-100	10h 55mins	69.34	90.97
DeiT-S on CIFAR-100	24h 42mins	72.72	91.70
DeiT-S* on ImageNet-1k	9d 14h 37mins	70.85	90.21
DeiT-S** on CIFAR-10	4h 24mins	97.54	100.00
DeiT-S** on CIFAR-100	4h 48mins	85.40	98.13

H. Summary of Pre-training Time

We will not talk about training times in this section, since transferring models often converge very fast at low epochs and we do not think they are very valuable to discuss in details with our results. In fact, we have documented fine-tuning times for all fine-tuned models in our GitHub repository either in Readme file or printed in our notebooks. The most important time comparison, such as the inference time comparison, is already discussed in prior sections when we compared inference speeds of different models. Thus, we will focus on talking about training times for pre-trained models, since they took the most time and effort to train. Table IV shows training times until convergence for pre-training DeiT-Ti and DeiT-S from scratch both on CIFAR-100 dataset only. All models are trained with one Tesla T4 GPU on google cloud platform. We can see that training DeiT-S took significantly longer time than DeiT-Ti but yield better accuracy. The longer training time is as expected since DeiT-S has a few times more parameters than DeiT-Ti. Although the original paper only pre-trained DeiT-B on CIFAR-10 instead, the training results can be comparable to ours. The paper documents it

takes 53 hours using 8 V100 GPUs to pre-train DeiT-B for 300 epochs, approximately 424 hours for one V100 GPU. [6] Thus, considering the difference in sizes of DeiT variant and that our dataset is more complex, we think the training time is reasonable short for learning curves to converge. Thus, these training time results demonstrate the conclusion in DeiT paper that DeiT is very data efficient compared to ViT, because we did generate reasonable results by training DeiT from scratch on a small dataset and quite quickly for 300 epochs with a single Tesla T4 GPU. Lastly, since the team wants to optimize our results as much as possible, we decided to pre-train one DeiT-S model from scratch for longer time. Table IV shows it took about ten days to pre-train a DeiT-S from scratch on ImageNet-1k, and a few hours to transfer it to CIFAR-100. As discussed in the previous section, the results have shown to be very close to directly transferring the pre-trained DeiT-S by the author.

V. Discussion and Insights

A. Comparison with other Papers

We compare our results with the data in the original three papers. Specifically, we tried to compare the training time and training accuracy, as well as some conclusions. In our project, we attempted to pre-train DeiT-S of 22M parameters and DeiT-Ti of 5M parameters from scratch on a small dataset of CIFAR-100 on a single Tesla T4 GPU. DeiT-Ti takes 2 hours for training and 1 hour and fifty minutes for validation in one epoch. DeiT-B is much more demanding, as it takes about five hours to train one epoch. The original paper showed that it typically took 53 hours to train a DeiT-B of 87M parameters on 8 V100 GPUs on CIFAR-10 dataset for equivalent 300 ImageNet epochs. Due to our budge limit, we could not afford training with V100 GPU for days, and thus we eventually decided to work on a single Tesla T4 GPU to train all our models. [6] Moreover, observe that DeiT starts to converge at around 300 epochs and only improve slightly afterwards. Thus, we decided to terminate training at 300 epochs instead of 1000 epochs in the original paper to save some time, since we also needed to gather other results. Although the original paper only pre-trained DeiT-B on CIFAR-10 instead, the training results can be comparable to ours. The paper documents it takes 53 hours using 8 V100 GPUs to pre-train DeiT-B for 300 epochs, approximately 424 hours for one V100 GPU. [6] Thus, considering the difference in sizes of DeiT variant and that our dataset is more complex, we think the training time is reasonable short for learning curves to converge. All in all, although we can not replicate the exact training setting in the paper with the resources at hand, we still showed that we are able to train DeiT from scratch on a small dataset and acquired reasonable performance.

Table V summarizes our results on direct evaluation of different models on ImageNet-1k and it shows that our experiment results are very consistent with the accuracy results from the papers. We are able to achieve the same fine-tuning results of ViT on CIFAR-10 and CIFAR-100 datasets as the paper for only three epochs. Moreover, Table V shows Swin Transformers outperforms than DeiT and ViT on ImageNet-1k evaluation, which the original paper uses to reach the same conclusion.

TABLE V
Accuracy Compared with Papers

Model	No. of Params	Dataset	Experiment	Paper
ViT	86M	CIFAR-100	91.81	91.67
ViT	86M	CIFAR-10	98.96	98.95
DeiT-Ti	6M	ImageNet-1k	72.1	72.2
DeiT-S	22M	ImageNet-1k	79.8	79.8
DeiT-B	87M	ImageNet-1k	81.8	81.8
SwinT-T	29M	ImageNet-1k	81.2	81.3
SwinT-S	50M	ImageNet-1k	83.2	83.0
SwinT-B	88M	ImageNet-1k	83.5	83.5
SwinT-B-384	88M	ImageNet-1k	84.5	84.5

Moreover, Figure 12 and Figure 11 both show the comparison of accuracy and inference speed for SwinT and DeiT. In the original paper of SwinT, the author stated that SwinT-T has higher accuracy and faster inference speed than DeiT-S due to the fact that the time complexity for SwinT is linear but quadratic for DeiT. However, our results do not completely support this statement. For example, if we compare SwinT-T and DeiT-S that have a similar number of parameters, we can see that SwinT-T does not have higher accuracy than DeiT-S, and the DeiT-S has much faster inference speed than SwinT-S.[4] This significant contradiction could be due to several factors. Our comparison is on CIFAR-10 and CIFAR-100 dataset, whereas the author performed his experiment on COCO dataset. Additionally, the author used DeiT and SwinT as backbones to concatenate them with other frameworks to perform classification, whereas we directly fine-tuned and evaluated the models on our dataset. Lastly, our results are evaluated on Tesla T4 GPU, which is different from the V100 GPU used by the author. Therefore, we suspect the difference could be a result of several factors, but our results show that a general statement about a model's performance may not hold true under certain circumstance.

B. Discussion of faced Problems

There are several problems that the team faced during our project. We faced with extremely long training time, limited data, problems of executing existing codes from online GitHub repository. The total training and fine-tuning time is further exacerbated by the fact that we need to preform the controlled analysis on all three models, each of those has several versions. All major problems that the team encountered will be discussed in details in the following paragraphs.

Training time and resource have always been one of the biggest problem for the team. Initially the team planned to build a ViT from scratch and train it on ImageNet and compare results with those from the original paper. However, the team realize training time is so long and training resource is so demanding that we could not produce meaningful results in a few weeks and we would have run out of money on GCP very quickly. Even after we decided to switch to more manageable

model such as DeiT to pre-train by ourselves, the original paper uses several GPUs to distributively train the models for a few days. Thus, with all of these time and resource limitations, we have to forfeit some ambitious reproductions of certain parts from the three papers. For example, we terminate training at 300 epochs instead of 1000 epochs in the original paper, when we felt we gathered enough information for analysis.

Secondly, transformer models for vision need a substantial amount of data to perform well. In the original ViT paper, the author trained the ViT on JFT-300M dataset that comprises of 300 million label data private to Google to get a performance that was truly state-of-art. The amount of data plays a crucial rule in generating promising results from ViTs, but we could not get access to that huge dataset nor could we have the resource to train with that many images. Even though we decided to use a more data efficient model DeiT, its original paper also used a variety of different image transformations and data augmentation techniques, from which we could only select a few. Although we eventually were able to train the DeiT on ImageNet-1k, the results were not as good as that presented in the paper due to limited data size and computation power.

Thirdly, we also encountered problems in running codes from online repositories. For example, one team member had to spent days configuring the shell environment in order to fine tune the ViT on Imagenette and Imagewoof that are acquired from online GitHub repository. Thus, there is a learning curve to use online materials but luckily we were able to figure out solutions and acquire results.

C. Discussion of Insights

In this project, we have worked on three different models, ViT, DeiT, and SwinT, of which we have trained and fine-tuned many variants. The first insight we gained is the level of difficulty to train a vision transformer from scratch and gain good results. We learned the amount of training data and computation power required to train a transformer even on a small dataset, with which a CNN model can learn much more efficiently. Even though DeiT provides us the possibility to train a vision transformer model on a small dataset, the training process still took very long and the results were not impressive after 300 epochs. We learned that it is generally not possible to train a giant neural network from scratch due to limited computation resource, which is even less likely for training ViT that requires gigantic dataset. Thus, fine-tuning an existing model is the best bet to test if the model is suitable for a certain task in practice. Our fine-tuned models can achieve impressive results on a small dataset with much shorter training time. Additionally, from imagenette and imagewoof, we learned that adding data augmentation significantly improves the generalizability of ViT for fine-tuning on small dataset.

Our second insight is that it is extremely difficult to reproduce results from papers except directly evaluating the pre-trained models or fine-tuning the models on datasets suggested in the papers. As discussed in the previous paragraph, it is almost impossible to pre-train a millions parameters model from scratch without access to unlimited hardware accelerators. In addition to that, the statements or conclusions in the paper may not always hold. For example, the SwinT paper shows that SwinT-Ti outperforms DeiT-S in accuracy and inference speed for transfer learning on COCO dataset. However, when we directly fine tune the models on CIFAR-10 and CIFAR-100 dataset, the results show the exact opposite conclusion can be made. All of these variations may be a result of different problem settings, hardware accelerators, and datasets. Thus, we need to try different models for the problem at hand, because the trends or conclusions in the papers may not hold true for all cases.

Lastly, the biggest insight we gained from this project is the balance between speed and accuracy for practical applications. Based on our results, more sophisticated models usually can achieve higher accuracy but at the expense of inference speed. Thus, we can imagine we will have to balance the importance of accuracy and speed, and choose the model variant that best suits the problem. we need to consider all model variants instead of a single model from an architecture, since Figure 12 has shown us even for the same architecture, models of different sizes can vary a lot in their performances. Thus, when selecting models for practical uses, it is necessary to compare and balance inference speed, model size and all other relevant factors for all model variants instead of only considering the one model with the highest inference accuracy.

VI. Conclusion

Our project studies the recent breakthroughs in applying Transformer architectures in computer vision. We started from the Vision Transformer architecture, one of the earliest successful attempt, and fine tune the pretrained the model on small dataset and compared it with other models. Moreover, we also analyzed its robustness to noise in the fine-tuning data, by training it on Imagenette and Imagewoof, datasets with wrong labels. While understanding the limitations of vanilla Vision Transformer, we then studied and implemented two famous ViT successors, DeiT and SwinT. DeiT improves the data efficiency of the ViT by introducing a teacher network during training, and SwinT introduces novel shifted window multi-head self-attention algorithms to improve computation efficiency. Our implementations of DeiT and SwinT show great performance boost from ViT, especially in training efficiency. Additionally, our implementation of DeiT shows that it is possible to train a DeiT from scratch on a small dataset while achieving reasonable accuracy. Overall, the team conclude that Transformer architectures are generally data hungry, and does not perform as well as CNNs for limited data and computing power if they are trained from scratch. However, there are increasingly more advances in improving data and computation efficiency from the original ViT such as the DeiT and SwinT. Thus, we believe Transformer based backbones are still important models to consider for solving many computer vision tasks in the future.

VII. Contributions

Team member	Percentage	Contributions
Jiawei Lu	33	Coding, GitHub, Report
Zihui Ouyang	33	Coding, Github, Report
Jianfei Pan	33	Analysis, Report, PPT

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[2] Jeremy Howard. Imagenette.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

[5] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4055–4064. PMLR, 2018.

[6] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 2021.