CS152 Neural Networks Final Project

Sophie Bekerman
Harvey Mudd College
sbekerman@hmc.edu

Mukta Ubale
Harvey Mudd College
mubale@g.hmc.edu

Hayley Walters
Harvey Mudd College
hwalters@hmc.edu

Francine Wright
Harvey Mudd College
fwright@hmc.edu

Abstract

Our project explores the capabilities of using transformers for image classification problems. We use the model for a Vision Transformer (ViT) as detailed in the paper *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* [4] and compare its performance against other ViT architectures, then examine it when the value of hyperparameters is varied. We found that Simple ViT continued to perform better than ViT, even on small datasets. With regards to the hybrid architectures, we found that the Simple-ViT model had the most success. We observed that low patch sizes and low number of heads in the attention layer stop the ViT from training. And as expected the convolutional neural network performed the best given our limited computational resources.

1 Introduction and related work

The goal of this project is to explore how transformer-based neural networks can be used for image classification problems. Previous approaches to image classification have often used the convolutional neural network structure, but a recent research team has demonstrated that a pure transformer is capable of attaining comparable results to SOTA CNNs [4]. We would like to verify these results by testing out the visual transformer on a sample dataset, and compare it against a conventional CNN. To understand this architecture better, we will also test models with varying hyperparameters, like the number of attention heads in the attention layer and the number of image patches used during the image processing step. Since the publishing of this paper, many hybrid approaches using the visual transformer show varying degrees. For example, a recent paper has shown that employing a distillation token to transfer knowledge from convolutional networks to vision transformers can result in compact and effective vision transformers [7]. We also seek to compare the performance of these hybrid models to the original simple transformer described by Dosovitskiy et al [4].

Image classification is a useful technique in many situations. For example, image classification can aid in identifying relevant features in medical scans, recognizing obstacles for a self-driving car, or tracking plant growth in aerial landscape images. Applying a transformer-based neural network instead of conventional image classification methods may yield similar results but with a much lower computational cost [4]. As computational resources are a major limiting factor for machine learning, this technique should be explored for its applications in robotics and other low-resource environments.

This topic is interesting to us because we recognize that transformers have been very useful in NLP contexts. They are often used due to their computational efficiency, scalability, and ability to learn contextual information and relationships. We are excited about the idea of using transformers for potential computer vision applications. Current approaches to image classification are also fairly defined, but Dosovitskiy et al demonstrate that their transformer architecture performs better with

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

large datasets. This opens the door to many other potential applications for transformers and hybrid architectures besides the one described here.

2 Datasets

For this project, we used the CIFAR-10 dataset. The CIFAR-10 dataset, commonly used in machine learning and computer vision research, comprises 60,000 32x32 color images categorized into 10 classes. Each class contains 6,000 images, representing objects such as airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The dataset is divided into 50,000 training images and 10,000 test images, providing a standardized foundation for assessing and training image classification algorithms. In PyTorch, it's readily available, serving as an invaluable tool for researchers developing and experimenting with image recognition models. This dataset was chosen because it is pre-loaded in PyTorch and was used by the researchers behind the original vision transformer paper. The paper notes that the model performed better on datasets with more classes, like the CIFAR-100, but because of computational restraints and for simplicity, we decided to opt for the CIFAR-10.

3 Methods

In this section, we detail the models and datasets we used, as well as the approaches we took for our various experiments.

3.1 Standard Vision Transformer (ViT)

For the regular vision transformer, we utilized an implementation created in PyTorch [8] that was based off of the original ViT structure [4]. All three variations were trained using Adam [6] with a batch size of 64 for 20 epochs. All models ran with a seed of 42 and a learning rate of 0.00003.

3.2 Datasets

Input images from the CIFAR-10 dataset are of low resolution (32 x 32) and thus, we conducted some transforms to the training, testing, and validation sets prior to running. The training transformation resized to 224 x 224 and then a random portion of the image was cropped and again resized to 224. This resulting image was then randomly horizontaly flipped and then converted to a tensor for model training purposes. For the test and validation transforms, images were resized to 256 x 256, then a 224 x 224 center crop was applied. The result was again converted to a torch tensor.

As mentioned earlier, the CIFAR-10 dataset was 60000 training images, so we split the dataset into train and validation using a 5:1 ratio. When creating variations of this original implementation of the vision transformer, we decided to vary hyperparameters, specifically looking at the input patch size and number of heads. This followed some of the model variance conducted by Dosovitskiy et al [4]. and base configurations of the model were derived from the original repository [8].

3.3 Varying Input Patch Sizes

The values were chosen for input patch size were based on the parameters presented to us by the architecture. Specifically, it was required that image size be divisible by patch size and since we had an image size of 224, we were limited to those factors. Additionally, the number of patches is given by the quotient of the image size and patch size squared, and this value was required to be greater than 16. Thus, we decided to run with the following patch sizes: 7, 8, 14, 16, 28, 32, and report cross entropy loss for both training and validation sets every 200 mini-batches.

3.4 Varying Number of Heads

The values chosen for the number of heads are also based on the parameters presented in the architecture. The number of heads had to be divisible by the dimension of the token embeddings. We set the dimension of our token embeddings to be equal to 128. Our architecture requires that the number of heads is a divisor of the embeddings, ensuring an even split and concatenation of the information across the heads, maintaining compatibility and preventing issues related to uneven splitting or concatenation. Therefore, we selected the following number of heads: 2, 4, 8, 16, 32, 64, and 128. We proceed with the tests in an identical way to the patch size tests, reporting cross entropy loss for validation and training sets every 200 mini-batches.

3.5 Distillation ViT

We use a distillation ViT, as presented by Touvorn et al.[7] to compare to the standard ViT model. While regular ViT models are pretrained with hundreds of millions of images, Touvorn et al. record competitive results after training the distillation Vit on only ImageNet. This is achieved by using teacher-student strategy specific to transformers. During training, in addition to training on class labels, the transformer trains on the outputs of a convolution neural network. A major detractor of vision transformer models is that they lack appropriate inductive biases that convolutional networks have that make them ideal for images. However, as the original ViT paper showed, large enough data trumps inductive bias [4]. The distillation vision transformer gets around this problem by learning the inductive bias from the convolutional neural network, while still using a transformer, convolution-free architecture. This makes it trainable with much less data. We wanted to test this model to see how it would compare to the standard ViT on our small dataset.

3.6 Simple ViT

We also utilized an implementation of another model, Simple ViT, to compare to the standard ViT, Distillation ViT, and Convolutional Neural Network. The implementation was from the ViT-Pytorch library [8], and we adapted it to function for CIFAR-10 and our experiments.

Simple ViT was proposed as an update to the original ViT in [1]. According to the paper [1] and the implementation in [8], Simple ViT trains faster and better. It also has a simplified model structure which changes the positional embedding, does not use dropout, uses smaller batch sizes, and a linear layer at the end instead of an MLP [8], among other modifications.

We decided to compare Simple ViT to the standard ViT to see if these improvements hold true for the smaller CIFAR-10 dataset, as they were originally tested on ImageNet-1k [2]. The Simple ViT model seemed to be much improved over the original, so we hope to see a similar improvement on CIFAR-10.

We will compare the performance of the default Simple ViT model with the default standard ViT model. The default standard ViT model uses a patch size of 32, a depth of 6, 16 heads, and a dropout rate of 0.1 [8]. The default Simple ViT model uses a patch size of 32, a depth of 6, 16 heads, and no dropout. We will compare their training and validation loss over 20 epochs.

3.7 CNN

We implement a convolutional neural network on the CIFAR10 dataset. CNNs are considered state of the art and are ubiquitous in computer vision, so it is a natural baseline to compare against the vision transformer methods.

4 Results and experiments

In this section, we detail the results that we obtained from the experiments we conducted to evaluate standard ViT with different configurations and hyperparameters and compare it to other ViT models.

4.1 Low Patch Sizes Stop ViT Training

As discussed earlier, to study the effects of varying input patch sizes on training and validation loss, we ran the original implementation of the vision transformer using patch sizes = [7, 8, 14, 16, 28, 32]. We calculated cross entropy loss and accuracy for both datasets (training & validation) at each epoch. As demonstrated by our results in Figure 1, we see that the models do not begin to learn until the patch sizes increase to 28 and 32.

When analyzing these curves, it is important to note that the patch size is inversely related to the number of patches used. Thus, for lower patch sizes, we expect each input image to be split into more

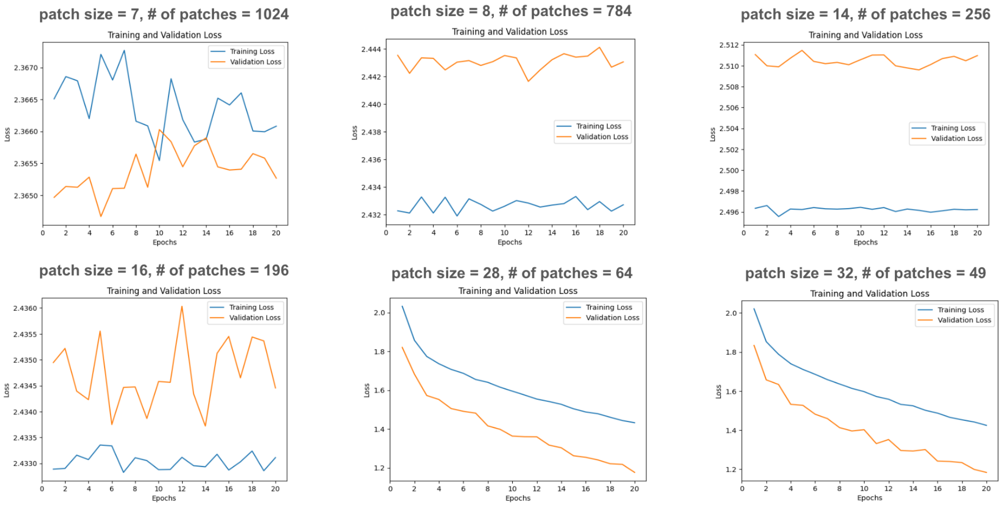

Figure 1: Training and validation loss with varying patch sizes

patches. These image patches can be treated as the word inputs to a NLP transformer. Our input data from CIFAR-10 contained low resolution 32 x 32 images which were then transformed to be of size 224 x 224. These images will be broken into the defined number of patches and then fed into the linear projection layer used to create the patch and positional embeddings for each patch. Similar to the case of NLPs and other vector-based representations in ML, we see that, for ViT, an embedding with `n` values can represent `n` unique features.

Therefore, one potential explanation for the inability to train we see in the lower patch size cases might just be the lack of representation of unique features. For the cases where patch size < 28, we see that there is a maximum of 256 pixels, or features, for the model to train using. In the case of patch size = 7, we would only have 49. When we consider how we increased the pixel count of the original low resolution images, we can also infer that the amount of unique features will decrease significantly as well. Therefore, the model's inability to learn that is depicted in the first plot of Figure 1 follows what we might intuitively expect. We infer that the amount of randomness that we see in the first plot of Figure 1 may be a result of the lack of available 1D data due to the small dataset size and low resolution of the training images. When we increase the patch size to 28, the model now has 784 features to learn from, which seems to be an indicator of a threshold required for the model to learn. Further studies will have to be conducted to see how loss scales with patch size and whether there is any relationship with training dataset size and input image resolution.

4.2 Low Head Quantity Stops ViT Training

For this set of tests, we held the patch size of the image embeddings constant at 16. Here, we alter the number of heads in the attention layer of our ViT architecture. It is important to note that in comparison to the original paper on the ViT, we are operating with much more limited computational resources, which may be impacting our results. In looking at these results, we can see that the models failed to train within 20 epochs unless they had 64 or more heads. Several factors could contribute to the varying loss values: ViTs use self-attention mechanisms in their architecture. Increasing the number of heads allows the model to capture more diverse and fine-grained patterns in the data. Higher head counts can enhance the model's ability to attend to different parts of the image simultaneously and at multiple scales, potentially leading to better feature extraction and representation. With a small dataset and limited computational resources, models with higher complexity (more heads in this case) might have an advantage in capturing more intricate patterns. Models with fewer heads may struggle to generalize well due to overfitting, which may be why the models with less heads failed to train. The fewer the heads, the more constrained the model might be in learning representations, leading to poorer generalization to unseen data. Models with fewer heads might require more epochs

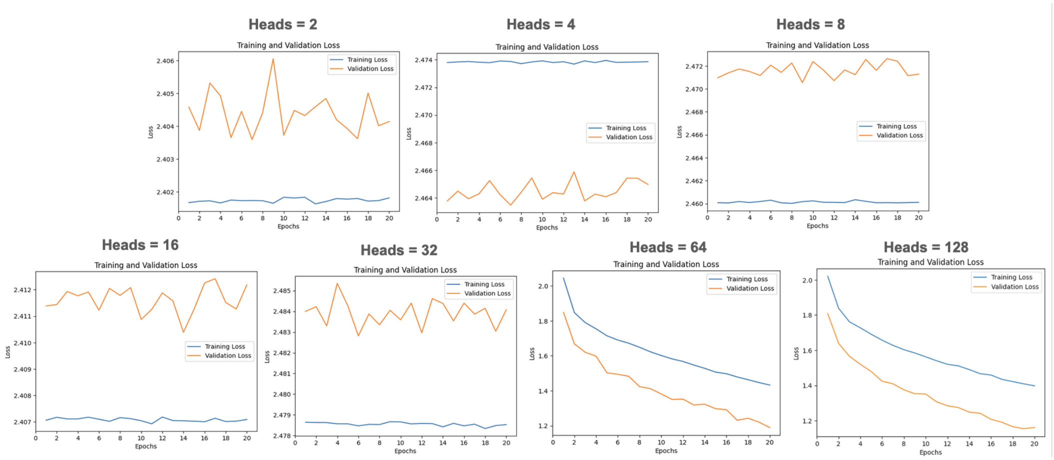

Figure 2: Training and validation loss with varying number of heads

to converge to an optimal solution, especially if they struggle to capture complex relationships within the data. With limited computational resources, these models might not have had enough time to converge within the 20 epochs.

4.3 Simple ViT Training Compared to Standard ViT

We trained both Simple ViT and the standard ViT on the CIFAR-10 dataset. Simple ViT training is shown in Figure 3, and standard ViT training is shown in Figure 4. In every epoch, Simple ViT had a smaller loss for both training and validation than the standard ViT.

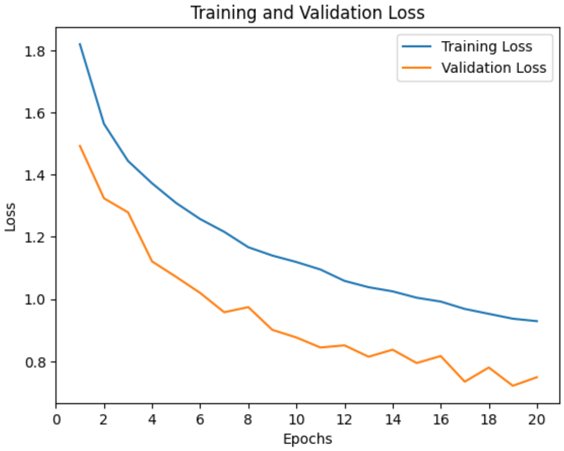

Figure 3: Simple ViT training and validation loss over 20 epochs. Uses default configuration of model.

Simple ViT obtained a final validation loss of roughly 0.8 and training loss of about 1.0, and the standard ViT reached a final validation loss of about 1.2 and training loss of 1.3. Both were trained for 20 epochs, but the curve did not flatten out for either, so it is likely that the loss would continue to decrease if the model were trained for longer. Thus, we cannot conclude that Simple ViT will have a lower final loss than the standard ViT, but it does seem likely given that the loss at every iteration was lower. At the very least, Simple ViT trains faster and more effectively than the standard ViT, even on smaller datasets (CIFAR-10).

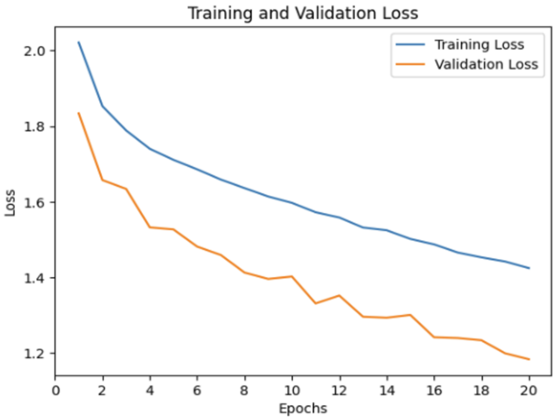

Figure 4: Standard ViT training and validation loss over 20 epochs. Uses default configuration of model.

4.4 Distillation ViT Performs Poorly on CIFAR-10

The Distillation ViT starts out with fairly low training and validation loss; however, validation loss increases as the training loss decreases (see Figure 5). This indicates that the model is overfitting. The model might not be modeling the data well due to insufficient complexity in the model. It also could be that the pretrained resnet model that acts as the teacher does not model CIFAR-10 very well. However, the model does achieve fairly low validation loss early in its training, indicating potential for this method.

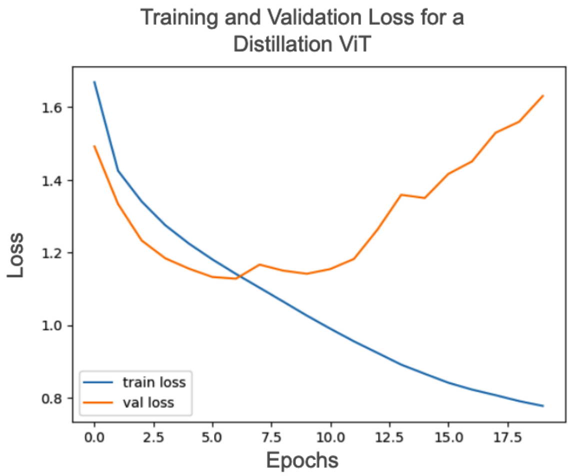

Figure 5: Standard Distillation ViT training and validation loss over 20 epochs. Uses default configuration of model[8]

.

4.5 Convolutional Neural Network Beats out ViT

The convolutional neural network achieved the lowest loss of any of the models tested with the CIFAR-10 over 20 epochs. The model obtained a training loss of below 0.2 and a validation loss of about 0.3 after 20 epochs (as can be seen in 6). This was expected, as Dosovitskiy et al describe in "An image is worth 16x16 words" the ViT network is outperformed by conventional CNN's over a small dataset. It's only when given lots of data to work with (data sets with 14M-300M images) that the ViT model begins to outperform the CNN [4]. It should be expected that when trained over a more expansive dataset, like the ImageNet-21k dataset [2], that the ViT might outperform the CNN in image classification.

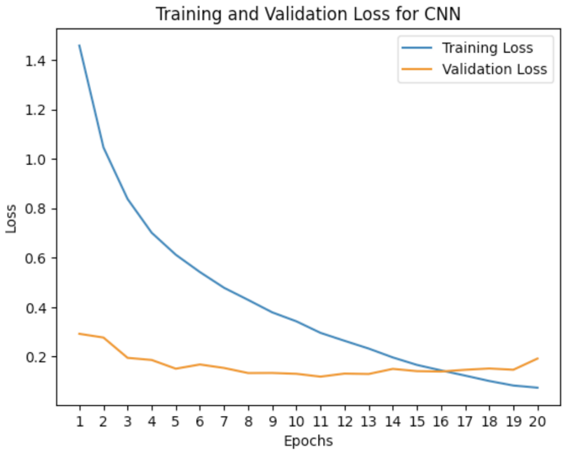

Figure 6: Convolutional Neural Network training and validation loss over 20 epochs. Uses default configuration of model[5]

5 Conclusion and future work

In this project, we evaluated the performance of the ViT model with different model configurations and hyperparameters. We also compared the standard ViT model to two other ViT models, Simple ViT and Distillation ViT. We saw that the CNN performed much better and trained much faster than all ViT models we tested, which is expected for such a small datatset. However, the CNN's validation loss did not seem to improve as the training loss decreased, indicating that the model may be overfitting or contain some other problem. We found that when we made the patch size or head quantity too small, the ViT model wasn't able to effectively train. We also found that above a certain patch size and head count, increasing the patch size or head count further did not seem to have a significant impact on the model's performance. When comparing the standard ViT to Simple ViT, we saw that Simple ViT continued to perform better (as suggested in the paper), even on smaller datasets. While Distillation ViT did not perform as well as the Simple ViT, it performed better than Standard ViT early in its training, before validation loss began to increase.

As future work, we believe that it would be valuable to evaluate more different ViT variations on CIFAR-10. The original Visual Transformers paper presents the model as something that only works well for large quantities of data [3]. However, the PyTorch implementation we used of ViT models contains 33 different variations of ViT [8]. While some of those models are for more specialized tasks, a full evaluation of their performance relative to each other could reveal interesting trends with different configurations of the model. The strange behvaior of Distillation ViT on CIFAR-10 could also be investigated in future work.

6 Broader impacts

Vision transformers have shown remarkable performance in various visual recognition tasks. They've demonstrated competitive or even superior performance compared to convolutional neural networks (CNNs) on tasks like image classification, object detection, and semantic segmentation [4]. One key advantage that supports using ViTs over CNNs and other image recognition/analysis networks is their scalability and flexibility. Previous work has shown that vision transformers are highly scalable to different input sizes and resolutions, making them adaptable to various image sizes without architectural changes. This has also been seen to be very beneficial in the case of higher resolution images; through our study, we aim to illustrate some of the effects of using lower resolution images and how we can vary hyperparameters and the architecture of the neural network to offset some of the inbuilt weaknesses of using lower resolution input data.

This information from this project can be used in various real-world applications, such as autonomous vehicles and healthcare, where gathering large quantities of high-quality data is not always possible or computationally efficient. If it is possible to use low resolution images (e.g. the 32 x 32 Cifar10 datasets) to accurately perform computer vision tasks, then this could help with lowering the space in memory and storage that input data takes and the energy required to collect high resolution images.

We do see, from the extremely preliminary trials we have run, that there are a significant amount of fluctuations in the loss outcomes and so it is hard to generalize any specific details about hyperparameters or how we can fix potential underfitting / overfitting issues.

For further researchers, we trained the model using a 2× NVIDIA GeForce RTX 4090 GPU with a RAM size of 256 GB. For each variable changed, we ran overnight for 20 epochs with the Cifar10 dataset. It is significant to note that at smaller patch sizes and with more heads in the attention layer, we expect an increase in computational energy expended.

7 Code

All code for this project can be found in this repository: https://github.com/f-wright/ViT-implementation/tree/d7c5d79dc8656cc55ae0ce606a4aefad32eaa22e

The ViT-Pytorch implementation in [8] was also used in development of the code of this project.

References

[1] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. "Better plain ViT baselines for ImageNet-1k". In: arXiv:2205.01580 (May 2022). arXiv:2205.01580 [cs]. URL: http://arxiv.org/abs/2205.01580.

[2] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". en. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL: IEEE, June 2009, pp. 248–255. ISBN: 978-1-4244-3992-8. DOI: 10.1109/CVPR.2009.5206848. URL: https://ieeexplore.ieee.org/document/5206848/.

[3] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: arXiv:2010.11929 (June 2021). arXiv:2010.11929 [cs]. URL: http://arxiv.org/abs/2010.11929.

[4] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[5] Shonit Gangoly. *CNN on CIFAR10 data set using pytorch*. Mar. 2021. URL: https://shonit2096.medium.com/cnn-on-cifar10-data-set-using-pytorch-34be87e09844.

[6] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[7] Hugo Touvron et al. *Training data-efficient image transformers distillation through attention*. 2021. arXiv: 2012.12877 [cs.CV].

[8] Phil Wang. *ViT PyTorch Implementation*. https://github.com/lucidrains/vit-pytorch. 2023.