DS-265 DLCV 2025: Assignment 2 Report

Godwin Khalko

SR No. 24006

1 Implementation Details

1.1 Vision Transformer Model

We implemented a ViT model based on the paper recommended by in the assignment. The model splits an image into fixed-size patches, adds the cls token(for image classification), adds the positional embeddings, projects them into a latent space, and processes them using self-attention layers. Image augmentations such as horizontally flip with 50 % chance, random crop with padding, color jitter, random rotation by $\pm$ 15° and normalization.

1.2 Training Setup

Parameters	Value
Dataset	CIFAR-10 (10 classes, 60,000 images)
Training	45000
Valiation	5000
Test	10000
Optimizer	AdamW LR: 1×10^{-4}, WD: 1×10^{-5}
Loss Function	Cross-entropy loss
Batch Size	128
Epochs	30
Self-attention heads	4
Transformer encoder layers	4
Dropouts	0.1 (Encoder and MLP outputs)

Table 1: Hyperparameter setup

2 Experiments and Results

2.1 Experiment 1: Baseline Training

We trained the ViT model with 4 attention heads on the full CIFAR-10 dataset. Hyper-parameters were as stated in the table above. The model achieved an accuracy of 59.91% on the test set.
This shows that the ViT isn't able to perform as good as the CNNs(Eg. ResNet, which achieves 90+ accuracies even on simpler models) on the smaller datasets, something that is stated even in the ViT paper.
Another observation which was noticed was that the role of image augmentation in the test performance. Without any image augmentation, the average test accuracy was 55 % whereas with image augmentation, the average test accuracy was jumped to 59 %

2.2 Experiment 2: Effect of Training Data Size

We trained the model on different fractions of the dataset (5%, 10%, 25%, 50%, 100%). The validation data and the test data remain the same size as mentioned above. Table 2 summarizes the results.

Though, another thing to notice is that the test accuracy only increases by 5.36% even though we're basically doubling the data, a similar increase is seen when we increase from 25 % to 50%.Something along the similar lines could be observed in the training vs validation graphs of the different dataset percentages. In the Figure 2, the validation losses has significant decrease in going from 25% to 50% but not a huge decrease going from 50% to 100%.

Notice in Figure 1 that the validation loss decreases to a minimum of 1.2 value before stagnating and sometimes increasing. Thought the effect of dropout could be seen, which helps the validation loss reduce further before stagnating.

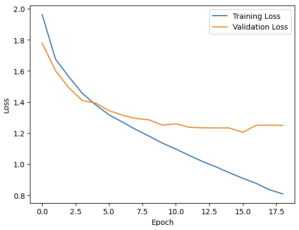


Figure 1: Training vs Validation graphs for Experiment 1

We can, obviously, see that as the training data increases, the model is able to learn more about the data and this result in better performance, as evident in the continuous increase in the test accuracy. The biggest increase happened in the case when we increase the data size from 10 % to 25 %(By 7.38 points).

Data Size	Test Accuracy
5%	35.46%
10%	39.88%
25%	47.26%
50%	52.93%
100%	58.29%

Table 2: Performance of ViT with varying dataset sizes.

2.3 Experiment 3: Effect of Patch Size

We evaluated patch sizes (4x4, 8x8, 16x16). Table 4 shows the test accuracy when there's a 50% overlap in the patches, while Table 3 is for no-overlap cases. Notice that in the case of overlapping patches, patch size 16 produces the worst test accuracy whereas the same produces the best test accuracy in the case of non-overlapping patches. Conversely, same is true for patch size of 8. Though, it is to be noted that the margin for best accuracy is much higher in the overlapping case, than in the overlapping case(where the best and worst are depared by just by 0.89 points)
Figure 3 shows the Train vs Validation plots for the Overlapping case and Figure 4 is for the Non-overlapping case. In term of the plots, both of them look identical.

Patch Size	Test Accuracy
4	57.15%
8	60.36%
16	56.54%

Table 3: Performance of ViT with varying overlapping patch sizes.

Patch Size	Test Accuracy
4	58.33%
8	58.13%
16	59.02%

Table 4: Performance of ViT with varying non-overlapping patch sizes.

2.4 Experiment 4: Effect of Attention Heads

We varied the number of attention heads and observed accuracy changes. The results are in Table 5.

The overfit of the model can be confirmed by taking a look at the validation curve that form as U-curve that is typically seen when the model overfits. Another thing to note is that we've applied dropouts after every transformer encoder layer, which has helped significantly in increasing test accuracy and model learning. So, all in all, as we increase the number of attention heads, we increase the complexity of the model that helps the model learn more global features to predict better.

2.5 Experiment 5: CLS Token from Different Layers

The model was first trained with 6 transformer encoder layers normally. During the testing, instead of just passing the CLS token from the last layer, we pass the CLS Token from every layer to the MLP to get the logits which is then used to generate the prediction and calculate the accuracy. We show how

As we can see from the table, the test accuracy increases linearly with the addition of the number of the heads. A significant jump could be seen when we switch from 2 to 4, but a minimal one when we switch from 4 to 8. Figure 5 shows that the model tends to overfit as we increase the number of heads.

Data Size	Test Accuracy
2	57.10%
4	57.95%
8	58.07%

Table 5: Performance of ViT with varying Attention Head numbers.

taking CLS from different layers affect the performance as summarized in Table 6.
Additionally, I've also trained the transformer with varying number of encoder layer to compare the performance when we take the CLS token from a middle layer and a final layer, which can be seen in Table 7. Notice that when we train with different transformer encoder end to end, we get much better accuracies as compared to taking CLS tokens from the middle layers. This might be because in the first case the transformer tends to optimize over the entirety of the encoder layers rather than making sure every layers shows better accuracy, as in the latter case.

Encoder Layer	Test Accuracy
1	10.46%
2	17.03%
3	28.26%
4	39.32%
5	51.11%
6	58.22%

Table 6: Performance of ViT with CLS taken from middle layers.

Encoder Layers	Test Accuracy
1	51.10%
2	55.14%
4	57.20%
8	58.59%

Table 7: Performance of ViT with CLS taken from final layers.

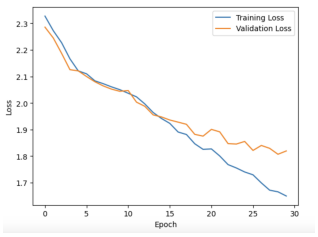
2.6 Experiment 6: Attention Map Visualization

We visualized attention maps for two test images per class. We have trained the model on the best hyperparameters that was found across all of these experiments and was used to test the images. We have used random images per class from the test dataset and tried to visualize those attention scores. For each image, we would get the attention score from each of the Transformer encoder layer attention scores in the form of (Number of Layer, Batch, Number of Heads, Sequence Length, Sequence Length). There are two ways of showing the attention score, Raw Attention scores and Rolling Attentions. The Raw attention scores provide information regarding attention scores for a patch with respect to all the other individual patches present in the images. Roll-out attention is a technique used to aggregate attention scores across multiple layers of a transformer model, typically to visualize which input tokens (or image patches in ViTs) influence the final prediction. Figure 6 shows the rolling attention scores for the two random images from each class. Notice how the attention maps highlight the important parts of the image that help classify the object in the image.

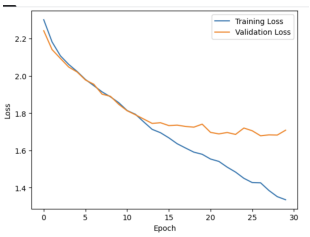
3 Conclusion

In this assignment, we explored the implementation and evaluation of a Vision Transformer (ViT) for image classification. Through a series of controlled experiments, we analyzed how different factors—such as dataset size, patch size, number of attention heads, and CLS token extraction from different layers—affect model performance.Overall, this assignment provided valuable hands-on experience with self-attention mechanisms in transformers and their effectiveness for vision tasks. The results reinforce the importance of architectural choices in optimizing model performance. Future work could involve pretraining on larger datasets, experimenting with hybrid CNN-ViT architectures, or applying these techniques to more complex real-world tasks.

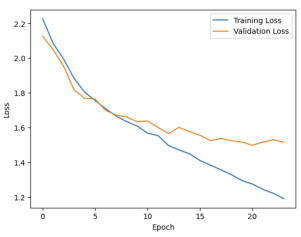
4 Appendix



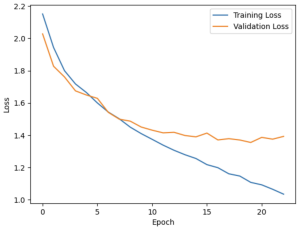
(a) (5%)



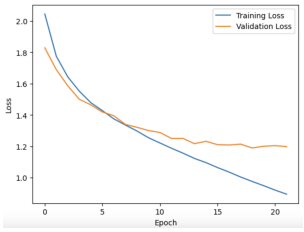
(b) (10%)



(c) (25%)

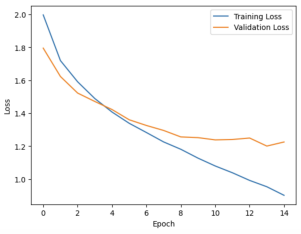


(d) (50%)

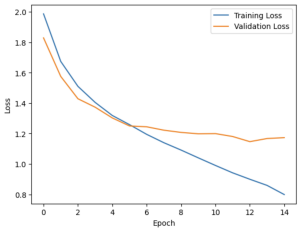


(e) (100%)

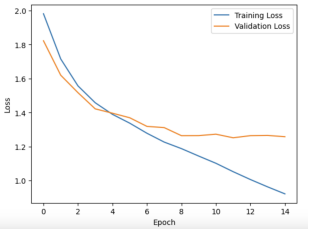
Figure 2: Training vs Validation plots for Experiment 2



(a) (Patch size: 4)

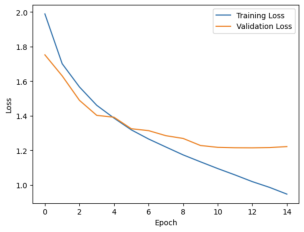


(b) (Patch size: 8)

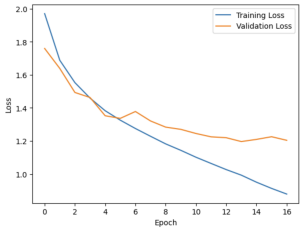


(c) (Patch size: 16)

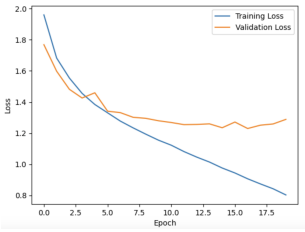
Figure 3: Training vs Validation plots for Experiment 3 Overlapping



(a) (Patch size: 4)

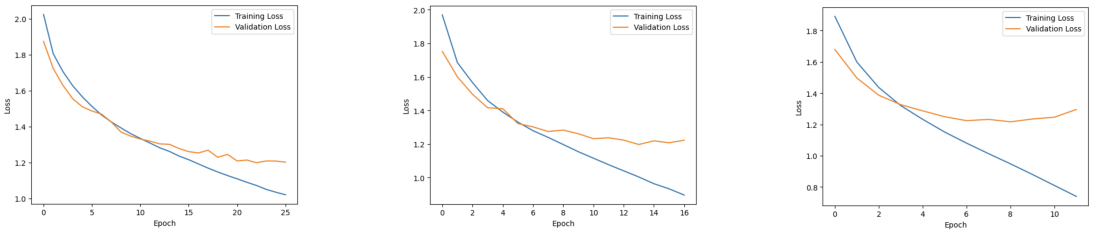


(b) (Patch size: 8)



(c) (Patch size: 16)

Figure 4: Training vs Validation plots for Experiment 3 Non-Overlapping



(a) # Heads: 2 (b) # Heads: 4 (c) # Heads: 16

Figure 5: Training vs Validation plots for Experiment 4

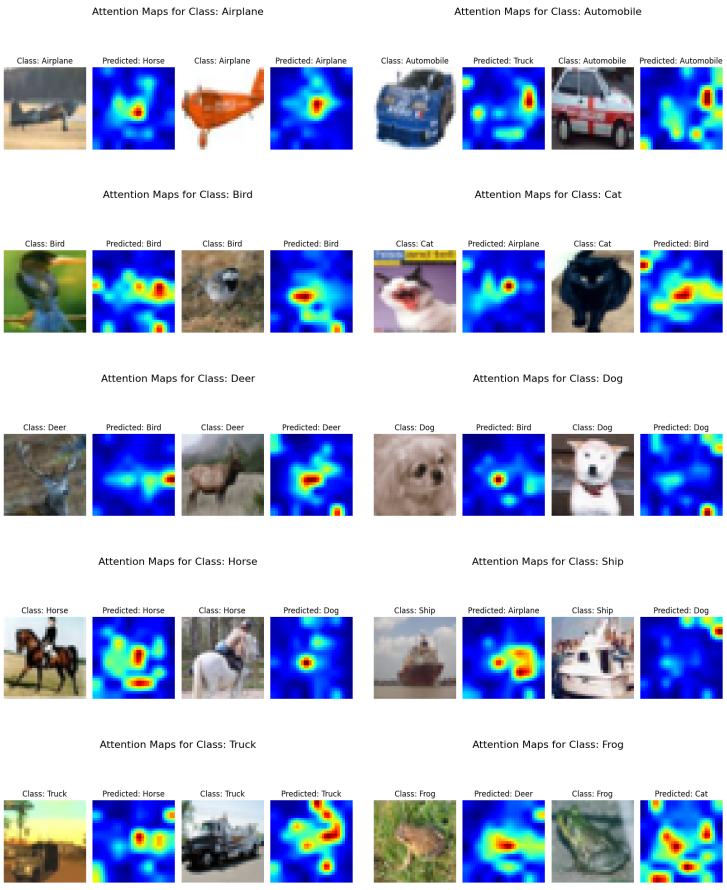


Figure 6: Rolling Attention maps for Experiment 6