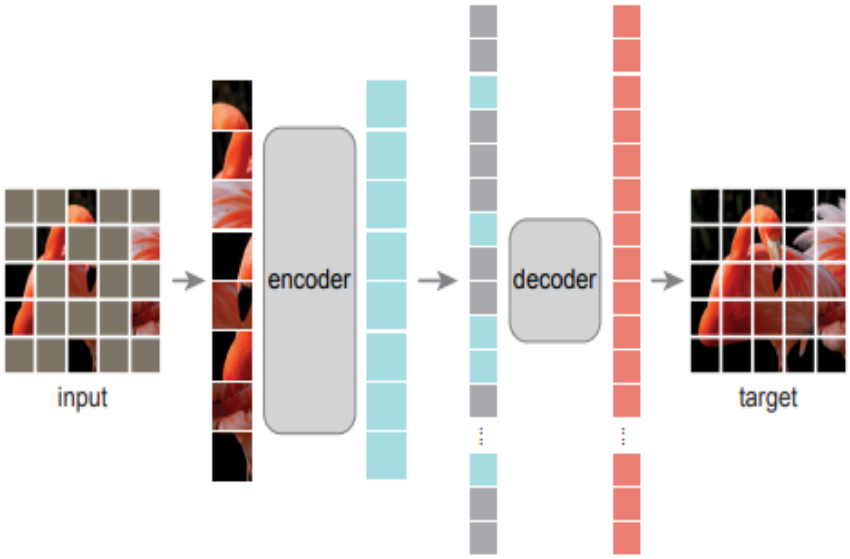
Advanced Machine Learning
final report

3조
Min geon,
Jung sung hoon,
Kulboboev Shukhrat

Topic : self-supervised learning
Target paper : Masked Autoencoders Are Scalable Vision Learners(CVPR 2022)
Reference paper : Wave-VIT: Unifying Wavelet and Transformers for Visual Re presentation Learning (ECCV 2022).



index

1. problem statement

we think that reconstruction image is too blurry. it is result that MAE's represenati

on learning is still lacking. The reconstucted image is blurry because high frequencies are lost. Therefore we focus on visual representation learning through frequency to s olve problem.

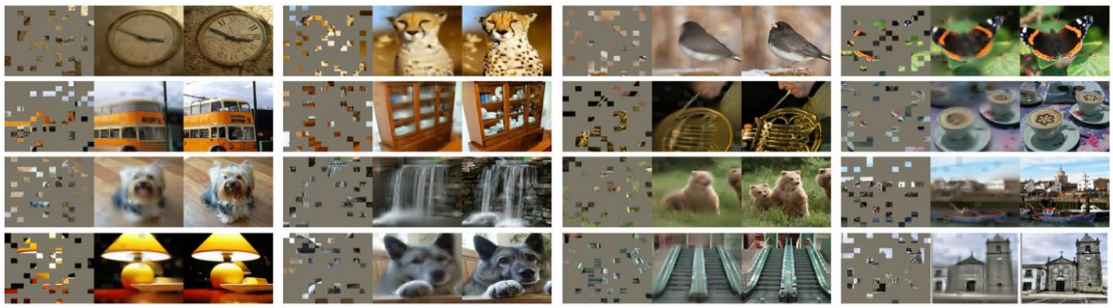
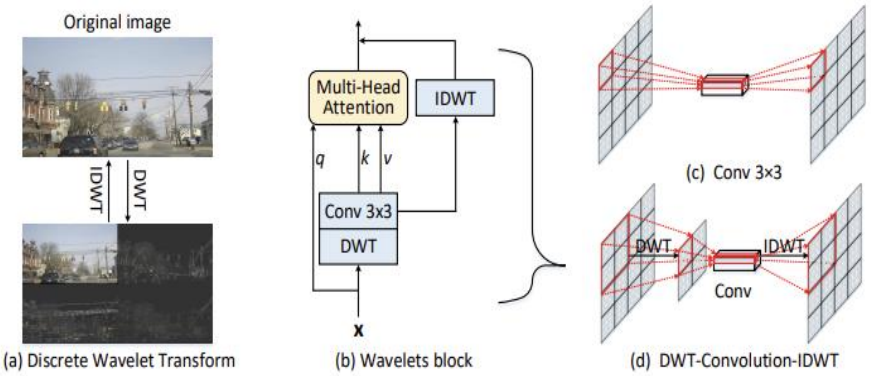


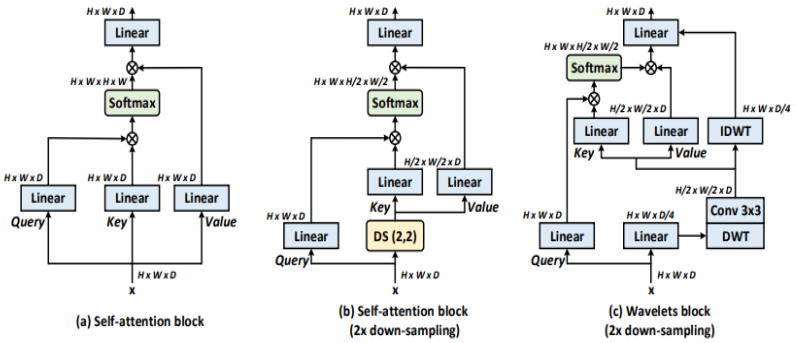
Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction† (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix. †*As no loss is computed on visible patches, the model output on visible patches is qualitatively worse. One can simply overlay the output with the visible patches to improve visual quality. We intentionally opt not to do this, so we can more comprehensively demonstrate the method's behavior.*

2. proposed method

We proposed a method of changing the module from the existing self-attention method to the wavelet transform attention method. we reference at Wave-VIT: Unifying Wavelet and Transformers for Visual Representation Learning (ECCV 2022).
Wavelet transform is effective for time-frequency analysis and is widely used in compression. this paper is proposed that using discrete wavelet transform(DWT), inverse discrete wavelet transform(IDWT) because multi-head self-attention is more computational cost. We tried to apply this method. we expect training speed, fine-tuning speed fast and reconstructed image more sharp. It is more improve performance to downstream task.



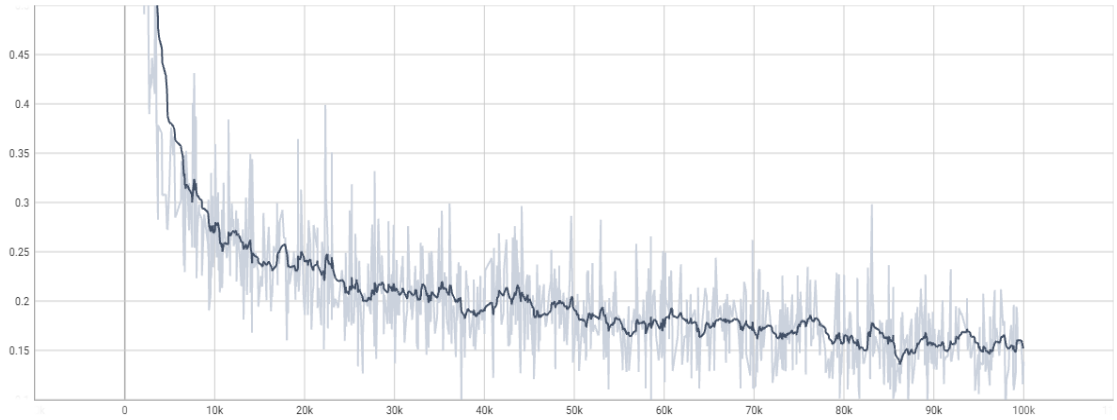
(a) Discrete Wavelet Transform (b) Wavelets block (d) DWT-Convolution-IDWT



3. experiment

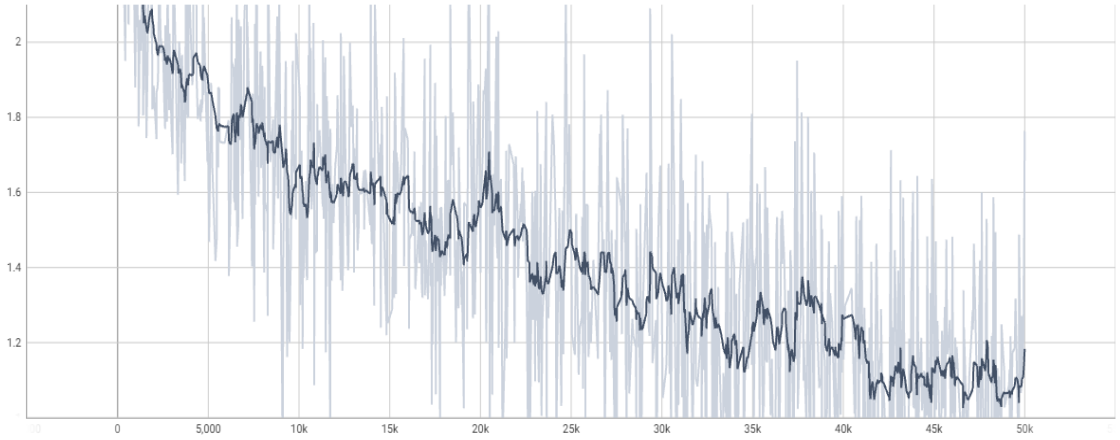
In the case of the dataset, a large-scale dataset must be used when using a vision t ransformer, but due to the environmental limitations of the experiment, we chose cifar -10 to easily check our idea.

<pretraining train loss>

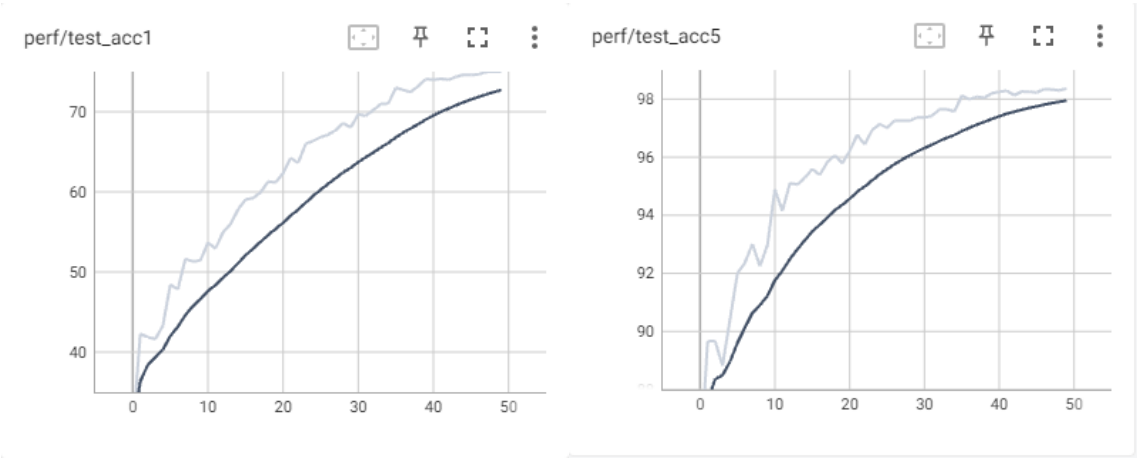


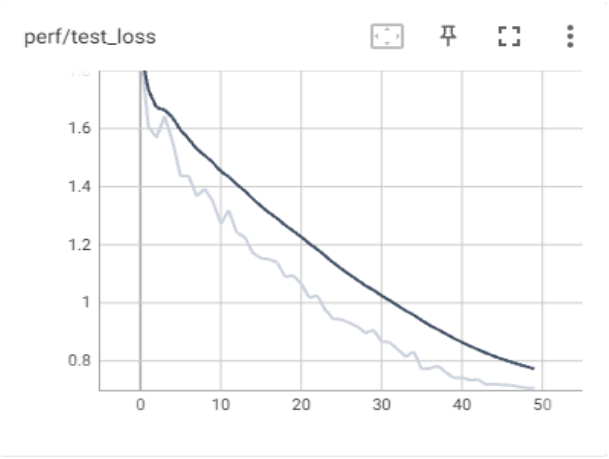
{"train_lr": 6.24857298672417e-05, "train_loss": 0.20023345728635789, "epoch": 40}
{"train_lr": 6.240012860749686e-05, "train_loss": 0.19810505717992782, "epoch": 41}
{"train_lr": 6.222914701045804e-05, "train_loss": 0.19756853072166441, "epoch": 42}
{"train_lr": 6.197325372479759e-05, "train_loss": 0.19557023706197738, "epoch": 43}
{"train_lr": 6.163315013622454e-05, "train_loss": 0.19516970252513885, "epoch": 44}
{"train_lr": 6.120976844503516e-05, "train_loss": 0.19362956388950348, "epoch": 45}
{"train_lr": 6.07042691110159e-05, "train_loss": 0.1933357198166847, "epoch": 46}
{"train_lr": 6.01180376727042e-05, "train_loss": 0.19190957434415817, "epoch": 47}
{"train_lr": 5.945268094972429 4e-05, "train_loss": 0.19005368705034256, "epoch": 48}
{"train_lr": 5.871002263860584e-05, "train_loss": 0.1897548224377632, "epoch": 49}

<finetuning train loss>

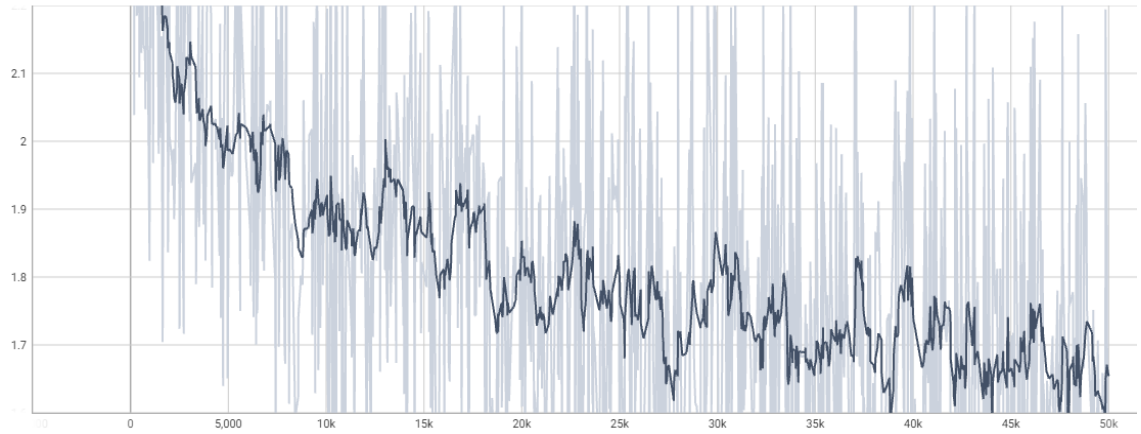


{"train_lr": 7.523801309717829e-06, "train_loss": 1.168001580324173, "test_loss": 0.7425702916622162, "test_acc1": 74.0, "test_acc5": 98.25, "epoch": 40, "n_parameters": 41116772}
{"train_lr": 6.262463417986463e-06, "train_loss": 1.1589121054649354, "test_loss": 0.7336947041034698, "test_acc1": 74.12, "test_acc5": 98.29, "epoch": 41, "n_parameters": 41116772}
{"train_lr": 5.125298242484447e-06, "train_loss": 1.153238209552765, "test_loss": 0.73589951171875, "test_acc1": 73.99, "test_acc5": 98.13, "epoch": 42, "n_parameters": 41116772}
{"train_lr": 4.117845937639396e-06, "train_loss": 1.1466462660312653, "test_loss": 0.7183035012245178, "test_acc1": 74.35, "test_acc5": 98.26, "epoch": 43, "n_parameters": 41116772}
{"train_lr": 3.245014709811729e-06, "train_loss": 1.1394743491077424, "test_loss": 0.7200762296199799, "test_acc1": 74.62, "test_acc5": 98.25, "epoch": 44, "n_parameters": 41116772}
{"train_lr": 2.5110569050067765e-06, "train_loss": 1.131714769001007, "test_loss": 0.7173528614521026, "test_acc1": 74.58, "test_acc5": 98.23, "epoch": 45, "n_parameters": 41116772}
{"train_lr": 1.9195482918723448e-06, "train_loss": 1.128947275094986, "test_loss": 0.7166503818273544, "test_acc1": 74.66, "test_acc5": 98.34, "epoch": 46, "n_parameters": 41116772}
{"train_lr": 1.4733706409133818e-06, "train_loss": 1.1282100876903534, "test_loss": 0.711397427725792, "test_acc1": 74.97, "test_acc5": 98.33, "epoch": 47, "n_parameters": 41116772}
{"train_lr": 1.1746976847957525e-06, "train_loss": 1.1213003156757355, "test_loss": 0.7066548881053925, "test_acc1": 75.01, "test_acc5": 98.3, "epoch": 48, "n_parameters": 41116772}
{"train_lr": 1.0249845281391598e-06, "train_loss": 1.122831041355133, "test_loss": 0.7058345557451248, "test_acc1": 74.98, "test_acc5": 98.36, "epoch": 49, "n_parameters": 41116772}

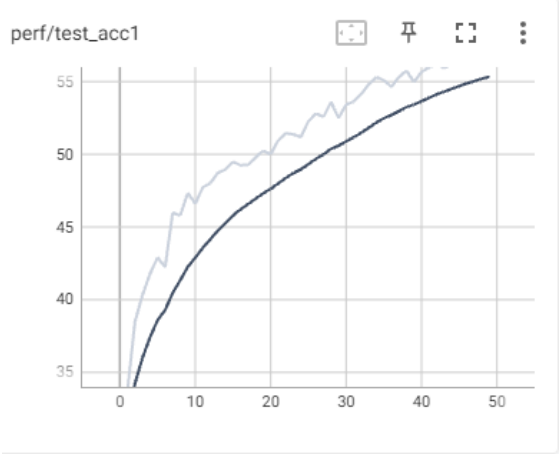


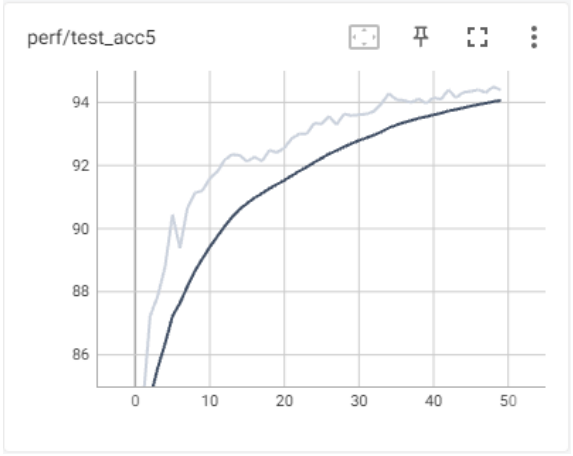


<finetuning train loss_our proposed>



{"train_lr": 7.523801309717829e-06, "train_loss": 1.6864059208679198, "test_loss": 1.251656888771057, "test_acc1": 55.67, "test_acc5": 94.15, "epoch": 40, "n_parameters": 126241604}
{"train_lr": 6.262463417986463e-06, "train_loss": 1.684573376350403, "test_loss": 1.2398381196975707, "test_acc1": 55.98, "test_acc5": 94.1, "epoch": 41, "n_parameters": 126241604}
{"train_lr": 5.125298242484447e-06, "train_loss": 1.6822495053482056, "test_loss": 1.2370990285873413, "test_acc1": 56.24, "test_acc5": 94.4, "epoch": 42, "n_parameters": 126241604}
{"train_lr": 4.117845937639396e-06, "train_loss": 1.6804218548965455, "test_loss": 1.2477351434707642, "test_acc1": 55.93, "test_acc5": 94.15, "epoch": 43, "n_parameters": 126241604}
{"train_lr": 3.245014709811729e-06, "train_loss": 1.6729083525085449, "test_loss": 1.2429365970611572, "test_acc1": 56.24, "test_acc5": 94.32, "epoch": 44, "n_parameters": 126241604}
{"train_lr": 2.5110569050067765e-06, "train_loss": 1.677544997367859, "test_loss": 1.233582684135437, "test_acc1": 56.4, "test_acc5": 94.35, "epoch": 45, "n_parameters": 126241604}
{"train_lr": 1.9195482918723448e-06, "train_loss": 1.6722665362930298, "test_loss": 1.23357429561615, "test_acc1": 56.53, "test_acc5": 94.4, "epoch": 46, "n_parameters": 126241604}
{"train_lr": 1.4733706409133818e-06, "train_loss": 1.6729297211837768, "test_loss": 1.2326972568511962, "test_acc1": 56.43, "test_acc5": 94.31, "epoch": 47, "n_parameters": 126241604}
{"train_lr": 1.1746976847957525e-06, "train_loss": 1.6774164756774903, "test_loss": 1.2303087787628173, "test_acc1": 56.72, "test_acc5": 94.5, "epoch": 48, "n_parameters": 126241604}
{"train_lr": 1.0249845281391598e-06, "train_loss": 1.6691475969696046, "test_loss": 1.2331407499313354, "test_acc1": 56.46, "test_acc5": 94.39, "epoch": 49, "n_parameters": 126241604}





4. discussion

The results of the experiment did not change as we thought.
Since masked autoencoder(MAE) uses only unmasked patches as input to the encoder, it seems that wavelet transform did not help visual representation learning.
In the case of the dataset, Vision Transformer uses a large-scale dataset, but due to the limitations of the experimental environment (ex. GPU), the CIFAR-10 dataset was used to easily verify the idea we proposed.
In addition, it would have been nice to test a wide range of tasks such as object detection and semantic segmentation, but in our case, only classification was confirmed.
contrary to our expectations, there is an no improvement in terms of speed despite the characteristic that wavelet transform helps image compression without loss. speed is 2 hours 37minutes to 5 hours 10minutes and max accuracy is 75% to 56%.

```
Test: Total time: 0:00:19 (0.0308 s / it)
* Acc@1 56.460 Acc@5 94.390 loss 1.233
Accuracy of the network on the 10000 test images: 56.5%
Max accuracy: 56.72%
Training time 5:10:07
```

<finetuning_our proposed>

```
Test: Total time: 0:00:12 (0.0197 s / it)
* Acc@1 74.980 Acc@5 98.360 loss 0.706
Accuracy of the network on the 10000 test images: 75.0%
Max accuracy: 75.01%
Training time 2:37:52
```

<finetuning>