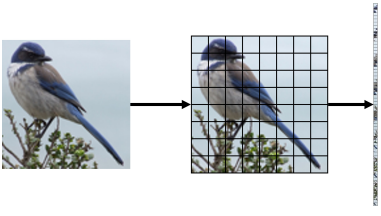
PatchRot: Self-Supervised Training of Vision Transformers by Rotation Prediction - Supplementary

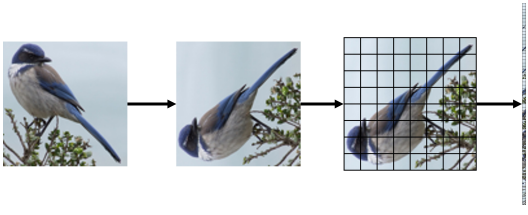
Sachin Chhabra[1]
sachin.chhabra@asu.edu
Hemanth Venkateswara[2]
hvenkateswara@gsu.edu
Baoxin Li[1]
baoxin.li@asu.edu

[1] Arizona State University
699 S Mill Ave.
Tempe, AZ, USA- 85281

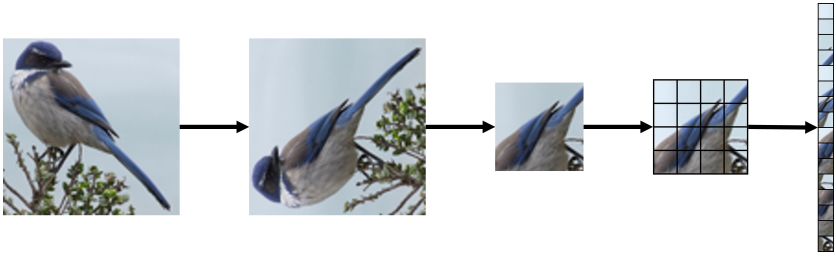
[2] Georgia State University
25 Park Pl NE
Atlanta, GA, USA - 30303



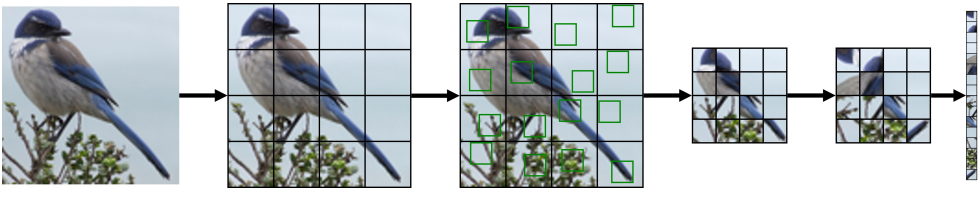
(a) Original Input



(b) Original ImageRot Input



(c) Our ImageRot Input



(d) Our PatchRot Input

Figure 1: Processing of different types of inputs by Vision Transformer.

1 Step by Step PatchRot Image Construction

In Figure 1, we show the construction of different input types including our proposed PatchRot image. Figure 1a shows the processing of an original input by a Vision Transformer. Let's assume the image is 32×32 and the patch size is 4×4. This results in $\frac{32}{4} \times \frac{32}{4} = 8 \times 8 = 64$ patches. Figure 1b shows the original image rotation input. It rotates the image before splitting it into patches.

Figure 1d shows the creation of the PatchRot image. PatchRot image uses a patch size of $P' = P + B = 8$. This results in $\frac{32}{8} \times \frac{32}{8} = 4 \times 4 = 16$ patches. From each 8×8 patch, we extract the patch of the original patch size 4×4 using random crops (marked as green boxes). This results in 16 patches of size 4×4, which makes the PatchRot input of size 16×16. Next, the patches are rotated by random rotation angles to produce the PatchRot input. Based on the PatchRot input we modify the image rotation prediction input. Like the original rotation prediction input, we rotate the whole image but extract a random crop of 16×16 to match the PatchRot input size. Figure 1c shows our image rotation input.

Vision Transformer processes both our image rotation and PatchRot images together. Note the original input has 64 patches, whereas the input of our method has only 16 ($\frac{1}{4}th$) patches. Our approach saves on computation per image by reducing the input size and drastically decreasing the number of patches. This process of creating the inputs is done in the dataloaders, resulting in negligible overheads.

2 More Attention Maps

Here, we present more attention maps of our ViT network trained using PatchRot on the validation set of Tiny-ImageNet in Figure 2. We can see that the ViT learns to attend to the main object in the image without any need for the training labels.

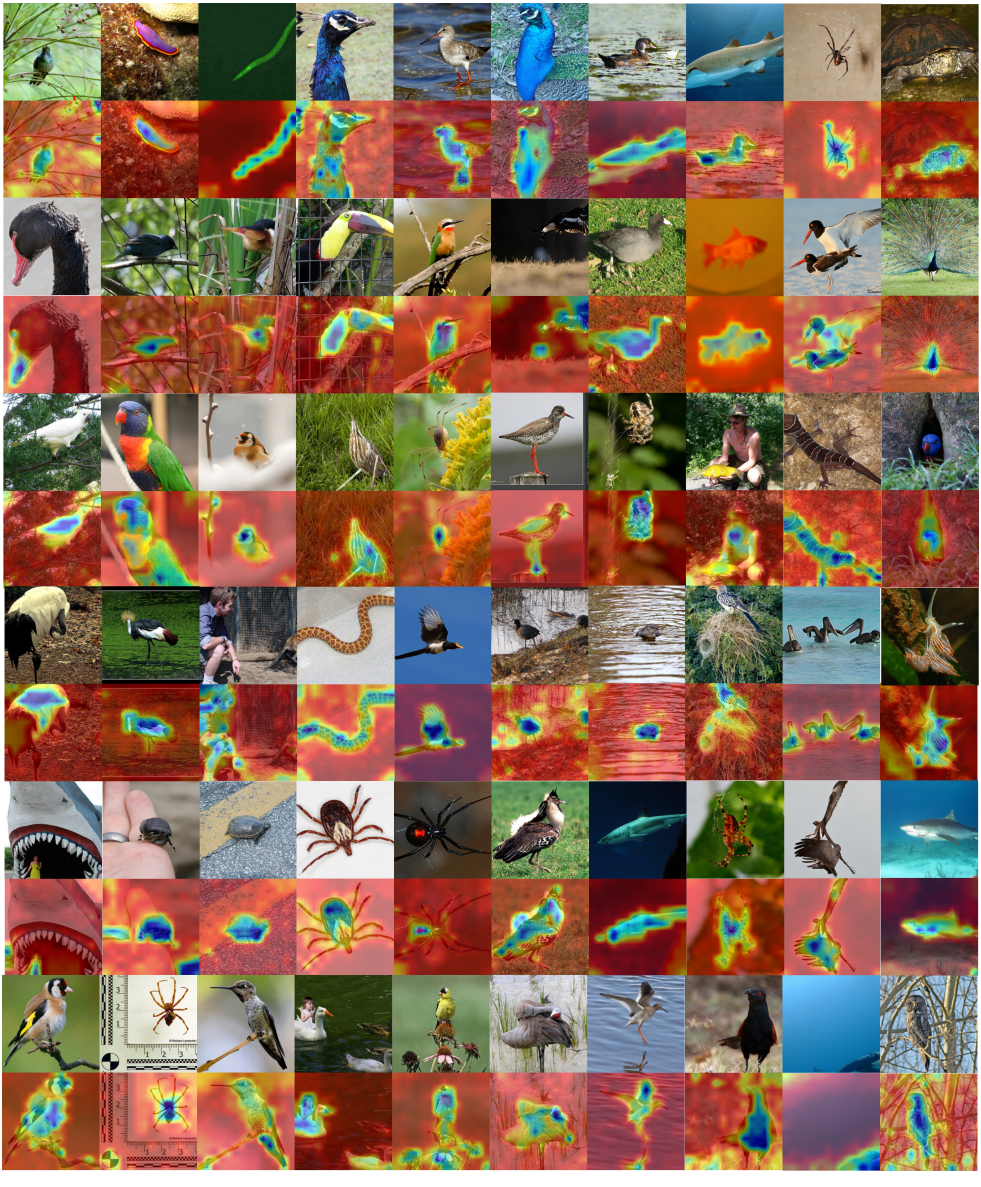


Figure 2: More attention maps of ViT trained using PatchRot on the validation set of Tiny-ImageNet. Odd row: Input and Even row: Attention Map.

3 Training Details

This section discusses training details for our approach and the baselines for all the experiments.

3.1 Network Details

We used the DeiT-Tiny Vision Transformer for all our experiments [12]. It is a much smaller version of the original ViT [5] and contains only 5.8 million parameters instead of 86 million (Original ViT). While training the network from random initialization (pretraining and supervised training from scratch), we used Adam Optimizer with a learning rate and a weight decay of 5×10^{-4} and 5×10^{-2}, respectively. While fine-tuning the network, we used a learning rate of 1×10^{-3} with a layer-wise learning rate decay of 0.75 (similar to Masked AutoEncoder [8]). We always warmed up the learning rate and then decayed it with a cosine scheme. We used basic augmentations (mentioned below) during pretraining and fine-tuning; no advanced augmentations (like Mixup [15], CutMix [14], PatchSwap [3] etc.) were used for pretraining/fine-tuning.

3.2 Baseline Details

3.2.1 Rotation Prediction [6]

We used four rotation angles of 0°, 90°, 180°, or 270°of an image together in a mini-batch. This effectively increases the batch size to four times.

3.2.2 SIMCLR [1]

We used two views of each input image. We applied the following augmentations on each view: RandomResized Crops, Random Horizontal flips (wherever applicable), Random Grayscale with 0.2 probability, and Color Jitter of strength 0.4, 0.4, 0.4, 0.1 with 0.8 probability. The MLP heads of SIMCLR consist of two fully connected layers with 96 output neurons in both and a ReLU non-linearity after the first layer.

3.2.3 MoCo v2 [2, 7]

We used two views of each input image. We applied the following augmentations on each view: RandomResized Crops, Random Horizontal flips (wherever applicable), Random Grayscale with 0.2 probability, and Color Jitter of strength 0.4, 0.4, 0.4, 0.1 with 0.8 probability. The MLP heads of Moco consist of two fully connected layers with 96 output neurons in both and a 1D-BatchNorm with ReLU non-linearity after the first layer and an L2-normalization after the last layer. The hyperparameter memory bank, momentum, and temperature are set to 4096, 0.999, and 0.2, respectively.

3.2.4 Masked AutoEncoder [8]

We used a masking ratio of 0.75 to create the input images. We pretrained ViT with double the epochs for this method as it took longer to converge.

3.2.5 PatchRot

Our approach uses four rotation angles of 0°, 90°, 180°, or 270°for the Image Rotation prediction together in a mini-batch along with the PatchRot image. This effectively increases the batch size to five times. However, our method uses a buffer size equal to the patch size which results in reducing the input size to half the original image size. Due to this, the number of patches per input in our method is decreased to just $\frac{1}{4}th$. The number of patches increases computations quadratically in ViTs, and our approach saves computing by using just $\frac{1}{4}th$ number of patches.

3.3 Dataset-Specific Details

3.3.1 CIFAR10 and CIFAR100 [9]

- **Input size**: 32×32
- **Patch size**: 4×4
- **Batch size**: 256
- **Pretraining epochs**: 300
- **Fine-tuning epochs**: 50
- **Supervised training epochs**: 300
- **Learning rate warm epochs for pretraining and supervised training**: 10
- **Learning rate warm epochs for fine-tuning**: 5
- **Augmentations**: Padding of size 4 with Random Crops, and random horizontal flips.

3.3.2 FashionMNIST [13]

- **Input size**: 32×32
- **Patch size**: 4×4
- **Batch size**: 256
- **Pretraining epochs**: 150
- **Fine-tuning epochs**: 25
- **Supervised training epochs**: 125
- **Learning rate warm epochs for pretraining and supervised training**: 10
- **Learning rate warm epochs for fine-tuning**: 5
- **Augmentations**: Padding of size 4 with Random Crops, and random horizontal flips.

3.3.3 SVHN [10]

- **Input size**: 32×32
- **Patch size**: 4×4
- **Batch size**: 256
- **Pretraining epochs**: 150
- **Fine-tuning epochs**: 25
- **Supervised training epochs**: 125
- **Learning rate warm epochs for pretraining and supervised training**: 10
- **Learning rate warm epochs for fine-tuning**: 5
- **Augmentations**: No augmentations.

3.3.4 Animals-10N [11] and TinyImageNet [4]

- **Input size**: 64×64
- **Patch size**: 8×8
- **Batch size**: 256
- **Pretraining epochs**: 150
- **Fine-tuning epochs**: 25
- **Supervised training epochs**: 150
- **Learning rate warm epochs for pretraining and supervised training**: 10
- **Learning rate warm epochs for fine-tuning**: 5
- **Augmentations**: Padding of size 4 with Random Crops, and random horizontal flips.

3.3.5 ImageNet100 [4]

- **Input size**: 224×224
- **Patch size**: 16×16
- **Batch size**: 128
- **Pretraining epochs**: 150
- **Fine-tuning epochs**: 25
- **Supervised training epochs**: 150
- **Learning rate warm epochs for pretraining and supervised training**: 10
- **Learning rate warm epochs for fine-tuning**: 5
- **Augmentations**: Random resized crops of 224 along with random Horizontal flips.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[2] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[3] Sachin Chhabra, Hemanth Venkateswara, and Baoxin Li. Patchswap: A regularization technique for vision transformers. In *BMVC*, page 996, 2022.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[6] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[10] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[11] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pages 5907–5915. PMLR, 2019.

[12] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[13] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[14] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6022–6031. IEEE, 2019. doi: 10.1109/ICCV.2019.00612. URL https://doi.org/10.1109/ICCV.2019.00612.

[15] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.