PatchSwap: A Regularization Technique for Vision Transformers

Sachin Chhabra[1]
schhabr6@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

Abstract

Vision Transformers have recently gained popularity due to their superior performance on visual computing tasks. However, this performance is based on training with huge datasets, and maintaining the performance on small datasets remains a challenge. Regularization helps to alleviate the overfitting issue that is common when dealing with small datasets. Most existing regularization techniques are designed keeping ConvNets in mind. As Vision Transformers process images differently, there is a need for new regularization techniques crafted for them. In this paper, we propose a regularization called PatchSwap, which interchanges the patches between two images, resulting in a new input for regularizing the transformer. Our extensive experiments showcase that PatchSwap yields superior performance than existing state-of-the-art methods. Further, the simplicity of PatchSwap makes a straightforward extension to a semi-supervised setting with minimal effort.

1 Introduction

Transformers were originally designed for natural language processing [28] but their application to other domains is rapidly gaining traction [1, 16]. In computer vision, Convolution neural networks (ConvNets) have been the traditional choice of deep learning framework for image recognition task for almost a whole decade [3, 4, 12, 17, 22, 25]. However, in 2020, Vision Transformers (ViT) has created a new benchmark by outperforming ConvNets [9] on ImageNet dataset. Nevertheless, this is the case only when there is abundant training data available. Multiple attempts are being made to adapt a Vision Transformer to small datasets by modifying the transformer architecture [11, 14, 20], distillation [27], etc.

The major challenge when dealing with small datasets is that a Vision Transformer often overfits and results in poor generalization. To combat overfitting, commonly used regularization solutions are dropout [23], weight decay [10], label smoothing [24], batch normalization [15], data augmentations [5, 6]. Other advanced augmentation techniques like Mixup [32], Cutmix [31] create intermediate images by combining multiple images. All of these have become a staple part of training ConvNets as well as Vision Transformers [11, 13]. Although

Mixup and Cutmix work well for both these network types, they were originally designed for ConvNets, which raises the question - can we design data augmentation specialized for Vision Transformers to boost their performance? Both the network types take image input and predict its label but differ in the way they process the images. ConvNets process an image spatially like a grid and uses kernels to extract features whereas Vision Transformer divides the image into fixed size patches and uses self-attention mechanism. ConvNets have constraints of spatial equivariance inbuilt into them which is essential for modeling vision data. On the other hand, Vision Transformers do not have such constraints and must learn to model spatial equivariance from large amounts of data [9].

Keeping Vision Transformers in mind, we propose PatchSwap, a simple yet novel data augmentation technique that interchanges the patches of images to increase the amount of training data and thereby regularizes the performance. PatchSwap shares multiple similarities with Mixup and Cutmix, including (i) preventing overfitting by regularizing the network by mixing images and labels; (ii) linearly interpolating the image consistently within the label space. However, Mixup and Cutmix do not fully utilize the global receptive field of Vision Transformers. ConvNets grow their receptive field with depth whereas a Vision Transformer can learn to interact between any pair of pixels from the beginning at the input layer encoders [9]. Hence, the related patches can be any where in the image and Vision Transformer extract relevant information from them. Based on this understanding, our approach divides two images into patches and then randomly swaps patches between them to create a PatchSwap image (Figure 1). Similar to Cutmix, PatchSwap images contain regions from both classes, but the objects are scattered throughout the PatchSwap image, and the Vision Transformer is trained to predict the objects with their mixing ratios.

In this paper, we showcase PatchSwap as an effective regularization technique for Vision Transformers. It outperforms state-of-the-art methods for datasets like CIFAR-10 and CIFAR-100. We also show that PatchSwap not only regularizes effectively but can also be utilized with unlabeled data (extending to a semi-supervised learning setting). Most of the existing semi-supervised techniques are based on consistency regularization, where a network is trained to produce the same output for two versions of an input image. Unsupervised PatchSwap works on the same principle. Since the PatchSwap images contain a mix of objects from different images, the consistency regularization between the original and the PatchSwap image cannot be implemented. However, if we create two different PatchSwap images of two inputs and ensure that their mixing coefficients are the same, we can train the Vision Transformer to produce consistent outputs for these PatchSwap images. In essence, unsupervised PatchSwap applies the consistency regularization between two PatchSwap images.

The organization of our paper is as follows: Section 2 discusses the work related to our approach. Section 3 presents our proposed approach in supervised and unsupervised settings. Section 4 discusses our experiments and its comparison with the baseline approaches. Section 5 consists of the analysis of our approach and Section 6 outlines the conclusions.

2 Related Work

2.1 Vision Transformer

Images have a continuous grid-like structure while the transformers require sequential series data as input, making them incompatible initially. However, [9] fixed this issue and intro-

duced Vision Transformers to the world. They split images into 16×16 square patches and flatten them into a vector to form series-like input data. Vision Transformers processes each patch using a fully connected layer to learn its embedding. A learnable or a fixed (sinusoidal) positional embedding is added to this feature embedding at the input level to provide spatial information. Transformers project the embeddings into queries, keys, and values and compute self-attention between the patches. Each layer of the transformer consists of a self-attention block, fully-connected layers followed by a normalization layer. The overall architecture of the Vision Transformer is similar to a BERT encoder [7]. Vision Transformer also uses a learnable classification token which is concatenated to input patches. This token is considered to represent the content of the entire image while the patches contain the local spatial information. At the output layer, a classification token is used to classify the input.

2.2 Regularization

When the size of training data is not large enough for a network, it tends to overfit and generalize poorly on unseen data. Several regularization techniques like dropout [23], label smoothing [24], and various data augmentations have been proposed in the past to alleviate this problem. Most of these techniques prevent high confidence predictions on samples. Label smoothing divides a pre-defined probability evenly among all the classes to form a smooth probability vector instead of a one-hot vector for training the network [24]. Cutout is another regularization technique inspired by dropout [8]. It randomly removes a portion of the image and makes the network focus on other parts of the image. This ensures that the overall image is considered while making a prediction instead of just a small portion of it.

Some data augmentation techniques combine multiple inputs to create a new input for training. Mixup is a technique that combines two random samples using $x_{mixup} = \lambda x_a + (1-\lambda)x_b$ where x_a and x_b are two input images and $\lambda \in [0,1]$ is their mixing ratio [32]. The network is trained to linearly interpolate the predictions according to the input. Similarly, Cutmix uses a binary mask M on an image to stitch portions of two images together using $x_{cutmix} = M \cdot x_a + (1-M) \cdot x_b$ [31]. This results in aesthetically better images and higher performance as well.

2.3 Semi-supervised Learning

Semi-supervised learning techniques aim to utilize unlabeled data along with the labeled data for better generalization. A popular semi-supervised learning technique, Pseudo-label, utilizes the network prediction as the ground truth if the confidence is above a certain threshold [19]. Other techniques use a constraint on unlabeled data during training in such a way that it does not require its labels. Π-model proposed that a network should produce consistent outputs despite small changes in the network or the input [18]. This was achieved by reducing the mean-squared error between the outputs obtained by passing either an input twice through a network with stochasticity like dropout or by augmenting an image to create its different versions. MeanTeacher showcased that a teacher network trained with exponential moving weights average provides better targets for unlabeled data [26]. Mixup along with consistency regularization was used in MixMatch [2]. Consistency regularization between a weak and a strong augmentation was proposed in [30].

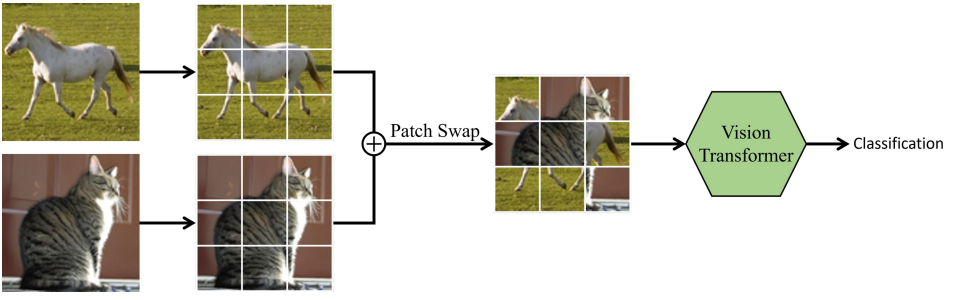

Figure 1: Overview of the PatchSwap technique. Images are divided into patches and the patches between two images are swapped (without changing their relative positions) to create a PatchSwap image.

3 Proposed Approach - PatchSwap

3.1 Regularization

PatchSwap is a simple regularization technique tailor-made for Vision Transformers. It combines two input images and swaps patches between them to produce a PatchSwap image. The PatchSwap image is then used to train a Vision Transformer to predict the mixing ratio as well as the categories of the original images.

Let $x_a, x_b \in [0,1]^{C\times H\times W}$ be two input images, where C, H, W are the number of channels, the height and the width of the images, respectively. Let y_a and y_b be their respective labels. Given a patch size P, we divide the images into patches of equal size I_a and I_b, where $I_a = [x_a^1, x_a^2, \ldots, x_a^N]$, $I_b = [x_b^1, x_b^2, \ldots, x_b^N]$, and $x^i \in [0,1]^{C\times P\times P}$ is the i-th patch of image x. The number of patches is $N = \frac{H}{P} \times \frac{W}{P}$, where P is a factor of H and W to ensure that N is an integer.

PatchSwap generates a new image x_{ps} using the patches I_a and I_b and a mixing ratio λ. We sample the mixing ratio λ from a Beta distribution $\lambda \sim Beta(\alpha, \alpha)$, where α is a constant that defines the Beta distribution. λ is converted to a discrete value $\lambda' \in \{0, 1, \ldots, N\}$, where $\lambda' = \texttt{round}(\lambda.N)$ to estimate the number of patches to be mixed. We generate a random binary mask $M = [M^1, M^2, \ldots, M^N] \in \{0,1\}^N$ where $M^i = 0$ indicates the i-th patch is not selected and $M^i = 1$ indicates the patch is selected in the mix and $\lambda' = \texttt{sum}(M)$. We mix the patches from the two images to generate a PatchSwap image x_{ps},

$$x_{ps}(x_a; x_b; \lambda) = I_a \cdot M + I_b \cdot (1 - M), \tag{1}$$

where '$\cdot$' denotes the element-wise multiplication between a patch x^i and the corresponding mask element M^i. The '$+$' is overridden to denote a mixing operation combining the two images. Figure 1 displays a Patchswap between two images with a mixing ratio $\lambda = 0.4$ and $N = 9$.

The PatchSwap image x_{ps} has image components from x_a and x_b. We use a cross-entropy loss to train a Vision Transformer to predict both the labels y_a and y_b. The loss term is given as,

$$\mathcal{L}_{ps}(x_a; y_a; x_b; y_b; \lambda) = \lambda \mathcal{L}_{ce}(f(x_{ps}; \theta_t), y_a) + (1 - \lambda)\mathcal{L}_{ce}(f(x_{ps}; \theta_t), y_b), \tag{2}$$

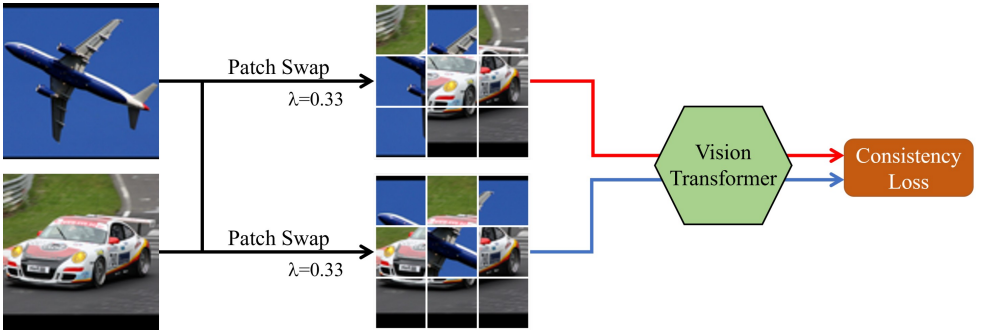

Figure 2: Overview of the Unsupervised PatchSwap regularization for semi-supervised learning. Two images are combined to create two distinct versions of PatchSwap images with same mixing coefficients. Vision Transformer is trained to produce consistent output for both the inputs.

where $\mathcal{L}_{ce}$ represents the standard cross-entropy loss, $f(x;\theta_t)$ is the output prediction for image x for the Vision Transformer with parameters θ_t.

3.2 Unsupervised PatchSwap Applied to Semi-supervised Learning

PatchSwap is a simple regularization technique for labeled data. However, it can also be used for unlabeled data, extending it to semi-supervised learning applications. Popular semi-supervised learning methods are based on consistency regularization [2, 18, 26]. Consistency regularization states that two distinct versions of the same input should give consistent results. Two distinct versions can be generated by either variations in the network, like Dropout or by modifying the input in two different ways. The network is trained to output the same predictions for the two distinct inputs. Standard loss functions like mean-squared error, Kullback-Leibler-divergence, etc., are used to guide the training.

Unsupervised PatchSwap is inspired by the above principle. Given two unlabeled images x_1 and x_2, we generate two patch swap masks M_1 and M_2 using the same mixing ratio $\lambda \sim Beta(\alpha,\alpha)$. We ensure $M_1 \neq M_2$. Using x_1, x_2 and M_1, we generate the PatchSwap image x_{ps_1}. Similarly, with x_1, x_2 and M_2, we generate the 2nd PatchSwap image x_{ps_2}. Since M_1 and M_2 are generated using the same mixing ratio λ, the ratio of the number of patches from x_1 and x_2 is identical in x_{ps_1} and x_{ps_2}, even though the same patches are not swapped since $M_1 \neq M_2$. This ensures x_{ps_1} and x_{ps_2} are different. This is illustrated in Figure 2 where two different images are generated with a mixing ratio $\lambda = 0.33$. We want to train a Vision Transformer $f(.;\theta_t)$ that generates the same output for x_{ps_1} and x_{ps_2} given that their swapping ratios are identical. We define an unsupervised loss to enforce this consistency regularization using,

$$\mathcal{L}_{cr}(x_1;x_2;\lambda) = ||f(x_{ps_1};\theta_t) - f(x_{ps_2};\theta_t)||^2. \tag{3}$$

In the semi-supervised context, we have a labeled pool of data D_l and an unlabeled pool of data D_u. We apply PatchSwap regularization loss on the labeled data D_l and unsupervised PatchSwap on the unlabeled data D_u. The final equation for the semi-supervised training

Dataset	CIFAR-10			FashionMNIST			SVHN		
Patch Size	4	8	16	4	8	16	4	8	16
Cross Entropy	83.3	78.3	69.8	92.1	92.8	91.2	96.4	94.7	92.7
Label smoothing [24]	83.0	79.0	69.6	92.0	92.9	91.5	96.5	94.8	92.8
Cutout [8]	84.0	79.2	70.1	94.2	93.5	91.4	96.8	96.2	94.5
Mixup [32]	87.4	82.3	74.3	93.0	93.4	92.2	97.0	95.7	94.2
Cutmix [31]	88.0	82.7	73.8	94.0	93.8	92.5	96.9	96.2	**94.8**
PatchSwap	**88.3**	**84.7**	**74.9**	**94.4**	**93.9**	**92.6**	**97.2**	**96.8**	**94.8**

Table 1: Comparison of Top-1 classification accuracies on CIFAR-10, FashionMNIST and SVHN datasets using different patch sizes. **Bold** numbers represent the highest accuracy.

loss is,

$$\mathop{\mathbb{E}}_{(x_a,y_a),(x_b,y_b)\sim D_l} \mathcal{L}_{ps}(x_a;y_a;x_b;y_b,\lambda) \quad + \quad \gamma \mathop{\mathbb{E}}_{x_1,x_2\sim D_u} \mathcal{L}_{cr}(x_1;x_2;\lambda), \tag{4}$$

where γ is a hyper-parameter that balances the two loss components.

4 Experiments

4.1 Regularization

4.1.1 Datasets

To assess the performance of PatchSwap, we test it on various datasets: CIFAR-10, CIFAR-100, SVHN [21], FashionMNIST [29] and Tiny-ImageNet as these datasets represent different types of images. Training Vision Transformers requires a huge amount of data and robust regularization [9]. However, the chosen datasets are tiny and we apply standard augmentation techniques during training. For CIFAR-10 and CIFAR-100, we use a random-crop with zero padding of 4 and a horizontal flip with a probability of 0.5. Tiny-ImageNet is a subset of Imagenet with 200 classes and image size of 64×64 pixels. We use the same augmentations as that of CIFAR datasets for it. We also test Tiny-Imagenet images with RandAugment augmentation (strong augmentation) [5]. FashionMNIST consists of grayscale images which we resize to 32×32 pixels. We use a random-crop with zero padding of 2 and a random horizontal flip as the augmentations. For SVHN, we resize the images to 32×32 pixels and use a random-crop with zero padding of 2. We also evaluate the proposed method under different augmentations and present those results in the supplementary material.

4.1.2 Training Details

For our experiments, we use ViT-Lite from [11] which is a scaled-down version of the original Vision Transformer. Specifically, we use 6 encoder blocks with 256 hidden dimension size and 0.1 dropout. The forward expansion layer is set to 512 and the number of attention heads is reduced to 4. This results in about 3.7 million parameters as compared to 86 million in the original Vision Transformer. We train the Vision Transformer from scratch. Due to the absence of results for baseline approaches, we use the official code from their respective repositories to report the results. We use 8×8 cutout size for CIFAR-100, 16×16 for CIFAR-10 and FashionMNIST, 20×20 for SVHN and 32×32 for TinyImagenet. Cutmix is

Patch Size	4		8		16	
Method	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Cross Entropy	57.9	81.5	50.6	76.2	39.3	65.0
Label smoothing [24]	58.3	77.0	51.5	71.7	39.8	62.3
Cutout [8]	57.0	81.1	50.2	76.1	39.1	64.7
Mixup [32]	63.5	85.0	56.8	80.0	45.3	70.6
Cutmix [31]	63.7	85.2	57.0	80.4	44.2	69.5
PatchSwap	**64.9**	**86.4**	**58.5**	**82.5**	**45.7**	**71.6**

Table 2: Comparison of Top-1 and Top-5 classification accuracies on CIFAR-100 dataset using different patch sizes for a Vision Transformer. **Bold** numbers represent the highest accuracy.

Augmentation	Standard				RandAugment [5]			
Patch Size	8		16		8		16	
Method	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Cross Entropy	41.9	65.2	34.4	57.7	46.2	70.4	39.1	63.4
Label smoothing [24]	42.8	63.0	34.6	56.5	47.0	69.7	39.3	62.7
Cutout [8]	42.8	66.6	33.8	58.1	47.5	71.5	40.2	65.1
Mixup [32]	46.6	69.0	38.5	62.4	49.9	73.5	43.2	67.5
Cutmix [31]	48.4	71.6	39.5	63.5	48.4	74.9	44.0	68.0
PatchSwap	**49.9**	**73.4**	**41.8**	**66.3**	**52.8**	**77.0**	**45.6**	**70.8**

Table 3: Comparison of Top-1 and Top-5 classification accuracies on TinyImagenet dataset with standard (Random crop with padding and Random horizontal flip) and RandAugment agumentations using different patch sizes for a Vision Transformer. **Bold** numbers represent the highest accuracy.

applied with 0.5 probability. For label smoothing, ε is always set to 0.1 unless specified. We set $\alpha = 1.0$ for all the experiments.

ViT-Lite uses a smaller patch size than the original Vision Transformer. We performed our experiments with 4, 8, and 16 patch sizes, and the best performance was observed with a patch size of 4. We report the results for 4, 8, and 16 patch sizes, except for Tiny-Imagenet where we use only 8 and 16 due to computation overhead. A smaller patch size increases the number of patches and in turn, increases the data available to the network for training but it also increases the computation quadratically. All experiments including the baselines follow the same training procedure. We train the network for 300 epochs with a batch size of 128 and 0.03 weight decay. We use a learning rate of 5×10^{-4} which is warmed up for the first 10 epochs and then decayed per epoch using a cosine schedule. The code for PatchSwap is available at: https://github.com/s-chh/PatchSwap

4.1.3 Results

We compare our approach with state-of-the-art regularization techniques - label smoothing, Cutout, Mixup, and Cutmix. The results for CIFAR-10, SVHN and FashionMNIST are in Table 1, for CIFAR-100 in Table 2, and for Tiny-Imagenet in Table 3. Our approach outperforms all the baselines for all patch sizes. PatchSwap gains approximately 1.5% and 2.5% over Cutmix and Mixup respectively, and about 9% over the standard cross-entropy

Dataset	CIFAR-10			SVHN		
Patch Size	4	8	16	4	8	16
Cross Entropy	56.0	53.5	45.4	87.9	86.7	76.0
Pseudo Label [19]	58.1	54.0	46.3	91.2	88.9	78.0
MeanTeacher [26]	62.6	56.5	48.2	96.2	95.1	90.1
PatchSwap (Labeled only)	63.2	60.6	51.3	89.7	87.2	81.0
PatchSwap (Full)	**67.6**	**62.9**	**54.2**	**96.4**	**96.7**	**90.9**

Table 4: Classification accuracies on CIFAR-10 and SVHN datasets in a semi-supervised setting. **Bold** numbers represent the highest accuracy.

loss. In addition, PatchSwap outperforms RandAugment augmentation for Tiny-Imagenet. Combining RandAugment with PatchSwap further boosts its performance over the baseline approaches.

4.2 Semi-supervised Learning

We perform semi-supervised learning experiments on CIFAR-10 and SVHN [21] using 4000 labeled training samples and all the training samples in the unlabeled set. Pseudo-label training uses a threshold of 0.9 probability [19]. MeanTeacher uses a teacher network with an exponential moving average of the student network to generate output targets for the unlabeled data [26]. We used two augmented versions of the inputs - the first one is used for generating the output targets using the teacher network and the other one is used to train the student network.

Our approach also utilizes the exponential moving average similar to [26]. However, it does not require multiple augmented versions of an image. The consistency regularization is imposed on the two patch versions of the same image. The teacher network is used to generate targets from the first PatchSwap image and the student network is trained to match outputs using the second PatchSwap image. The γ is set to 100 based on [26] for all the experiments. The unlabeled loss is linearly increased over the first 10 epochs. The rest of the setup for semi-supervised learning experiments is the same as regularization experiments.

The results for these experiments are shown in Table 4. We also showcase the results of training with just the labeled data - PatchSwap (Labeled only). PatchSwap (Full) combines PatchSwap and Unsupervised PatchSwap. Our approach outperforms the baselines methods. The PatchSwap with labeled loss outperforms MeanTeacher on CIFAR-10 only and unsupervised PatchSwap provides additional gain.

5 Analysis

5.1 Regularization Intensity

The hyperparameter, α controls the regularization intensity of PatchSwap. The mixing coefficient generated by the Beta distribution is rounded to the closest multiple of $\frac{1}{N}$ where N is the number of patches, as it can take discrete values only. A small value of α in Beta distribution generates values close to 0 or 1 and due to rounding, the mixing coefficient will end up being 0 or 1. This will result in PatchSwap reducing down to cross-entropy loss.

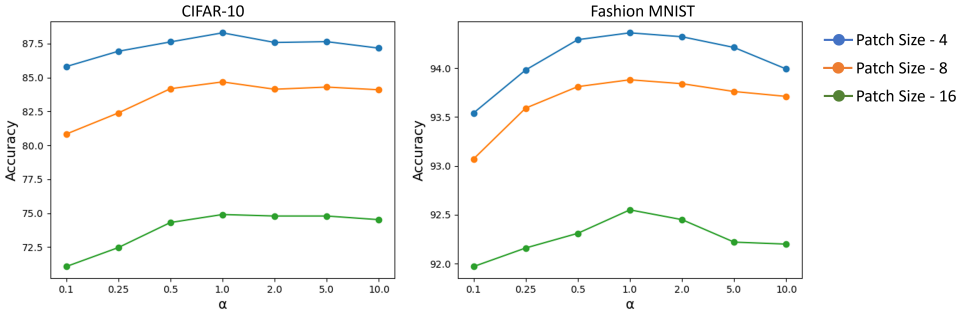

Figure 3: Impact of α on the CIFAR-10 (left) and FashionMNIST (right) datasets. Different patch sizes used for training the transformers are denoted by different colors.

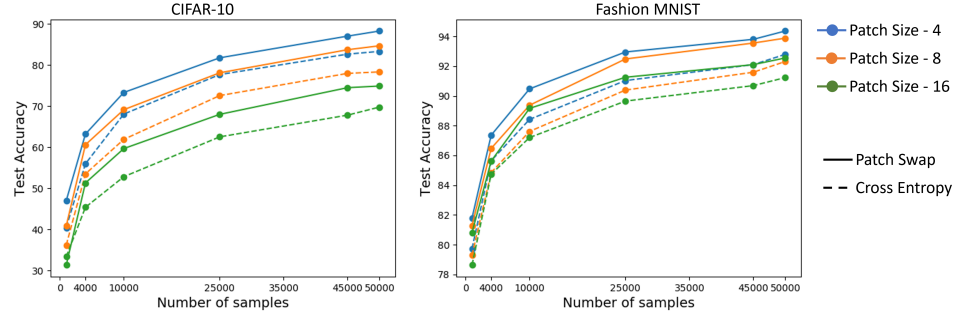

Figure 4: Comparison of Number of samples vs Test Accuracy for CIFAR-10 and FashionMNIST dataset. Different patch sizes used for training the Vision Transformers are denoted by different colors.

Similarly, a high value of α will result in mixing ratio of 0.5. Thus, α parameter handles the balance between cross-entropy and regularization in such a way.

We experiment with different values of α on CIFAR-10 and FashionMNIST datasets. The results are displayed in Figure 3. As expected, a low value of α results in significantly poor performance (close to cross-entropy loss). Also, a high value of α leads to decrease in the performance. $\alpha = 1$ (as used in all our experiments) results in a uniform distribution and functions as cross-entropy loss with $\frac{1}{N}$ probability.

5.2 Number of Samples

In this section, we reduce the available training data to assess the strength of our regularization. We perform these experiments with CIFAR-10 and FashionMNIST datasets and the results are shown in Figure 4. We set the available number of training samples to 1000, 4000, 10000, 25000, 45000, full set, and report the test accuracy. We compare the results with the standard cross-entropy loss (shown using dashed lines) for various patch sizes. PatchSwap with only 10,000 CIFAR-10 labeled training samples achieves performance equivalent to supervised training with 25,000 samples. Similarly, for FashionMNIST, PatchSwap signifi-

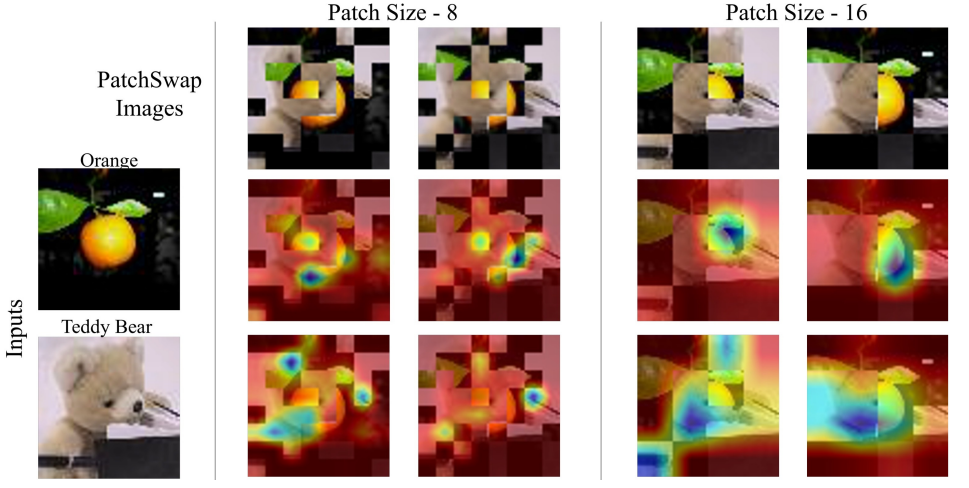

Figure 5: Class-specific Attention Maps for PatchSwap images. The first column shows the input images. We generate PatchSwap images for two different patch sizes - 8 (2^{nd} and 3^{rd} column) and 16 (4^{th} and 5^{th} column). The 2^{nd} row displays the attention map for Orange and the last row shows for Teddy Bear.

cantly reduces the amount of labeled data required for training.

5.3 Attention Maps

We visualize class-specific attention maps for the PatchSwap images in Figure 5. We can observe that the network has learned to focus on the correct patches for classification. For example, for the orange class, the network focuses on the patches in the middle (2^{nd} row) where most of the orange is in the original and PatchSwap images. Similarly, the network focuses on the patches belonging to the Teddy Bear image while classifying it.

6 Conclusions

In this paper, we presented the PatchSwap technique for regularizing Vision Transformers. Our approach swaps image patches between two images to create a regularized input for training. Also, it can be further extended to Unsupervised PatchSwap for semi-supervised applications by applying consistency regularization on two PatchSwap images. Through extensive experiments, we showcase the strength of PatchSwap over existing state-of-the-art techniques on various datasets.

Acknowledgements The work was supported in part by a grant from ONR. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ONR.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Sachin Chhabra, Prabal Bijoy Dutta, Baoxin Li, and Hemanth Venkateswara. Glocal alignment for unsupervised domain adaptation. In *Multimedia Understanding with Less Labeling on Multimedia Understanding with Less Labeling*, pages 45–51. 2021.

[4] Sachin Chhabra, Hemanth Venkateswara, and Baoxin Li. Iterative image translation for unsupervised domain adaptation. In *1st Workshop on Multimedia Understanding with Less Labeling, MULL 2021, co-located with ACM MM 2021*, pages 37–44. Association for Computing Machinery, Inc, 2021.

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[6] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *CoRR*, abs/1805.09501, 2018. URL http://arxiv.org/abs/1805.09501.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

[8] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017. URL http://arxiv.org/abs/1708.04552.

[9] 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.

[10] Stephen Hanson and Lorien Pratt. Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems*, 1, 1988.

[11] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *CoRR*, abs/2104.05704, 2021. URL https://arxiv.org/abs/2104.05704.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] 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.

[14] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11936–11945, 2021.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[16] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pages 4651–4664. PMLR, 2021.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[18] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BJ6oOfqge.

[19] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, 2013.

[20] 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*, pages 10012–10022, 2021.

[21] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[25] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[26] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[27] 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.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

[30] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

[31] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[32] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.