Enforcing Style Invariance in Patch Localization

Elior Ben Arous [*1] **Dustin Brunner** [*1] **Jonathan Manz** [*1] **Felix Yang** [*1]

Abstract

This project employs style invariance concepts from RELIC on the patch localization pretext task with the goal of attaining style-invariant image embeddings for downstream tasks. We investigate multiple augmentation schemes and analyze the resulting embedding spaces. Our proposed pretext task achieves significant improvements over the standard patch localization method on downstream tasks. The code is available at https://github.com/brunnedu/DeepLearning2022.

1. Introduction

Self-supervised learning (SSL) has grown to be a very successful tool in certain fields of machine learning, especially where large quantities of unlabeled data are easily acquirable. In these areas, the cost of human annotation poses the main bottleneck for upscaling learning datasets. SSL circumvents this issue by extracting relevant information from the data through so-called pretext or proxy tasks where labels are generated in an automated manner. In (Mitrovic et al., 2020), the authors introduce a theoretical framework for SSL applied to contrastive learning pretext tasks. Their work tries to explain the success of mutual information methods in particular.

In this project, we adopt their style invariance toolkit and apply it to patch localization, as first described in (Doersch et al., 2015). We present a general framework for style invariance in patch localization (SIPL). Specifically, we train three different models, each exploring a different way of modifying patch localization to utilize style invariance. We then evaluate the quality of our learned embeddings by training a shallow head on the downstream task of image classification. Our SIPL approach significantly outperforms a baseline implementation of the original patch localization method (OPL) on said downstream task. Finally, we discuss some possible reasons for this performance increase and the implications we see for some other ideas from (Mitrovic et al., 2020).

[*]Equal contribution [1]Department of Computer Science, ETH Zürich, Switzerland. Correspondence to: Elior Ben Arous <ebenarous@student.ethz.ch>, Dustin Brunner <brunnedu@student.ethz.ch>, Jonathan Manz <manzjo@student.ethz.ch>, Felix Yang <fyang@student.ethz.ch>.

Deep Learning, ETH Zürich, 2022

2. Models and Methods

We will first briefly explain the key concepts we adopt from (Doersch et al., 2015) and (Mitrovic et al., 2020) because these papers are the basis of this project. Based on that, we will explain our approach and explain the details of our process.

2.1. Patch Localization

We adopt the patch localization pretext task as presented in (Doersch et al., 2015) for our model. The target of this task is to predict the relative position of two patches sampled from the same image. Specifically, the model is handed a center patch as well as one of the eight possible neighbors randomly sampled from a 3×3 grid surrounding the center patch. The label is an index from 0 to 7 representing the relative position of the neighboring patch. To prevent the algorithm from learning trivial shortcuts, some additional precautions must be taken (Doersch et al., 2015). We discuss these in section 2.5.

2.2. RELIC

RELIC (Representation Learning via Invariant Causal Mechanisms) is a theoretical framework that attempts to explain the success of mutual information targets in contrastive learning (Mitrovic et al., 2020). To do this, they employ a causal framework wherein images are generated from two independent latent variables: their style and their content. Furthermore, the assumption is that all relevant downstream tasks would only depend on the content variable and thus be invariant to changes in the style of an image. Of course, the assumed style and content variables are not known a priori and the style is implicitly defined by a set of image augmentations. Applying these augmentations is thus implicitly understood as varying the style variable of the image while leaving the content variable unchanged. This style invariance is then applied as a divergence regularizer

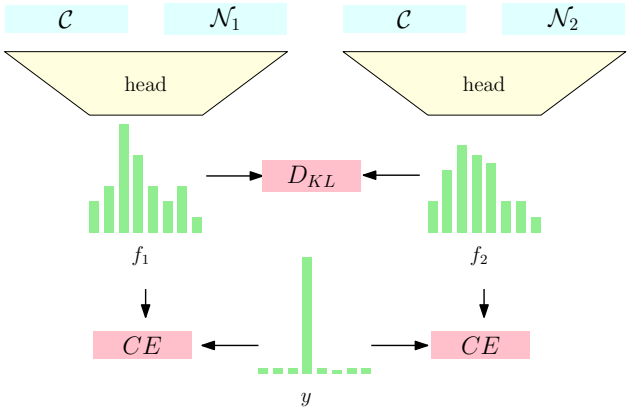

Figure 1. The architecture of our pretext task. $\mathcal{C}$ and $\mathcal{N}$ are the embeddings of center and neighboring patches respectively. D_{KL} is the symmetrized Kullback-Leibler divergence and CE is the cross-entropy loss with respect to the true label y.

between two differently augmented instances of the same image in contrastive learning. (Mitrovic et al., 2020) also introduce the concept of refinements as a tool to reason about transfer learning to downstream tasks, which will be relevant to the discussion of our results.

2.3. Proposed Idea

In this project, we use the causal framework from RELIC to modify the patch localization pretext task. We generate two different patch localization tasks from the same two patches through an augmentation scheme described in section 2.5. By applying a softmax to our outputs, we can treat them as probability distributions, on which we can use divergence measures.

2.4. Architecture

We split our network into separate "encoder" and "head" networks. A ResNet-18 is used for encoder and a multi-layer perceptron with a single hidden layer is used as the head to avoid dimensional collapse (Jing et al., 2021). We generate our patch embeddings $\mathcal{C}$ and $\mathcal{N}$ from the center and neighboring patches respectively using encoder, as shown in Figure 2. A pair of embeddings is concatenated prior to being fed into head, as shown in Figure 1. The output of the head is softmaxed to produce probability distributions f. The loss is calculated as follows:

$$\mathcal{L} = CE(f_1, y) + CE(f_2, y) + \alpha\, D_{KL}(f_1, f_2) \quad (1)$$

where CE is the cross-entropy loss to the true label y, $D_{KL}(f_1, f_2) := D_{KL}(f_1 \parallel f_2) + D_{KL}(f_2 \parallel f_1)$ is the symmetrized Kullback-Leibler divergence, and α is a hyperparameter used to weigh the style invariance condition against the patch localization task.

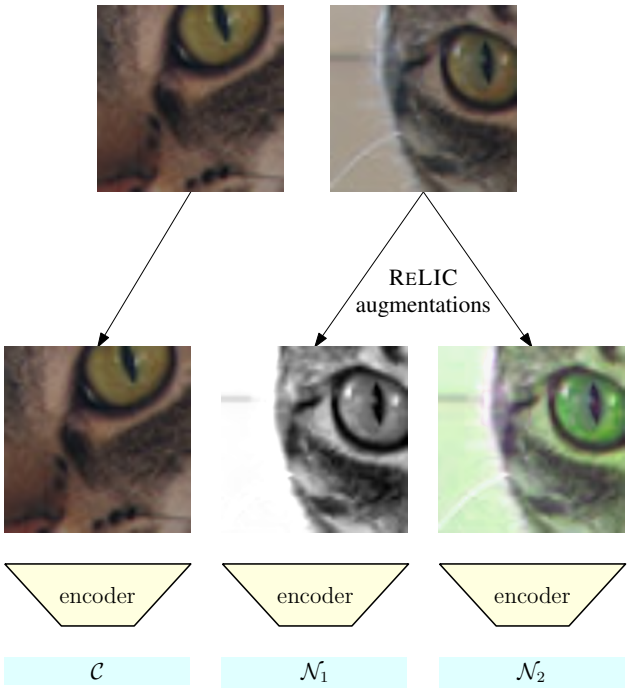

Figure 2. The augmentation scheme of SIPLV1 leaves the center patch unaltered and generates two localization problems by applying two varying augmentations to the outer patch. Feeding a patch to the encoder generates an embedding.

2.5. Augmentations

In order to generate the patches, which we hand to encoder as seen in Figure 2, we employ an augmentation pipeline shown in Algorithm 1. To make better use of smaller-scale images, we sample patches by cutting the image into a 3×3 grid. We crop each patch down to 56×56 pixels randomly to ensure a gap between the center and neighboring patches. This is lifted from (Doersch et al., 2015) and should prevent the use of trivial patterns, such as continued textures, for patch localization. If a patch should be augmented with RELIC-like augmentations, these transforms are applied now. The transforms used are adopted from (Mitrovic et al., 2020) with hand-tuned parameters as listed in Table 2. However, we discarded random horizontal flips because, unlike RELIC 's pretext task, patch localization is not invariant to them.

Finally, it is shown in (Doersch et al., 2015) that neural networks can learn to exploit chromatic aberration artifacts to locate an image patch globally. Chromatic aberration occurs in color photography due to the difference in refraction between different wavelengths of light. The aforementioned problem can be addressed by projecting patches onto the green-magenta color axis and subtracting the projection out to make hue differences along that axis disappear (Doersch et al., 2015). We adopt this approach and apply it to all patches prior to embedding them.

Algorithm 1 Augmentation Pipeline

Input: image
patches ← center(image)
patches ← randomNeighbor(image)
for all patches **do**
  Crop and random jitter.
  Resize to 224×224.
  if patch is RELIC **then**
    Random resized crop
    Color jitter
    Random grayscale
    Gaussian blur
    Random solarize
  end if
  Color projection
  Normalization
end for

Table 1. Common architecture details for the pretext training. For the downstream task, everything remains the same, except for the head's hidden layer size which becomes 1000.

INPUT SIZE	$224 \times 224 \times 3$
encoder ARCHITECTURE	RESNET-18[1]
EMBEDDING SIZE	1000
head HIDDEN LAYER SIZE	4096
head ACTIVATION	RELU

3. Results

3.1. Experimental Setup

We train four different models for evaluation, each following a unique augmentation scheme. The first is a reimplementation of OPL, which does not involve RELIC augmentations. We then have SIPLV1, where only neighbors are augmented as seen in Figure 2. We'll write this as $\langle c, A_1(n)\rangle, \langle c, A_2(n)\rangle$. Here, c and n denote the center and neighboring patches respectively, and $A(\cdot)$ are different augmentations. In SIPLV2, the center and neighbor patches are augmented in pairs with the same RELIC transforms applied to both, i.e. $\langle A_1(c), A_1(n)\rangle, \langle A_2(c), A_2(n)\rangle$. In SIPLV3, we augment the center patches and neighbor patches differently, specifically $\langle A_1(c), A_2(n)\rangle, \langle A_1(c), A_3(n)\rangle$.

All models share an identical architecture for both encoder and head, summarized in Table 1. The loss function from Equation 1 is also used for all SIPL models. OPL is trained with only one patch localization problem and uses cross-entropy $CE(f, y)$ as its loss. We train each model on the ILSVRC 2012 (ImageNet) validation set (Deng et al., 2009), excluding non-RGB images. Table 3 reports the selected values of training parameters. We find empirically that SIPLV1 performs best downstream with $\alpha = 5$ and keep it identical for SIPLV2 and SIPLV3. The learning rate, batch size and weight decay are optimized using Optuna (Akiba et al., 2019), a Bayesian hyperparameter tuning library.

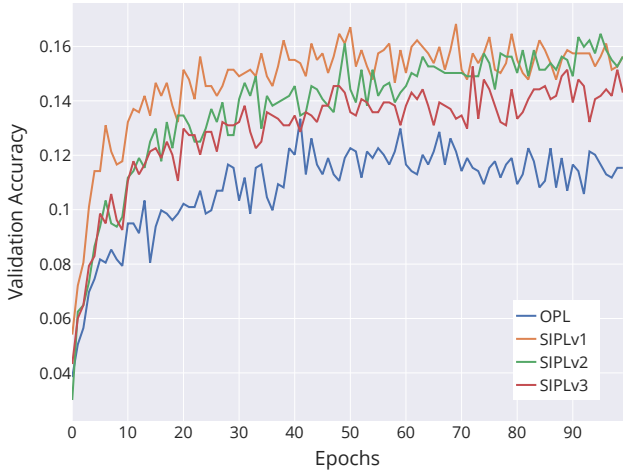

Figure 3. The image classification accuracies of our models on a held-out validation set plotted against the number of epochs spent training on Tiny ImageNet.

3.2. Image Classification

To evaluate the quality of the embeddings generated by our encoder, we test their performance on the Tiny ImageNet validation set (Deng et al., 2009). Again, only images with all 3 color channels are used. Each embedding network has its weights frozen and a shallow multilayer perceptron head with 1000 hidden units is appended to them. The head weights are then trained on 9000 images with 832 held out as a validation set. In Figure 3 we plot the classification accuracies of all four models on the validation set. We see that all models trained with RELIC-like style invariance conditions converge to a higher accuracy than OPL.

4. Discussion

Before comparing the different downstream experiments, we want to point out that all of them performed relatively poorly. This is due to multiple factors. First of all, we solely wanted to evaluate the expressiveness of the raw embeddings generated by encoder. Therefore, we deliberately did not fine-tune encoder during the downstream training, i.e. we froze encoder's weights, and only trained a very simple head with a single hidden layer of 1000 neurons. Secondly, we only trained on the validation set of the Tiny ImageNet dataset to preserve the usual setting of SSL where the pretext dataset is much larger than the downstream dataset.

As shown in Figure 3, our new approach SIPLV1 outperforms the baseline OPL by about 30%. However, it is important to note that in (Doersch et al., 2015), they used a

[1](He et al., 2015)

Figure 4. 50 images from two different classes in two different styles embedded by OPL (left) and SIPLv1 (right). The 1000 dimensional embeddings are projected to 2D using t-SNE.

different downstream task for evaluation, namely object detection instead of image classification. We primarily picked image classification as an evaluation task due to its simplicity. In principle, this should not matter as we evaluate both the baseline and our models on the same task. However, it could be that the style invariance condition applied in our models somehow intrinsically favors classification as a downstream application. Investigating the expressiveness of embeddings generated by an encoder trained on our pretext task for other downstream tasks such as object detection would be interesting future work.

4.1. Embedding Space Analysis

We want to visualize the embedding spaces generated by these different models. For this purpose, 50 images from two different classes each are randomly chosen from Tiny ImageNet. Using only the RELIC transforms listed in Algorithm 1 and normalization, we generate two distinct augmentations and apply each to both sets of images. This yields four sets of 50 images depicting both classes transformed under both augmentations. We perform dimensionality reduction on the embeddings produced by OPL and SIPLv1 by first applying PCA down to 50 dimensions and then t-SNE to 2. This yields the plots seen in Figure 4. Neither model clusters classes seemingly well, however OPL heavily clusters styles together. Contrastingly, SIPLv1 does not have such clustering according to style. In fact, we see that almost all embeddings from SIPLv1 are grouped in pairs of different styles. These are the embeddings of the same image under two different styles, which demonstrates that style invariance in SIPLv1 is indeed achieved in the embedding space, even though it is only enforced at a low dimensional output space.

This result is notable in so far as it is poorly explainable with the concept of refinements employed in (Mitrovic et al., 2020). As patch localization is in general not a refinement of any nontrivial downstream task, this observation also holds for the successful results from section 4. This could draw into question the merit of the concept of refinements for explaining the success of mutual information approaches in self-supervised learning. Of course, this would need to be investigated more rigorously.

5. Summary

Introducing RELIC-like style invariance to the pretext task of patch localization leads to embeddings that facilitate the downstream task of image classification. Though multiple different models were tried, all performed similarly when evaluated. The results we produce in this project are difficult to interpret on a large scale, as it is hard to say how our results will scale to industry-standard datasets. Additionally, enforcing style invariance at the pretext stage may introduce a bias towards our particular choice of downstream evaluation task. It would be interesting to see how our models behave in object detection, as was used by (Doersch et al., 2015) in their original paper.

However, we believe that these results provide an interesting perspective on RELIC. For instance, it is not clear how enforcing style invariance at the output of patch localization, a pretext task with only 8 classes, improves the expressiveness of embeddings for such a fine-grained downstream task as image classification. This connects again to our mention of refinements in section 4.1, which the authors of RELIC used to explain downstream performance. It is thinkable, that the concept of refinements can be expanded to explain this transfer learning, but currently it cannot. It should be interesting to explore exactly when style invariance conditions at the pretext task output lead to improved latent representations.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902, 2019. URL http://arxiv.org/abs/1907.10902.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015. URL http://arxiv.org/abs/1505.05192.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Jing, L., Vincent, P., LeCun, Y., and Tian, Y. Understanding dimensional collapse in contrastive self-supervised learning. *CoRR*, abs/2110.09348, 2021. URL https://arxiv.org/abs/2110.09348.

Mitrovic, J., McWilliams, B., Walker, J. C., Buesing, L., and Blundell, C. Representation learning via invariant causal mechanisms. *CoRR*, abs/2010.07922, 2020. URL https://arxiv.org/abs/2010.07922.

A. Additional Parameter Listings

Table 2. The parameters used for our augmentation pipeline. For random transformations we give the permitted range.

RANDOM RESIZED CROP:	
AREA SCALE	$[0.32, 1]$
ASPECT RATIO	$[\frac{3}{4}, \frac{4}{3}]$
COLOR JITTER:	
BRIGHTNESS RATIO	$[0.2, 1.8]$
CONTRAST RATIO	$[0.2, 1.8]$
SATURATION RATIO	$[0.2, 1.8]$
HUE JITTER	$[-0.2, 0.2]$
GAUSSIAN BLUR:	
KERNEL SIZE	23×23
STANDARD DEVIATION	$[10^{-10}, 0.2]$
RANDOM SOLARIZE:	
THRESHOLD	0.5
PROBABILITY	0.2
RANDOM GRAYSCALE:	
PROBABILITY	0.05
NORMALIZATION:	
MEAN PER CHANNEL	(0.485, 0.456, 0.406)
STANDARD DEVIATION PER CHANNEL	(0.229, 0.224, 0.225)

Table 3. The hyperparameters used to train all four models.

MODEL	α	PRETEXT LEARNING RATE	DOWNSTREAM LEARNING RATE	WEIGHT DECAY	BATCH SIZE	OPTIMIZER
OPL	-	5E-5	1E-4	0	64	ADAM
SIPLV1	5	5E-5	1E-4	0	64	ADAM
SIPLV2	5	5E-5	1E-4	0	64	ADAM
SIPLV3	5	5E-5	1E-4	0	64	ADAM

B. Model Performance Evaluation

Table 4. The best validation accuracies of all four models on the pretext and downstream tasks.

MODEL	PRETEXT ACCURACY (%)	DOWNSTREAM ACCURACY (%)
OPL	**50.5**	13.3
SIPLV1	35.4	**16.8**
SIPLV2	35.9	16.5
SIPLV3	32.0	15.3