# Color-$S^4L$: Self-supervised Semi-supervised Learning with Image Colorization

Anonymous ECCV submission

Paper ID 3156

**Abstract.** This work tackles the problem of semi-supervised image classification task with the integration of several effective self-supervised pretext tasks. Different from widely-used consistency regularization within semi-supervised learning, we explored a novel self-supervised semi-supervised learning framework (**Color-**$S^4L$) especially with image colorization surrogate task and deeply evaluate performances of more various neural architectures in such special pipeline. Also, we demonstrate its effectiveness and optimal performance on CIFAR-10, SVHN and CIFAR-100 datasets in comparison to previous supervised and semi-supervised methods.

**Keywords:** Semi-supervised learning, Recognition, Self-supervised learning, Image Colorization

## 1 Introduction

Vision serves as the most promising way to explore machine learning instances within multiple benchmarks, such as image classification [43], face recognition [45], video segmentation [16]. To address the fundamental weakness of supervised learning that demands for a vast amount of human-labeled data which is costly to collect and scale up, an emerging body of research on semi-supervised learning [35], few-shot learning [13], self-supervised learning [43,16], and transfer learning [25] have dedicated towards learning paradigms that enable machines to recognize novel perception concepts by leveraging limited labels.

Within this effort, semi-supervised learning (SSL) matches our human learning patterns that conduct tasks well after developing compatible concepts by mastering some correct label information. Also, self-supervised visual representation learning has demonstrated the most promising results [43,42,14,47] on tough computer vision tasks. In general, the destination of self-supervised learning is utilizing pretext tasks (i.e., Image Rotation [8], Geometric Transformation [11] to learn intermediate semantic or structural features from large-scale unlabeled data and then transfer the representation to diversified downstream tasks like image classification [43,14,47], video segmentation [16,39], and keypoint tracking [16,39] effectively. Inspired by the reasonable results of semi-supervised learning with self-supervised regularization [37], our work focuses on such novel but challenging topic, even explores the ingenious combination of various self-supervised pretext tasks with semi-supervised framework and design an effective algorithm to tackle the issue for semi-supervised image classifiers.

In this work, we further investigate multiple pretext tasks (e.g., image rotation [8], image colorization [49,18,2], geometric transformation [8,11]) as self-supervised regularization for semi-supervised learning. Unlike [37], our research speculated that the classification-based self-supervised tasks (e.g., predicting image transformations and image rotation) may fail to capture semantic features like image colors and textures. Thus, we specifically employ the popular reconstruction-based pretext task—image colorization [49,2] in addition to the regular classification-based tasks including image rotation [8] and geometric transformation [8,11], then we even excavate the influence of multiple tasks and various neural networks in Color-$S^4L$. In this regard, our algorithm to semi-supervised learning belongs to the approach that creates auxiliary labels from unlabeled data by implementing surrogate self-supervised tasks which serve as targets along with labeled data.

Our key contributions and observations can be summarized as follows:

- First, we proposed a novel self-supervised semi-supervised learning algorithm Color-$S^4L$ especially with the image colorization pretext task to generate surrogate labels on unlabeled data, and experimentally self-trained several well-performed colorization models on CIFAR-10, SVHN and CIFAR-100 datasets.
- Second, we conducted extensive experiments to validate the performance of various neural network trunks like Convolution Neural Networks (ConvNet), Wide Residual Network (WRN), etc. on the SESEMI model [37] and Color-$S^4L$, even observed the outstanding performance of certain architectures.
- Finally, we further dived into our novel Color-$S^4L$ model to evaluate its performance and discovered it really achieves competitive or state-of-the-art predictive performance against previous SSL methods for both supervised and semi-supervised image classification with no additional hyper-parameters to tune.

## 2 Related work

### 2.1 Semi-supervised Learning

**Semi-supervised learning (SSL)** aims to counter the defects of the tedious hand-labeling process in supervised learning and the limited application spectrum of unsupervised learning. Indeed, several state-of-the-art semi-supervised algorithms [43,3,28,38] have approached the performance of supervised baselines based on deep neural networks. The most pervasive technique for SSL is **consistency regularization**, which randomly perturbs the input data by means of dropout [33] or data augmentation [42], then adding consistency regularization losses to measure discrepancy between predictions on the distorted unlabeled data. Initially, consistency regularization was popularized by Oliver et al. [24] and they provide an overview and evaluation of several consistency regularization based approaches including $\prod$ model [28], Pseudo-Ensembles [1], MixMatch [3], Temporal Ensembling [17] and Mean Teacher [34]. Specifically, Pseudo-Ensembles [1] aims to stabilize ensembles of predictions through minimizing the mean squared error over the augmented or randomly perturbed training examples. Mean Teacher [34] utilizes two neural networks — the student and the teacher, then encourages consistency of their individual predictions.

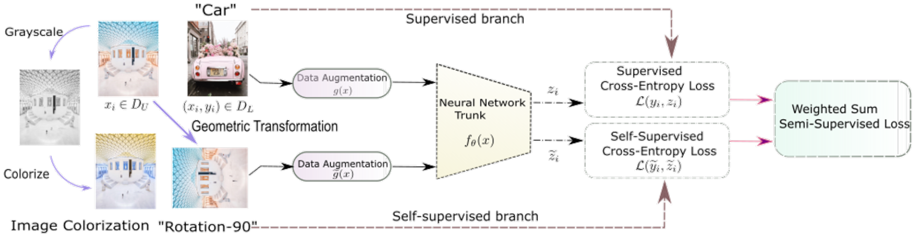


**Fig. 1.** Color-$S^4L$ architecture for semi-supervised image classification. The left of pipeline denotes two kinds of self-supervised pretext tasks which include image colorization and image rotation. Also, we employ **Geometric Transformation** function $g(x)$ to produce 6 proxy labels defined as image rotation in multiples of 90 degrees ($[0°, 90°, 180°, 270°]$) along with horizontal(left-right) and vertical(up-down) flips like [37]. In addition, we especially utilize the special **Image Colorization** function $h(x)$ to create the $7^{th}$ auxiliary label and strengthen the existing self-supervisions on unlabeled data within semi-supervised learning paradigm.

Along side the popular consistency regularization and its smoothing derivations [34,28,17], adversarial training [29,32] represents huge potential to solve the SSL issue. Miyato et al. introduced virtual adversarial training (VAT) [20] mechanism in which adversarial samples serve as perturbations on unlabeled data and create an auxiliary unsupervised loss term. Recently, a powerful method MixMatch [3] manages to unify the current dominant approaches for SSL and guesses low-entropy labels for stochastically augmented unlabeled examples with MixUp [48]. Similarly, the novel Interpolation Consistency Training (ICT) [38] algorithm from Verma et al. combined Mean teacher [34] and applied MixUp on unsupervised branch with consistency regularization. Thus, the integration of multiple learning paradigms represents significant improvements for SSL problems.

## 2.2 Self-supervised learning

**Self-supervised learning** is considered as an unsupervised representation learning paradigm whose common workflow is training a model with one or multiple pretext tasks on unlabeled data itself and extract one intermediate feature layer of this model to fine-tune another one on downstream tasks like action classification [36], object tracking [39,40] or video segmentation [16,39]. To avoid explicit manual supervision, many brilliant pretext tasks [8,49,18] have been proposed on unlabeled images and videos. The popular image-based auxiliary tasks can be classified into two types of self-supervisions: reconstruction-based tasks (e.g., Image Reconstruction [31], Image Colorization [49,18,50,2], Image

Inpainting [26]) and classification-based tasks (Image Rotation [8], Exemplar [6] and Geometric Tansformation [8]). In detail, Gidaris et al. proposed a 4-classification model RotNet [8] to predict the transformation degree of the input image which has been randomly rotated by a multiple of 90 degrees, corresponding to $[0^\circ, 90^\circ, 180^\circ, 270^\circ]$. Also, Exemplar-CNN [6] was designed to generate surrogate unlabeled image patches with random transformations [8] and classify the distorted images into the same class. What's more, image colorization [49,18,2] is another powerful self-supervised auxiliary task to colorize the grayscale input data and map it into a distribution over quantized color values.

A plenty of unlabeled videos grow explosively in our modern life and they represent huge potential in self-supervised learning paradigm [14,47,23,9]. Wang & Gupta [40] came up with an unsupervised visual representation method via tracking dynamic moving objects of videos. Anyway, colorization can also be utilized effectively on videos and obtain extraordinary performance in various benchmarks. Vondrick et al. introduced video colorization [39] to copy colors from a normal colorful reference frame to another target grayscale frame by keeping track of correlated pixels within the video. Furthermore, multi-task fashion of self-supervised learning [44,5] also achieved significant successes in unsupervised representation learning. Within our Color-$S^4L$, we applied multiple auxiliary tasks like image colorization and geometric transformation to exert the ingenuity of the self-supervised learning, which will be introduced at **Section 3.3**.

### 2.3 Combination of two learning pipelines

Nowadays, new research on the combination of self-supervised learning and semi-supervised learning has achieved significant success. Zhai et al. [47] has explored a new algorithm for semi-supervised learning (SSL) by training the model with pretext tasks including image rotation [8] and exemplar [6] on unlabeled images, then feed one intermediate feature layer to a multinomial logistic regression classifier for ImageNet classification task. Also, such $S^4L$ algorithm demonstrates that semi-supervised learning is complementary to the existing self-supervised techniques, even outperforms the carefully-tuned baselines and previous SSL methods [43,1,17]. On the other hand, [37] has introduced a simple and efficient SSL method with a novel self-supervised regularization which trains the model end-to-end for the multi-task learning of both labeled and unlabeled data with no additional hyper-parameters to tune. Inspired by [37], our work creates a novel algorithm Color-$S^4L$ and take the color & textures of unlabeled data into consideration, then utilize the similar evaluation protocol and terminal goal to decrease the error rate of semi-supervised image classification. We also explored more network trunks in Color-$S^4L$ model and observe that the new-designed neural architectures even achieve better results than [37].

In the following section, we will describe our Color-$S^4L$ methodology in § **3**, provide experimental results in § **4**, and we conclude in § **5**.

# 3 Methodology

## 3.1 Overview of Color-$S^4L$ Model

We present Color-$S^4L$, a novel model based on SESEMI in [37] and it belongs to a class of approaches [19,20,1] that produce proxy or auxiliary labels from unlabeled data without manual annotations, which are utilized as credible targets along with labeled samples. The whole design of Color-$S^4L$ is depicted in **Fig. 1**. Consider the common semi-supervised framework, the input of Color-$S^4L$ contains a training set of labeled data-target pairs $(x, y) \in D_L$ and unlabeled inputs $x \in D_U$. Typically, we make the default assumption that the labeled dataset $D_L$ and unlabeled ones $D_U$ are sampled from the same distribution $p(x)$, in which case $D_L$ is a labeled subset of $D_U$. Whereas, in real-world settings it seems infeasible to make our assumption fit well with the potential existence of class-distribution mismatch [24]. Thus, in the model we sample $D_L$ and $D_U$ from consistent distribution in identical datasets like CIFAR-10 [15], CIFAR-100 [41] and SVHN [21].

As for the training phase, the whole framework is represented in **Fig. 1**. Obviously, we split the input data into two mini-batches of training sets with the same number of examples at each training step, one is the labeled input-target pairs $(x, y) \in D_L$ and another contains unlabeled inputs $x \in D_U$. The objective of Color-$S^4L$ is training a prediction function $f_\theta(x)$ which is parameterized by $\theta$, where the combination of $D_L$ and $D_U$ may benefit the model to achieve significantly better prediction performance than the single utilization of $D_L$. Therefore, we input two training samples separately in each supervised branch and self-supervised branch with a shared CNN backbone $f_\theta(x)$. Since there is a large amount of unlabeled data in the self-supervised line, labeled examples will represent a mini-batch repeatedly in Color-$S^4L$. Also, we operate forward-propagation on $f_\theta(x)$ both in the labeled branch $x_i \in D_L$ and the unlabeled one $x_i \in D_U$, resulting in softmax prediction vectors $z_i$ and $\tilde{z}_i$. Next we compute the Color-$S^4L$'s loss function both with the supervised cross-entropy loss $L_{super}(y_i, z_i)$ applying ground truth labels $y_i$ and the self-supervised cross-entropy loss $L_{self}(\tilde{y}_i, \tilde{z}_i)$ using proxy labels $\tilde{y}_i$ generated from image colorization and geometric transformation pretext tasks. Following [37] the parameters $\theta$ can be learned via backpropagation technique by minimizing the multi-task Color-$S^4L$ objective function defined as the weighted sum semi-supervised loss:

$$L_{Color-S^4L} = L_{super}(y_i, z_i) + \omega * L_{self}(\tilde{y}_i, \tilde{z}_i) \tag{1}$$

In [37] the formulation of SESEMI's objective function treats the self-supervised loss as a regularization term, whereas we don't deliberately consider the self-supervised cross-entropy loss as the self-supervised regularization and we think it is a normal component of the objective function. What's more, $\omega$ serves as a crutial hyper-parameter to control the self-supervision's contribution to the overall weighted loss function. Thus, we implemented experiments to explore the performance of different weighting values of the parameter $\omega$ such as 1, 1.2, 1.5, 2, etc. on previous SESEMI model and we found $\omega = 1$ yields consistent results across all datasets and CNN architectures similar to [37], suggesting that both the supervised and self-supervised losses can be relatively balanced and they almost play the identical role in the compatible training setting. Furthermore, $\omega = 1.2$ or 2 could produce more fluctuating error rates on testing sets which means that we don't make sure if the $\omega$ value is available to construct a high-quality SSL model. Thus, we evidently choose $\omega = 1$ in our proposed Color-$S^4L$ model and also reduce one hyper-parameter to tune. At last, we update $\theta$ by backpropagating gradients in two branches of the network.

### 3.2 Choosing Self-supervised Tasks

Given the weighted sum of semi-supervised loss, we know that self-supervised loss can not only allow Color-$S^4L$ to learn complementary visual features from unlabeled data but also act as a strong regularizer with multiple pretext tasks to improve the model's generalization. In addition, effective self-supervisions [8,18,6,31,2] can generate much more powerful surrogate labels which equal to the supervision of ground-truth labels in $D_L$. Based on previous self-supervised learning literature, we find each pretext task has various advantages and certain applications (**Section 2.2**). Due to successful results of the SESEMI model [37] which utilizes image rotation along with horizontal (left-right) and vertical (up-down) flips in geometric transformation as pretext tasks, we faithfully follow it and additionally apply the powerful reconstruction-based image colorization task in Color-$S^4L$ as our model's nomenclature.

Consider that each pretext task seems independent in their individual backgrounds, we design to apply multiple self-supervised pretext tasks to generate auxiliary labels for unlabeled data independently while training all data simultaneously. Thus, we could match different tasks with different labels respectively in the self-supervised branch. In detail, for image rotation we enable it to produce 4 labels such as $\tilde{y} = 0, 1, 2, 3$ because it serves as a 4-classification model to predict degrees corresponding to $[0°, 90°, 180°, 270°]$. Next, we chose the horizontal (left-right) and vertical (up-down) flips within geometric transformation to produce another two surrogate labels. To emphasize, we especially embed our self-trained image colorization model into the Color-$S^4L$ to recognize color changing within the self-supervised branch. Different from the techniques that produce transformation operations on input images directly, we trained a new image colorization model with Encoder-Decoder Network and transferred it on unlabeled data to yield an auxiliary label (i.e. $\tilde{y} = 7$) for the colorized samples. In **Section 3.3**, we will introduce our image colorization training model meticulously.

### 3.3 Training Image Colorization Model

Since the role of pretext tasks is assisting unlabeled data to learn some reliable supervisory signals, the image colorization model is supposed to yield optimal results for colorizing the grayscale pictures. From previous self-supervised literature, Zhang's [49] model has arisen considerable attentions on image colorization and we firstly evaluate the colorization results on CIFAR-10 dataset but we find it produce very monotonous and retro colors on colorful CIFAR-10 pictures which we do not expect. Thus, we choose to self-train a reasonable image colorization model inspired by [2] and **Fig.2** represents the overall architecture of our Encoder-Decoder model. Similarly, we consider images of size $H \times W$ in the CIE $L*a*b$ color space [27] because it can separate the color characteristics from the luminance that contains the main image features [4,22]. Given an input luminance component $X_L \in R^{H \times W \times 1}$, the objective of our model is to estimate the remaining sections to generate a fully colored version $\tilde{X} \in R^{H \times W \times 3}$. In brief, we assume that there is a mapping $F$ such that $F : X_L \to (\tilde{X_a}, \tilde{X_b})$, where $\tilde{X_a}, \tilde{X_b}$ are the a*, b* components of the reconstructed image, which combined with the input data and give us the estimated colored image $\tilde{X} = (X_L, \tilde{X_a}, \tilde{X_b})$. With regards to the architecture, we train a CNN to map a grayscale input towards a distribution over quantized color value outputs by using the model displayed in **Fig.2**. Unlike [2], our network is logically divided into two main components: the Encoder and Decoder. Before the training, we applied image preprocessing to make the pixel values of all three image components within the interval of [-1,1] to ensure correct learning. Aimed to attain the optimal image colorization by minimizing the loss function defined over the target output and the estimated output, we employed the Mean Square Error between the estimated pixel colors in a*b* space and their real values. For a picture **X**, the MSE is given by (2):

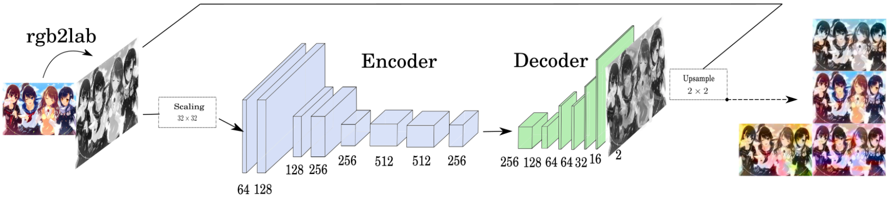


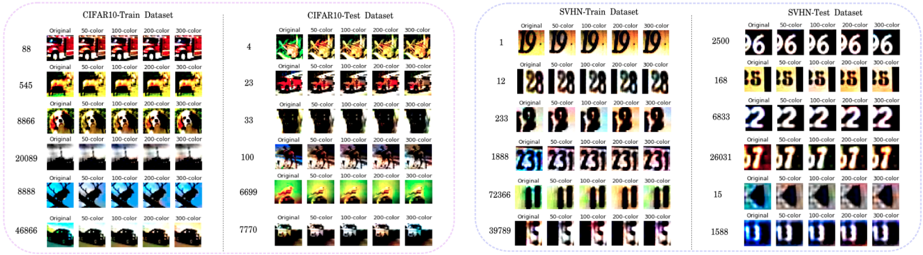
**Fig. 2.** An overview of the Encoder-Decoder image colorization model architecture.

$$C(X, \theta) = \frac{1}{2HW} \sum_{k \in \{a,b\}} \sum_{i=1}^{H} \sum_{j=1}^{W} (X_{k_{i,j}} - \tilde{X}_{k_{i,j}})^2 \tag{2}$$

Where $\theta$ demonstrates all model parameters, $X_{k_{i,j}}$ and $\tilde{X}_{k_{i,j}}$ respectively denote the ij:th pixel value of the k:th component of the target and reconstructed image. Such loss function can easily be extended to a batch $\beta$ by averaging the cost among all images in each batch, i.e. $1/|\beta| \sum_{X \in \beta} C(X, \theta)$. The above is the training framework of our image colorization pretext task and more experimental results for Color-$S^4L$ model will be analyzed detailedly in **Section 4.2**.

# 4 Experimental Evaluation and Analysis

To begin with, we will analyse our image colorization empirical results and present how we embed it into the Color-$S^4L$ model. Then, we follow the standard evaluation protocol for semi-supervised learning (SSL) which randomly samples varying fractions of the training data as labeled items while treating the rest examples that have discarded label information as the unlabeled data. Next, we train the previous SESEMI model [37] with a variety of network architectures (e.g., ConvNet [17], Wide Residue Network(WRN) [46], VGG16 [30], etc.) and make comparisons of their different performances. After the further research on SESEMI, we manage to master more valuable information and knowledge for our Color-$S^4L$ model which is similarly affected by those neural architectures because the neural network trunk and pretext tasks in **Fig.1** play independent roles within the model from our greatest knowledge. Thus, we generally employ three best network architectures on Color-$S^4L$ model with CIFAR-10, SVHN, CIFAR-100 datasets and discover that Color-$S^4L$ could deliver competitive performance in most cases with the optimal network architectures.



(a) (b)

**Fig. 3. Left(a)**: Comparison of several results obtained from our CIFAR-10 colorization networks with various training epochs on CIFAR-10 dataset. ("Original" means the data samples, "S-color" denotes the model has been trained for S epochs and S belongs to the set {50,100,200,300}). **Right(b)**: Performance comparison of CIFAR-10 colorization models on SVHN datasets.

## 4.1 Datasets and Baselines

We empirically evaluate the performance of our proposed Color-$S^4L$ model on three widely adopted semi-supervised image classification datasets: Street View House Numbers (SVHN) [21], CIFAR-10 and CIFAR-100 [15]. Briefly speaking, the CIFAR-10 dataset consists of 50000 training and 10000 testing 32*32 natural color images in 10 classes. Similarly, CIFAR-100 includes 60000 pictures but it extends to 100 classes containing 600 images each. As for SVHN, it contains 73257 train and 26032 test samples categorized over 10 digits (0-9) in natural scene images. To train our Color-$S^4L$ model efficiently, we just utilize the official train/test splits without the provided 531131 extra images and our classification task is to recognize the centermost digit in each natural image of 32*32 pixels. In sum, we utilize such three datasets to conduct further research of [37] and evaluate the performance of our proposed Color-$S^4L$ model under the condition of class-distribution mismatch systematically.

Consider that our model focuses on semi-supervised learning (SSL), we empirically compare our Color-$S^4L$ model trained with effective self-supervised pretext tasks against two state-of-the-art baselines for supervised and semi-supervised learning like [37]: (a) the retrained RotNet models of [8] which employed the self-supervised rotation loss to pre-train unlabeled data followed by a separate step of supervised fine-tuning on labeled samples. Also, we apply all labels of data in the supervised branch of Color-$S^4L$ model to imitate the standard supervised learning framework with data augmentation [17]. (b) the SESEMI SSL model [37] and previous SSL effective approaches with consistency regularization, namely $\prod$ model [28], Mean Teacher [34], Temporal Ensembling [17], VAT [20], and MixMatch [3].

### 4.2 Image Colorization Experiments and Analysis

Since our datasets of Color-$S^4L$ lie in CIFAR-10, SVHN, and CIFAR-100, we chose to train image colorization models (**Section 3.3**) on the training sets of CIFAR-10 and CIFAR-100 datasets separately, then transfer the saved model on the testing data of three datasets to evaluate the performance of the colorization model. While training, the MSE loss function (1) is backpropagated to update model parameters $\theta$ using Adam Optimizer [12] and we impose a fixed input image size of 32*32 to allow for batch processing. **Fig.3** and **Fig.4** demonstrate our training models' image colorization results on CIFAR-10 & CIFAR-100 respectively. As MSE will be lower as the training epochs increase, we explored a variety of training epochs S to attain an optimal colorization model for our Color-$S^4L$ framework, where S belongs {50,100,200,300} and we get final accuracies on CIFAR-10 successively corresponding to {79.14%, 81.80%, 83.06%, 83.96%}. Also, we utilized two metrics to explore the empirical results of colored pictures including Structural Similarity Measure(SSIM) and Mean Squared Error(3) [41] as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{((\mu_x)^2 + (\mu_y)^2 + c_1)((\sigma_x)^2 + (\sigma_y)^2 + c_2)}; MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \qquad (3)$$

To brief, we will provide more details about the series of such metric eveluations in **Appendix** and we just conclude several rules of our colorization models: (a) Different pictures correspond to distinct S-color models with the optimal evaluation values, in which the SSIM value is close to 1 and MSE approaches 0. (b) Sometimes four S-color models return the same values of two metrics on images like CIFAR-10-Test[33] and SVHN-Train[12] (**Fig.3**), which means each model is reasonable to be utilized in our Color-$S^4L$ architecture. (c) These trained colorization models could predict our datasets' color components well and meet the demands for self-supervised pretext task, whereas it seems impossible and complex to choose the best colorization inference model. Thus, we decided to embed the saved CIFAR-10-{100,200,300}-color & CIFAR-100-{100,200}-color H5 Keras models into our Color-$S^4L$ model and employ them to infer unlabeled data within each epoch, then generate proxy labels for data samples with image rotation and geometric transformation independently.

### 4.3 Model Architectures.

In this part, we'll introduce the neural architectures of our "Neural Network Trunk" in **Fig.1** and we explored a variety of network trunks to conduct further research on SESEMI [37], then applied the most effective and reasonable ones to Color-$S^4L$ model. To make further comparison and analysis of diverse neural architectures, we conduct experiments with 6 high-quality and widely-used architectures: (i) the 13-layer max-pooling ConvNet [37,17]; (ii) the modern wide residual network with depth 28 and width 4 (WRN-28-4) [24,46]; (iii) ResNet34 network with Shake-Shake regularization (Shake-WRN) [7]; (iv) the popular deep residual network ResNet50 [10]; (v) the 13-layer max-pooling Network-in-Network (NIN) [8]; (vi) the very deep convolution networks VGG16 [30].

To emphasize, we are the first to explore model effects of Shake-WRN, ResNet50 and VGG16 in our novel self-supervised semi-supervised learning framework and we will compare them within the following experiments from multiple perspectives. Practically speaking, we faithfully follow the original specification of the ConvNet, WRN, and NIN architectures. As for ResNet50 and VGG16, we simply utilized the normal ResNet50 and VGG16 models from Keras. To employ Shake-WRN, we implemented the Shake-Shake regularization on ResNet34 network inspired by [7]. Then, we make all architectures embody convolutional layers followed by batch normalization and ReLU non-linearity to embed the "Neural Network Trunk" into our Color-$S^4L$ model, which can be viewed as a multi-task architecture that contains a common CNN "backbone" to learn a shared representation of both labeled and unlabeled data.

### 4.4 Results and Analysis

**Further Research on SESEMI model.** Since our inspiration derives from SESEMI model [37], we think it's better to conduct more experiments to validate such novel semi-supervised algorithm which combined self-supervised regularization. Also, we implemented similar empirical details of [37] and more information will be provided in **Appendix**. Specially, we evaluate different model architectures and various number of proxy labels with additional pretext tasks in geometric transformation like Affine Transformation and Translation.

**Table 1** represents the error rates for 6 neural architectures on CIFAR-10 with disparate labeled information and training epochs. According to [37], we find one obvious advantage of $S^4L$ framework—the training epochs can be significantly fewer than millions of training iterations in [43,3] which means it will cost a lot less time and computing power. Within our experiments, we computed the average value of optimal error rates over three runs since error rates will

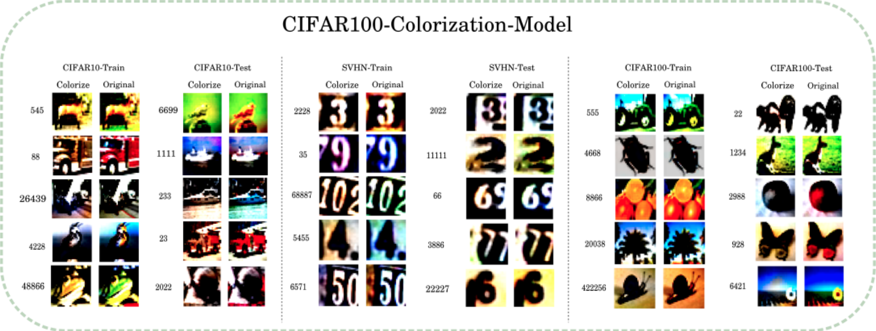


**Fig. 4.** Transfer the CIFAR-100 Image Colorization with 100 epochs on CIFAR-10, SVHN and CIFAR-100 datasets.

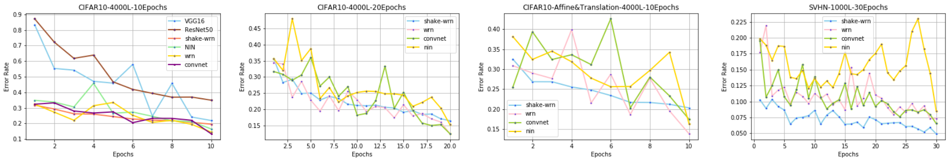
decrease gradually in our training process and the lowest error that usually occurs at the last training epoch can reflect each trunks' performance much better. Following the methodology, we chose top 3 CNN trunks in each situation and we observed that ConvNet and WRN can achieve satisfactory behaviors on three datasets. Surprisingly, Shake-WRN outperforms all the CNN trunks especially on SVHN datasets. Also, NIN represents intermediate performance over CIFAR-10 and CIFAR-100. On the contrary, ResNet50 seems to be the worst architecture of $S^4L$ framework and VGG16 merely perform a little well with full labeled information like 50000 or 73257 labels. Thus, we chose top 3 neural architectures with lower error rates on various labeling situations.

**Fig. 5** dives into several specific training processes of 6 architectures to demonstrate the CNN trunk's training stability and explore additional pretext tasks like Affine Transformation & Translation. Definitely, we found that Shake-WRN denotes the best training stability in each training phase (especially **Fig. 5(d)**) due to its Shake-Shake regularization which could combat overfit by decorrelating the branches of multi-branch networks. Furthermore, other architectures demonstrate reasonable value fluctuations no matter 10 or 20 epochs in the training phase (e.g. **Fig. 5(a)(b)**). We also add the Affine or Translation tasks in geometric transformations along with 6 previous pretext tasks to generate 7 or 8 proxy labels, and discover that they return normal performance at the last training epoch but there exist several huge fluctuations while training (e.g. **Fig. 5(c)**) which seems not available in our Color-$S^4L$ model that needs to control the number of surrogate tasks much carefully and efficiently.

**Color-$S^4L$ model Results and Analysis.** Based on the further research of SESEMI model [37], we tend to take investigations of our novel Color-$S^4L$ algorithm's performance and analyse it separately on CIFAR-10, SVHN and CIFAR-100 datasets compared with previous supervised and semi-supervised learning approaches. Reviewing the research results on model architectures, we especially

**Table 1.** Test classification error rates (%) for SESEMI model on 6 different neural architectures with various training epochs (shown as '10/20' & '10/30') over CIFAR-10, SVHN and CIFAR-100 datasets with data augmentation averaged over three runs.

| CNN Trunk | CIFAR-10 (10/20) | | | SVHN (10/30) | | | CIFAR-100 (10/20) | |
|---|---|---|---|---|---|---|---|---|
| | 50000L | 4000L | 2000L | 73257L | 1000L | 500L | 50000L | 20000L |
| ConvNet | **6.42/4.87** | **13.40/12.57** | **16.15/15.27** | **2.67/2.49** | **6.78/6.65** | **9.73/8.52** | **25.61/25.07** | **33.07/33.61** |
| WRN | **5.96/5.58** | **14.48/12.54** | **17.66/16.11** | **2.60/2.52** | **7.56/7.34** | **9.82/8.30** | **27.01/24.87** | **34.43/32.94** |
| NIN | **8.03/6.53** | **16.51/15.36** | **20.49/18.59** | 6.53/4.89 | 8.91/8.54 | 12.18/10.69 | **30.02/28.56** | **37.18/35.76** |
| Shake-WRN | 8.98/7.51 | 19.59/16.57 | 25.73/23.00 | **2.90/2.68** | **5.07/4.94** | **8.56/6.99** | 38.11/32.74 | 40.25/38.93 |
| ResNet50 | 27.92/29.34 | 34.98/32.79 | 39.69/37.95 | 9.50/8.74 | 16.83/14.92 | 18.79/16.32 | 69.86/63.67 | 71.25/68.39 |
| VGG16 | 8.84/7.37 | 21.92/20.34 | 26.89/25.48 | 3.73/3.26 | 15.95/13.67 | 18.46/15.97 | 55.50/52.94 | 57.39/55.26 |



(a) (b) (c) (d)

**Fig. 5.** Selected training processes with different epochs or pre-text tasks on datasets.

chose 3 outstanding stable CNN trunks: **ConvNet, WRN and Shake-WRN** to be applied in Color-$S^4L$ model, and the implementation details will be provided at **Appendix**, except that we train the inference model on test dataset with 128 batch size for 30 training epochs. More importantly, our Color-$S^4L$ model could produce image classification results competitive with, and in some cases exceeding prior state-of-the-art methods.

**CIFAR-10**: **Table 2** represents the comparison between Color-$S^4L$ model with other supervised & semi-supervised methods based on consistency baselines. To comprehensively investigate each CIFAR-10 & CIFAR-100 S-epochs image colorization model effects within Color-$S^4L$, we have conducted extensive experiments with diverse colorization models including CIFAR-10-{100,200,300}-color & CIFAR-100-100-color. Following our methodology in **Section 3**, we found that ConvNet outperforms WRN and Shake-WRN on CIFAR-10 no matter with any colorization model and it accompanies especially best with the CIFAR-10-300-color model. Thus, we just show the best results from ConvNet model with the CIFAR-10-300-color model in **Table 2**. Whereas, WRN represents unstable results with most colorization models so that it can't represent trusty results like Shake-WRN and ConvNet, which means that somtimes image colorization technique may not be complementary to certain model architecture in Color-$S^4L$.

**SVHN**: Experiments on SVHN denotes a different story. The optimal performance we achieve comes from the CIFAR-10-100-color model with 30 Color-$S^4L$

**Table 2.** Color-$S^4L$: CIFAR-10, '1000L' means 1000 labels.

| Method(CIFAR-10) | 1000L | 2000L | 4000L | 50000L |
|---|---|---|---|---|
| Supervised [34] | 46.43±1.21 | 33.94±0.73 | 20.66±0.57 | 5.82±0.15 |
| Mixup [38] | 36.48±0.15 | 26.24±0.46 | 19.67±0.16 | —— |
| Manifold Mixup [38] | 34.58±0.37 | 25.12±0.52 | 18.59±0.18 | —— |
| SESEMI ASL(ConvNet) [37] | 29.44±0.24 | 21.53±0.18 | 16.15±0.12 | 4.70±0.11 |
| VAT SSL [20] | —— | —— | **11.36±0.34** | 5.81±0.02 |
| Π Model SSL [17] | —— | —— | 12.36±0.31 | 5.56±0.10 |
| Mean Teacher SSL [34] | 21.55±1.48 | 16.73±0.31 | **12.31±0.28** | 5.94±0.15 |
| Color-$S^4L$(ConvNet) | **20.45±0.34** | **16.22±0.20** | 13.05±0.25 | **5.07±0.06** |
| SESEMI(ConvNet) | **18.45±0.26** | **15.84±0.40** | **12.95±0.34** | **5.21±0.24** |

training epochs. From the **Table 3**, our newly-used Shake-WRN architecture surpasses other neural architectures both in SESEMI and Color-$S^4L$ models. Also, we are the first to explore and excavate the good performance of Shake-WRN neural architecture in our novel $S^4L$ framework. Whereas, the Color-$S^4L$ model can not achieve satisfactory results when compared against Mean Teacher which averages model weights instead of label predictions. Also, as the number of labels in the supervised branch increase, Color-$S^4L$ with ConvNet architecture even outperforms the optimal methods (e.g. SESEMI(Shake-WRN), Color-$S^4L$(Shake-WRN)) in other benchmarks with fewer labels. Later, we will discuss the characteristics and limitations of both SESEMI and Color-$S^4L$ models for semi-supervised learning on SVHN in **Section 4.5**.

**Table 3.** Color-$S^4L$: SVHN

| Method(SVHN) | 250L | 500L | 1000L | 73257L |
|---|---|---|---|---|
| Supervised [34] | 27.77±3.18 | 16.88±1.30 | 12.32±0.95 | 2.75±0.10 |
| Mixup [38] | 33.73±1.79 | 21.08±0.61 | 13.70±0.47 | —— |
| Manifold Mixup [38] | 31.75±1.39 | 20.57±0.63 | 13.07±0.53 | —— |
| SESEMI ASL(ConvNet) [37] | 23.60±1.38 | 15.45±0.79 | 10.32±0.16 | 2.26±0.07 |
| ∏ Model SSL [17] | —— | 6.65±0.53 | 4.82±0.17 | 2.54±0.04 |
| Mean Teacher SSL [34] | **4.35±0.50** | **4.18±0.27** | **3.95±0.19** | 2.50±0.05 |
| Color-$S^4L$ (ConvNet) | 17.97±0.72 | 12.92±1.26 | **5.05±0.25** | **2.57±0.06** |
| SESEMI(ConvNet) | 16.11±1.38 | 8.65±0.18 | 5.59±0.34 | **2.26±0.07** |
| Color-$S^4L$(Shake-WRN) | **8.81±0.51** | **6.37±0.26** | **5.13±0.12** | 3.31±0.05 |
| SESEMI (Shake-WRN) | **10.52±1.36** | **7.23±0.24** | 5.68±0.23 | **2.34±0.08** |

**Table 4.** Color-$S^4L$: CIFAR-100

| Method(CIFAR-100) | 20000L | 50000L |
|---|---|---|
| Supervised [34] | 42.83±0.24 | 26.42±0.17 |
| SESEMI ASL(ConvNet)[37] | 38.62±0.31 | 22.49±0.15 |
| ImageNet-32 Fine-tuned | **30.48±0.27** | **22.22±0.25** |
| ∏ Model SSL [17] | 36.18±0.32 | 26.32±0.04 |
| TempEns SSL [17] | 35.65±0.41 | 26.30±0.15 |
| Color-$S^4L$(ConvNet) | **33.59±0.30** | **24.80±0.62** |
| SESEMI(ConvNet) | **34.09±1.54** | **25.32±0.39** |
| SESEMI(WRN) | **34.69±0.10** | **24.83±0.12** |

**CIFAR-100**: Semi-supervised image classification on CIFAR-100 (100 classes) seems much more challenging than CIFAR-10 and SVHN. Reviewing the image colorization section, we decided to embed CIFAR-100-100-color into CIFAR-100 Color-$S^4L$ model since this model has learned color information from the CIFAR-100 data itself and the pretext task seems more available to extract image features and generate reasonable proxy labels (**Fig. 4**). Similar to CIFAR-10, ConvNet is the best CNN trunk on CIFAR-100 dataset and we obtained competitive performance for 30 training epochs, even our Color-$S^4L$ model could achieve slightly lower error rates than our retrained SESEMI model (**Table 4**).

### 4.5 Discussions.

From the above extensive experiments, we not only observed good characteristics of our Color-$S^4L$ model but also conclude several limitations of such $S^4L$ training mode. Obviously, the trained Color-$S^4L$ model could obtain best performances in most cases and we evaluate the predictive results averaged over 4 runs with different seed numbers (e.g., seed = 1,5,10,15). Also, our special model with certain CNN trunks on specific datasets demonstrate consistent good results without random seeds' effects, such as the powerful Shake-WRN model works best on SVHN datasets and ConvNet for CIFAR-10 & CIFAR-100. Whereas, **Table 3** shows that our performances are not satisfactory compared against Mean Teacher. One possible reason is that images of SVHN do not contain a whole object picture like CIFAR-10, and presents complex additional information with the centermost identified numbers surrounding by "distractor" digits. What's more, sometimes image colorization may cause a few perturbations on classification error rates or bad results with WRN architecture due to a little instability or non-compatibility of multimodal image colorization models, but mostly it acts as a trusty supervision to generate proxy labels on unlabeled data.

Compared with concurrent research on semi-supervised learning over multiple regularization like self-supervised techniques [47] and Mixup [3], our framework is similar to $S^4L$ [47] but with different evaluation protocols. Specifically, we are the first to self-train good colorization models for CIFAR-10, SVHN and CIFAR-100, then we embed it as an effective self-supervision in Color-$S^4L$ to train the model recognizing color changes of images along with geometric transformations. In principle, our work is potentially complementary to MixMatch and Label Propagation like [37] by integrating a self-supervised loss term. Also, in the future research we can try to extract features of the trained Color-$S^4L$ model by using feature clustering methods and transfer the learned representation to more challenging tasks like object detection and semantic segmentation.

## 5 Conclusions

We proposed a novel Color-$S^4L$ model which combine multiple self-supervised pretext tasks into the common semi-supervised learning framework. Additionally, we embedded our self-trained effective image colorization model into the SSL pipeline to establish a new supervision along with image rotation and geometric transformation. Furthermore, we explored 6 CNN architectures' performance both in SESEMI [37] and Color-$S^4L$ model, even discovered our first-applied Shake-WRN neural network surpasses other trunks on SVHN datasets. In sum, we dived into the research on the novel integration of the quickly-advancing self-supervised visual representation learning and semi-supervised learning, even provide competitive or best results from our Color-$S^4L$ model in comparison to previous semi-supervised learning methods.

# References

1. Bachman, P., Alsharif, O., Precup, D.: Learning with pseudo-ensembles. In: NIPS (2014)
2. Baldassarre, F., Morín, D.G., Rodés-Guirao, L.: Deep koalarization: Image colorization using cnns and inception-resnet-v2. CoRR **abs/1712.03400** (2017), http://arxiv.org/abs/1712.03400
3. Berthelot, D., Carlini, N., Goodfellow, I.G., Papernot, N., Oliver, A., Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. In: NeurIPS (2019)
4. Cao, Y., Zhou, Z., Zhang, W., Yu, Y.: Unsupervised diverse colorization via generative adversarial networks. In: Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Dzeroski, S. (eds.) Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10534, pp. 151–166. Springer (2017). https://doi.org/10.1007/978-3-319-71249-9_10, https://doi.org/10.1007/978-3-319-71249-9_10
5. Doersch, C., Zisserman, A.: Multi-task self-supervised visual learning. CoRR **abs/1708.07860** (2017), http://arxiv.org/abs/1708.07860
6. Dosovitskiy, A., Fischer, P., Springenberg, J.T., Riedmiller, M.A., Brox, T.: Discriminative unsupervised feature learning with exemplar convolutional neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(9), 1734–1747 (2016). https://doi.org/10.1109/TPAMI.2015.2496141, https://doi.org/10.1109/TPAMI.2015.2496141
7. Gastaldi, X.: Shake-shake regularization. CoRR **abs/1705.07485** (2017), http://arxiv.org/abs/1705.07485
8. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. CoRR **abs/1803.07728** (2018), http://arxiv.org/abs/1803.07728
9. Goyal, P., Mahajan, D., Gupta, A., Misra, I.: Scaling and benchmarking self-supervised visual representation learning. CoRR **abs/1905.01235** (2019), http://arxiv.org/abs/1905.01235
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
11. Jing, L., Tian, Y.: Self-supervised spatiotemporal feature learning by video geometric transformations. CoRR **abs/1811.11387** (2018), http://arxiv.org/abs/1811.11387
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
13. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition (2015)
14. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. CoRR **abs/1901.09005** (2019), http://arxiv.org/abs/1901.09005
15. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
16. Lai, Z., Xie, W.: Self-supervised learning for video correspondence flow. CoRR **abs/1905.00875** (2019), http://arxiv.org/abs/1905.00875
17. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. ArXiv **abs/1610.02242** (2016)

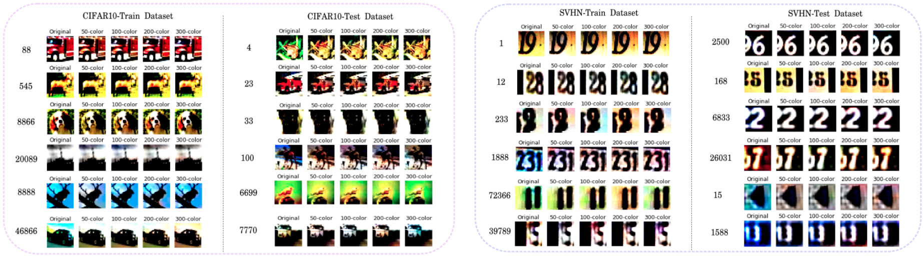
18. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 840–849. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.96, https://doi.org/10.1109/CVPR.2017.96
19. Lee, D.H.: Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks (2013)
20. Miyato, T., ichi Maeda, S., Koyama, M., Ishii, S.: Virtual adversarial training: A regularization method for supervised and semi-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**, 1979–1993 (2017)
21. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. NIPSW
22. Nixon, M.S., Aguado, A.S.: Feature extraction  image processing for computer vision (2002)
23. Noroozi, M., Vinjimoor, A., Favaro, P., Pirsiavash, H.: Boosting self-supervised learning via knowledge transfer. CoRR **abs/1805.00385** (2018), http://arxiv.org/abs/1805.00385
24. Oliver, A., Odena, A., Raffel, C., Cubuk, E.D., Goodfellow, I.J.: Realistic evaluation of deep semi-supervised learning algorithms. In: NeurIPS (2018)
25. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010). https://doi.org/10.1109/TKDE.2009.191, https://doi.org/10.1109/TKDE.2009.191
26. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 2536–2544 (2016). https://doi.org/10.1109/CVPR.2016.278, https://doi.org/10.1109/CVPR.2016.278
27. Robertson, A.: The cie 1976 color-difference formula. Color Research  Application 2(1) (1977)
28. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: NIPS (2016)
29. Salimans, T., Goodfellow, I.J., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. ArXiv **abs/1606.03498** (2016)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1409.1556
31. Singer, J.R., Grünbaum, F.A., Kohn, P., Zubelli, J.P.: Image reconstruction of the interior of bodies that diffuse radiation. Science **248**(4958), 990–993 (1990), http://www.jstor.org/stable/2874402
32. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. CoRR **abs/1511.06390** (2015)
33. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014), http://jmlr.org/papers/v15/srivastava14a.html
34. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NIPS (2017)
35. Thomas, P.: Semi-supervised learning by olivier chapelle, bernhard schölkopf, and alexander zien (review). IEEE Trans. Neural Networks **20**(3),  542 (2009), http://dblp.uni-trier.de/db/journals/tnn/tnn20.html#ChapelleSZ09

36. Tran, D., Wang, H., Torresani, L., Feiszli, M.: Video classification with channel-separated convolutional networks. CoRR **abs/1904.02811** (2019), http://arxiv.org/abs/1904.02811
37. Tran, P.V.: Semi-supervised learning with self-supervised networks. CoRR **abs/1906.10343** (2019), http://arxiv.org/abs/1906.10343
38. Verma, V., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. In: IJCAI (2019)
39. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking emerges by colorizing videos. CoRR **abs/1806.09594** (2018), http://arxiv.org/abs/1806.09594
40. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. CoRR **abs/1505.00687** (2015), http://arxiv.org/abs/1505.00687
41. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE TRANSACTIONS ON IMAGE PROCESSING **13**(4), 600–612 (2004)
42. Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: When to warp? In: 2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016, Gold Coast, Australia, November 30 - December 2, 2016. pp. 1–6. IEEE (2016). https://doi.org/10.1109/DICTA.2016.7797091, https://doi.org/10.1109/DICTA.2016.7797091
43. Xie, Q., Dai, Z., Hovy, E.H., Luong, M.T., Le, Q.V.: Unsupervised data augmentation. ArXiv **abs/1904.12848** (2019)
44. Yamaguchi, S., Kanai, S., Shioda, T., Takeda, S.: Multiple pretext-task for self-supervised learning via mixing multiple image transformations. CoRR **abs/1912.11603** (2019), http://arxiv.org/abs/1912.11603
45. Yan, M., Zhao, M., Xu, Z., Zhang, Q., Wang, G., Su, Z.: Vargfacenet: An efficient variable group convolutional neural network for lightweight face recognition. CoRR **abs/1910.04985** (2019), http://arxiv.org/abs/1910.04985
46. Zagoruyko, S., Komodakis, N.: Wide residual networks. CoRR **abs/1605.07146** (2016), http://arxiv.org/abs/1605.07146
47. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: $S^4l$: Self-supervised semi-supervised learning. CoRR **abs/1905.03670** (2019), http://arxiv.org/abs/1905.03670
48. Zhang, H., Cissé, M., Dauphin, Y., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. ArXiv **abs/1710.09412** (2017)
49. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III. Lecture Notes in Computer Science, vol. 9907, pp. 649–666. Springer (2016). https://doi.org/10.1007/978-3-319-46487-9_40, https://doi.org/10.1007/978-3-319-46487-9_40
50. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 645–654. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.76, https://doi.org/10.1109/CVPR.2017.76

# 6 Appendix

## 6.1 Image colorization model evaluation.

Intuitively, we can observe several colored example pictures obtained from a variety of S-epochs colorization models separately in **Fig. 6**. The **left (a)** displays the colorization performance over CIFAR-10's training and testing set from models trained on CIFAR-10 50000 pictures with S-epochs and the **right (b)** aims at SVHN dataset. Overall, different models certainly represent disparate changes for the predicted color pixels of natural images compared to the ground truth("original") ones, but they generate the reconstructed pictures with natural colors and complete object semantics, which have achieved what we expect for the Color-$S^4L$ model. Take CIFAR-10-Train[88][545][8866] & CIFAR-10-Test[33] as example, most of the new constructed pictures even seems identical as the original ones from our eyes. Whereas, some of other images demonstrate obvious color changes that can not significantly influence the role of image colorization pretext task to generate proxy labels. Evidently, CIFAR-10-Train[46866] altered the car's background from blue to yellow or white, CIFAR-10-Test[7770] established the white trunk to be blue with CIFAR-10-300-color-model. What's more, CIFAR-10-Test[4] has colorized those leaves under the frog to be orange instead of green so that it may cause some fluctuations in the self–supervised branch of image classification process. As for CIFAR-10-Test[6699], it presents uneven color distributions that make the bird a little unclear and fuzzy. Whereas, such alternatives are very normal while colorizing pictures and they produce auxiliary labels on unlabeled data much reasonably.



(a) (b)

**Fig. 6. Left(a)**: Comparison of the results obtained from our CIFAR-10 colorization networks with various training epochs on CIFAR-10 dataset. ("Original" means the data samples, "S-color" denotes the model has been trained for S epochs and S belongs to the set {50,100,200,300}). **Right(b)**: Performance comparison of CIFAR-10 colorization models on SVHN datasets.

Since these models have learned comprehensive and natural color layers' predictions on 10-classes objects, it is reasonable to transfer them into SVHN dataset which just contains digits of street views. From **Fig. 6 (b)**, we could find each “S-epochs” model preserve the original contour of digits and some of them produce alternative color tones for the image background (i.e., SVHN-Train [233] [72366] ; SVHN-Test [15]). Obviously, the change of color tones do not affect our Color-$S^4L$ model to learn complete and clear semantic features by utilizing multiple self-supervised pretext tasks.

As we have mentioned in **Section 4.3**, to quantify the evaluation of S-epochs models, we utilized two metrics including Mean Squared Error (MSE) and Structural Similarity Measure(SSIM)(4) [41] as follows:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2; SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{((\mu_x)^2 + (\mu_y)^2 + c_1)((\sigma_x)^2 + (\sigma_y)^2 + c_2)}; \quad (4)$$

Different from each other, the gist of SSIM is to model the perceived change in the structural information of images, whereas MSE is actually estimating the perceived errors. In addition, MSE value can be very large values where 0 means equal pictures but the SSIM value can vary between -1 and 1, where 1 indicates perfect similarity. We computed two values for randomly-selected 1000 example images with all the S-epochs colorization models and treat them as references for us to embed suitable models into Color-$S^4L$ much better. The following **Table 5** and **Table 6** show the detailed computing values of SSIM and MSE on our randomly selected CIFAR-10 samples.

**Table 5.** CIFAR-10 Train

| Train-index | 50-color | 100-color | 200-color | 300-color |
|---|---|---|---|---|
| 88 | (MSE)1.62/0.08(SSIM) | 1.59/0.08 | 1.59/0.08 | **1.57/0.08** |
| 545 | 1.62/0.17 | 1.60/0.17 | **1.59/0.18** | 1.61/0.19 |
| 8866 | 1.71/0.26 | 1.70/0.26 | **1.69/0.27** | **1.69/0.27** |
| 8888 | 1.89/0.18 | 1.80/0.20 | 1.82/0.19 | **1.79/0.20** |
| 20089 | 1.80/0.50 | 1.80/0.50 | **1.79/0.50** | **1.79/0.50** |
| 46866 | **1.74/0.10** | **1.74/0.10** | 1.76/0.11 | 1.81/0.10 |

**Table 6.** CIFAR-10 Test

| Test-index | 50-color | 100-color | 200-color | 300-color |
|---|---|---|---|---|
| 4 | (MSE)1.92/0.25(SSIM) | 1.91/0.24 | **1.87/0.25** | 1.91/0.25 |
| 23 | 1.75/0.15 | 1.77/0.14 | **1.62/0.15** | 1.80/0.15 |
| 33 | **1.43/0.07** | **1.43/0.07** | **1.42/0.07** | **1.43/0.07** |
| 100 | 2.07/0.39 | **2.04/0.40** | 2.06/0.38 | 2.06/0.39 |
| 6699 | 1.86/0.34 | **1.83/0.34** | 1.87/0.33 | 1.91/0.29 |
| 7770 | 1.52/0.27 | 1.52/0.28 | **1.51/0.28** | 1.54/0.27 |

To illustrate, the lower value of MSE or higher SSIM means better similarities between the reconstructed image and the original one. Thus we chose the best results for each image and investigate which colorization model can achieve the optimal performance, then it can provide us with some well-performed reference colorization models which can be embedded into our Color-$S^4L$. As we can see, different pictures correspond to distinct S-color models with the optimal evaluation values, in which the SSIM value is close to 1 and MSE approaches 0. For example, CIFAR-10-Test[4][23][33][7770] company best with the CIFAR-10-200-color model, but CIFAR-10-Train[88][8888] work well with 300-color. Sometimes four S-color models return the same values of two metrics on images like CIFAR-10-Test[33] and CIFAR-10-Train[20089]. From **Table 7 &**

**8**, SVHN shows the preferences to 50-color and 100-color models, even SVHN-Train[12] SVHN-Test[2500][6833] return the same evaluation values of two metrics. Generally speaking, MSE and SSIM just assist us to quantitatively evaluate our self-trained image colorization models and give us some reference models for Color-$S^4L$. Definitely, each of MSE and SSIM possess drawbacks respectively and we employ both of them to evaluate models much accurately. MSE has the major drawbacks that can be applied globally and only estimate the perceived errors of the image. On the other hand, SSIM is slower than MSE and it perceives the change in structural information of the picture by comparing local regions of the image instead of globally. Thus, in most cases we may find SSIM performs much more seriously than MSE.

**Table 7.** SVHN Train

| Train-index | 50-color | 100-color | 200-color | 300-color |
|---|---|---|---|---|
| 1 | (MSE)1.94/-0.02(SSIM) | 1.90/0.01 | **1.86/0.03** | 1.88/0.01 |
| 12 | **1.78/0.40** | **1.79/0.40** | **1.79/0.40** | **1.79/0.40** |
| 233 | 1.86/0.19 | 1.86/0.20 | 1.90/0.20 | **1.86/0.21** |
| 1888 | **2.03/0.15** | 2.07/0.13 | 2.10/0.11 | 2.10/0.12 |
| 72366 | **1.99/0.16** | 2.00/0.11 | 2.03/0.14 | 2.00/0.13 |
| 39789 | **1.62/0.34** | **1.62/0.34** | **1.61/0.35** | **1.62/0.34** |

**Table 8.** SVHN Test

| Test-index | 50-color | 100-color | 200-color | 300-color |
|---|---|---|---|---|
| 2500 | (MSE)1.40/0.27(SSIM) | **1.40/0.28** | 1.40/0.27 | **1.40/0.28** |
| 168 | 1.77/0.32 | 1.83/0.30 | 1.78/0.33 | 1.80/0.31 |
| 6833 | **1.47/0.39** | **1.47/0.39** | 1.48/0.39 | **1.47/0.39** |
| 26031 | 1.71/0.25 | 1.74/0.24 | 1.68/0.26 | **1.67/0.28** |
| 15 | 2.41/0.33 | **2.36/0.34** | 2.39/0.33 | 2.41/0.32 |
| 1588 | 1.84/0.17 | **1.82/0.18** | 1.93/0.16 | 1.87/0.17 |

According to the multiple and complex results of model evaluations, we decided to embed CIFAR-10-100,200,300-color & CIFAR-100-100,200-color models into our Color-$S^4L$ model to compare their respective performances for the whole dataset instead of some random samples. As image colorization pretext task needs to be independent with others in our self-supervised branch, we just employed the saved H5 Keras models to inference unlabeled data within each epochs and generate proxy labels for data samples with image rotation and geometric transformation.

### 6.2 Implementation Details.

Our Color-$S^4L$ algorithm is implemented by using Keras with GPU-enabled Tensorflow backend. We will follow the standard practice including data preprocessing & augmentation and hyper-parameter search respectively to introduce our training protocol.

**Data Pre-processing and Augmentation.** We utilized global contrast normalization to scale all data to have zero mean and unit L2 norm like [37]. Also, we pre-process CIFAR-10, CIFAR-100 with Zero Components Analysis (ZCA) whitening apart from SVHN. For Color-$S^4L$, we utilized standard augmentation on CIFAR-10 and CIFAR-100, such as random translations by up to 2 pixels on each side $\{\triangle x, \triangle y\} \in [-2, 2]$, Gaussian noise with $\sigma$=0.15, and horizontal (left-right) flip, whereas SVHN is limited to random translations and Gaussian noise. In practice, data augmentation is applied independently to both supervised and self-supervised branches, except that we do not employ horizontal flip on the self-supervised branch.

**Hyper-parameters.** Similar to [37], our Color-$S^4L$ model faithfully follows the same optimal hyper-parameters that have been evaluated on 10 percent of the provided SVHN training examples. Such hyper-parameters contain the mini-batch size, percentage of dropout regularization, initial learning rate, and number of training epochs that minimizes the classification error. We used the series of hyper-parameters subsequently across supervised and semi-supervised settings for all datasets with various architectures (ConvNet, WRN, Shake-WRN, etc.). Generally speaking, these hyper-parameters are tuned for CNNs and not aimed at Color-$S^4L$, which is a notable advantage of our method. On the contrary, approaches based on consistency regularization almost need to carefully tune specific hyper-parameters for optimal performance, like exponential moving average decay in Mean Teacher, the consistency coefficient of $\prod$ model, and the norm constraint $\epsilon$ for the adversarial direction in VAT.

**Training protocol.** We train our Color-$S^4L$ model using Nesterov accelerated gradient descent on mini-batches of 32 examples, also with the initial learning rate of 0.05, momentum of 0.9, weight decay of 0.0005, and dropout rate of 0.5 in all experiments. Creatively, we implement four rotations, two flips and image colorization on a given image in a mini-batch for improved training. Thus, our Color-$S^4L$ models receive two effective mini-batches having the same number of $32 \times 7 = 224$ unlabeled and labeled samples. For the semi-supervised learning setting, we train Color-$S^4L$ over unlabeled data in $D_u$ for 30 epochs on all three datasets. While training, we anneal the base learning rate according to the polynomial decay of the form: $lr(t) \leftarrow base - lr \times (1 - t/t_{max})p$, where base-lr = 0.05, t is the current iteration, $t_{max}$ is the maximum number of iterations, and $p = 0.5$ controls the rate of decay. Following such learning rate schedule, our Color-$S^4L$ could achieve the best performance improvement in few epochs which is a notable advantage by contrast to other SSL methods. To predict error rates of testing datasets, we set the batch size in our inference model to be 128, then we compute the mean and standard deviation of optimal error rates from four independent runs with random seed numbers (e.g., seed = 1,5,10,15) to evaluate our novel Color-$S^4L$.