COMPUTER SCIENCE

Self supervised Visual Representation Learning by Reconstructing Images

Donggun Lee [1]

Dongmin Kang [2]

Jonghyun Choi [3]

[1] Gwangju Institute of Science and Technology (GIST College), S.Korea, unlimitlife@gist.ac.kr

[2] School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST College), S.Korea, skydmk21@gist.ac.kr

[3] School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST College), S.Korea, jhc@gist.ac.kr

Abstract

In this paper, we explored the problem of performing image representation learning without human annotation processing. It proposes a new way to allow the model to understand the overall information of the image by solving a pretask that predicts the remaining area with a limited area of the image. As a network to solve the task, we use Generative Adversarial Network which has Auto-Encoder as generator. The GAN's generator receives only the center of the original image as input, and then learns to trick the discriminator by making it look like a picture that actually exists to regenerate the rest of the area. We can induce an understanding of invisible images by training the network to solve this task, which allows the model to learn more meaningful visual representations. We have tried to show better performance than model which is learned from the scratch for some transfer learning benchmarks as our method for learning visual representation.

1. INTRODUCTION

The computer vision model has undergone significant research and is currently experiencing significant performance advances. Several models that perform object recognition, semantic segmentation, and object detection are competing with human performance. However, a lot of supervision is required for this tremendous performance, and the process of collecting it is very expensive. For this reason, studies such as tarnsfer learning, domain adaptation, weakly supervised and unsupervised learning have been conducted to reduce the dependency of human data labeling. One of the highlights of these recent studies is self-supervised learning. In this paper, we aim to improve the performance of existing visual recognition tasks while reducing the cost of labeling through self-supervised visual representation learning.

Self-supervised visual representation learning is divided into a pretask task that learns a visual representation with a largely unlabeled data set, and a target task that collectively refers to ordinary computer vision tasks that generally expect good performance from the model. To be good at pretext tasks, networks need to have a good understanding of the model. The model then has the ability to encode images very precisely, which can be used as a good quality encoder that leads to high performance gains in typical computer vision tasks.

2. Solving Image Reconstruction Task

To learn the visual representation, we need to design an appropriate pretext task. We thought that if the network could look at part of the image and reconstruct the whole image, it would be able to get a good understanding of the image and learn a good quality visual representation. As we can see in Fig 1, the original image (left) is cut horizontally and vertically once (red box), and the middle area of a certain ratio (sky box) is cut out to create an input image. After all, if we take an input image and produce a ground truth iamge (bottom) or something similar, we can say that the network have solved this task well.

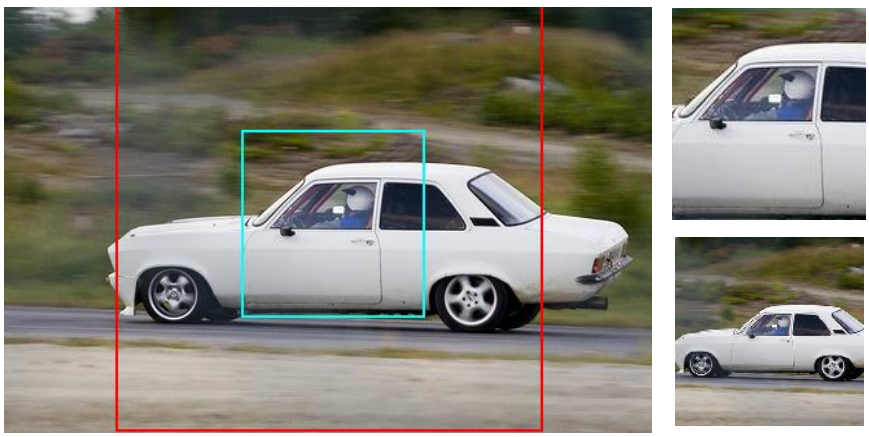

Fig. 1: The samples of original image (left), input image (top), ground truth image (bottom).

2.1. Generating scale and bias image

The ideal direction we want is to reconstruct images containing objects similar to ground truth. To do that we've tried a bit of tricks: using scale image and bias image (in Fig. 2) to help the generator to focus on only rest of center region that occupied by the input image originally. As a result, the generator only needs to reconstruct the portion of the groundtruth image except for the area occupied by the input image, and the learning process can be made more advantageous by reducing the size of the image to be generated. To apply the above process, we define scale image α and bias image β:

$$I_f = \alpha \cdot I_{in} + \beta \quad (1)$$

Fig. 2:The samples of scale image (left) and bias image (right)

2.2. Network Architecture

Our GAN [1] consists of Generator part of AutoEncoder [2] structure and Discriminator of CNN structure. When the input image enters the generator blocks, features are extracted by convolution filters, and the number of channels is increased as the height and width of the feature map decreases in the encoder section of the generator to minimize information loss. After the encoder, the upsampling process is performed while passing through the decoder part of the generator. The final output is the same size as the input image. Eventually, the output image produced by the generator is multiplied by the scale image first, the bias image added, and put into the input of the Discriminator, as mentioned in Section 2.1.

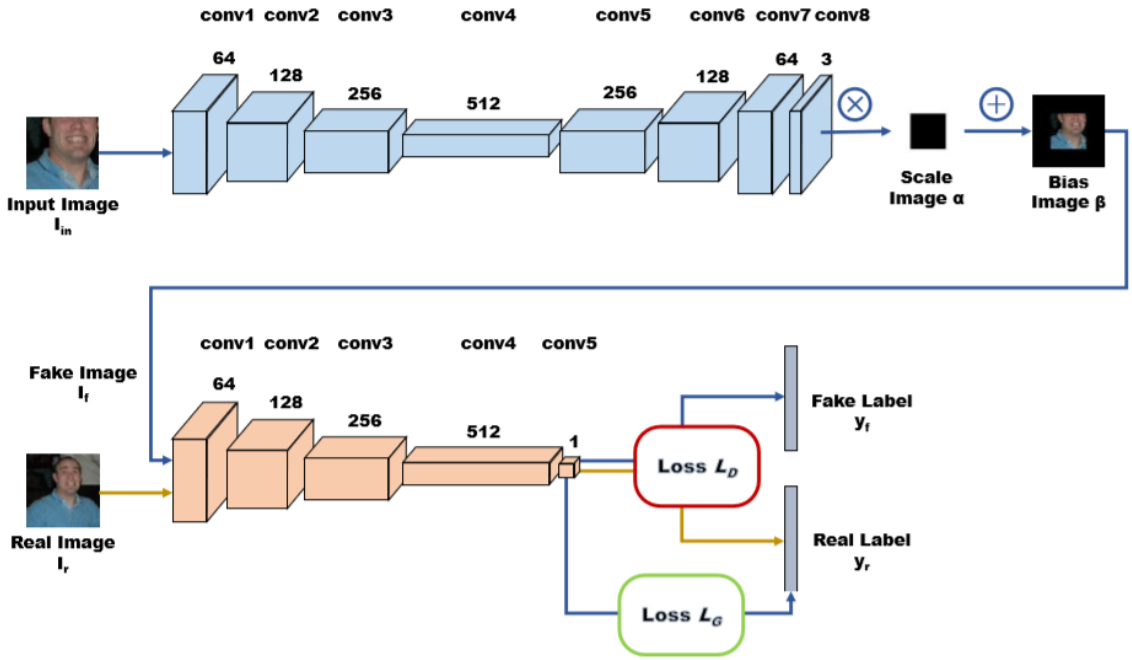

Fig. 3: Overall architecture of our model.

2.3. Learning the Model

Image Adversarial Loss. At least two losses are required to train the GAN. That is Generator loss L_G and Discrinimator loss L_D. As shown in Fig. 3, Generator Loss L_G is defined as binary cross entropy loss between real label which is the vector filled with 1 and the output of discriminator which the input is Fake Image generated by the generator, where $L_{bce}(\hat{z}, z) = -z\ln(\hat{z}) - (1-z)\ln(1-\hat{z})$. Discriminator Loss L_D is defined as the summation of two binary cross entropy losses which the pairs of input is (fake image, fake label) and (real image, real label). Therefore, Generator Loss and Discriminator Loss are defined as follows,

$$L_G(G,D,I) = L_{bce}(D(I_f), 1) \quad (2)$$

$$L_D(G,D,I) = L_{bce}(D(I_f), 0) + L_{bce}(D(I_r), 1) \quad (3)$$

$$L_I(G,D,I) = L_G(G,D,I) + L_D(G,D,I) \quad (4)$$

Identity Loss. In order to guarantee the central part of the image generated by the generator of the GAN, as mentioned in Section 2.1, using scale and bias images. However we also proposed the Identity loss which help generator to guarantee the central part of the image generated without scale and bias image. The loss is defined as the L1 divergence between the real image and the generator's output $G(I_{in})$. The loss is as follows.

$$L_{idt}(G,I) = ||G(I_{in}) - I_r||_1 \quad (5)$$

Full Loss. The total loss for training our proposed GAN is expressed as the sum of the two losses as suggested below.

$$L = L_I(G,D,I) + \lambda_{idt} L_{idt}(G,I) \quad (6)$$

where λ_{idt} has a value of 0 or 1 depending on the training setting.

2.4. Implementation Details

Our generator is based on the variation of AutoEncoder in Colorization [3], which has shown good performance in self-supervised visual representation learning. The characteristic of this generator is that there is no pool layer. Every conv layer is preceded by a convolution filter for upsampling or downsampling according to the role of the layer. After that, a convolution filter is placed that computes the feature map while maintaining its width and height. Next, batch normalization is placed in all layers except the last layer to improve training stability. Finally, in case of downsampling layer, Leaky ReLU and ReLU are placed in case of upsampling layer. Especially in the case of conv4, which extracts the feature of the largest number of channels from the middle layer, shown as Fig3, Dropout (rate = 0.5) is placed on the ReLU front end to prevent overfitting, and in the last layer, the Tanh function is placed as a nonlinear function. The discriminator used the discriminator structure used in DCGAN [4].

Our model was trained using the VOC2012 dataset [5]. Center crop the VOC2012 training image to the length of the smaller side, either width or height. Then resize it to 80 * 80, which is the ground truth image of our proposed image reconstruction task. After center cropping by the method described above, center crop so that it has one third of the length of both sides, and then resize it to 80 * 80, which is the input image of image reconstruction. There are two settings for training the model. The total loss mentioned in Section 2.3 is the case where λ_{idt} is 0 and 1. When $\lambda_{idt} = 1$, the scale and bias image are not used during training. On the other hand, when $\lambda_{idt} = 0$, the scale and bias image are used in training process. We set up a Adam optimizer with a learning rate of 0.0005 for the generator and 0.0001 for the discriminator, and set the weight decay to 0.00001. The batch size is 512, trained for 1000 epochs. The model takes 12 hours to train with a double GeForce GTX 1080 Ti GPUs.

3. EXPERIMENTS

3.1. Baseline

The GAN model we proposed earlier in Fig. 3 is a network for visual representation learning. What we ultimately want to show is that the Encoder learns a good visual representation in the Generator section of the GAN, improving performance in the usual visual recognition task. Finally, we present a baseline model of image classification training with a network with two fully connected layers attached to an encoder that has learned the image reconstruction task.

3.2. Datasets

As mentioned earlier, our GAN model is trained by the VOC2012 dataset. We import only the GAN Encoder and build the baseline model proposed in Section 3.1 and evaluate visual representation learning through image classification task. Evaluated by three datasets, CIFAR-10 [6], SVHN [7] and STL-10 [8]. Details are given in Table 1.

Dataset	*CIFAR-10*	*SVHN*	*STL-10*
# classes	*10*	*10*	*10*
# training images	*50000*	*73257*	*5000*
# testing images	*10000*	*26032*	*8000*
# classes/batch	*32*	*32*	*32*
eval. metric	*top 1*	*top1*	*top 1*

Table 1: The statistics of the datasets used in our experiments.

3.3. Experimental Settings

We performed the experiment with a total of three different settings including the baseline. Details are shown in Table 2. The baseline is the case where the output of the generator shown in Figure 3 is used as the input of the discriminator without any processing. In this case, the baseline + L1 is the case where the L1 loss between the groundtruth image and the output is added to the generator loss. Finally, as in the overall process shown in Figure 3, the case of identity loss using scale image and bias image is considered as baseline + scale & bias. All experiments were conducted for each setting.

Experiment ID \ option	*using L_{idt}*	*using scale & bias image*
baseline	✗	✗
baseline + L1	✓	✗
baseline + scale & bias	✗	✓

Table 2: Experimental settings for visual representation learning.

3.4. Reconstruction quality

We evaluate the performance qualititively using output image of our proposed network (see Fig. 4). In the case of the baseline model, the input image is often created as it is. With the addition of L1 loss, the central region is preserved like the center of the ground truth image, and the color is restored to some extent similar to the rest of the ground truth image. In the case of using scale and bias images, the center area is restored to the center of the ground truth, and the color is adjusted to the other areas, but the blur is severe.

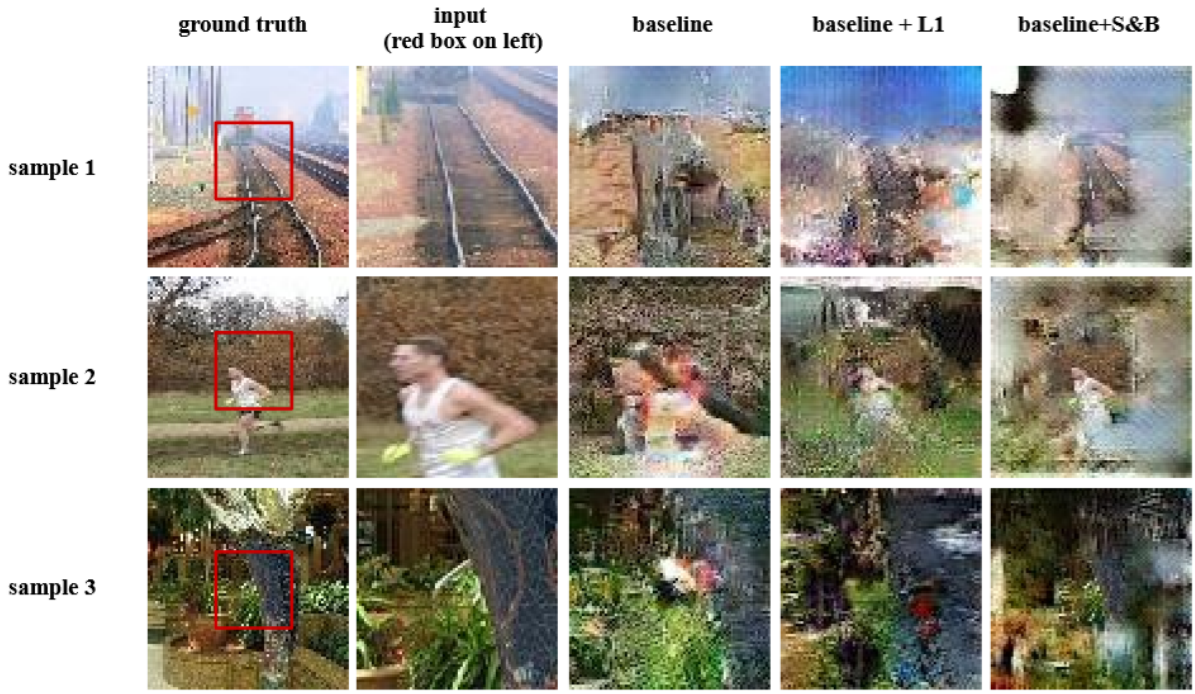

Fig. 4: Quantitive result for each experimental settings

3.5. Evaluating Visual Representation

The visual representation learned earlier was evaluated through a classification task. For all three datasets, we did not get a higher top1 accuracy than from the scratch. The three settings we tested showed nearly similar results, but baseline + s & b showed slightly better performance than the rest of the settings for all datasets. The detailed values are shown in table 3.

Dataset	*from the scratch*	*baseline*	*baseline + L1*	*baseline + s&b*
CIFAR-10	*76.51% ± 0.43*	*68.47% ± 0.87*	*68.67% ± 1.26*	*70.46% ± 0.99*
SVHN	*92.03% ± 0.35*	*89.58% ± 0.47*	*88.55% ± 0.74*	*90.39% ± 0.49*
STL10	*11.76% ± 0.56*	*10.81% ± 0.71*	*10.92% ± 0.64*	*11.91% ± 0.88*

Table 3: Performance Evaluation of Encoder Learned Visual Representation: Classification Task

4. CONCLUSIONS AND FUTURE WORK

We trained the visual representation through a pretext task that is our own image reconstruction task. The pretext task we proposed expected to learn a better visual representation without the cost of data labeling. However, as mentioned earlier in Section 3.5, we found that the performance was lower than the model we learned from the scratch. For that reason, as we saw in Section 3.4, the difficulty of our pretext task is so high that the reconstruction quality is significantly reduced. In the end, this means that visual representation learning through our challenging pretext tasks was not very good.

As we suggest in the future, we can extract input sizes larger than one-third of the original, or perform pretext tasks on a Stanford Dogs[9] dataset that contains only samples containing one instance, So that we will devise a way to reduce the difficulty of the pretext task. Also, in order to prevent unstable training process of GAN, additional measures such as WGAN and WGAN-GP will be applied. As a result, if we can achieve good visual representation learning, we will evaluate not only classification but also other computer vision tasks such as detection and segmentation.

5. REFERENCES

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley,Sherjil Ozair, Aaron Courville, Generative Adversarial Nets. Yoshua Bengio. In NIPS, 2014.

[2] Pierre Baldi. Autoencoders, Unsupervised Learning, and DeepArchitectures. In ICML

workshop, 2012.

[3] Richard Zhang, Phillip Isola, Alexei A. Efros. Colorful Image Colorization. In ECCV, 2016.
[4] Alec Radford, Luke Metz, Soumith Chintala. Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks. In ICLR, 2016.

[5] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. In IJCV, 2010.

[6] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images, 2009.

[7] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. Deep Learning and Unsupervised Feature Learning Workshop, NIPS.

[8] Adam Coates, Honglak Lee, Andrew Y. Ng An Analysis of Single Layer Networks in Unsupervised Feature Learning AISTATS, 2011.

[9] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 2011.