25th INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION
Milan, Italy 10 | 15 January 2021

ICPR 2020

Image Representation Learning by Transformation Regression

Xifeng Guo, **Jiyuan Liu*,** Sihang Zhou, En Zhu, Shihao Dong

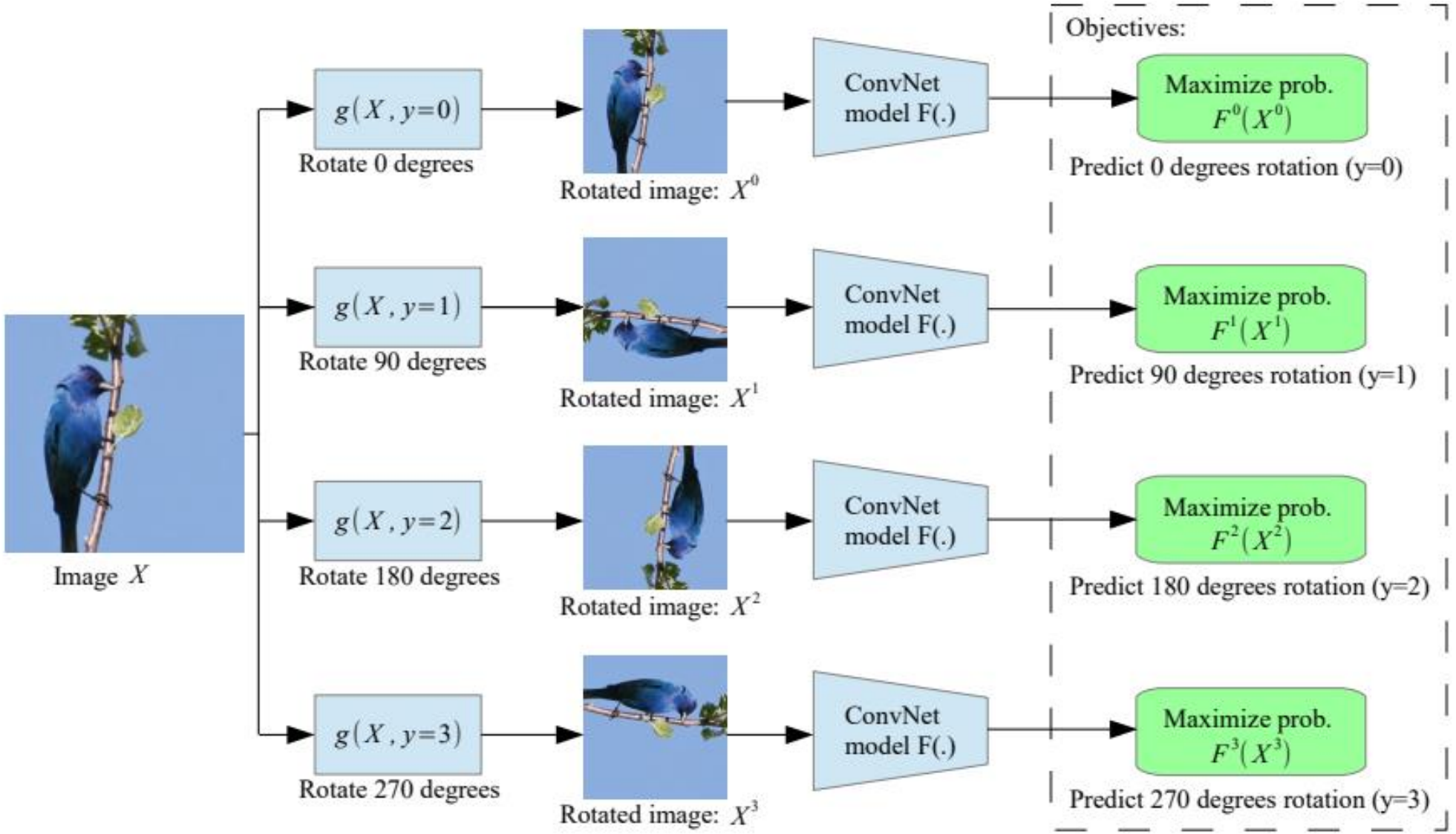
National University of Defense Technology

Related Work

- Current self-supervised representation learning methods can be roughly separated into two categories, i.e. content based and manipulation based representation learning.
- The content based methods take advantage of data itself as the supervisory signal. They are also known as conventional unsupervised representation learning, like AE, VAE and GAN.
- The manipulation based methods define a group of manipulations (e.g., image rotation, color jitter, and inpainting) employed on images and generate corresponding labels.

Related Work

- For example

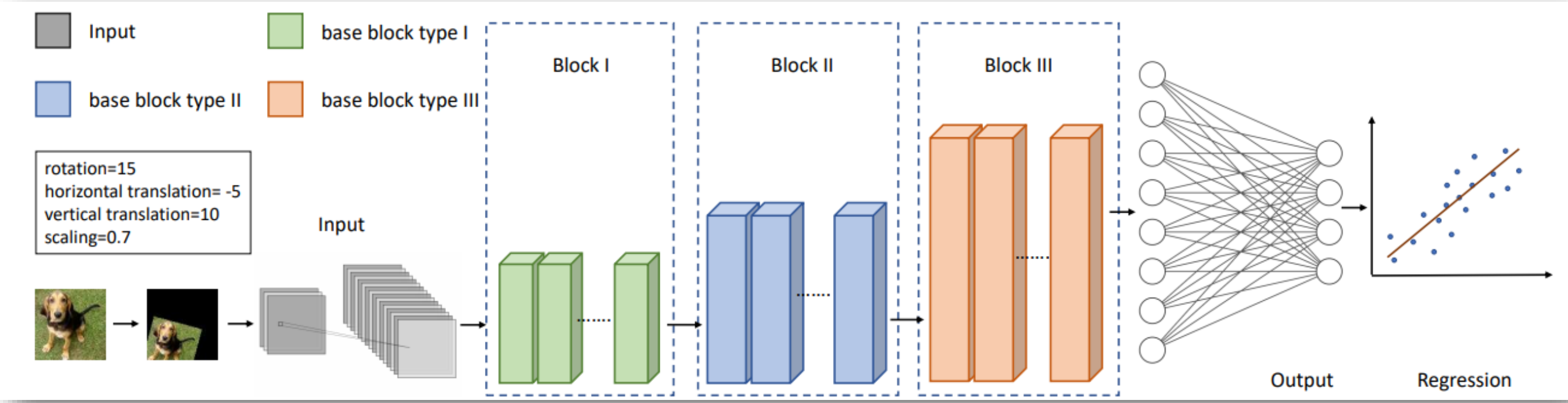


Motivation

- Self-supervised learning learns representations by developing an auxiliary learning task.
- Existing methods usually model the auxiliary learning tasks as classification tasks with finite discrete labels.
- Insufficient supervisory signals restrict the learning ability.
- To make full use of the supervision from data, we design a regression model to predict the continuous parameters of a group of affine transformations.

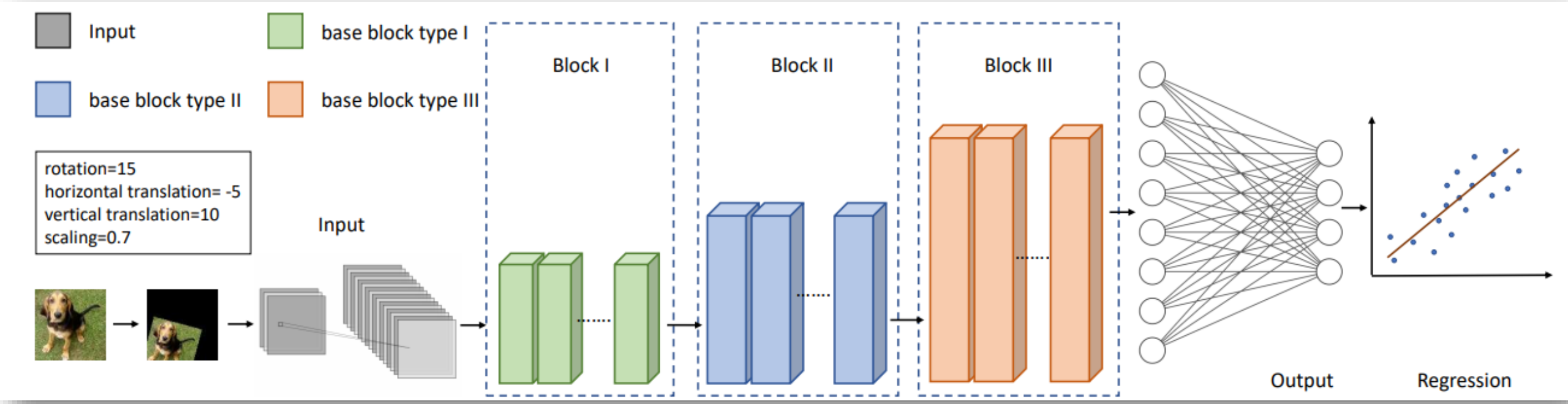
Method

- Transform images and generate continuous labels.
- Train the neural network by the regression task.



Method

$$\min_{\mathbf{W}} \frac{1}{n} \sum_{i=1}^{n} \left\| \mathcal{F}\left(\mathcal{T}\left(\mathbf{x}_i; y_i\right); \mathbf{W}\right) - y_i \right\|_2^2$$



Method

$$\min_{\mathbf{W}} \frac{1}{n} \sum_{i=1}^{n} \left\| \mathcal{F}\left(\mathcal{T}\left(\mathbf{x}_i; y_i\right); \mathbf{W}\right) - y_i \right\|_2^2$$

Algorithm 1 Image Representation Learning Algorithm by Transformation Regression

Input: Image Dataset X; Transformation function $\mathcal{T}(\cdot, y)$
Neural Network $\mathcal{F}(\cdot, \mathbf{W})$.

Output: The parameters of the neural network $\mathbf{W}$.

Initialize the neural network parameter $\mathbf{W}$;
for t in 1 to T do
  for i in 1 to n do
    Random sample a value y_i;
    Apply transformation: $\hat{\mathbf{x}}_i = \mathcal{T}\left(\mathbf{x}_i; y_i\right)$.
    Forward pass to get the output $\mathcal{F}\left(\hat{\mathbf{x}}_i; \mathbf{W}\right)$.
  end for
  Compute the loss $L = \frac{1}{n} \sum_{i=1}^{n} \left\| \mathcal{F}\left(\hat{\mathbf{x}}_i; \mathbf{W}\right) - y_i \right\|_2^2$.
  Update the W by gradient descent.
end for
return $\mathbf{W}$.

Experiment

- Datasets

	#examples	#training examples	#testing examples	#classes	image size
CIFAR10	60,000	50,000	10,000	10	$32 \times 32 \times 3$
CIFAR100	60,000	50,000	10,000	100	$32 \times 32 \times 3$
STL10	13,000	10,000	3,000	10	$32 \times 32 \times 3$
SVHN	99,289	73,257	26,032	10	$32 \times 32 \times 3$

- Protocol
  - Train a network in a self-supervised way to learn representations.
  - Extract representations from different layers.
  - Use a classification model to validate the quality of representations (higher accuracy corresponds to better quality)

Experiment

- Evaluation
  - Classification accuracy of a MLP on representations of different levels
- Observation
  - The method can learn good representations comparing Input with Block_i
  - The intermediate representation (Block_2) is the best.
  - More samples lead to better representation comparing STL10-10k and -100k

	Input	Block_1	Block_2	Block_3
CIFAR10	66.84	81.37	84.88	76.16
CIFAR100	39.60	53.19	57.06	44.07
STL10-10k	54.20	68.53	72.67	64.33
STL10-100k	54.20	71.00	77.03	67.80
SVHN	82.25	92.29	94.28	91.52

Experiment

- Evaluation
  - Classification accuracy of a CNN on representations of different levels
- Observation
  - The method can learn good representations comparing Input with Block_i
  - The intermediate representation (Block_1) is the best.
  - More samples lead to better representation comparing STL10-10k and -100k

	Input	Block_1	Block_2	Block_3
CIFAR10	93.37	93.65	90.75	81.44
CIFAR100	70.84	71.42	65.69	49.01
STL10-10k	77.93	79.93	77.20	65.50
STL10-100k	77.93	81.03	79.60	67.17
SVHN	96.12	96.54	95.67	92.63

Experiment

- Compare with SOTA
  - Supervised: Train the network with ground-truth labels then use a CNN to classify the learned representations of different levels.
  - Our method outperforms the other self-supervised methods and approaches the supervised method.

	CIFAR10	CIFAR100	STL10	SVHN
DCGAN* [12]	82.80	–	–	–
Split-Brain† [27]	67.10	39.00	–	77.30
Counting† [28]	50.90	18.20	–	63.40
AND† [21]	77.60	47.90	–	93.70
RotNet* [15]	91.16	–	–	–
TR (Ours)	93.65	71.42	79.93	96.54
Supervised	94.92	75.76	80.10	96.45

Experiment

- Ablation Study
  - Use different transformations to train the network.
  - The composition of any two types of transformations leads to good performance
  - All three types of transformations achieve the best performance.

Rotation	Translation	Scaling	ACC (%)
0	0	[0.5, 1.5]	90.97
0	[-10, 10]	1.0	87.97
0	[-10, 10]	[0.5, 1.5]	91.48
[-180, 180]	0	1.0	87.99
[-180, 180]	0	[0.5, 1.5]	92.21
[-180, 180]	[-10, 10]	1.0	93.24
[-180, 180]	[-10, 10]	[0.5, 1.5]	**93.65**

Conclusion

- We propose a new image representation learning method by constructing a regression task whose target is to predict the continuous parameters of some transformations applied to the input image.
- Extensive experiments on various image datasets validate the effectiveness and discriminability of representation learned by our proposed transformation regression method.
- Future work
  - Exploring other types of transformations like image flipping, cropping, and color jitter.
  - Eliminating the edge effect (artifact) when applying some transformations like image rotation.

THANK YOU!