Unsupervised Representation Learning by Predicting Image Rotations

Spyros Gidaris, Praveer Singh, Nikos Komodakis

Paper Published at ICLR 2018

Presented by:

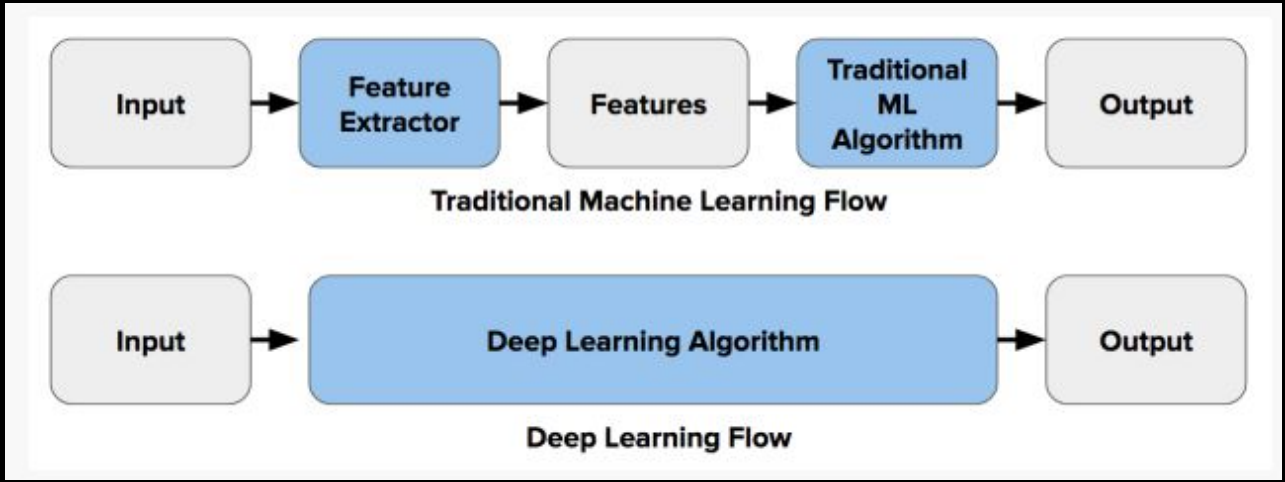
Arka Mukherjee

Balaji Balasubramanian

September 14th, 2021

Success of Convolutional Neural Networks

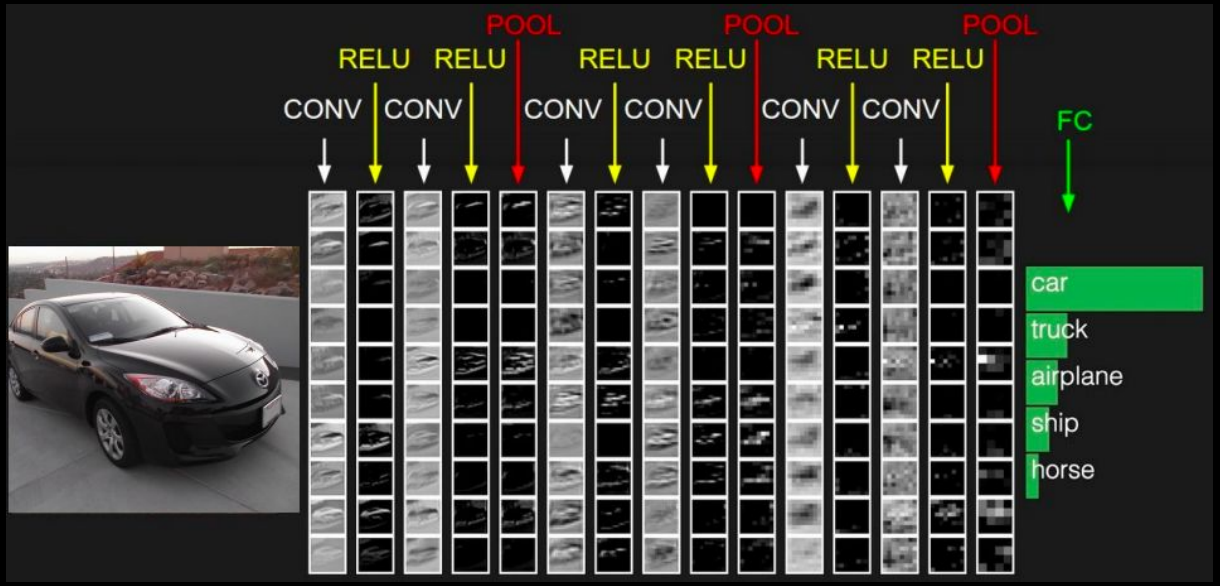
- **Deep Learning models like CNN have outperformed Classical Machine Learning approaches for Computer Vision that consists of a feature extractor following by ML algorithms such as Support Vector Machines (SVMs), KNN.**



https://naadispeaks.wordpress.com/2018/08/12/deep-learning-vs-traditional-computer-vision/

Success of Convolutional Neural Networks

Standard CNN architecture for image classification:



Success of Convolutional Neural Networks

- **CNN has also done well for more advanced computer vision tasks such as object detection image captioning or semantic segmentation.**

Motivation for SSL: But getting a hold of labels is costly! Creating massively labelled dataset on the scale of ImageNet is next to impossible!

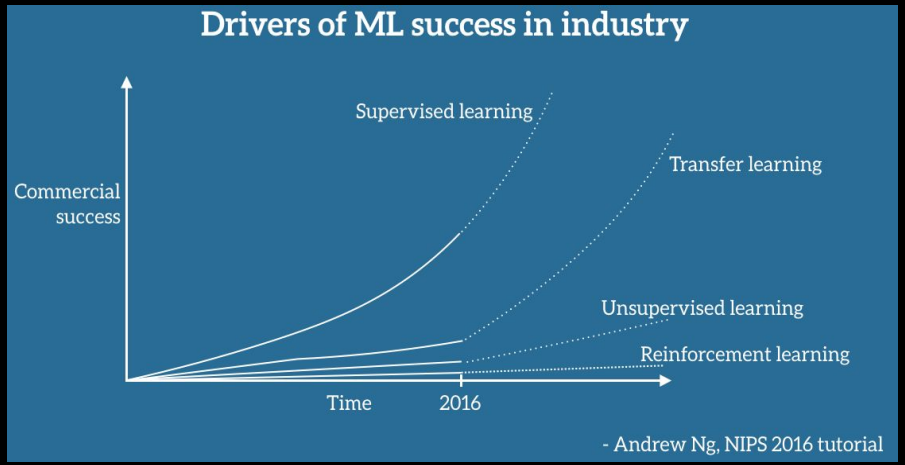
Can we get comparable results in Vision tasks without spending on labels?

Yes, we can define a pretext task. This way labels would be generated from the input data without human annotation.

Self Supervised Learning Motivation

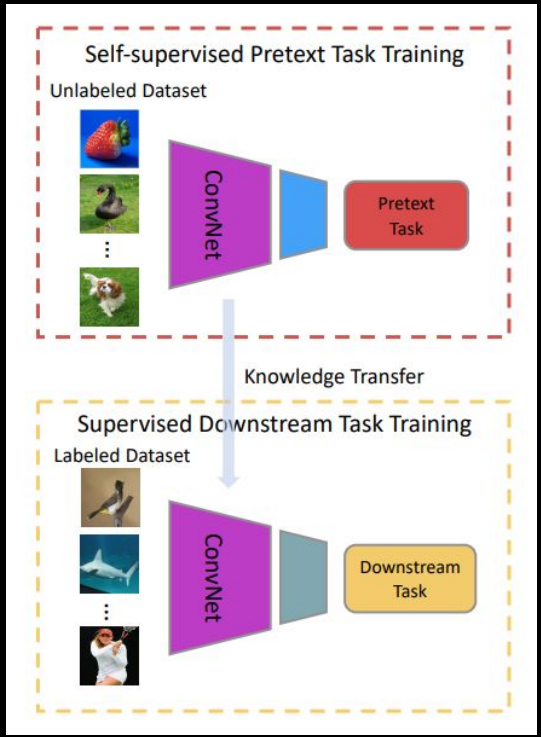
Motivations:

- Labelled datasets are expensive and require a lot of manual labour.
- Humans and other animals do not learn using supervised learning.
- Tasks like **Supervised Transfer Learning** for Image Classification (Using a supervised learning backbone) for another completely different supervised learning task perform great due to image patterns having a similar nature.



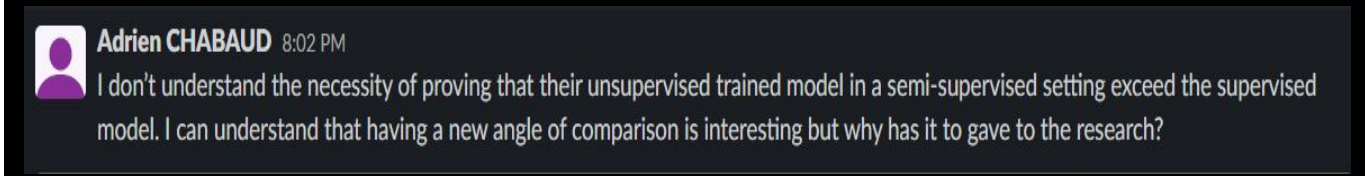
Transfer Learning was the next big DL methodology around the time of the paper, and SSL is a derivative of TL.

Self Supervised Learning - Pretext



- All the data leveraged for the training task is present in the image itself (Pseudo-labels are retrieved from the image).
- The learned weights from the pretext tasks usually do well for a task in a practical scenario using knowledge transfer (re-using learned weights and biases).

Motivation Question:

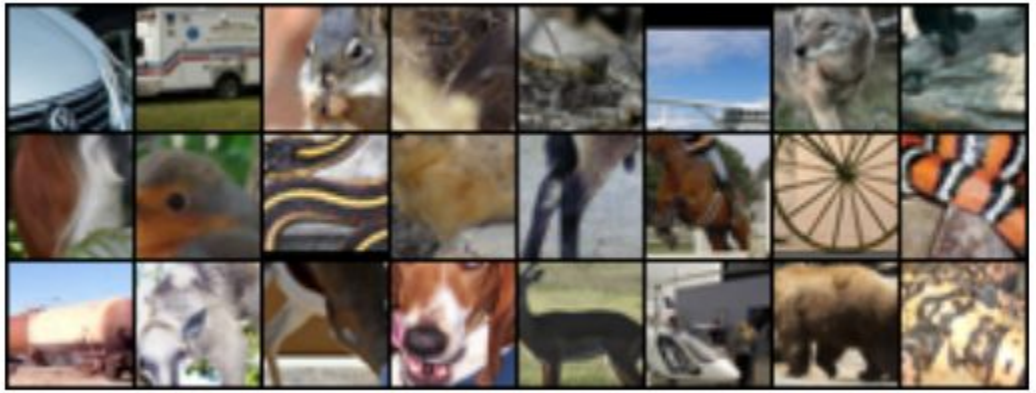


Reference: Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey, Longlong Jing and Yingli Tian* , Fellow, IEEE

Other pretext tasks in self-supervised learning

Similar Pretext tasks have been discussed the past few weeks, examples are:

Surrogate Class Prediction
Surrogate class = patch+augment



Other pretexts in self-supervised learning

Similar Pretext tasks have been discussed the past few weeks, examples are:

Context Prediction

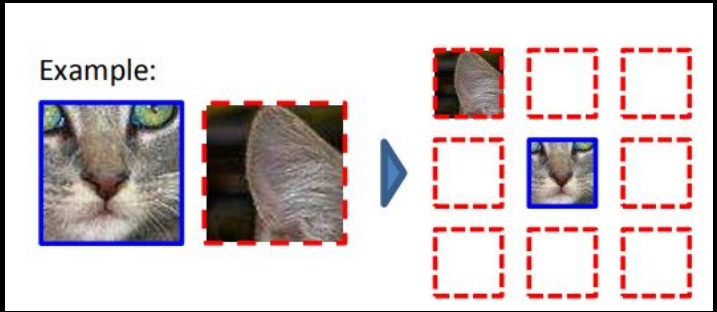
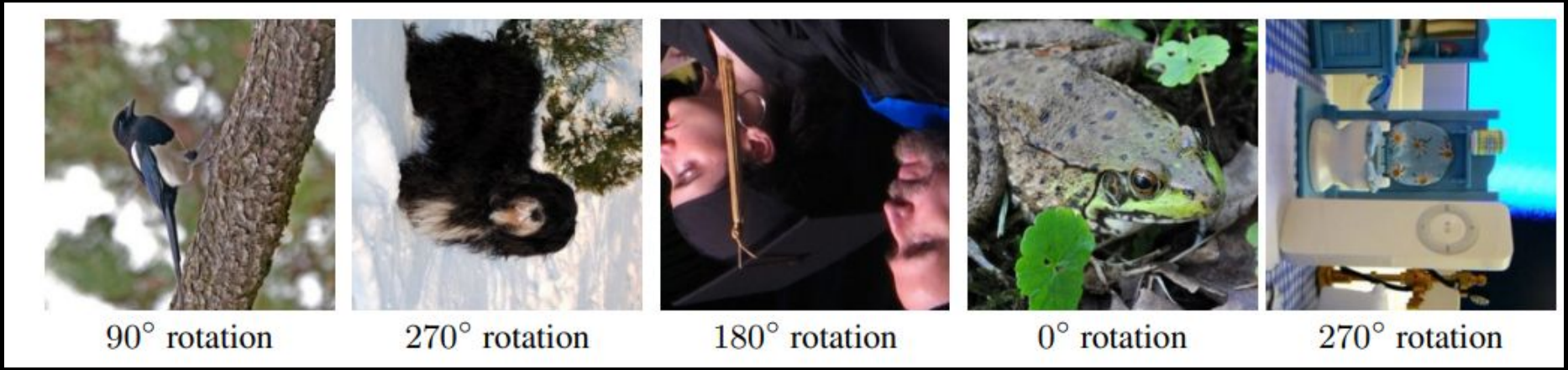


Image Inpainting

Pretext task in this paper

In this paper, the pretext task is predicting the image rotations (for simple images), and attempting to transfer the knowledge to other domains. Our goal is for the model to learn the context of the object itself, in this self-supervised learning process.



NOTE: One notable improvement could be the method to add rotations, the paper relies on only flips and transposes. We can instead use the rotation operation from Pillow and Open CV.

Mathematical Formulation

The Convnet model in question is given by $F(\ .)$

Transformations are given by: G

$$= \{g(.\,|y)\}_{y=1}^{k}$$

k = 1 to k = 4 refers to the transformations from 0 to 270 (0, 90, 180, 270), as a measurement in degrees.

If X is the image, and the geometric transformation is y, we can note the following is the transformation:

$$X^y = g(X|y)$$

Mathematical Formulation - continued

The function *g* is a geometric transformation function, which is formulated as:

$$g(X|y) = \text{Rotation}(X, (y-1)90)$$

Model predicts the probability distribution, which is formulated as:

$$F\left(X^{y*} \mid \theta\right) = \left\{F^{y}\left(X^{y*} \mid \theta\right)\right\}_{y=1}^{K}$$

Mathematical Formulation - Optimization

The Loss Formulation is given as follows (Cross Entropy Loss to maximize probability of a surrogate geometric transformation class, same flavour as Exemplar CNN):

$$\mathrm{DNN}_{objective} = \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathrm{loss}\left(X_i, \theta\right)$$

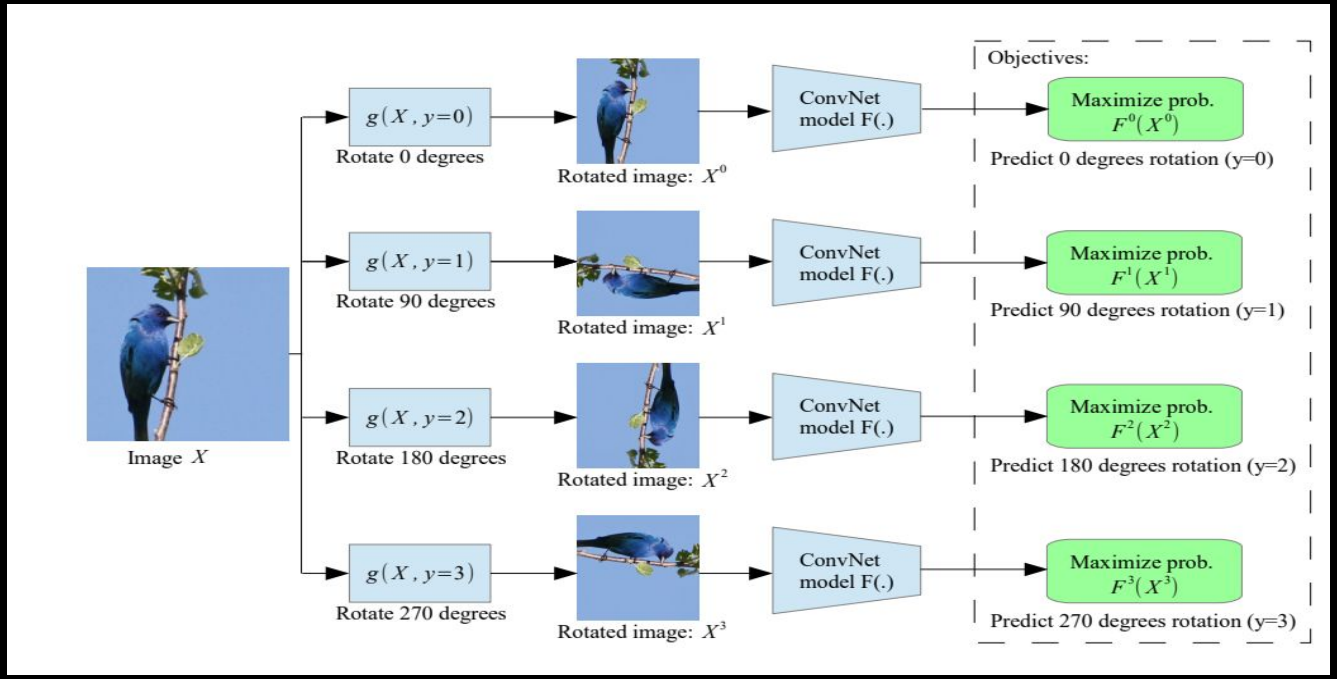
And, the Loss in the Objective is defined as follows:

$$\mathrm{loss}\left(X_i, \theta\right) = -\frac{1}{K} \sum_{y=1}^{K} \log$$

$$\left(F^y\left(g\left(X_i \mid y\right) \mid \theta\right)\right)$$

Self Supervised Methodology Overview

- Learn CONV features by maximizing the probability of the 4 different surrogate classes. Gist is given below:



Author's Experiments

CIFAR-10 Results:

Model	ConvB1	ConvB2	ConvB3	ConvB4	ConvB5
RotNet with 3 conv. blocks	85.45	88.26	62.09	-	-
RotNet with 4 conv. blocks	85.07	89.06	86.21	61.73	-
RotNet with 5 conv. blocks	85.04	**89.76**	86.82	74.50	50.37

The above metrics are observed when our given model (a.k.a RotNet) is trained with a non-linear classifier on top of it.

Author's Experiments

CIFAR-10 Results:

Model	ConvB1	ConvB2	ConvB3	ConvB4	ConvB5
RotNet with 3 conv. blocks	85.45	88.26	62.09	-	-
RotNet with 4 conv. blocks	85.07	89.06	86.21	61.73	-
RotNet with 5 conv. blocks	85.04	**89.76**	86.82	74.50	50.37

We believe the model performance get worse as we go deeper down the CNN blocks, as the model's inference task gets too 'specific' w.r.t. SSL pretext tasks.

Hyperparameters and Model Details

- Optimization: Vanilla Stochastic Gradient Descent using mini-batches (bs = 128).
- Momentum: 0.9
- Weight Decay: 5e-4
- Learning Rate: 0.1 (dropped by a factor of 5 after epochs 30. 60, 120).
- All 4 rotated copies of an image fed during training seemed to help the most.
- Based on Network in Network (NIN) - same architecture, but different learning methodology.

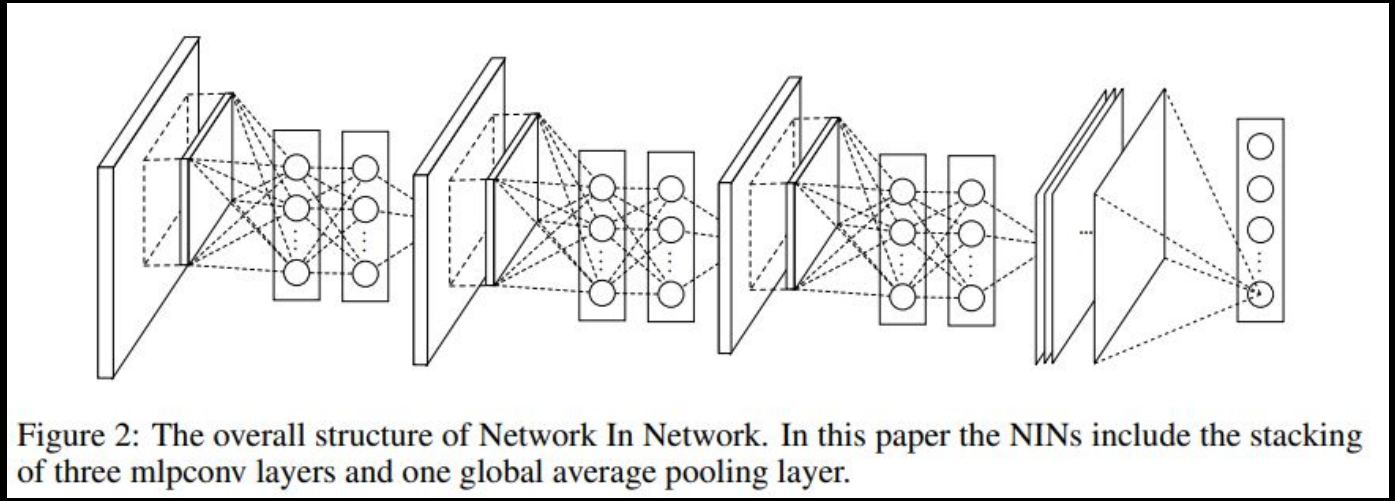


Figure 2: The overall structure of Network In Network. In this paper the NINs include the stacking of three mlpconv layers and one global average pooling layer.

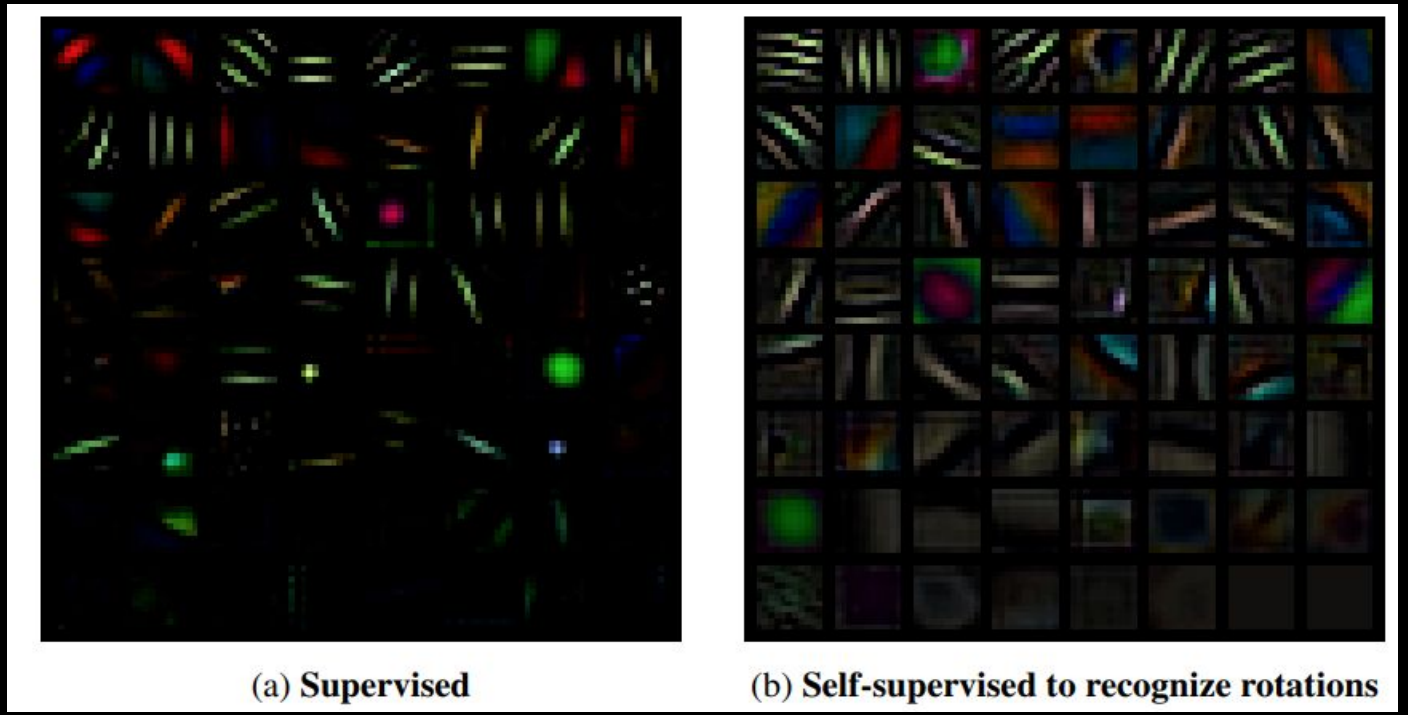
Image Reference: Network In Network, Min Lin, Qiang Chen , Shuicheng Yan

Angle for Geometric Transformation - analysis

# Rotations	Rotations	CIFAR-10 Classification Accuracy
4	0°, 90°, 180°, 270°	**89.06**
8	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°	88.51
2	0°, 180°	87.46
2	90°, 270°	85.52

Potential Reason for author's 8 Rotation Scenario being worse: Artifacting, or model learning dark pixel zones for classification.

A look at experimental filters - Supervised vs self-supervised



(a) **Supervised**

(b) **Self-supervised to recognize rotations**

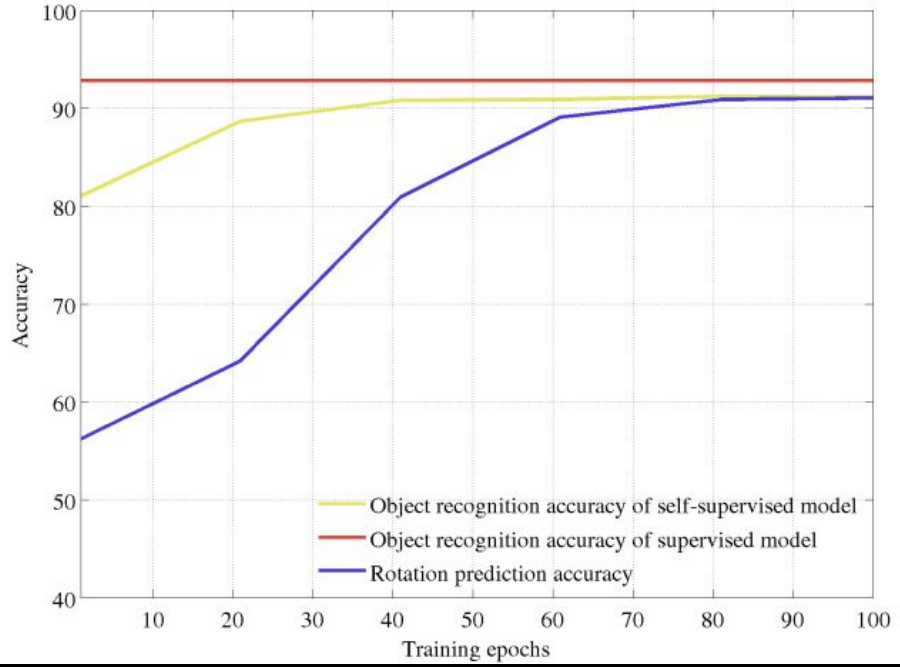
More defined Attention Maps - Self Supervised Learning

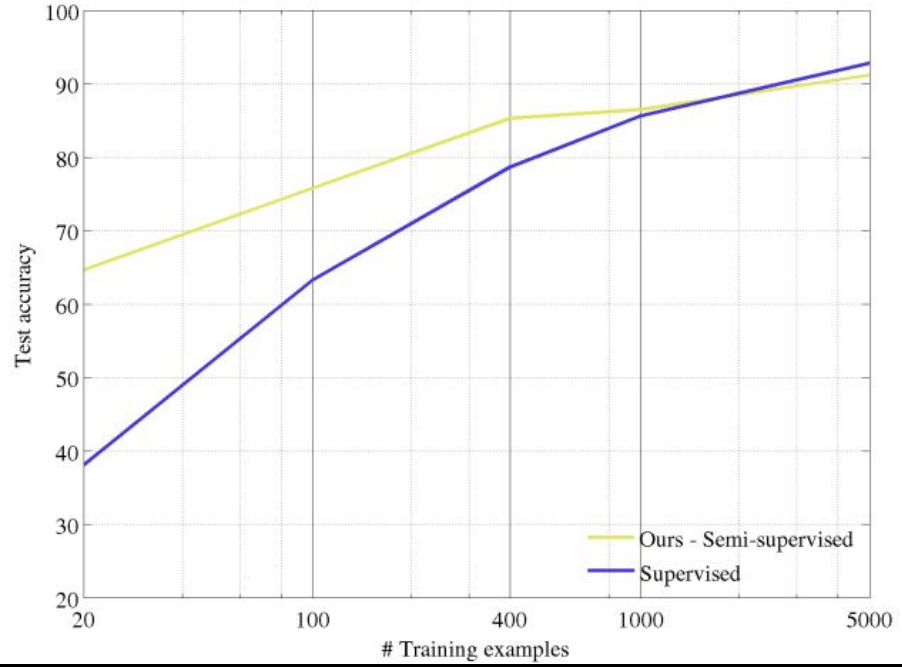
Supervised vs Self-supervised Attention Maps and Filters

The SSL task here has more context when it came to the objects that can help further in rotation, and in some cases, the filters + Attention Maps are more defined in the SSL case, and this is probably a benefit of the task being self-supervised. In supervised learning, we care more about the particular fed context than learning relevant filters.

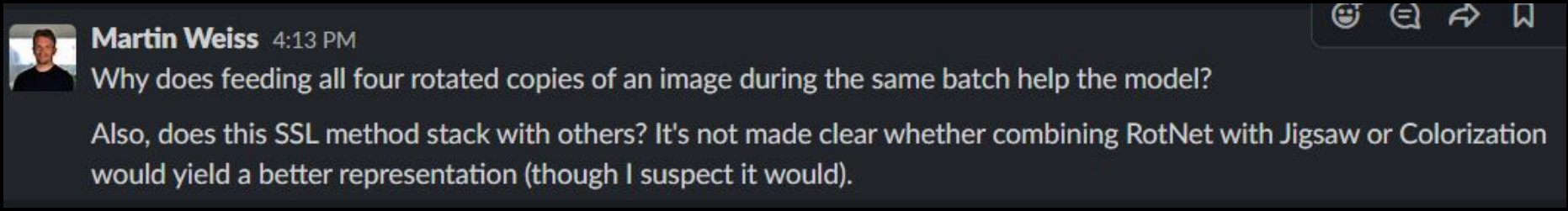
On a rough glance, it would seem like because some of the features obtained from the images have more 'context', self-supervised learning methods in general would be superior when it comes to task generalization (i.e backbones being more task agnostic) when pitted with supervised algorithms.

Considering SSL when you have less training time



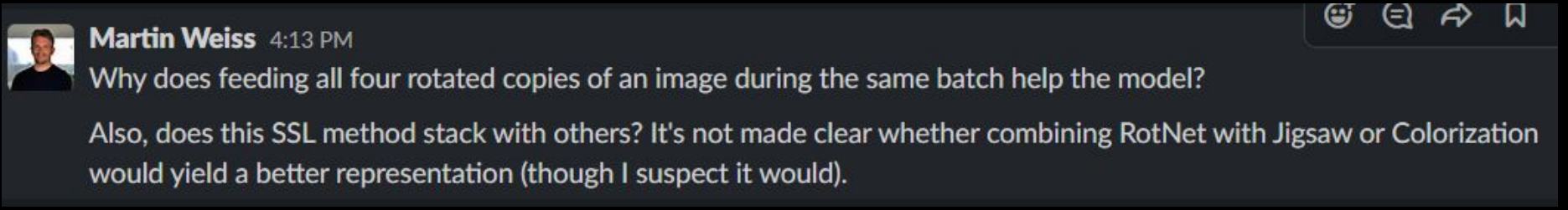


Questions and Answers!



- Batch follow-up: Since the goal is to learn robust features, 4 rotations of the same image per epoch improves the gradients during the model training. Imbalance of the labels between epochs can be a problem (eg: feeding images of 45 deg shift in epoch 1 + 90 shift in epoch 2).
- We could not find a paper that combines Jigsaw/Causality/Colourization + Rotation inference, but we found approaches that combine geometric approaches in the model backbone. We found a paper that combined rotations with other geometric transformations, and the result was better than having rotation supervisory signal alone.

Questions and Answers!



Martin Weiss 4:13 PM

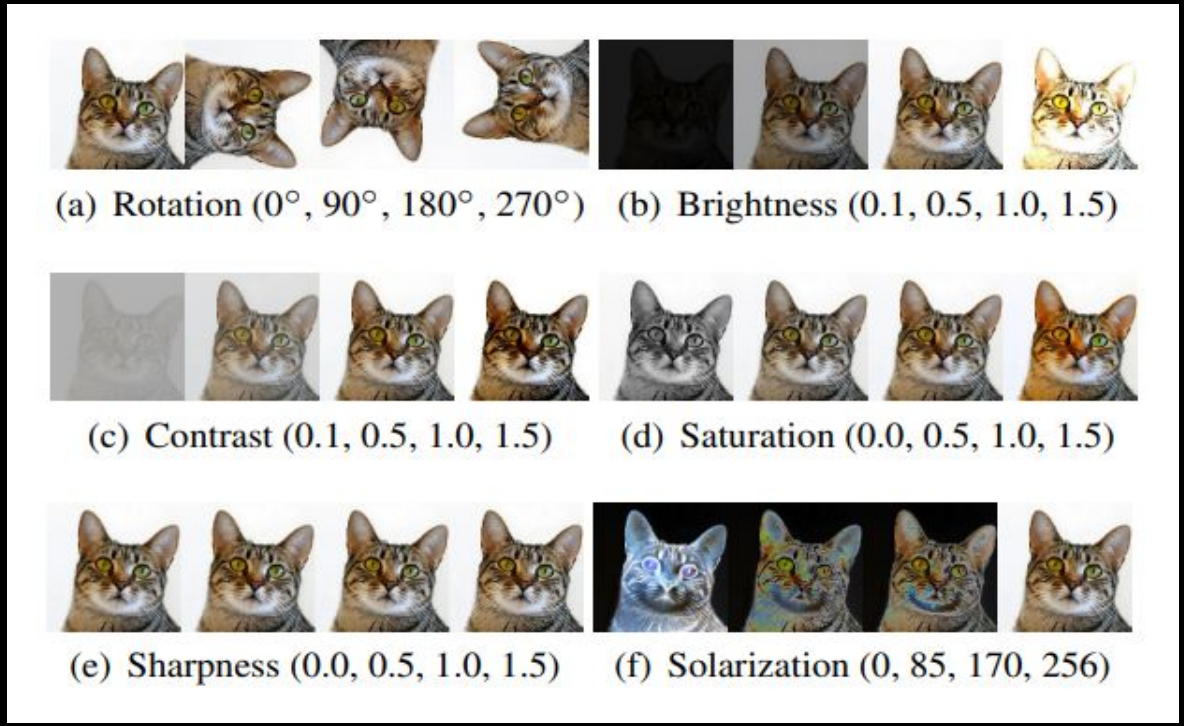
Why does feeding all four rotated copies of an image during the same batch help the model?

Also, does this SSL method stack with others? It's not made clear whether combining RotNet with Jigsaw or Colorization would yield a better representation (though I suspect it would).

Rotation Net performs better than Colorization and Jigsaw because the model gets to see the original image (0 deg rotation) which is close to the image the model will encounter in the downstream supervised task.

Method	Conv4	Conv5
ImageNet labels from (Bojanowski & Joulin, 2017)	59.7	59.7
Random from (Noroozi & Favaro, 2016)	27.1	12.0
Tracking Wang & Gupta (2015)	38.8	29.8
Context (Doersch et al., 2015)	45.6	30.4
Colorization (Zhang et al., 2016a)	40.7	35.2
Jigsaw Puzzles (Noroozi & Favaro, 2016)	45.3	34.6
BIGAN (Donahue et al., 2016)	41.9	32.2
NAT (Bojanowski & Joulin, 2017)	-	36.0
(Ours) RotNet	50.0	43.8

Paper - IMAGE ENHANCED ROTATION PREDICTION FOR SELF-SUPERVISED LEARNING

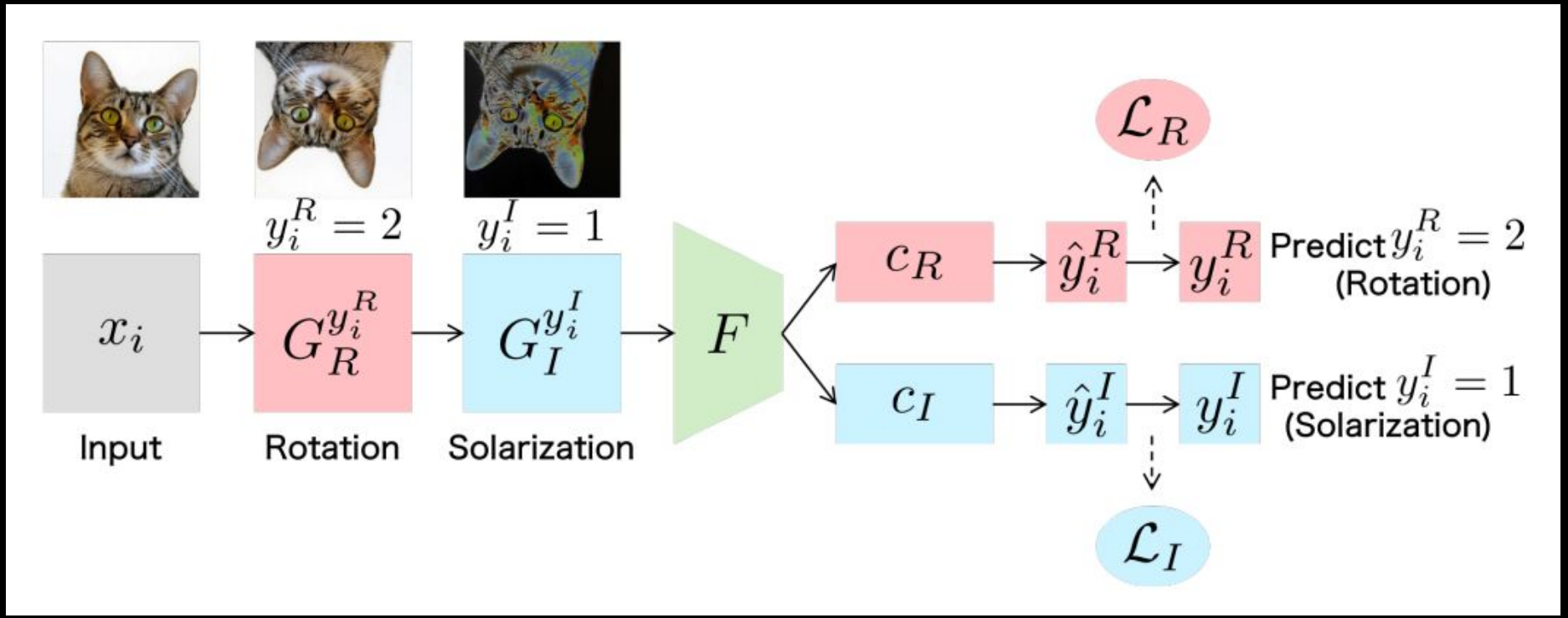


Paper - IMAGE ENHANCED ROTATION PREDICTION FOR SELF-SUPERVISED LEARNING

Table 1. Comparison of IE-Rot performances among multiple IEs on CIFAR-100. Each cell shows mean test top-1 accuracy of the linear classifier using feature maps generated by the pretrained (frozen) CNN (WRN-40-10, `block-3`).

Rotation	$43.0^{\pm 0.2}$
Rotation+Sharpness	$44.5^{\pm 0.4}$
Rotation+Brightness	$46.0^{\pm 0.6}$
Rotation+Contrast	$46.4^{\pm 0.9}$
Rotation+Saturation	$46.6^{\pm 0.1}$
Rotation+Solarization	$\mathbf{49.0}^{\pm 0.3}$

Supporting Paper Model Architecture



Supporting Paper Results

Table 4. Comparison of IE-Rot and Data Augmentation (DA). We used WRN-40-10 as the network architecture and Solarization as the IE.

	CIFAR-10	CIFAR-100	TinyImageNet
Rotation	$74.0^{\pm 0.5}$	$43.0^{\pm 0.2}$	$23.4^{\pm 0.3}$
Rotation + DA	$74.7^{\pm 0.1}$	$43.3^{\pm 1.2}$	$23.0^{\pm 0.4}$
IE-Rot	$\mathbf{75.4}^{\pm 0.2}$	$\mathbf{49.0}^{\pm 0.3}$	$\mathbf{26.1}^{\pm 0.6}$

IE-Rot performs better than the Rotation (RotNet) model.

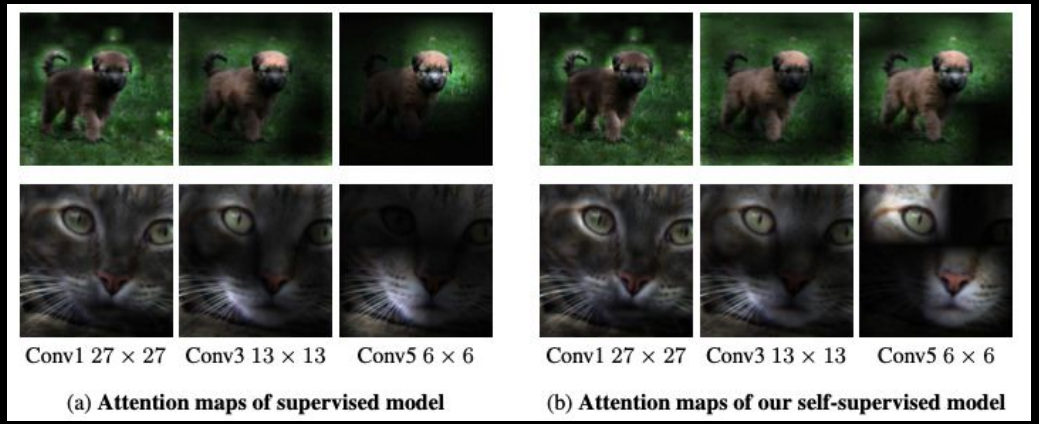
Questions and Answers!

Aniket Didolkar 7:40 PM

> Remarkably, these filters seem to have a greater amount of variety even than the filters learnt by the supervised object recognition task.

The authors say that one benefit of their method is that it learns attention maps with more variety, from fig 3 it seems that they mean that attention maps are more spread out. But do you think these maps are more of a consequence of rotations than a benefit? In the sense that the rotations cause the foreground object to cover many different spatial positions hence the overall attention maps would be more spread out by default and in the case of the supervised case the attention maps may be less spread out since for classifying an object it may require only a few features to differentiate between classes. What are your thoughts on this?

The supervised model only looks that the features that let it differentiate between the cat and the dog like looking at the whiskers of the cat and the face of the dog whereas the self supervised model looks at more areas of the image.



(a) **Attention maps of supervised model** (b) **Attention maps of our self-supervised model**

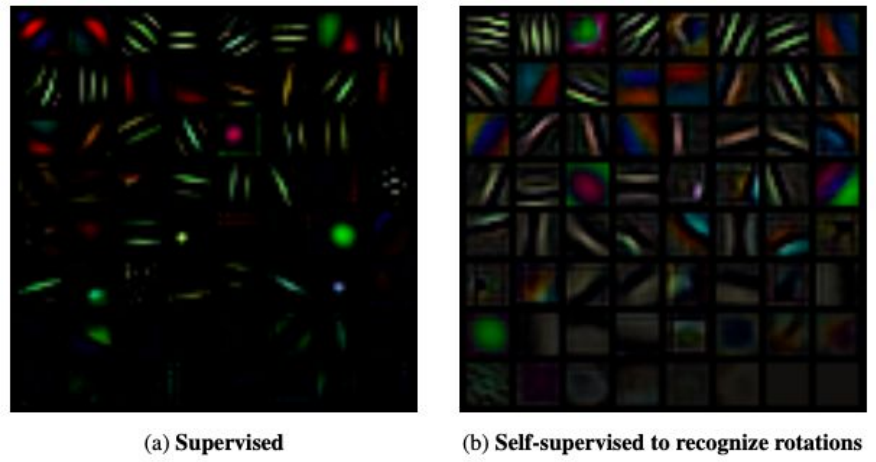
Questions and Answers!

Aniket Didolkar 7:40 PM

> Remarkably, these filters seem to have a greater amount of variety even than the filters learnt by the supervised object recognition task.

The authors say that one benefit of their method is that it learns attention maps with more variety, from fig 3 it seems that they mean that attention maps are more spread out. But do you think these maps are more of a consequence of rotations than a benefit? In the sense that the rotations cause the foreground object to cover many different spatial positions hence the overall attention maps would be more spread out by default and in the case of the supervised case the attention maps may be less spread out since for classifying an object it may require only a few features to differentiate between classes. What are your thoughts on this?

The filters learned by the first layer of the self supervised model are more varied/diverse compare the the filters learned by the first layer of supervised model.



(a) **Supervised** (b) **Self-supervised to recognize rotations**

Questions and Answers!

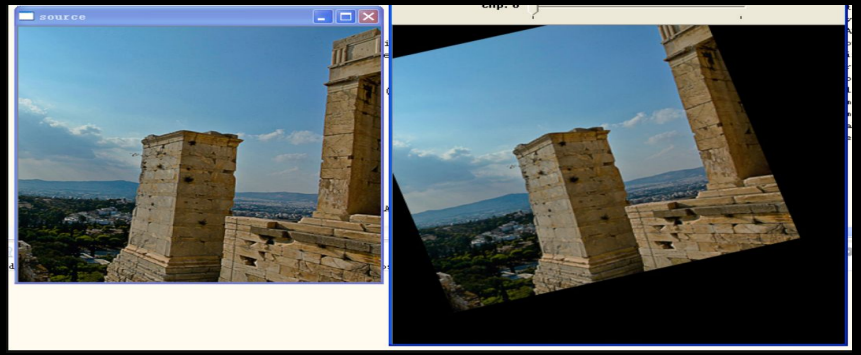
Naga Karthik 11:15 AM

The authors talk about "visual image artifacts", particularly,

> ... geometric transformations can be implemented by flip and transpose operations that do not leave any easily detectable low-level visual artifacts that will lead the ConvNet to learn trivial features with no practical value for the vision perception tasks.

Question 1 : What are some examples of these "low-level visual artifacts"?

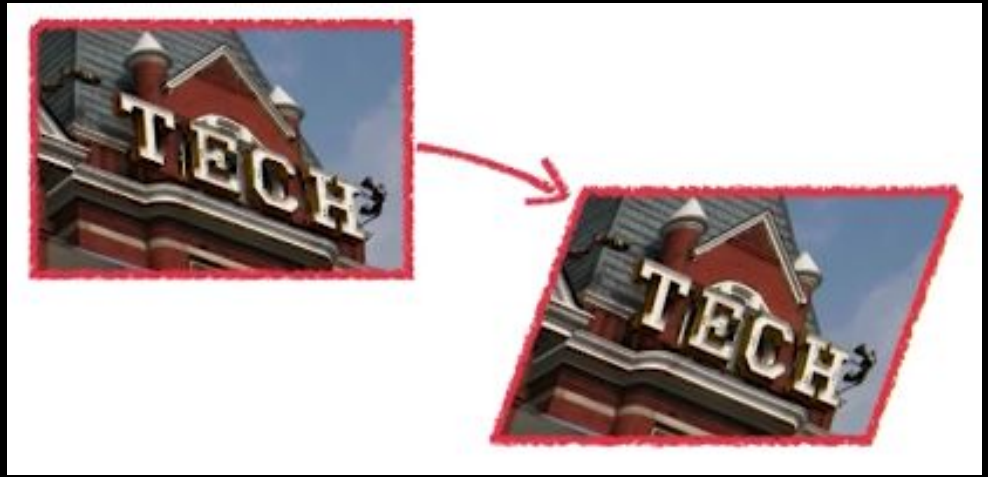
Secondly, in Table 2, the difference in CIFAR10 Classification accuracy for 4 and 8 rotations is just 0.55%, but the authors claimed that having 4 extra rotations "may lead to visual artifacts on the rotated images".



Questions and Answers!

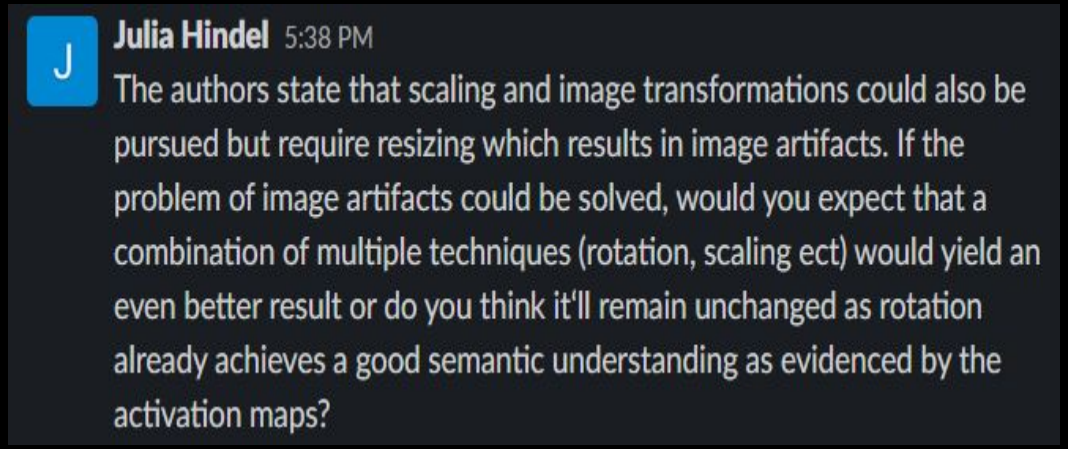
Question 2 : given the small percentage difference, it doesn't seem that these visual artifacts are not as bad as they are claimed to be, which suggests that the set of geometric transformations could at least be extended to include affine transformations. What are your thoughts on this? (edited)

If we add a simple affine transformation it will be easy to recognize the angle of rotation by looking at the boundary like we see in this image (i.e, the model can cheat). A technique like Nearest Neighbour Interpolation can be used to deal with the angle of the edges and it might work.



Udacity course "Computational Photography" https://www.youtube.com/watch?v=il6Z5LCykZk

Questions and Answers!

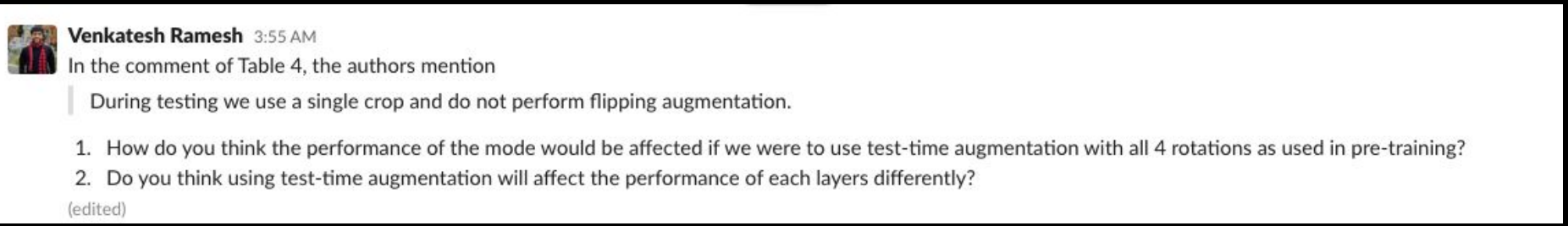


If scaling + aspect ratio transformations were possible to execute without visual artifacts, it would probably improve the performance in the downstream task .

As seen from the paper discussed in the previous slides while answering Martin's question, Rotation combined with another geometric transformation does seem to help.

In short: If we could leverage scaling/aspect ratio based SSL techniques without producing abysmal artifacts, would it help us?

Questions and Answers!



Venkatesh Ramesh 3:55 AM

In the comment of Table 4, the authors mention

> During testing we use a single crop and do not perform flipping augmentation.

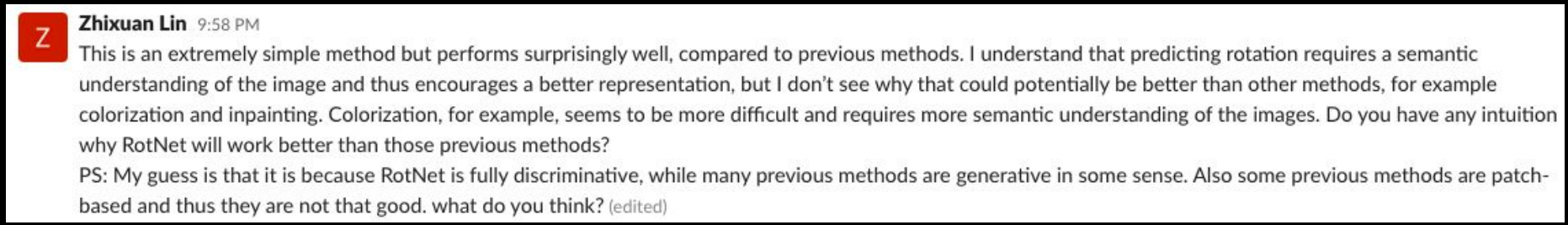
1. How do you think the performance of the mode would be affected if we were to use test-time augmentation with all 4 rotations as used in pre-training?
2. Do you think using test-time augmentation will affect the performance of each layers differently?

(edited)

What would happen if test time augmentation with the 4 rotations were performed?

The goal of the downstream supervised task is to check the generalizability of the model so we do not flip the images.
If we flip an image belonging to a dog for example, the facial features would be the same so the performance. The performance will probably remain the same. The performance might decrease for a few categories as the image orientation is changing (for eg :- tree), but the performance will not increase for sure as the object features have not improved.

Questions and Answers!



Zhixuan Lin 9:58 PM

This is an extremely simple method but performs surprisingly well, compared to previous methods. I understand that predicting rotation requires a semantic understanding of the image and thus encourages a better representation, but I don't see why that could potentially be better than other methods, for example colorization and inpainting. Colorization, for example, seems to be more difficult and requires more semantic understanding of the images. Do you have any intuition why RotNet will work better than those previous methods?

PS: My guess is that it is because RotNet is fully discriminative, while many previous methods are generative in some sense. Also some previous methods are patch-based and thus they are not that good. what do you think? (edited)

Why does rotation prediction perform better than other techniques like Colorization and Inpainting even though it is simpler?

In rotation prediction the model is able to view the original image i.e. 0 degree rotation. This original image is similar to the image the model will encounter in the downstream supervised task.
In case of exemplar method, we colour and augment the image. In case of context prediction and jigsaw, we cannot view the original image and the gaps between the parts of the image so that the model cannot find the pixel to pixel mapping to solve the task.

Questions and Answers!

Nader Asadi 9:14 AM

What do the authors mean by visual image artifacts?

Also, I understand how the rotation pretext task can help the model learn more object-centric semantics, compared to Jigsaw or Counting tasks where the low-level statistics of the images such as the texture or background of the image can be used as a shortcut. However, I think, with Rotation task, if we train the model on non-curated raw datasets, we observe significant decrease in the feature quality, because there might be many scenarios where the object shape is not rotation variant or the background can act as a shortcut for the model. What is your input on that?

What are visual image artifacts? Can rotation invariant image and background acting as a shortcut cause trouble?

The Visual Image Artifacts can cause a few weird shapes.
In case of spherical object like a ball, it is tricky to find the rotation by looking at the object because the object is symmetrical and looks similar at the different angles of rotation. If the background contains sky or table it can act as a shortcut to the model. These are limitations of this method.

Questions and Answers!

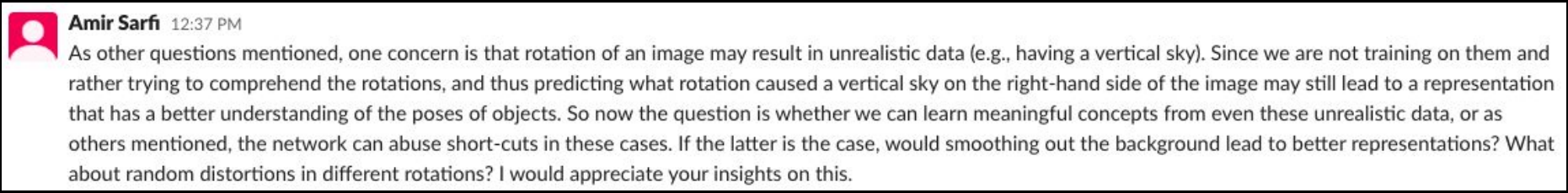
Abhinav Moudgil 11:45 AM

Authors motivate their approach by saying that doing well on this rotation prediction task requires semantic understanding and they provide evidence in Figure 3 (b) with attention maps visualization — as you move towards higher layers, attention regions become semantically relevant. However, in experiments they find that lower convolutional block (ConvB2) actually gives best performance. Do you have any justification for this?

Why does the performance reduce as we go deeper, despite the attentions being more semantically relevant?

The neurons in the later cnn layers like fourth or fifth layer have activations focused on giving high performance for the rotation specific task and are less generalizable to downstream supervised task. The second layer CNN performs better because it learns general features related to edges and curves of the image. It performs better if the number of CNN layers are 5 instead of 4 or 3 because if the numbers of are more than then the second layer focuses on building general set of features for the images and the later layers focuses on building features specific to the rotation specific task.

Questions and Answers!



Amir Sarfi 12:37 PM

As other questions mentioned, one concern is that rotation of an image may result in unrealistic data (e.g., having a vertical sky). Since we are not training on them and rather trying to comprehend the rotations, and thus predicting what rotation caused a vertical sky on the right-hand side of the image may still lead to a representation that has a better understanding of the poses of objects. So now the question is whether we can learn meaningful concepts from even these unrealistic data, or as others mentioned, the network can abuse short-cuts in these cases. If the latter is the case, would smoothing out the background lead to better representations? What about random distortions in different rotations? I would appreciate your insights on this.

Will background smoothing + random distortions help?

If we want to smooth the background of the image, we would **probably** require some kind of label to differentiate between the background and foreground which require manual labelling work.
Adding random distortions to different rotations can be tried and it looks promising.

Questions and Answers!

Kavin Patel 12:51 PM

1. I want to understand a specific use case. How will the round objects be evaluated? If such example(s) are given 4 geometric transformations and fed as input, my hunch is that the rotation prediction task will show lesser accurate results but the learned features will perform better for other vision tasks. So, in accordance to this, for round objects, can I say that performances for rotation prediction task and object recognition task are independent of each other?
2. "We observe that among the RotNet models trained with 2 discrete rotations, the RotNet model trained with 90 degree and 270 degree rotations achieve worse object recognition performance than the model trained with the 0 degree and 180 degree rotations, which is probably due to the fact that the former model does not "see" during the unsupervised phase the 0 degrees rotation that is typically used during the object recognition training phase." Can you please elaborate on this? Does the underlying assumption of human captured images being in the "up-standing" position an important one for this model?

1. Round Objects?

2. Why is 0+180 better than 90+270?

1. Round and other symmetrical object are hard to predict for the model and it is one of the limitations of the model, the model would potentially move to finer level features.
2. 0 degree image (original image) is very similar to the image the model would encounter during the downstream supervised learning stage. In case of the model trained with 90 degree and 270 degree image, it will not get to view the original image and performs a little worse on the downstream supervised learning task.

Questions and Answers!

Nehal Pandey 1:35 PM

1. In table 2, accuracy for images with 8 rotations is low than 4 rotations. So can we say that after 4 rotations the accuracy starts dropping and 4 is the ideal choice?
2. I wonder how would self-supervised model perform in case of images that would look the same after rotation too. For example any centered object like ball(sphere) or a square table would look the same even on rotations.

The model may not learn useful representations in the case of images like spheres, or polka dot fields or scenes like birds.