Scaling and Benchmarking of Self-Supervised Models

Review by:
Sudhan Wosti, Sai Kopparthi, Zixin Chi

Self-supervised learning

- Form of unsupervised learning where the data provides **supervision without any manual labelling**.
- Representations learned on a **large unlabelled dataset** as a **pretext task**.
- Can be fine tuned on a **much smaller amount of labelled data**.
- Usually is **comparable to the performance of supervised models.**

Why is it important?

- **>50,000 hours of video uploaded daily** on YouTube.
- **95 million photos and videos uploaded daily** on Instagram, many of which are public.
- **Downloadable for free! (for now)**

How is it done exactly?

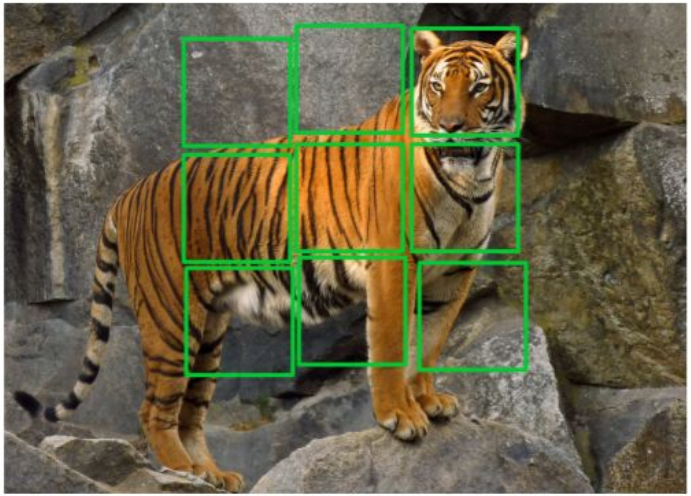
Two popular approaches discussed in the paper :

- **Jigsaw puzzles**
  (M. Noroozi and P. Favaro, ECCV 2016)

- **Colorization**
  (R. Zhang, P. Isola, and A. A. Efros, ECCV 2016)

Unsupervised learning by solving Jigsaw puzzles

- Take an image and slice it into N patches.
- A subset of the N! Permutations of these patches are fed to the model.
- Model returns a probability vector of the likelihood of each permutation being the correct one.



*Images obtained from the same paper by Mehdi Noroozi and Paolo Favaro

Unsupervised learning by solving Jigsaw puzzles

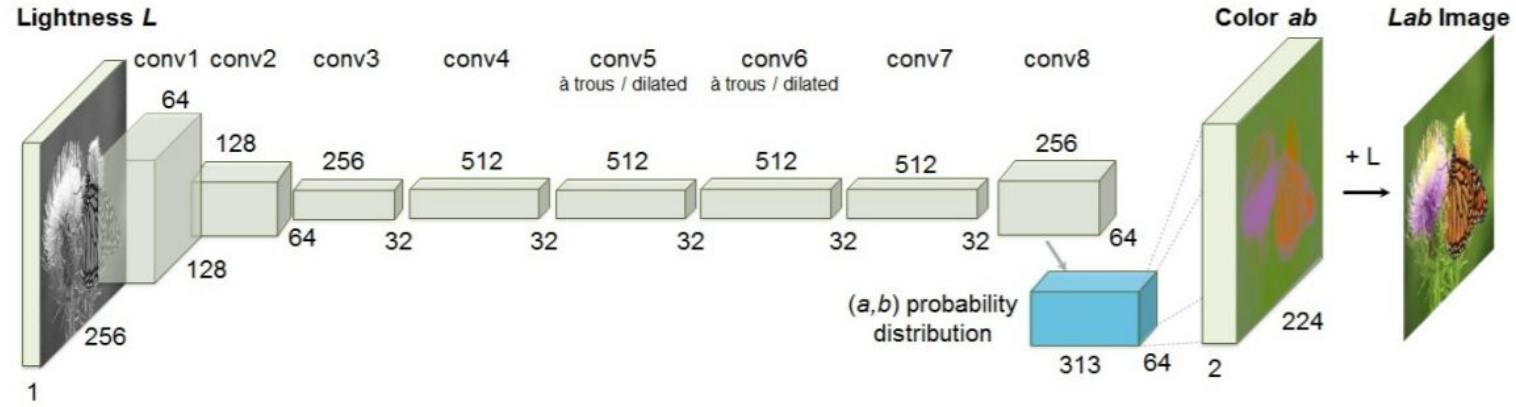
- The task is essentially classification on the number of permutations.
  (pick the best permutation out of all)

- **Number of permutations (|P|) controls the complexity!**

- Example :
  Ground truth permutation: {1, 2, 3, 4, 5, 6}
  Possible permutations: 6! = 720
  Permutations fed:
  {1, 4, 5, 2, 3, 6}, {5, 2, 3, 1, 4, 6}, **{1, 2, 3, 4, 5, 6}**, {3, 1, 6, 4, 5, 2}

Colorful Image Colorization

- Take a large number of normal RGB pictures as the dataset.
- Take the lightness(L) channel as input, and the color(ab) channels as the labels for the pretext task.

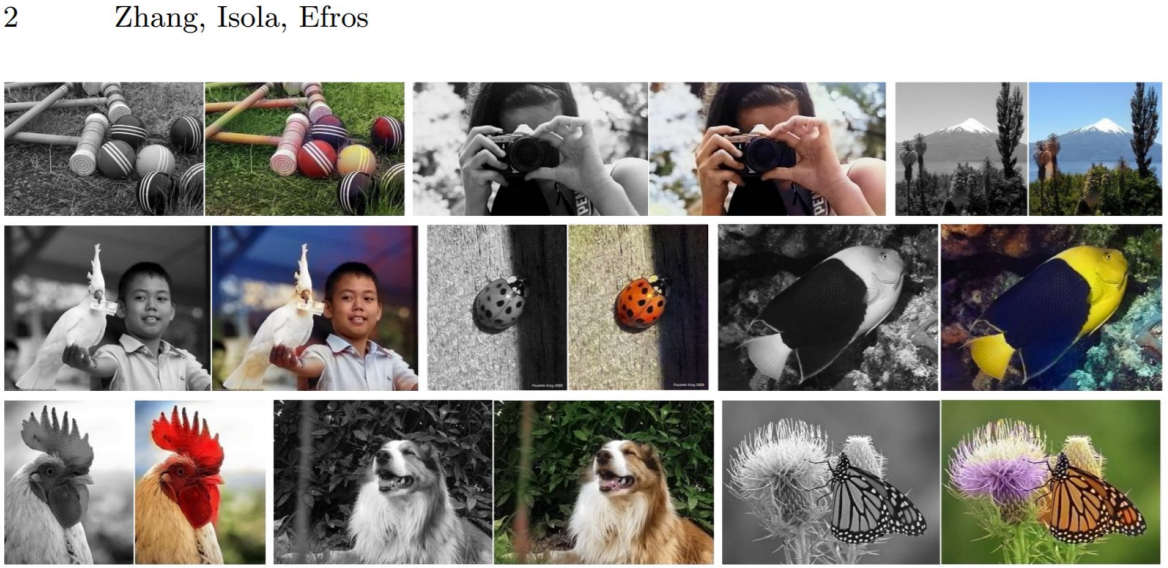


Colorful Image Colorization

- The color(ab) output space is quantized into K bins (= 10 in their paper) bins.
- Task is to assign each pixel into one of these K bins.
- **Value of K controls the hardness of the task!**
- Details about their approach is orthogonal to our paper; You may learn more at https://arxiv.org/pdf/1603.08511.pdf

Colorful Image Colorization

- Pretext task is to produce a **plausible** colorization.



The tennis ball may not be green in real-life, but it is believable.

So they also used a sort of “Color Turing test”, where they manage to fool 32% of people into thinking the generated colored picture is the ground truth.

Colorful Image Colorization

Fig. 8. Applying our method to legacy black and white photos. Left to right: photo by David Fleay of a Thylacine, now extinct, 1936; photo by Ansel Adams of Yosemite; amateur family photo from 1956; *Migrant Mother* by Dorothea Lange, 1936.

Performs well on ‘fake’ black and white photos as well as real ones.

Models used in this paper

- AlexNet
  ~62M parameters, 8 layers (small capacity)

- ResNet50
  ~25M parameters, 50 layers (large capacity)

- Depth of the network has more effect than the width.

How do we maximize performance?

Scale along the three axes **together** :

- Data
- Model capacity
- Task complexity

Scaling Self-Supervised Learning

- First, scaling the pre-training data to 100X the size commonly used in existing self-supervised methods.

- Second explore the model capacity by comparing ResNet-50 and AlexNet.

- Finally we check the how the hardness(Number of Permutation |p| , Number of nearest neighbors K) of pretext task controls the quality of the learned representation.

Investigation Setup

- We use task of image classification on PASCAL VOC2007.

- Then Train linear SVMs on fixed feature representation obtained from the ConvNet. Specifically choose the best performing layer: conv4 layer for AlexNet and the output of last res4 block for ResNet-50.

Axis 1: Scaling the Pre-training Data size

- This work studies scaling for both the Jigsaw and Colorization methods.
- Trained on various subsets of YFCC-100M dataset- YFCC[1,10,50,100] million images.
- Further, during the self-supervised pre-training, authors kept other factors that may influence the transfer learning performance such as the model, the problem complexity ($|P| = 2000$, $K = 10$) etc. fixed as a way to isolate the effect of data size on performance.

Observations

- We see that increasing the size of pre-training data improves the transfer learning performance for both the Jigsaw and Colorization methods on ResNet-50 and AlexNet.

- we make an interesting observation that the performance of the Jigsaw model saturates (log-linearly) as we increase the data scale from 1M to 100M.

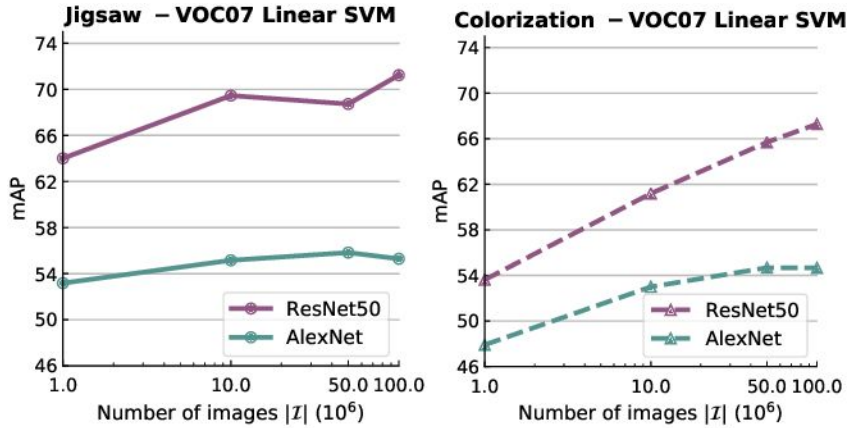


Figure 1: Scaling the Pre-training Data Size: The transfer learning per-

Axis 2: Scaling the model capacity

- Explore the relationship between model capacity and self-supervised representation learning.

- we observe this relationship in the context of the pre-training dataset size. For this, we use AlexNet and the higher capacity ResNet-50 model to train on the same pre-training subsets.

Observations

- An important observation is that the performance gap between AlexNet and ResNet-50 (as a function of the pre-training dataset size) keeps increasing.

- This suggests that higher capacity models are needed to take full advantage of the larger pre-training datasets.

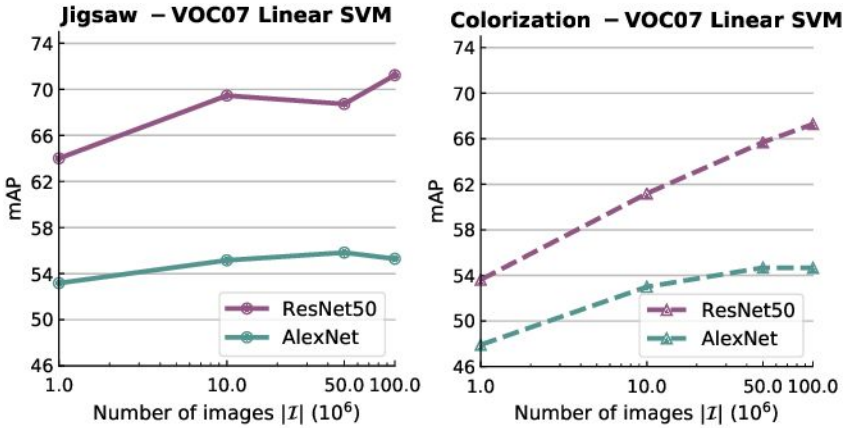


Figure 1: Scaling the Pre-training Data Size: The transfer learning per-

Axis3: Scaling the problem Complexity

Jigsaw: The number of permutations |P| determines the number of puzzles seen for an image. We **vary the number of permutations |P|** $\in$ [100, 701, 2k, 5k, 10k] to control the problem complexity. Note that this is a 10× increase in complexity compared to .

Colorization: We **vary the number of nearest neighbors K** for the soft-encoding which controls the hardness of the colorization problem. To isolate the effect of problem complexity, we fix the pretraining data at YFCC-1M.

Observation

- ResNet-50 shows a 5 point mAP improvement while AlexNet shows a smaller 1.9 point improvement.
- The Colorization approach appears to be less sensitive to changes in problem complexity. We see ~2 point mAP variation across different values of K.

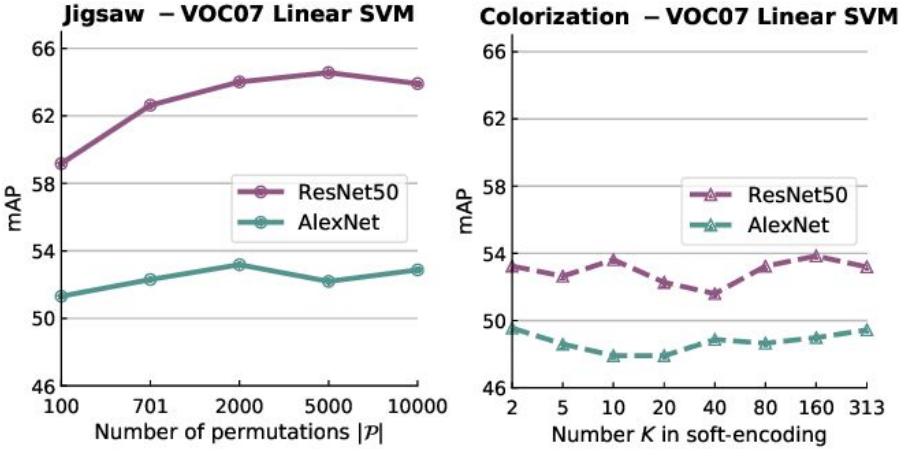
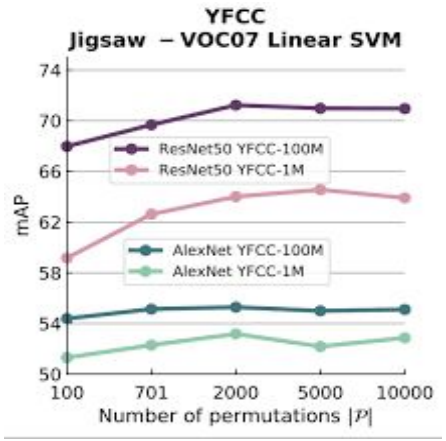


Figure 2: Scaling Problem Complexity: We evaluate transfer learning

Putting it all together

- We can see that transfer learning performance increases on all three axes, i.e., increasing problem complexity still gives performance boost on ResNet-50 even at 100M data size.
- But for best results, we should scale all three axes together.
- We can conclude that the three axes of scaling are complementary



Benchmarking Suite for self-supervision

- We need the model to **perform on real tasks, not pretext tasks.**
- Standardize the methodology for evaluating quality of visual representations
- A set of 9 tasks
- From semantic classification/detection, scene geometry to visual navigation.
- Two principles:
  - Transfer to many different tasks
  - Transfer with limited supervision and limited fine-tuning

Tasks and datasets

Task	Datasets	Description
Image classification		
§ 6.1	Places205	Scene classification. 205 classes.
(Linear Classifier)	VOC07	Object classification. 20 classes.
	COCO2014	Object classification. 80 classes.
Low-shot image classification		
§ 6.2	VOC07	$\leq$ 96 samples per class
(Linear Classifier)	Places205	$\leq$ 128 samples per class
Visual navigation		
§ 6.3 (Fixed ConvNet)	Gibson	Reinforcement Learning for navigation.
Object detection		
§ 6.4	VOC07	20 classes.
(Frozen conv body)	VOC07+12	20 classes.
Scene geometry (3D)		
§ 6.5 (Frozen conv body)	NYUv2	Surface Normal Estimation.

Common Setup

1. Perform self-supervised pre-training using a self-supervised pretext method.

Symbol	Description
YFCC-XM	Images from the YFCC-100M [70] dataset. We use subsets of size $X \in [1M, 10M, 50M, 100M]$.
ImageNet-22k	The full ImageNet dataset (22k classes, $14M$ images) [12].
ImageNet-1k	ILSVRC2012 dataset ($1k$ classes, $1.28M$ images) [61].

AlexNet and ResNet-50 is trained on these datasets

Common Setup

2. Extract features from various layers of the network

AlexNet:
after every conv layer.

ResNet-50:
from the last layer of every residual stage(res1, res2…)

Common Setup

3. Evaluate quality of these features by transfer learning

   Based on different self-supervised approaches.

   Benchmarking them on various transfer datasets and tasks.

Task 1. Image Classification

- 3 datasets are used: Places205, VOC07 and COCO2014.
- Batch size = 256; learning rate of 0.01 decayed by a factor of 10 after every 40k iterations.
- Train for 140 iterations using SGD on the train split.

Task 1. Image Classification

3 datasets are used: Places205, VOC07 and COCO2014.

Method	layer1	layer2	layer3	layer4	layer5
ResNet-50 ImageNet-1k Supervised	14.8	32.6	42.1	50.8	52.5
ResNet-50 Places205 Supervised	16.7	32.3	43.2	54.7	62.3
ResNet-50 Random	12.9	16.6	15.5	11.6	9.0
ResNet-50 (NPID) [77]◁	18.1	22.3	29.7	42.1	45.5
ResNet-50 Jigsaw ImageNet-1k	15.1	28.8	36.8	41.2	34.4
ResNet-50 Jigsaw ImageNet-22k	11.0	30.2	36.4	41.5	36.4
ResNet-50 Jigsaw YFCC-100M	11.3	28.6	38.1	44.8	37.4
ResNet-50 Coloriz. ImageNet-1k	14.7	27.4	32.7	37.5	34.8
ResNet-50 Coloriz. ImageNet-22k	15.0	**30.5**	37.8	44.0	**41.5**
ResNet-50 Coloriz. YFCC-100M	**15.2**	30.4	**38.6**	**45.4**	**41.5**

ResNet-50 top-1 center crop accuracy for linear classification

Method	layer1	layer2	layer3	layer4	layer5
AlexNet ImageNet-1k Supervised	22.4	34.7	37.8	39.2	38.0
AlexNet Places205 Supervised	23.2	35.6	39.8	43.5	44.8
AlexNet Random	15.7	20.8	18.5	18.2	16.6
AlexNet (Jigsaw) [52]	19.7	26.7	31.9	32.7	30.9
AlexNet (Colorization) [79]	16.0	25.7	29.6	30.3	29.7
AlexNet (SplitBrain) [80]	21.3	30.7	34.0	34.1	32.5
AlexNet (Counting) [53]	23.3	33.9	36.3	34.7	29.6
AlexNet (Rotation) [26]◁	21.5	31.0	35.1	34.6	33.7
AlexNet (DeepCluster) [9]	17.1	28.8	35.2	36.0	32.2
AlexNet Jigsaw ImageNet-1k	23.7	33.2	36.6	36.3	31.9
AlexNet Jigsaw ImageNet-22k	**24.2**	**34.7**	37.7	37.5	31.7
AlexNet Jigsaw YFCC-100M	24.1	**34.7**	**38.1**	**38.2**	31.6
AlexNet Coloriz. ImageNet-1k	18.1	28.5	30.2	31.3	30.3
AlexNet Coloriz. ImageNet-22k	18.9	30.3	33.4	34.9	34.2
AlexNet Coloriz. YFCC-100M	18.4	30.0	33.4	34.8	**34.6**

AlexNet top-1 center crop accuracy for linear classification

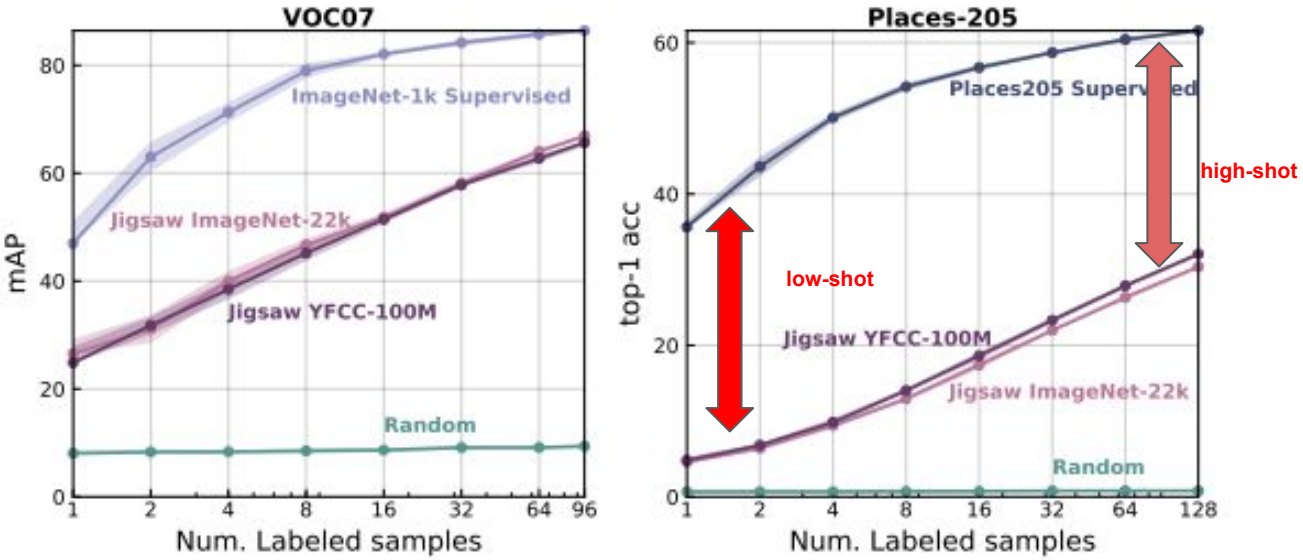
both the supervised pre-training and benchmark transfer tasks solve a semantic image classification problem.

Task 2. Low-shot Image Classification

What if the number of per-category examples are low?

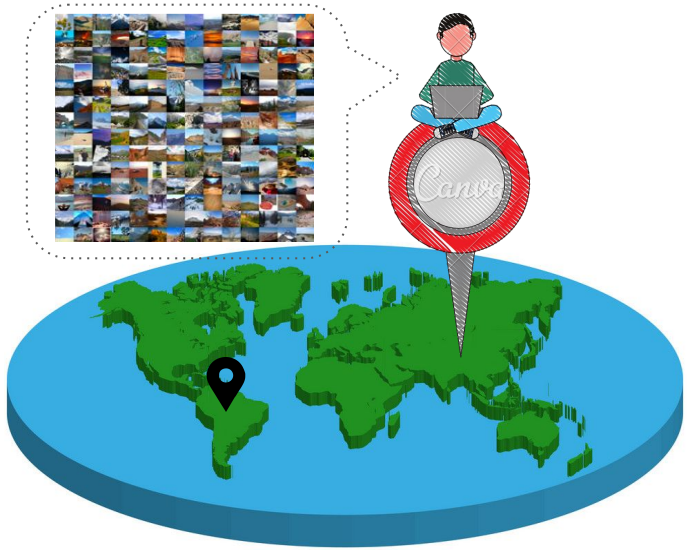
- Vary the number k of positive examples per class
- Evaluate only for ResNet50
- Average and standard deviation of 5 independent examples

Task 2. Low-shot Image Classification



Best performing Layer res4 for Resnet-50 on VOC07 and Places-205

Task 3. Visual Navigation



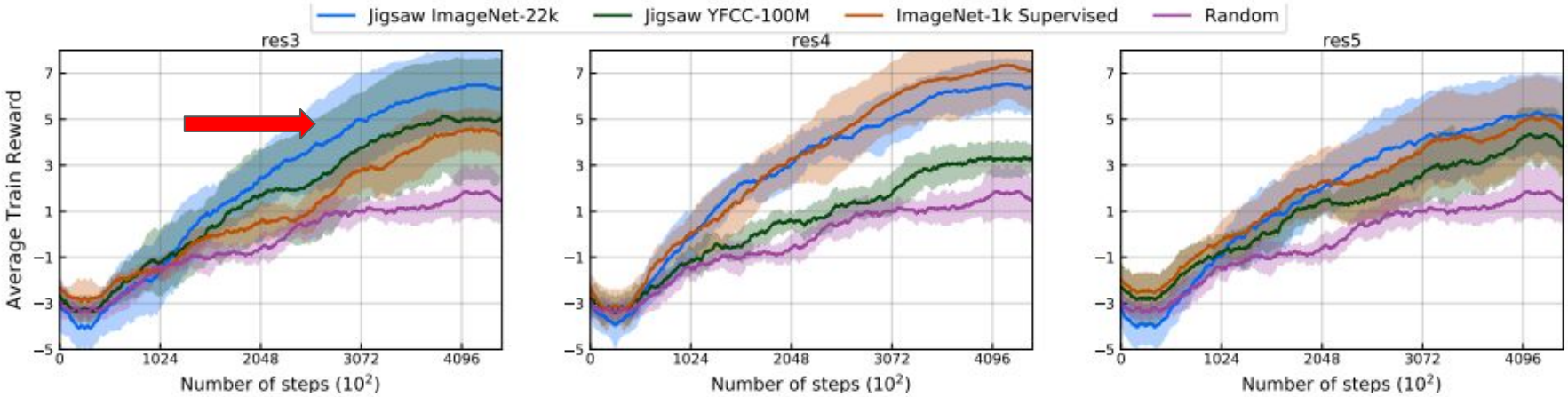
Scenario:

- An agent receives a stream of images as input
- navigate to a predefined location
- Spawned at a random start point
- How to build a map?

Setup:

- Train a agent using reinforcement learning in the Gibson environment
- Uses fixed feature representations from a ConvNet and only update the policy network
- Separately train agents for layers res3, res4, res5 of a ResNet-50

Task 3. Visual Navigation



Jigsaw ImageNet-22k has the highest rewards with fewer steps in Res3

Task 4. Object Detection

Setup:

- Detectron framework to train the fast R-CNN object detection model.
- Selective search on the VOC07 and VOC07-12 datasets
- Freeze the full conv budy of Fast R-CNN and only train the RoI heads
- Same training schedule for both supervised and self-supervised methods
- Slightly longer schedule to improve object detection performance
- 2 GPUs at 22k/8k(VOC07) and 66k/14k(VOC7_12)

Task 4. Object Detection

Method	VOC07	VOC07+12
ResNet-50 ImageNet-1k Supervised*	66.7 ± 0.2	71.4 ± 0.1
ResNet-50 ImageNet-1k Supervised	**68.5** ± 0.3	**75.8** ± 0.2
ResNet-50 Places205 Supervised	65.3 ± 0.3	73.1 ± 0.3
ResNet-50 `Jigsaw` ImageNet-1k	56.6 ± 0.5	64.7 ± 0.2
ResNet-50 `Jigsaw` ImageNet-22k	**67.1** ± 0.3	**73.0** ± 0.2
ResNet-50 `Jigsaw` YFCC-100M	62.3 ± 0.2	69.7 ± 0.1

the self-supervised initialization is competitive with the ImageNet pre-trained initialization on VOC07 dataset even when fewer parameters are fine-tuned on the detection task.

Task 5. Surface Normal Estimation

Setup:

- Use NYUv2 dataset which contains indoor scenes and PSPNet architecture
- Fine-tuned res5 onwards and train with same hyperparameters.
- Batchsize of 16, learning rate of 0.02 decayed with a power of 0.9 and SGD for optimization

Task 5. Surface Normal Estimation

Metrics: the angular distance(error) of the prediction and the percentage of pixels within t degree of the ground truth

Initialization	Angle Distance		Within t°		
	Mean	Median	11.25	22.5	30
	(Lower is better)		(Higher is better)		
ResNet-50 ImageNet-1k supervised	26.4	17.1	36.1	59.2	68.5
ResNet-50 Places205 supervised	23.3	14.2	41.8	65.2	73.6
ResNet-50 Scratch	26.3	16.1	37.9	60.6	69.0
ResNet-50 `Jigsaw` ImageNet-1k	24.2	14.5	41.2	64.2	72.5
ResNet-50 `Jigsaw` ImageNet-22k	22.6	13.4	43.7	66.8	74.7
ResNet-50 `Jigsaw` YFCC-100M	**22.4**	**13.1**	**44.6**	**67.4**	**75.1**

Summary

Self-supervised learned representation:

- Outperforms supervised on surface normal estimation
- performs competitively base on navigation tasks
- Match the supervised object detection baseline with limited fine-tuning
- Performs worse on image classification and low-shot classification.