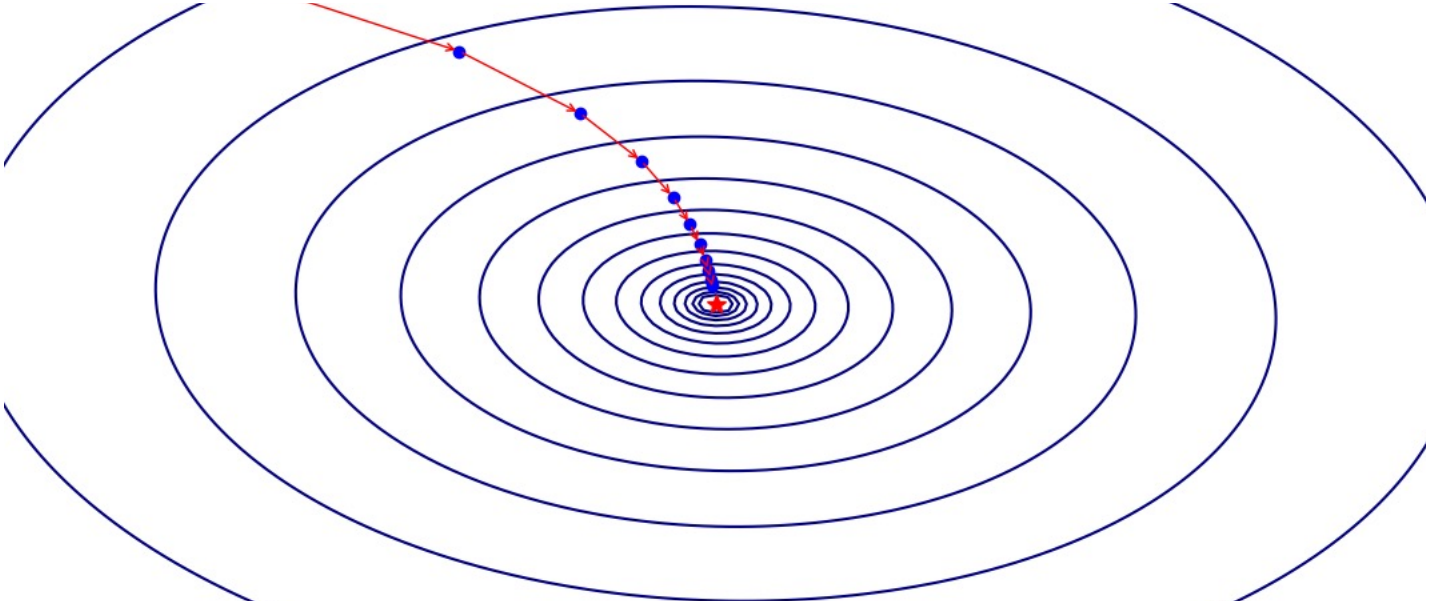


Advanced Topics in Deep Learning

Summer Semester 2024

3. Self-supervised Learning (Part I)

24.04.2023

Prof. Dr. Vasileios Belagiannis

Chair of Multimedia Communications and Signal Processing

Friedrich-Alexander-Universität
Technische Fakultät

Course Topics

Not for sharing (LMS, Friedrich-Alexander-Universität Erlangen-Nürnberg)

1. Interpretability.
2. Attention and Transformers.
3. **Self-supervised Learning.**
4. Similarity Learning.
5. Generative Models.
6. Model Compression.
7. Transfer learning, domain adaptation, few-shot learning.
8. Uncertainty Estimation.
9. Geometric Deep Learning.
10. Recap and Q&A.

- The exam will be written.
- We will have an exam preparation test.

Acknowledgements

- Special thanks Arij Bouazizi, Julia Hornauer, Julian Wiederer, Adrian Holzbock and Youssef Dawoud for contributing to the lecture preparation.

Recap

- Sequence modelling
- Language modelling
- Attention mechanism for RNNs
- Self-attention as dot-product
- Multi-head attention
- Encoding-Decoding architectures

Today's Agenda and Objectives

- Transfer learning.
- Self-supervised learning.
- Pretext tasks.
- Image-based self-supervision.

Supervised Learning

- Consider the problem of bird species classification and a deep neural network classifier.
- *What is the necessary for achieving high classification performance?*
- Supervised learning works very well in practice when there are enough annotated samples.
- However, sample annotation is a costly process and therefore not possible for every machine learning task.
- In addition, the samples available for specific applications may be limited. This can lead to poor model generalisation.
- *Is there a way to achieve good supervised learning performance when the available data is insufficient to train a deep neural network?*

Caltech-UCSD Birds*.

*Welinder, Peter, et al. "Caltech-UCSD birds 200." (2010).

Transfer Learning

- Consider a pre-trained model on a specific task, e.g. car segmentation, with a specific dataset. We want to apply this model to our task in hand, e.g. human segmentation, based on a different dataset.
- Transfer learning is the approach of adapting the pre-trained model to the new task using the new dataset.
- For instance, in computer vision, a common example is to pre-train a deep neural network to ImageNet dataset for classification and then use this model for transfer learning.
- In practice, we use the pre-trained model weights as initialisation for the new (downstream) task.

Transfer Learning (Step 1)

- Pre-training is normally happening with a large-scale dataset.

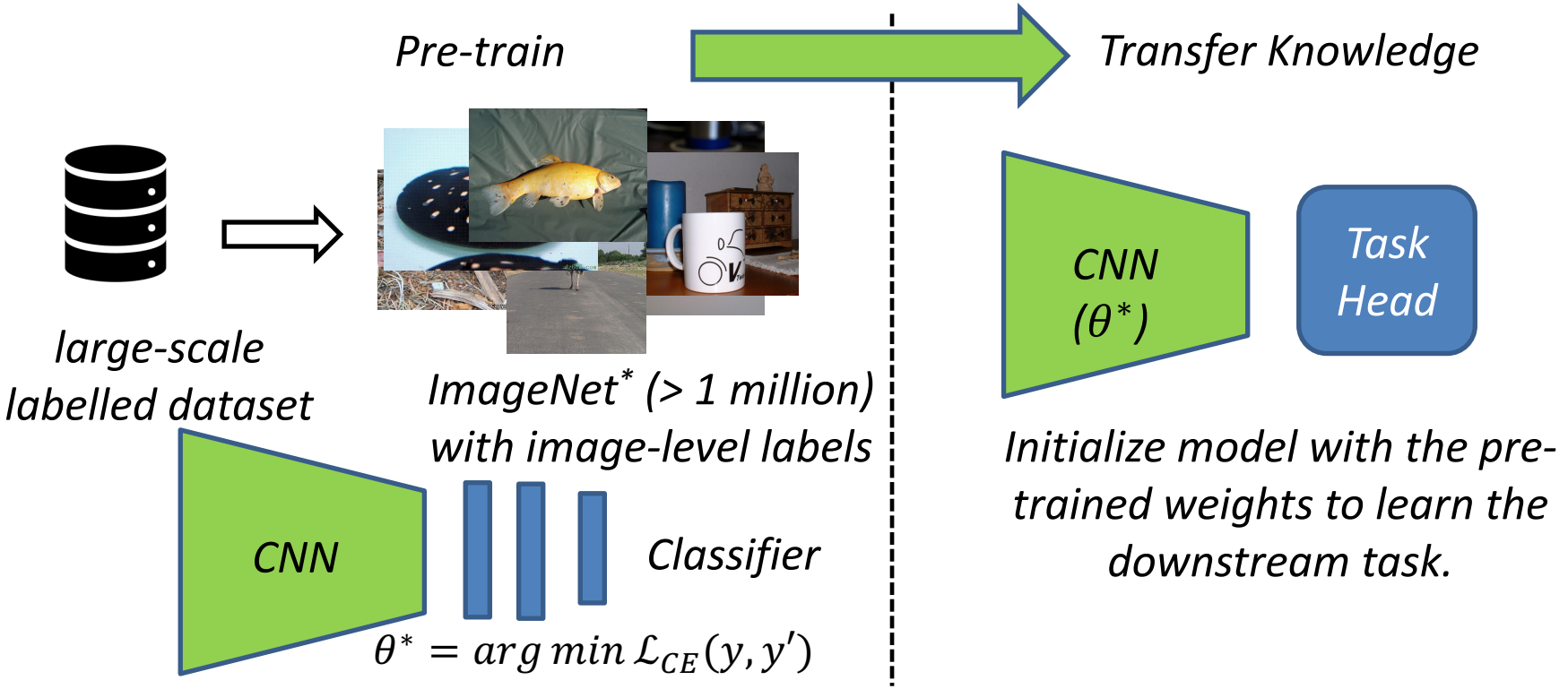
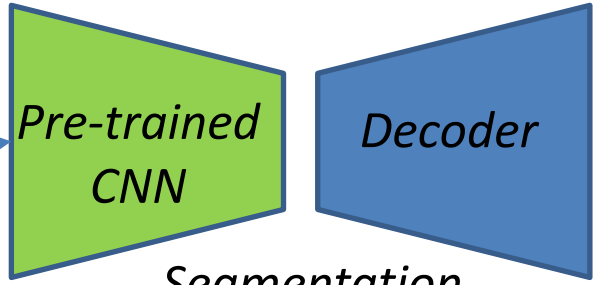


*Image Source: https://www.image-net.org

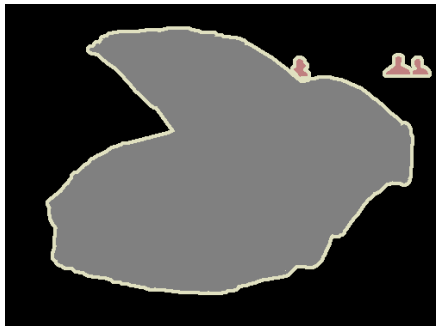
Transfer Learning (Step 2)

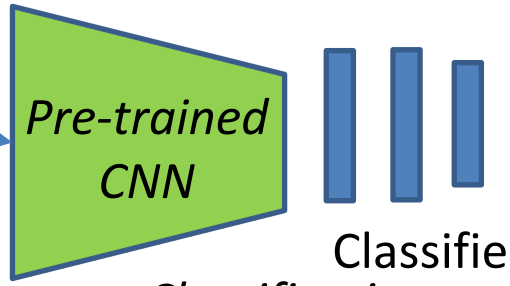
- The deep neural network is entirely or partially, in terms of network parameters, trained on the dataset and task of interest using the optimized weights from pre-training as prior knowledge.

*PASCAL VOC**



Segmentation (pixel-wise classification)





Classifier

Car
Not a Car

Classification

*Image source: http://host.robots.ox.ac.uk/pascal/VOC/

Transfer Learning Observations

- In deep learning, transfer learning normally happens with the pre-trained model approach.
- The process of training the pre-trained model on the task in place is called fine-tuning.
- Fine-tuning often requires different set of hyper-parameters, e.g. learning rate.
- Transfer learning is helpful when we lack sufficient amount of training data for the task in hand, e.g. we have only 100 annotated images for segmentation. That is the main motivation for transfer learning.
- It can also reduce the training time since we could start the optimisation from a ”better” set of initial parameter values.

Pre-training Limitations

- Enabling effective transfer learning comes at the cost of pre-training with large-scale labelled datasets.
- Pre-training on large-scale labelled data such as ImageNet may not generalise well to other domains such as medical image analysis. There can be domain and task misalignment and thus the pre-trained model might not be well-suited.
- Thus, transfer learning will not always lead to good performance for the task in place.
- Pre-training cannot prevent the task of annotating the dataset of interest. This is another costly process.
- *Can we replace the supervised pre-training with another task to obtain a good model initialisation?*

Training with Unlabelled Images

- There is a huge amount of unlabelled data available, for example on the internet. We could more easily create datasets that contain only training samples but no labels.
- Our objective is to a train deep neural network using unlabelled data to learn useful representations (features) that would benefit the target (downstream) learning task, e.g. image recognition.
- Self-supervised learning is a paradigm where the supervision stems from the data itself.

Unlabeled images*

* Image Source: http://host.robots.ox.ac.uk/pascal/VOC/

Self-Supervised Learning

- The main idea is to derive the supervision from a proxy / surrogate task.
- The proxy task allows the definition of a differentiable loss function and thus provides the necessary supervision for the training of the model.
- *How is the proxy task defined?*

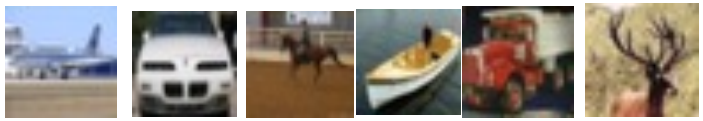
Unlabeled images*

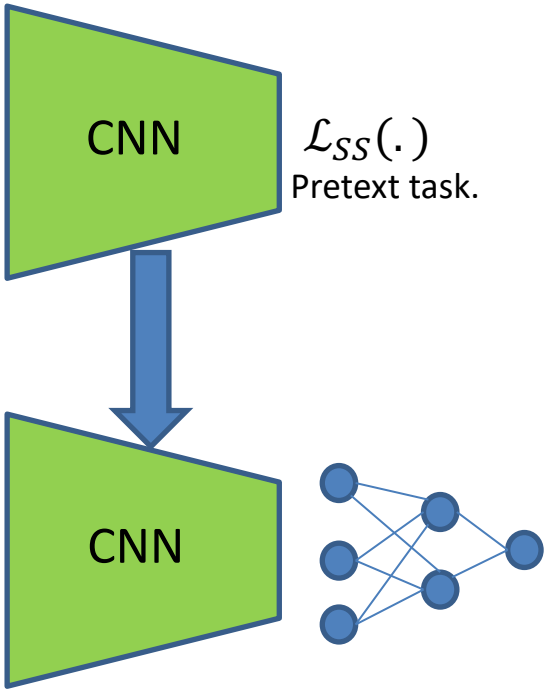
* Image Source: http://host.robots.ox.ac.uk/pascal/VOC/

Self-Supervised Learning Workflow

- Self-Supervised learning usually consists of two steps:
  - Pre-training of the backbone (encoder) on the pretext task with the unlabelled data.
  - Then fine-tuning the backbone on the downstream task, e.g. classification or object detection.
- The weights of the backbone are usually frozen in the fine tuning phase. We only train the head, e.g. the classifier or decoder.

Step 1: Pre-train using unlabelled dataset.





Step 2: Fine-tune classifier.

Self-Supervised Learning Workflow (Cont.)

- We optimize the parameters of the backbone network θ by training on the pretext task using the differentiable loss function $\mathcal{L}_{Pretext}$ as:

$$\theta^* = \arg\min \mathcal{L}_{Pretext}(\mathrm{x}; \theta)$$

- Afterwards, we rely on θ^* to initialize the same backbone model to learn the downstream task; and either fine-tune the task head only or the entire network. The downstream task has also the differentiable loss function $\mathcal{L}_{downstream}$. The optimization is summarized as:

$$\zeta^* = \arg\min \mathcal{L}_{downstream}(\mathrm{x}; \theta^*, \phi)$$

- where ϕ denotes task head parameters andζ^*denote the optimized parameters of the model.
- We use the terms “backbone” and “encoder” interchangeably.

Pretext Tasks

- Our focus is on computer vision tasks. We aim thus to define pretext tasks motivated from images.
- We can use the term proxy, pretext or surrogate task.
- The pretext tasks provide us with labels for free.
- We do not care about our performance in the pretext task. Instead, we care about learning a useful learning representation for the target task(s).
- *What is a good pretext task?*

Caltech-UCSD Birds*.

*Welinder, Peter, et al. "Caltech-UCSD birds 200." (2010).

Pretext Task: Surrogate Classes

- A set of exemplary image patches are sampled from an existing dataset in a random fashion.
- To sample informative patches, the image gradients are used to locate patches that contain parts of an object or texture in general.
- After randomly sampling an informative an image patch, it undergoes through a composition of geometric and photometric transformations.

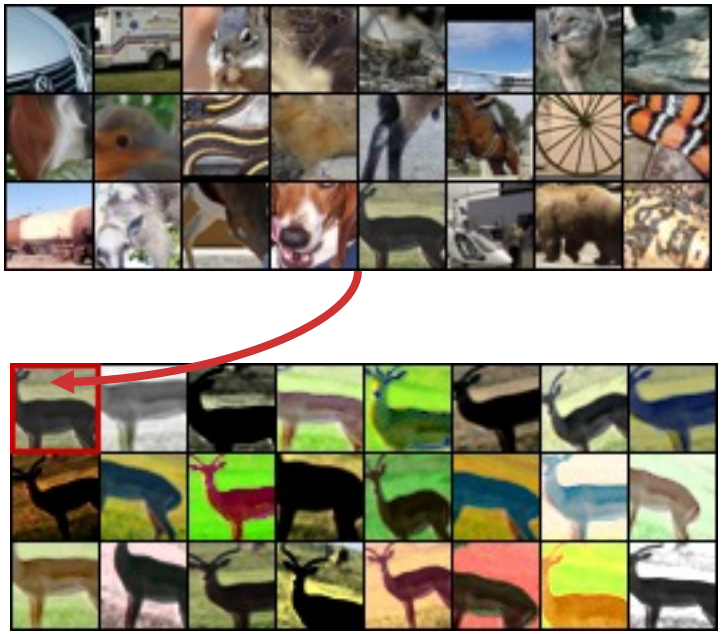


Image Source: https://arxiv.org/pdf/1406.6909.pdf

*Alexey, Dosovitskiy, et al. "Discriminative unsupervised feature learning with exemplar convolutional neural networks." IEEE TPAMI 38.9 (2016): 1734-1747.

Pretext Task: Surrogate Classes (Cont.)

- Each composition of transformations creates a new image patch that belongs to the same surrogate class with the original image patch.
- The transformations include translation, rotation, scaling, contrast, colour transformation.
- Essentially, based on the number of patches N, there will be N different surrogate classes.

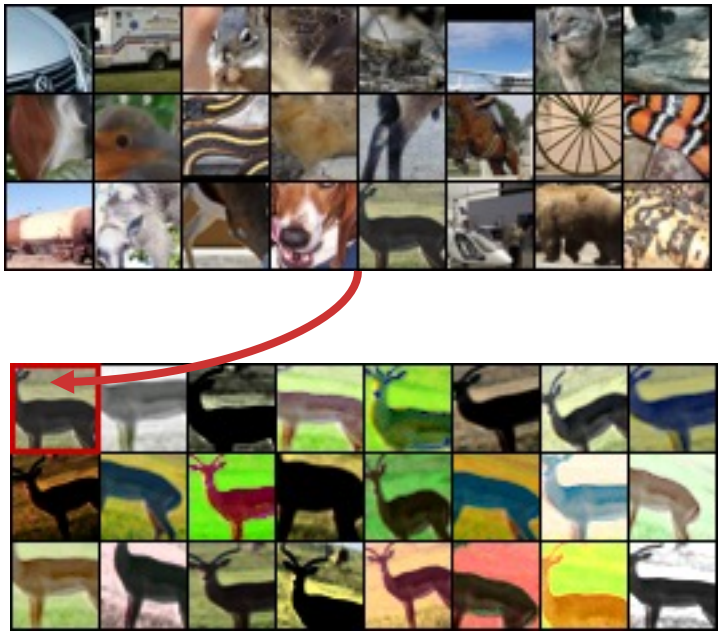


Image Source: https://arxiv.org/pdf/1406.6909.pdf

*Alexey, Dosovitskiy, et al. "Discriminative unsupervised feature learning with exemplar convolutional neural networks." IEEE TPAMI 38.9 (2016): 1734-1747.

Pretext Task: Surrogate Classes (Cont.)

- Discriminating between a set of N surrogate classes is the pretext task.
- The backbone network can be combined with a fully connected layer and a softmax output function to be trained. The loss function $\mathcal{L}_{Pretext}$ can be the negative loglikelihood.
- Afterwards the backbone can be trained on the target task, e.g. image classification.
- The pretext task performance is not always relevant to performance on the target task.

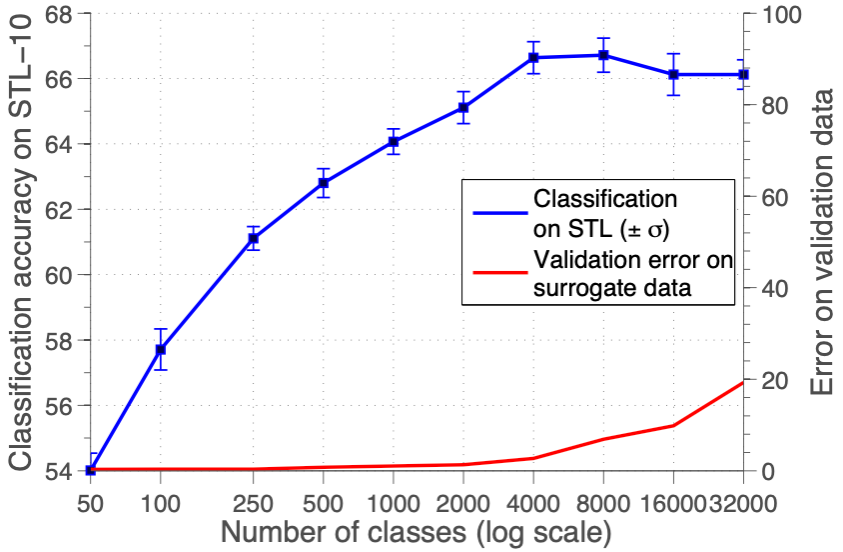


Image Source: https://arxiv.org/pdf/1406.6909.pdf

Pretext Task: Patch Relationship

- We can use the image context in forms of patch* information to learn a visual representation.
- The pretext task is to sample an image patch and a neighbouring patch after defining an image grid. Then the goal is to predict their relation within the grid.
- Both patches are processed by the same convolutional neural network (CNN) to predict their relationship as a classification problem.
- The relationships can be discretised into a certain number of classes.

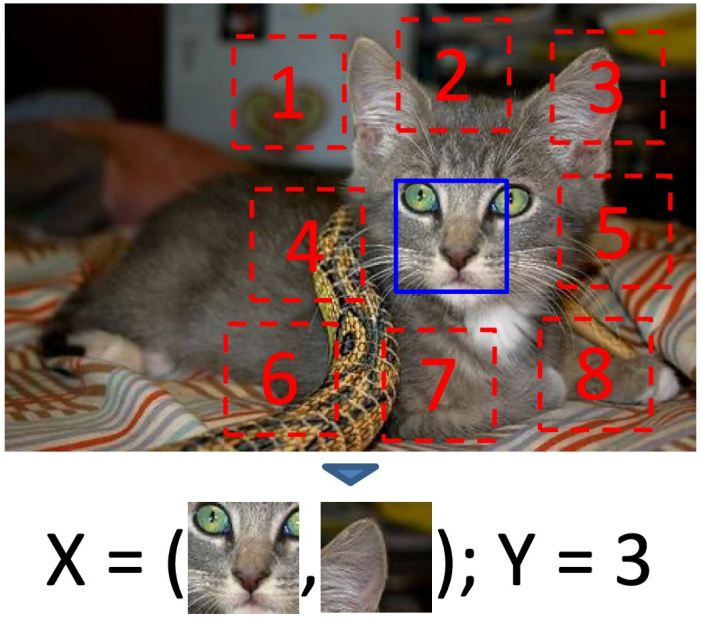


Image Source: https://arxiv.org/pdf/1505.05192.pdf

*Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." Proceedings of the IEEE international conference on computer vision. 2015.

Patch Relationship Learning Algorithm

- Randomly sample a patch from the image.
- Set up a gride with the random patch placed in the centre.
  - To avoid trivial solutions, e.g., same texture across two patches, add enough pixel gap between the grid cells. In addition, add pixelization augmentation (down-sample and up-sample) as well as jitter in the pixel location. Colour correction is also helpful to force the network not to memorise the patches.
- Given the reference patch and the randomly sampled patch, the network predicts where the randomly sampled patch is among the 8 neighbour locations.

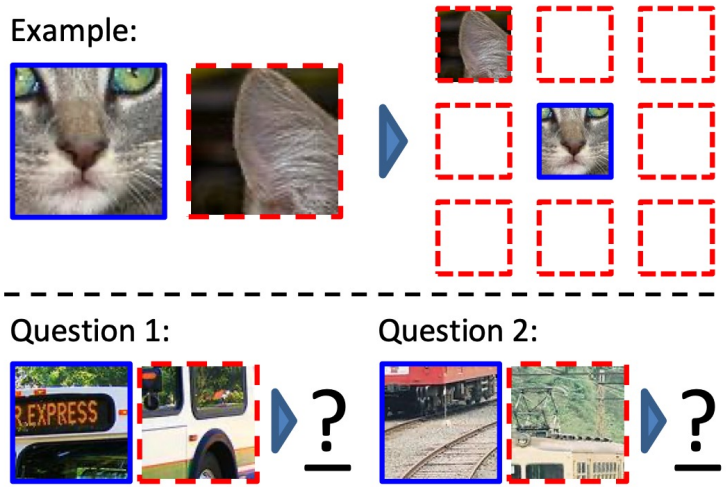


Image Source: https://arxiv.org/pdf/1505.05192.pdf

*Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." Proceedings of the IEEE international conference on computer vision. 2015.

Patch Relationship Training

- By learning the image context without the pretext task, the model should be able to learn a backbone that we can then use for a concrete task.
- The approach is evaluated for object detection on Pascal VOC dataset to show the ImageNet pre-training is not necessary.
- After pre-training with patch relationship learning task, the model is fine-tuned on Pascal VOC.

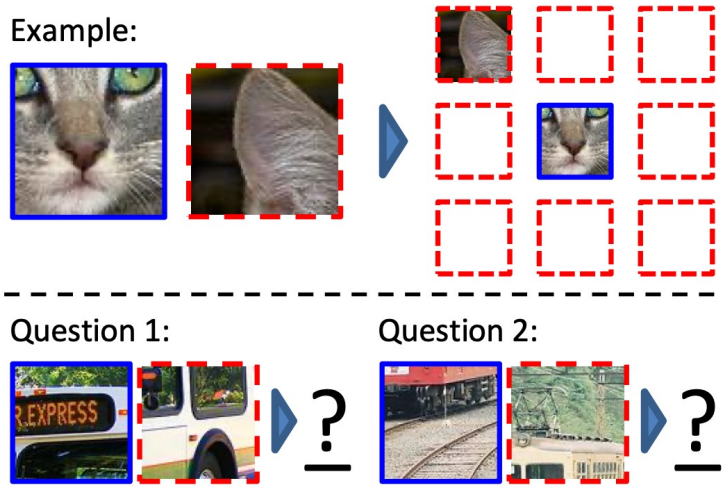


Image Source: https://arxiv.org/pdf/1505.05192.pdf

*Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." Proceedings of the IEEE international conference on computer vision. 2015.

Patch Relationship Learning (Cont.)

- *Check out the paper results on PASCAL VOC object detection. What can we conclude for the proposed approach compared to ImageNet pre-training?*

VOC-2007 Test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM-v5[17]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
[8] w/o context	52.6	52.6	19.2	25.4	18.7	47.3	56.9	42.1	16.6	41.4	41.9	27.7	47.9	51.5	29.9	20.0	41.1	36.4	48.6	53.2	38.5
Regionlets[58]	54.2	52.0	20.3	24.0	20.1	55.5	68.7	42.6	19.2	44.2	49.1	26.6	57.0	54.5	43.4	16.4	36.6	37.7	59.4	52.3	41.7
Scratch-R-CNN[2]	49.9	60.6	24.7	23.7	20.3	52.5	64.8	32.9	20.4	43.5	34.2	29.9	49.0	60.4	47.5	28.0	42.3	28.6	51.2	50.0	40.7
Scratch-Ours	52.6	60.5	23.8	24.3	18.1	50.6	65.9	29.2	19.5	43.5	35.2	27.6	46.5	59.4	46.5	25.6	42.4	23.5	50.0	50.6	39.8
Ours-projection	58.4	62.8	33.5	27.7	24.4	58.5	68.5	41.2	26.3	49.5	42.6	37.3	55.7	62.5	49.4	29.0	47.5	28.4	54.7	56.8	45.7
Ours-color-dropping	60.5	66.5	29.6	28.5	26.3	56.1	70.4	44.8	24.6	45.5	45.4	35.1	52.2	60.2	50.0	28.1	46.7	42.6	54.8	58.6	46.3
Ours-Yahoo100m	56.2	63.9	29.8	27.8	23.9	57.4	69.8	35.6	23.7	47.4	43.0	29.5	52.9	62.0	48.7	28.4	45.1	33.6	49.0	55.5	44.2
ImageNet-R-CNN[21]	64.2	69.7	50	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
K-means-rescale [31]	55.7	60.9	27.9	30.9	12.0	59.1	63.7	47.0	21.4	45.2	55.8	40.3	67.5	61.2	48.3	21.9	32.8	46.9	61.6	51.7	45.6
Ours-rescale [31]	61.9	63.3	35.8	32.6	17.2	68.0	67.9	54.8	29.6	52.4	62.9	51.3	67.1	64.3	50.5	24.4	43.7	54.9	67.1	52.7	51.1
ImageNet-rescale [31]	64.0	69.6	53.2	44.4	24.9	65.7	69.6	69.2	28.9	63.6	62.8	63.9	73.3	64.6	55.8	25.7	50.5	55.4	69.3	56.4	56.5
VGG-K-means-rescale	56.1	58.6	23.3	25.7	12.8	57.8	61.2	45.2	21.4	47.1	39.5	35.6	60.1	61.4	44.9	17.3	37.7	33.2	57.9	51.2	42.4
VGG-Ours-rescale	71.1	72.4	54.1	48.2	29.9	75.2	78.0	71.9	38.3	60.5	62.3	68.1	74.3	74.2	64.8	32.6	56.5	66.4	74.0	60.3	61.7
VGG-ImageNet-rescale	76.6	79.6	68.5	57.4	40.8	79.9	78.4	85.4	41.7	77.0	69.3	80.1	78.6	74.6	70.1	37.5	66.0	67.5	77.4	64.9	68.6

*Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." Proceedings of the IEEE international conference on computer vision. 2015.

Pretext Task: Jigsaw Puzzle

- In the jigsaw pretext task*, multiple patches are extracted in a grid format from the same image.
- Then the patches are shuffled and the model learns to predict the permutation index used to shuffle the patches.
- For example, the task is to place 9 shuffled image patches back in their original positions.

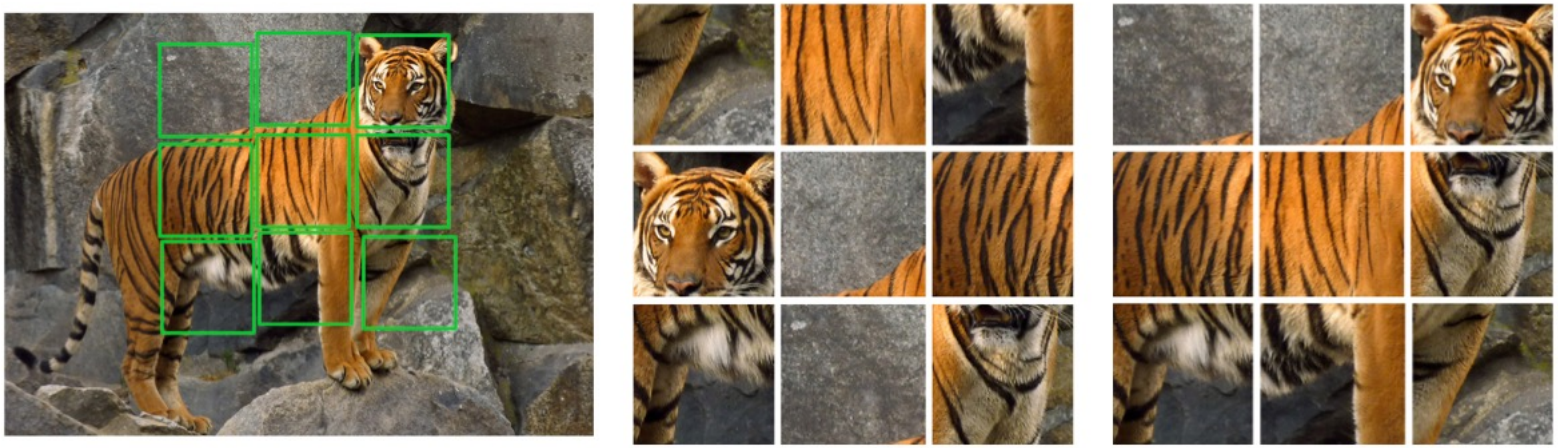


Image Source: https://arxiv.org/abs/1603.09246

*Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.

Jigsaw Puzzle Algorithm

- First an image is randomly cropped to 225x225 (see red dashed window).
- Then the it is divided into 3x3 grid cells, from each 75x75 cell.
- A 64x64 tile is randomly selected. A total of 9 tiles are extracted per image, which are then shuffled and used as input to a CNN with shared weights.
- The CNN processes each tile individually and produces one feature vector per patch index.
- The model is trained to place the shuffled patches back into place. This is achieved by training the model to predict the shuffle index used to generate the shuffled tiles. To do this, the pretext task could be treated as classification. The cross-entropy loss could be employed as the training objective.
- This task forces the network to become permutation invariant.

*Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.

Jigsaw Puzzle Algorithm (Cont.)

- The same CNN takes all patches to predict the correct permutation. For 3x3 grid, we have 64 permutation (classes).

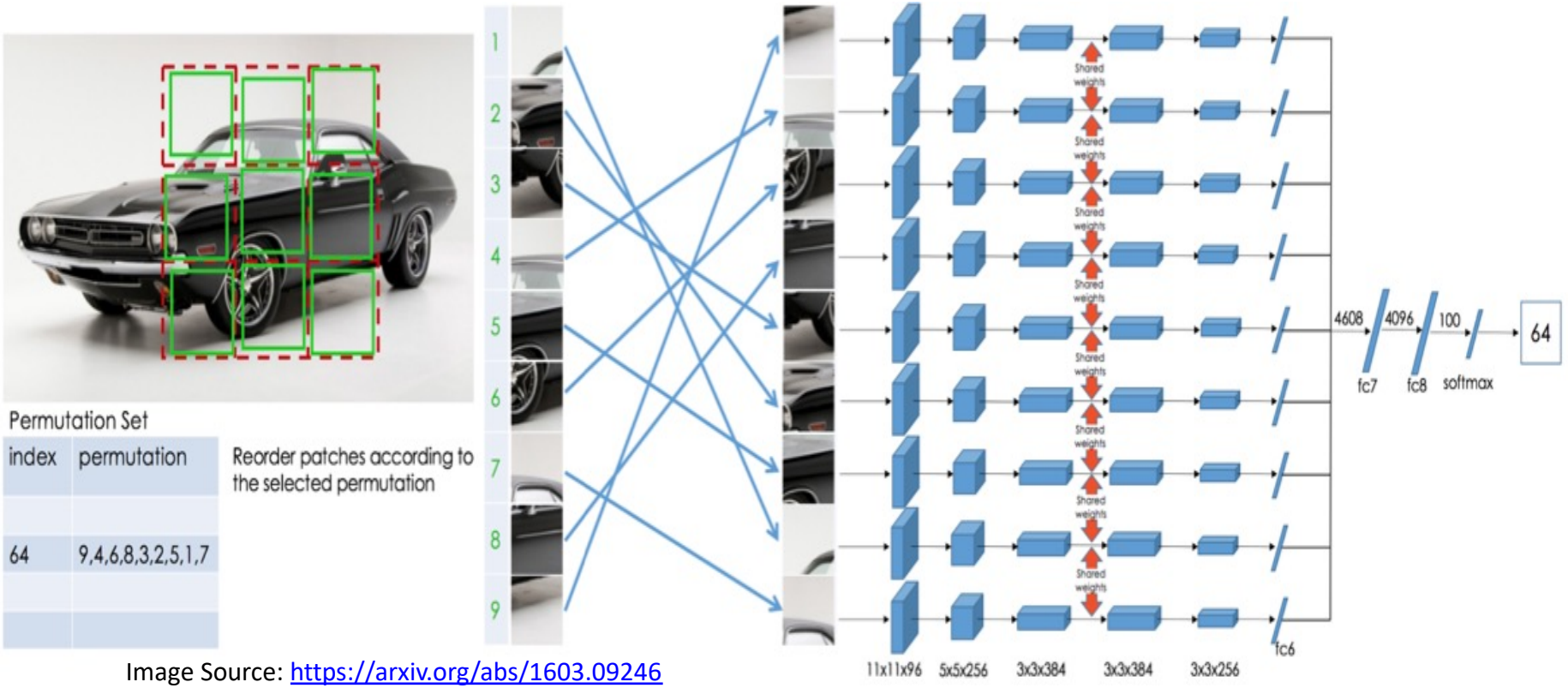


Image Source: https://arxiv.org/abs/1603.09246

*Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.

Jigsaw Puzzle Learning

- Predicting the permutation of 9 image patches can be a difficult task. Note that in the previous approach we were only concerned with predicting the relationship between two patches.
- For that reason, it is proposed to follow pre-defined permutations to regulate the task difficulty.
- After pre-training on the jigsaw puzzle task, the method is evaluated on the PASCAL VOC** dataset for classification, object detection, and semantic segmentation.
- The experimental motivation is to compare the results with the supervised learning pre-training and similar approaches.

*Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.
**Everingham, Mark, et al. "The pascal visual object classes challenge: A retrospective." International journal of computer vision 111 (2015): 98-136.

Jigsaw Puzzle Learning (Cont.)

- *What can we conclude for the proposed approach compared to ImageNet pre-training [25]?*
- *Which is the "best" self-supervised approach to choose?*

Method	Pretraining time	Supervision	Classification	Detection	Segmentation
Krizhevsky*et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**	**48.0%**
Wang and Gupta[39]	1 week	motion	58.4%	44.0%	-
Doersch *et al.* [10]	4 weeks	context	55.3%	46.6%	-
Pathak *et al.* [30]	14 hours	context	56.5%	44.5%	29.7%
Ours	2.5 days	context	**67.6%**	**53.2%**	**37.6%**

*Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.
**Everingham, Mark, et al. "The pascal visual object classes challenge: A retrospective." International journal of computer vision 111 (2015): 98-136.

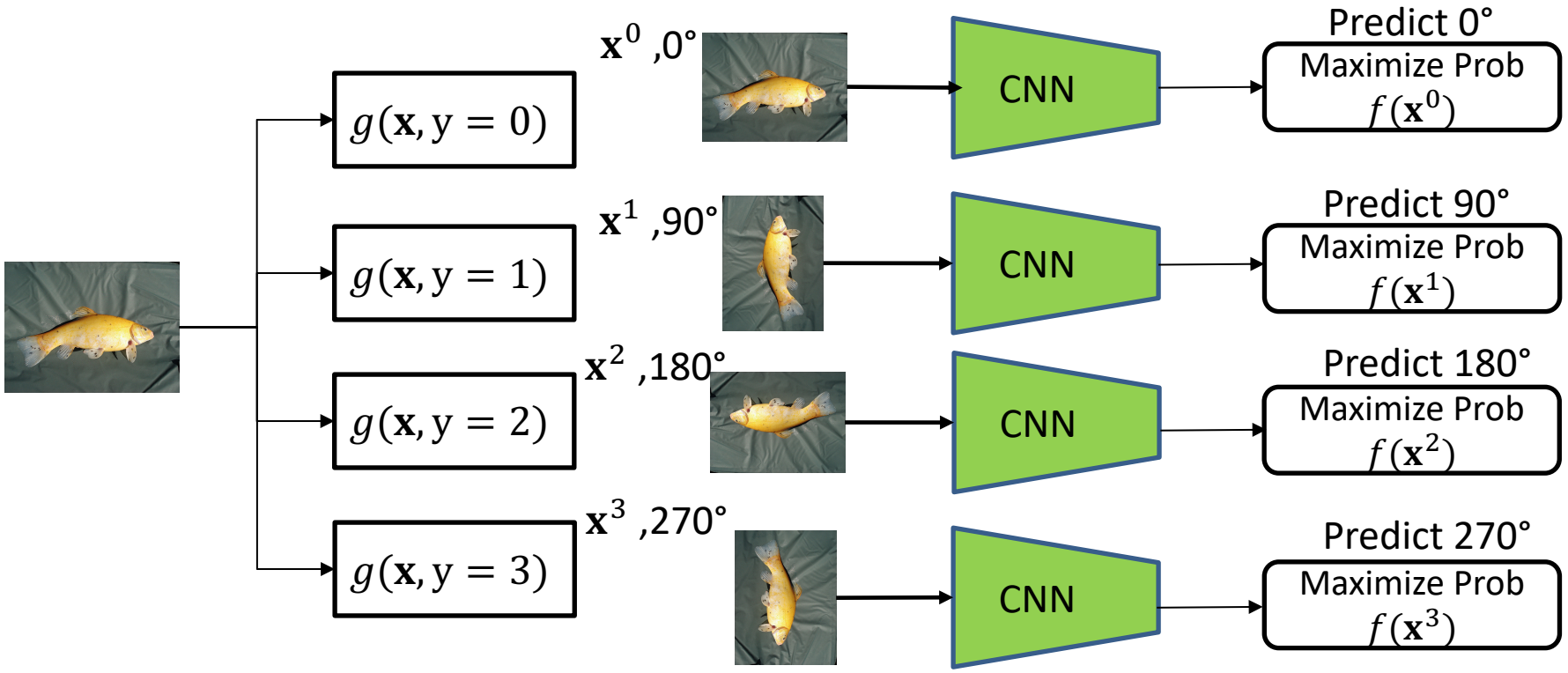
Pretext Task: Rotation

- Image Rotation* is another pretext classification task.
- The model is trained to predict the degree of image rotation.
- An image $\mathbf{x}$ is randomly rotated at multiples of 90°. $Rot(\mathbf{x}, \phi), \phi \in \{0°, 90°, 180°, 270°\}$. We have then a 4-class problem.
- By rotating the image, the context remains the same. At the same time, the model is forced to learn the same image context under different angles. In this way, the model has to learn object parts to conclude about the rotation.
- Image rotation is another computationally cheap approach to defining the pretext task.

* Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations." ICLR 2018.

Pretext Task: Rotation (Cont.)

- All transformed samples pass through the same network.



* Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations." ICLR 2018.

Rotation Learning

- A classification head is used on the backbone to learn the rotation prediction task.
- Formally, the rotation task is defined as follows:

$$G = \{g(\mathbf{x}|y)\}_{y=1}^{K},$$

- where $g(\mathbf{x}|y) = Rot(\mathbf{x}, (y-1)90°)$ is the rotation operator and $y \in \{0,1,2,3\}$.
- The classifier learns to predict (y') the corresponding label of the rotation degree. The output from the classifier is a probability distribution over all possible geometric transformations:

$$f_\theta(\mathbf{x}^y) = \left\{f_\theta^y(\mathbf{x}^y)\right\}_{y=1}^{K},$$

- where $f_\theta^y(\mathbf{x}^y)$ is the probability distribution for the geometric transformation with label y and θ denote the model's parameters.
- Empirically it is shown that $K = 4$ (i.e. $1 \rightarrow 0°$, $2 \rightarrow 90°$, $3 \rightarrow 180°$, $4 \rightarrow 270°$) works well in practice.

* Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations." ICLR 2018.

Rotation Learning (Cont.)

- Given a set of N training images, the network parameters are optimized to reduce the rotation loss $\mathcal{L}_{Rot}(y, y')$:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{Rot}(\mathbf{x}_i, \theta)$$

$$\mathcal{L}_{Rot}(\mathbf{x}_i, \theta) = \frac{-1}{K} \sum_{y=1}^{K} \log\left(f_{\theta}^{y}(g(\mathbf{x}|y))|\theta\right)$$

- After training the model, the rotation prediction head is thrown away.
- Finally, we use the backbone trained parameters to the downstream task.
- Like the previous approaches, the goal is to compare with ImageNet supervised pre-training on PASCAL VOC dataset.

* Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations." ICLR 2018.

Rotation Learning (Cont.)

- FC6-8 refers to fine-tuning only the classifier, while all to the whole model.

	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5	29.7
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4	
Context (Doersch et al., 2015)	55.1	65.3	51.1	
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9	35.6
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9	34.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2	37.6
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4	
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7	36.0
ColorProxy (Larsson et al., 2017)		65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4	36.6
(Ours) RotNet	**70.87**	**72.97**	**54.4**	**39.1**

* Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations." ICLR 2018.

Pretext Task: Colorization

- Colorization* is another pretext task where the model is trained to colour grayscale images.
- It is treated as a pixel-wise prediction task where each pixel is mapped to a quantized output colour value in the CIELAB colour space.
  - The CIELAB colour space is based on L^*, a^*, and b^* components. It approximates the human vision.
  - L^* can serve as input and a^*, and b^* as output.
- The problem could be treated as a regression task, where we could minimise the difference between the regressed output and the ground truth colour image. However, it is shown that in practice discretizing the colour space and treating the problem as a classification problem works better.
- The cross-entropy can be then employed as a loss function.

* Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." ECCV 2016.

Pretext Task: Colorization (Cont.)

- Colorizing pixels is another approach to learn context.

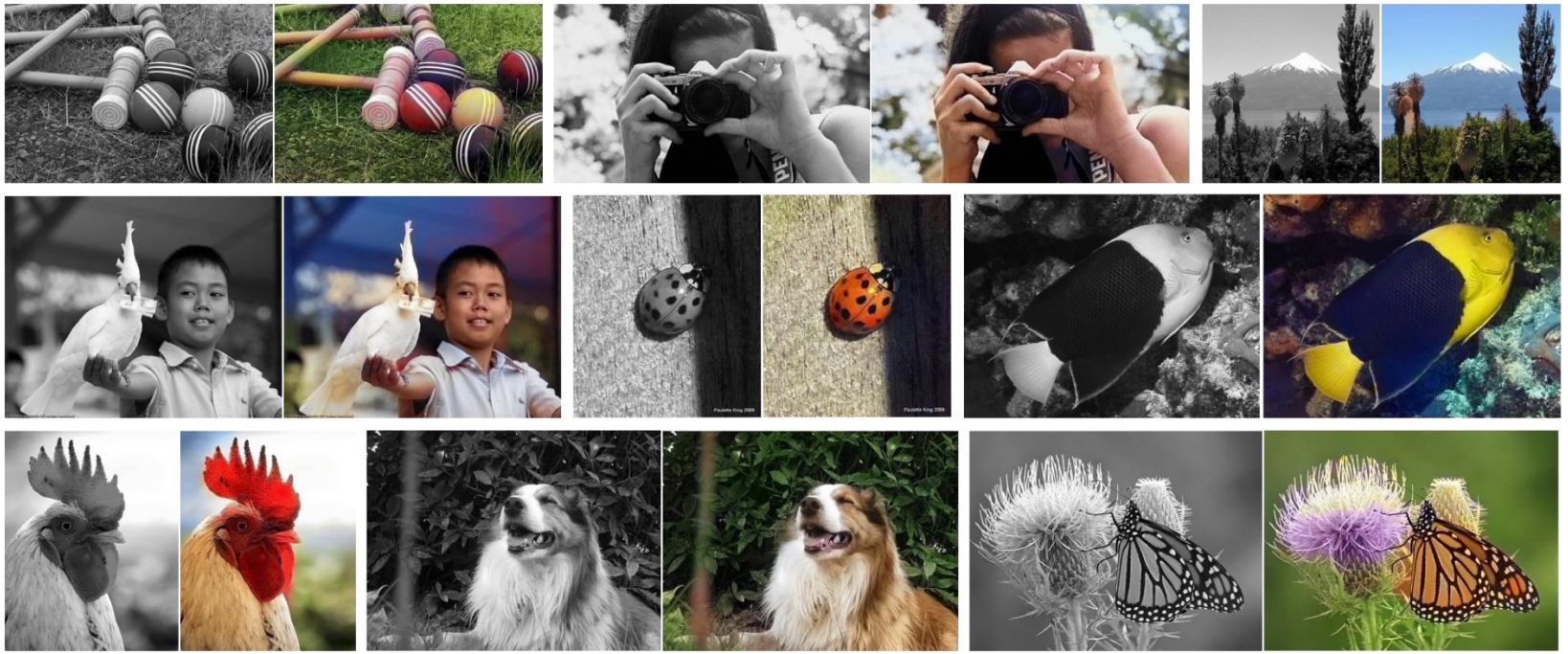


Image Source: https://arxiv.org/pdf/1603.08511.pdf

* Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." ECCV 2016.

Colorization Training

- The backbone is pre-trained with colorization supervision and then fine-tuned on PASCAL VOC dataset.

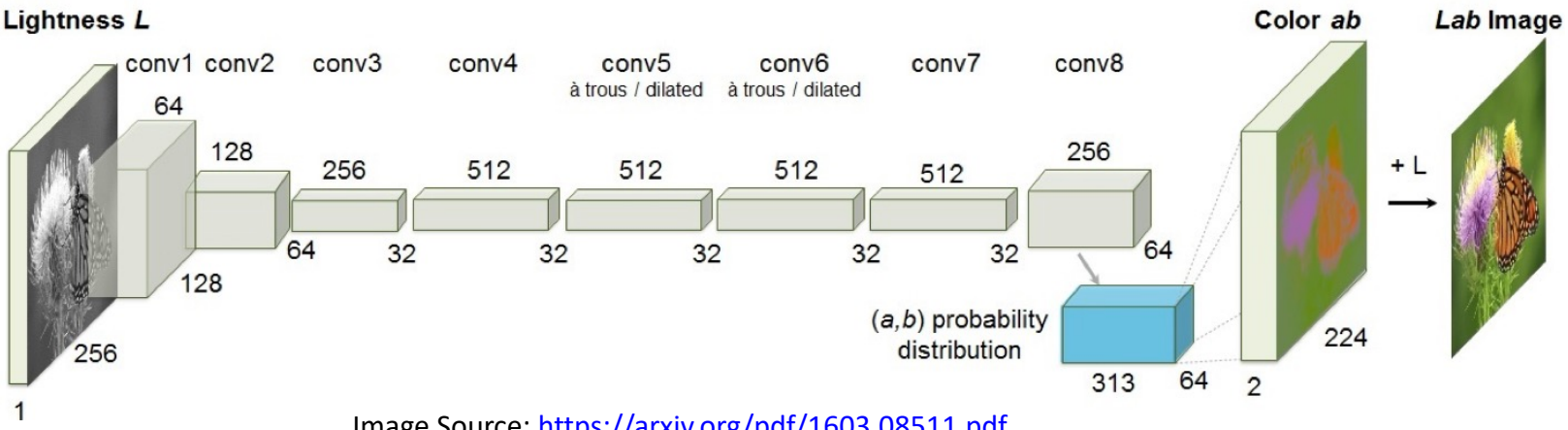


Image Source: https://arxiv.org/pdf/1603.08511.pdf

* Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." ECCV 2016.

More Pretext Tasks

- Additional tasks to use as proxy are:
  - Image inpainting and context encoding*.
  - Clustering**.
- Auto-encoders can be used as well for unsupervised pre-training.
- Overall, the present techniques are image-based and thus applied on specific data type.
- Nevertheless, there are more generic approaches which mainly rely on contrastive learning.

**Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
** Caron, Mathilde, et al. "Deep clustering for unsupervised learning of visual features." Proceedings of the European conference on computer vision (ECCV). 2018.

Study Material

Not for sharing (LMS, Friedrich-Alexander-Universität Erlangen-Nürnberg)

- *Alexey, Dosovitskiy, et al. "Discriminative unsupervised feature learning with exemplar convolutional neural networks." IEEE TPAMI 38.9 (2016): 1734-1747.*
- *Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." Proceedings of the IEEE international conference on computer vision. 2015.*
- *Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.*
- *Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations." ICLR 2018.*
- *Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." ECCV 2016.*
- *Blog: https://lilianweng.github.io/posts/2019-11-10-self-supervised/*

Next Lecture

Self-supervised Learning (Part II)