Koç University
COMP547 Deep Unsupervised Learning
Lecture Notes

Gürkan Soykan

May 07, 2021

8 Self-Supervised Learning

So far we have talked about generative modeling in general in terms of density modeling and implicit approaches. In density modeling we have covered Autoregressive Models, Normalizing Flows And Variational Approaches. As for implicit models we have specifically focused on Generative Adversarial Networks(GANs). Then applications of generative models have been studied.

Today's topic will be different from those. We will be looking at non-generated representation learning that stands for learning informative discriminative features from raw unlabeled data later to be used for other downstream tasks. To do that we will answer the following questions.

- How do learn rich and useful features from raw unlabeled data that can be useful for several downstream tasks?
- What are the various pretext (proxy) tasks that can be used to learn representations from unlabeled data?
- How can we improve data-efficiency and performance of downstream tasks with a good pre-trained network?

8.1 Motivation and Definition

Ultimate goal of Deep Learning is to be able to learn "really" useful representations without supervision.

However, having a good generative model does not mean we will learn useful features.

That is because generative models care about all the pixels in an image. Yet, not all the pixels contains much information and mostly a lot of unnecessary information is there. Then it can be said that only some parts of the image is interesting. As a result we need to get more informative features.

It needs to be also noted that generative models are not always ineffective. There are two exceptions for this case.

- Natural Language Modeling (all SOTA models are build on BERT-like representations)
- In the very small dataset regime, unsupervised learning can actually help.

Other than the exceptions gradient-based supervised training with the right model (e.g. CNNs for vision problems) has been very difficult to beat with unsupervised methods.

Why is this the case? If we have to speculate, it can be said that generative models learn extremely low level features whereas in supervised learning layers of representations can learn relevant axes of variance in the data.

On the other hand self-supervised learning is actually a branch of unsupervised learning that is based on the idea that data itself can be used for labeling. Most of the time task involves obscuring some part of the data from the model and developing the model in a way that could generate the missing part of data. With this way without any external annoations we would have "self" labels. If this is the case then we need to come up with methods to self label the data. Those are called proxy tasks(proxy loss - pretext task).

Goals of self-supervised learning:

- Learn equally good (if not better) features without supervision
- Be able to deploy similar quality systems without relying on too many labels for the downstream tasks
- Generalize better potentially because you learn more about the world

Self-supervised learning is also called Predictive learning. The name makes sense if one has to consider the following cases:

- Predict any part of the input from any other part
- Predict the future from the past
- Predict the future from the recent past
- Predict the past from the present
- Predict the top from the bottom
- Predict the occluded from the visible

As a generalization: Pretend there is a part of the input you don't know and predict that What is it good for?

- Can procedurally generate potentially infinite amounts of annotations
- We can borrow tricks from supervised learning without labels
- Focus on only the information that you need (e.g. not pixels).
- Answering these questions requires more fundamental understanding of the data

No so good! Designing good questions also requires some fundamental understanding of the data (e.g., structure)

8.2 Reconstruct from a corrupted (or partial) version

8.2.1 Denoising Autoencoder

Work in 2010, [11]. From corrupted version of the input image we would like to get original image.
corruption(noise) is introduced to original data by us.
In the paper, the authors show three sources of noise.

- Additive Isotropic Gaussian noise
- Masking noise
- Salt-and-Pepper Noise

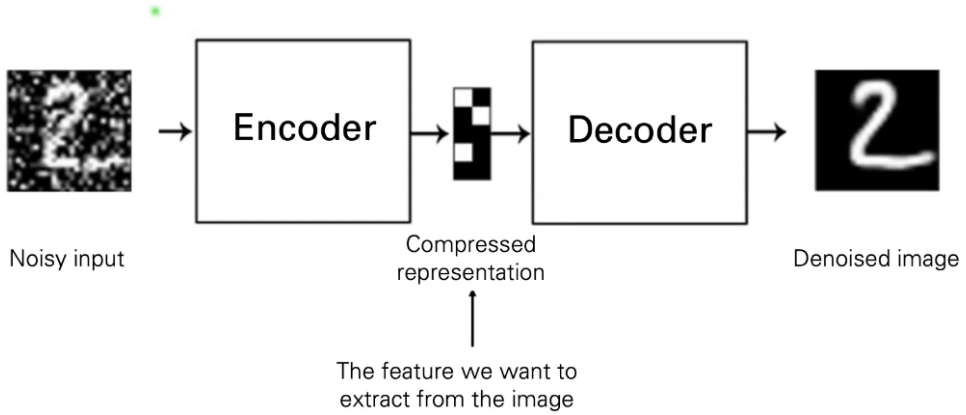
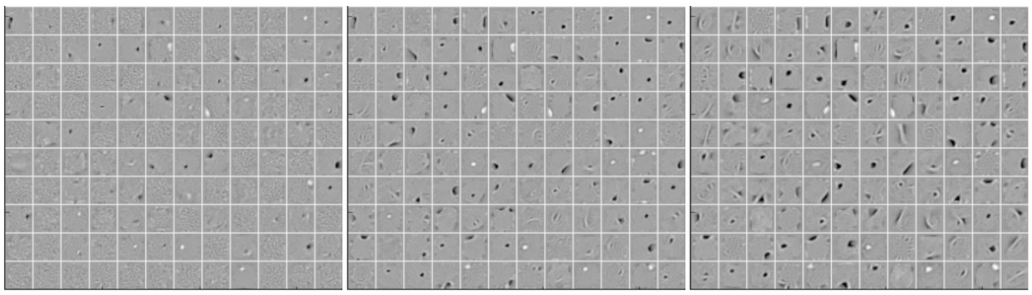


Figure 1: Denoising Autoencoder

Noises include stochasticity.
What is happening is we are deforming the existing data sample in the existing data space. The network should then separate noise from the actual image.
Loss function is weighted for this work in order to penalize reconstructions of noisy pixels.

$$L_{2,\alpha}(\mathbf{x}, \mathbf{z}) = \alpha \left(\sum_{j \in \mathcal{I}(\tilde{\mathbf{x}})} (\mathbf{x}_j - \mathbf{z}_j)^2 \right) + \beta \left(\sum_{j \notin \mathcal{I}(\tilde{\mathbf{x}})} (\mathbf{x}_j - \mathbf{z}_j)^2 \right) \tag{1}$$

Before 2010 the main dominant approach for training of neural networks was to have a layered pretrained steps where you incrementally increase the complexity of the network and this work is one such example. Again historically representations of neurons were included in the papers because people were trying to find commonalities between neural representations and mammal visual system.



(a) No destroyed inputs (b) 25% destruction (c) 50% destruction

Figure 2: Neuron Visualization of Denoising Autoencoder under different destruction of the data

Interestingly, when corruption level increases the network learns more from the data.

8.2.2 Predict Missing Pieces

Task is to strip of some part of the image and then try to inpaint stripped part with the neural networks. Prominent and pioneer paper for this task is "Context Encoders: Feature Learning by Inpainting" [9]

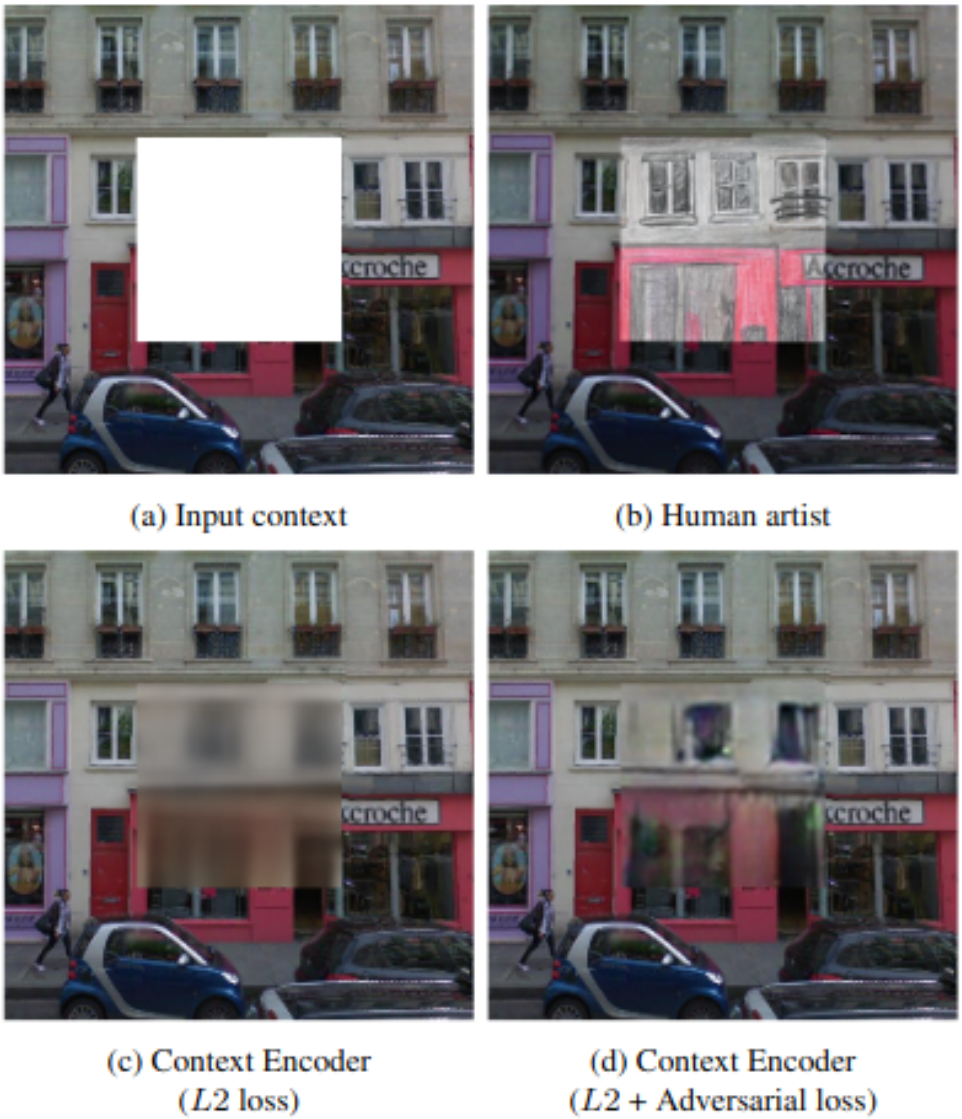


Figure 3: Qualitative illustration of the task. Given an im-age with a missing region (a), a human artist has no trouble inpainting it (b). Automatic inpainting using our context encoder trained with L2 reconstruction loss is shown in (c), and using both L2 and adversarial losses in (d)

Context Encoder can be seen in figure 4. The context image is passed through the encoder to obtain features which are connected to the decoder using channel-wise fully-connected layer. The decoder then produces the missing regions in the image.

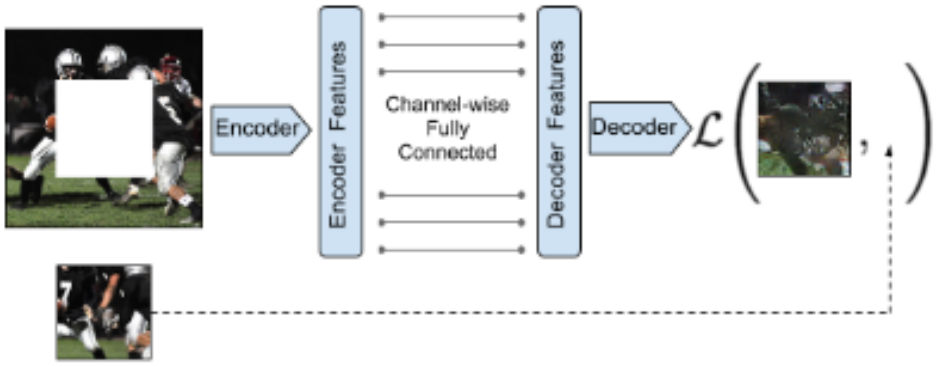


Figure 4: Inpainting Model Overview

Instead of just cutting the middle part of the image, random blocks and shapes can be sampled for recon-

struction.

Another contribution of the authors is also to use of adversarial loss along with the reconstruction loss for this task.

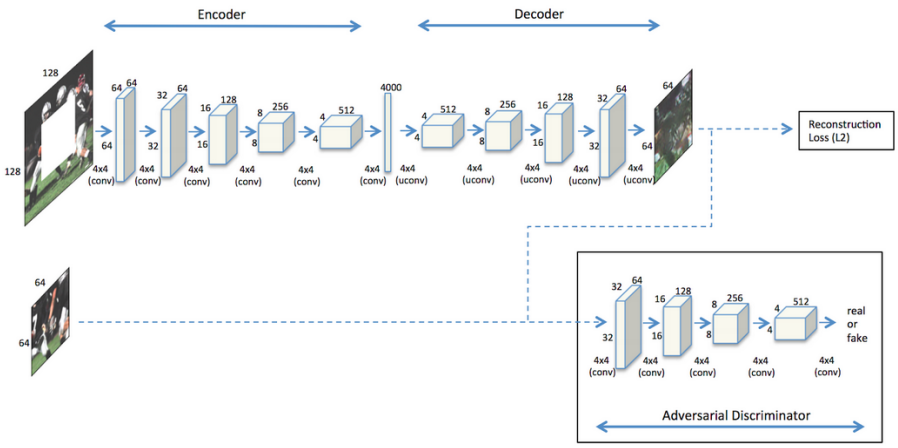


Figure 5: Inpainting Detailed Model Overview: Context Encoder

As we have stated earlier though the main purpose for this approach is not to be able to reconstruct or regenerate the pretext task but model to evolve in a way that it can be useful for other downstream tasks. Context encoders although a preliminary work it is actually reports promising results for that time. Although is pretty much below to the scores of ImageNet Pretrained models.

8.2.3 Cross-Channel Autoencoder

In Cross-Channel Autoencoder's [12] it is assumed that only single view of the input is accessed and the purpose of this network is to reconstruct the other views.

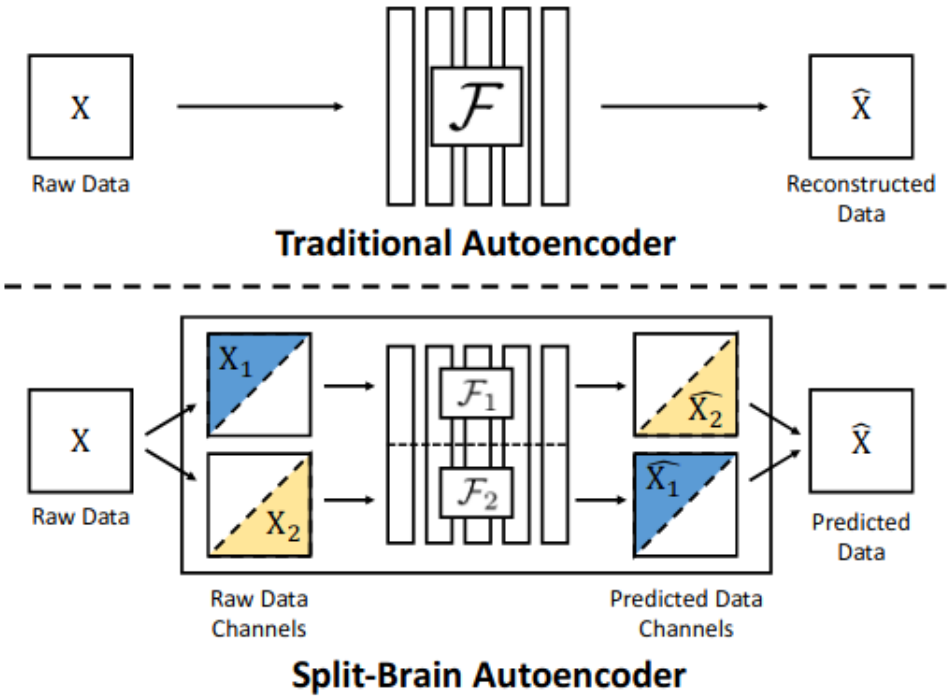


Figure 6: Traditional vs Split-Brain Autoencoder architectures. (top)Autoencoders learn feature representation F by learning to reconstruct input dataX. (bottom)The proposed split-brain autoencoder is composed of two disjoint sub-networks F1, F2, each trained to predict one data subset from another, changing the problem from reconstruction to prediction. The split-brain representation F is formed by concatenating the two sub-networks, and achieves strong transfer learning performance. The model is publicly available on https://richzhang.github.io/splitbrainauto.

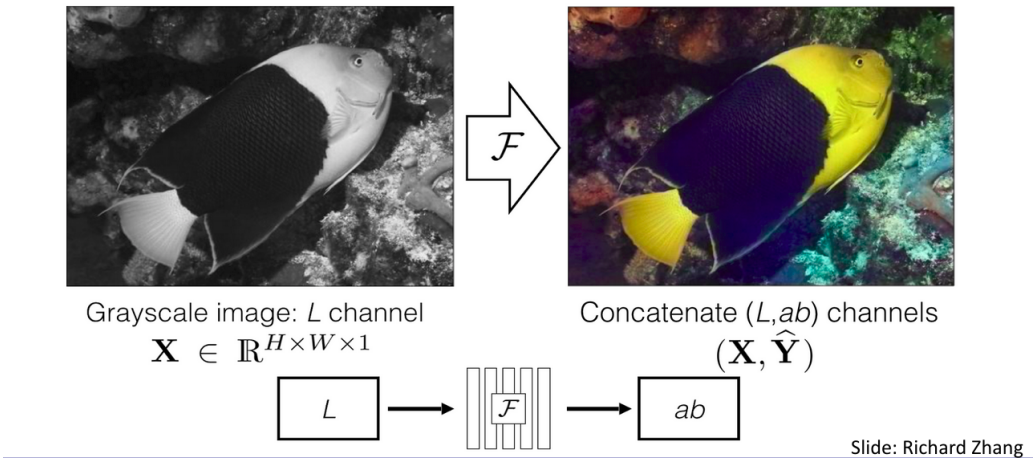


Figure 7: Image Colorization Example

In 7 image colorization task can be seen. In the figure the images are assumed to be in Lab color space. So given L channel ab channels are predicted and in the end all are concatenated.

For the objective again L2 loss is used and also additionally pixel values are treated as classes so they approached the problem as a pixel-wise classification problem.

In 6 you have basically two networks and each is given the other channel as input and tries to predict the other. So it can be said that those are complementing networks. And in the end their output is combined to reconstruct the original input image.

8.3 Proxy Tasks In Computer Vision

8.3.1 Relative Patch Prediction

Task is to predict the relative position of the second patch with respect to the first.

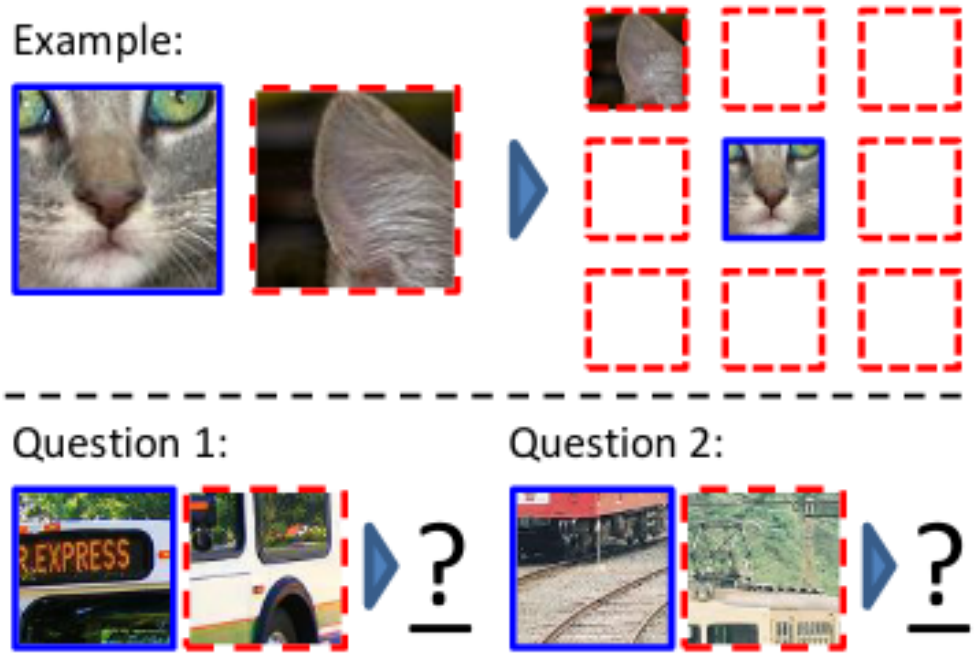


Figure 8: Learning Patch Representation

Architecture in the 9 is for pair classification. Dotted lines indicate shared weights and 'LRN' is a local response normalization layer. All conv and fc layers are followed by ReLU nonlinearities, except fc9 which feeds into a softmax classifier. So it is a pretty standard "Siamese" sort of architecture.

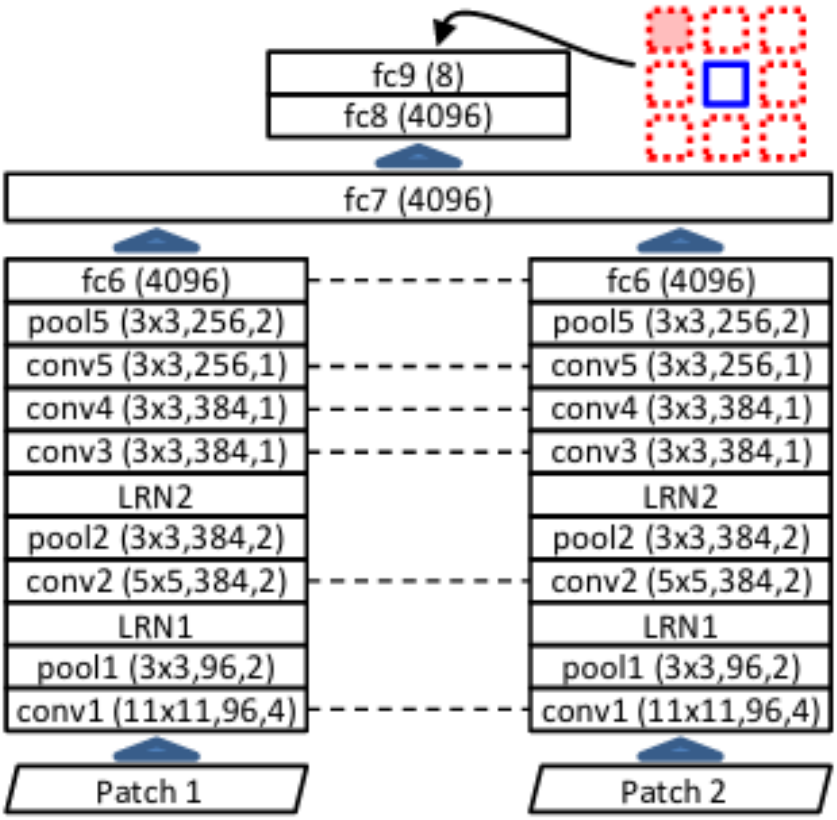


Figure 9: "Unsupervised Visual Representation Learning by Context Prediction" [2] Neural Network Architecture

8.3.2 Solving Jigsaw Puzzles

The tasks that we are dealing today actually consecutive works meaning that they share much incrementally improves the previous works.

Solving jigsaw puzzles follows this fashion. In that instead of just correctly guessing position of patches, this model is given with all the patches but all patches are shuffled. What this does is to correctly fix the position of the patches in order to form original image. This models network architecture is very similar to patch model and task is again classification.

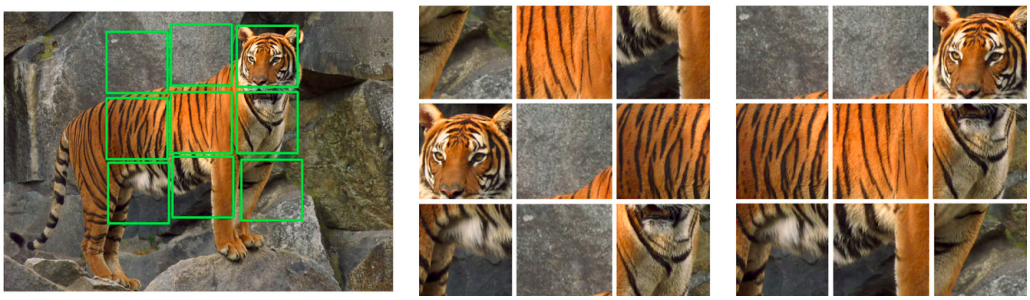


Figure 10: Solving Jigsaw Puzzle Task Representation

8.3.3 Rotation

Rotation task is to predict whether the given image is rotated or not and if it is rotated, the rotation angle should be determined as well. Again this problem is attacked in a way that is turned into a classification problem and the output is one of the four classes and multiples of 90 degrees. In order to correctly predict the rotations the model should have an understanding of the image and its correct rotation. So hopefully model learns what the object is maybe. And because of that although the approach is easy, its results are better than other self supervised models we learned so far.

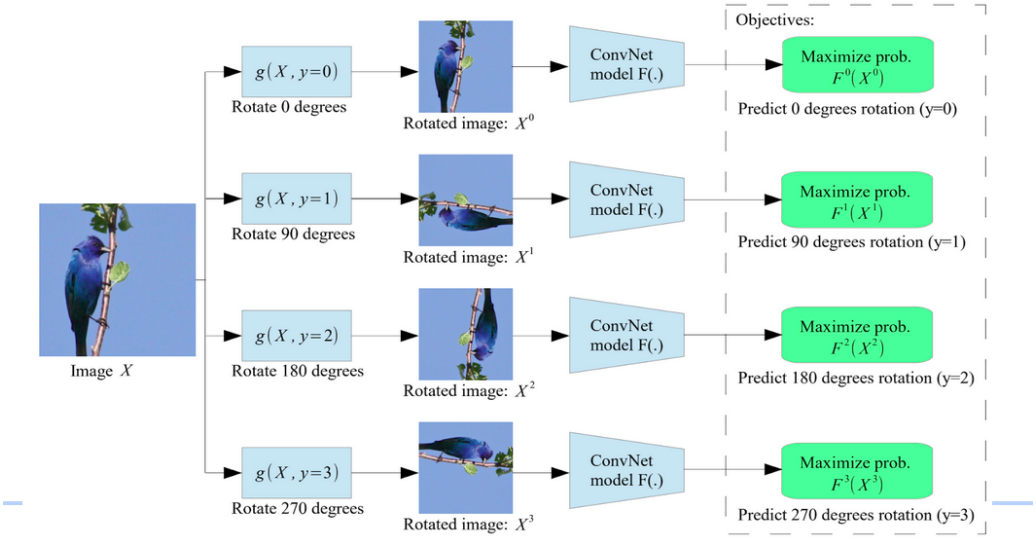


Figure 11: Rotation Task and The Model Architecture [3]

8.3.4 Temporal Coherence of Color

Task given a color video colorize all frames of a grayscale version using a reference frame for guidance. Main motivation for this task is to get an understanding of the environment and the objects in different frames and try to associate them with each other in different frames. So it can be said that tracking emerges from colorization.

Figure 12: Temporal Coherence of Color Task Visualization

8.4 Contrastive Learning

So far we needed to design a proxy task to have better representation learning schema. Now we would be mostly concentrating on a single objective trying to distinguish whether a pair of data samples are similar to each other or dissimilar to each other.

As a motivating example we will now focus on "Word2Vec" [8] which led to huge improvements in deep learning specifically in natural language processing.

8.4.1 Word2Vec

Forming word embeddings is a problem. One hot vector encoding approach can be speculated. But that is very sparse and requires a dimension for each new word and has no semantic relation because of orthogonality. So this is a naive way of embedding words.

Because of the inadequacy of one-hot word representations distributional representations of words had been investigated. It means each word gains meaning in response to its neighbors. So the meaning arises from context. This is one of the most successful ideas in NLP.

The meaning of a wampimuk(guess what this word is :)) is (can be approximated by, derived from) the set of contexts in which it occurs in texts.

In W2V model n-gram language modeling idea is used.

Unigram

$$P(w_1, w_2, \cdots, w_n) = \prod_{i=1}^{n} P(w_i) \tag{2}$$

Bigram

$$P(w_1, w_2, \cdots, w_n) = \prod_{i=2}^{n} P(w_i \mid w_{i-1}) \tag{3}$$

W2V paper involves two implementations. The first is a continuous bag of words. The other is skipgram. Importance negative sampling is used for propagation of supervision.

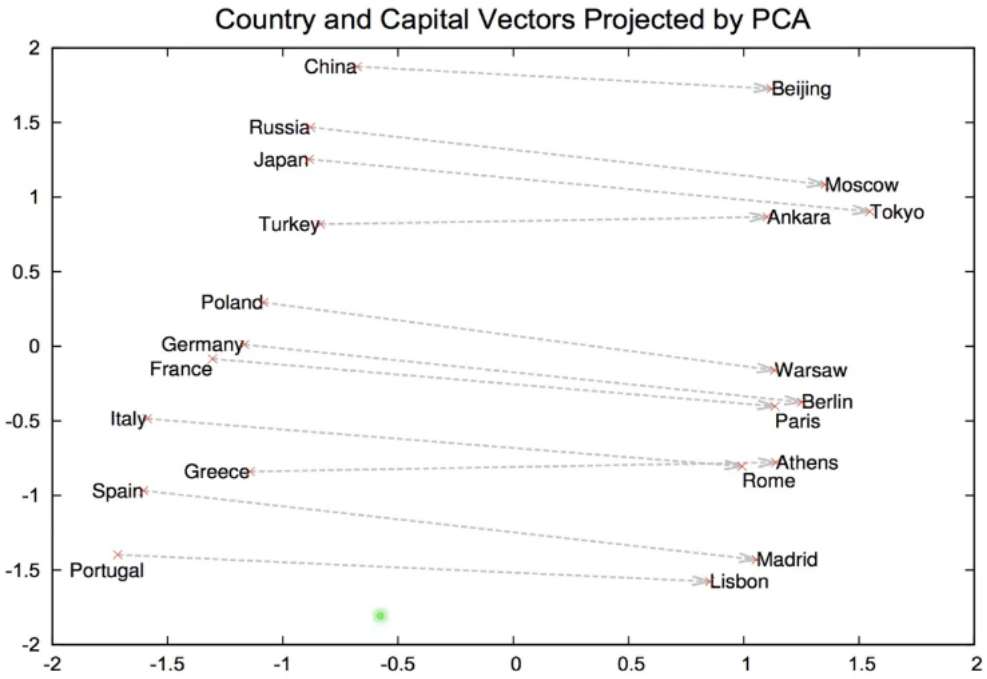


Figure 13: Semantic Embeddings of W2C

Learned embeddings from Word2Vec model are quite semantic. It captures semantic relations between words. In 13 it can be seen that vectoric relations are present in the differences of word embeddings.

Another important paper in this line of works is "Deep InfoMax" [7]. Input image is encoded into a grid structure by MxM feature map. Each feature is a vector. Vectors might be different for different classes of objects. And the idea is to maximize the encoder input and output. If you consider from the features corresponding features should be similar to the global feature of this input image. As a result we want to maximize shared information between global features and local features. Hence, local representations would become representatives of global features. So images would be distinguished by their local features. Classification accuracy results for this paper are superior for its time.

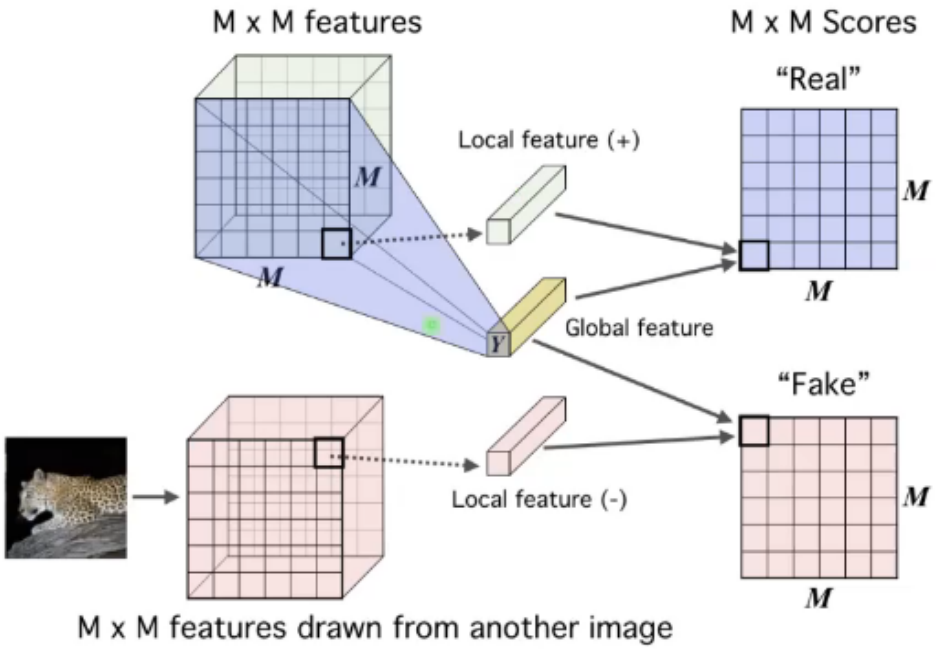


Figure 14: DeepInfo Max Model

Contrastive Loss, is the indication of similarity and dissimilarity between vectors and in "DeepInfoMax" this is what we are trying to reduce between local and global features.

8.4.2 Contrastive Predictive Coding

[10], the main idea of this paper is supposingly instead of having generative loss we have encoding and decoding. Very much like to DeepInfoMax approach, we may try to maximize mutual information between sequences of data. So again in the CPC approach we need to have negative samples as well.

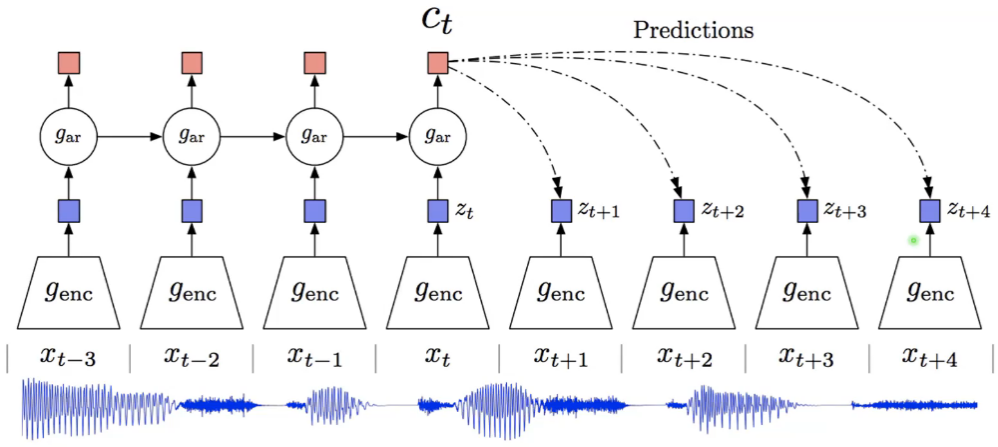


Figure 15: CPC Approach

With this manner you can learn much more semantic and slowly varying features. Really useful features can be extracted as a result.

Main idea in contrastive predictive coding is that we assume a particular context is observed regarding our data. Instead of just generating what comes next as in a classical Autoregressive sense, here we we would like to predict not the original input but latent codes that are obtained by some encoder. That constitutes the prediction stage. What makes it contrastive is the usage of contrastive loss.

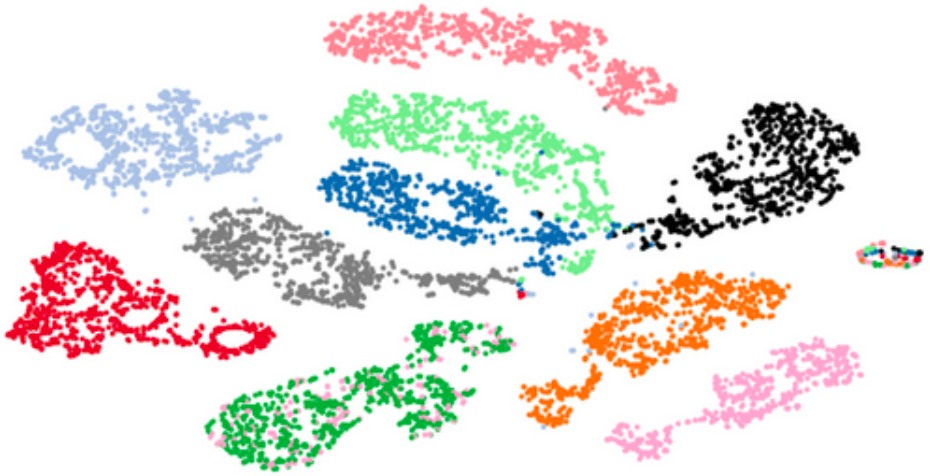


Figure 16: t-SNE visualization of audio (speech) representations for a subset of 10 speakers (out of 251). Every color represents a different speaker

CPC - Speech: In 16, it can be seen that the embeddings of input for the same users(speakers) clustered together. Thus we can infer that the model learned the characteristics of speakers from unlabeled data.

CPC - ImageNet: CPC framework is a generic one and can be used with other modalities. As an example in visual domain we can apply cpc by creating overlapping image patches and encoding them. After creating the encodings the idea is the same. Can we predict the other representations? Quite basic autoregressive task in the latent space. What expected is from our encoders then is to be able to encode what the object of patch is. So that it can predict the following image patches.

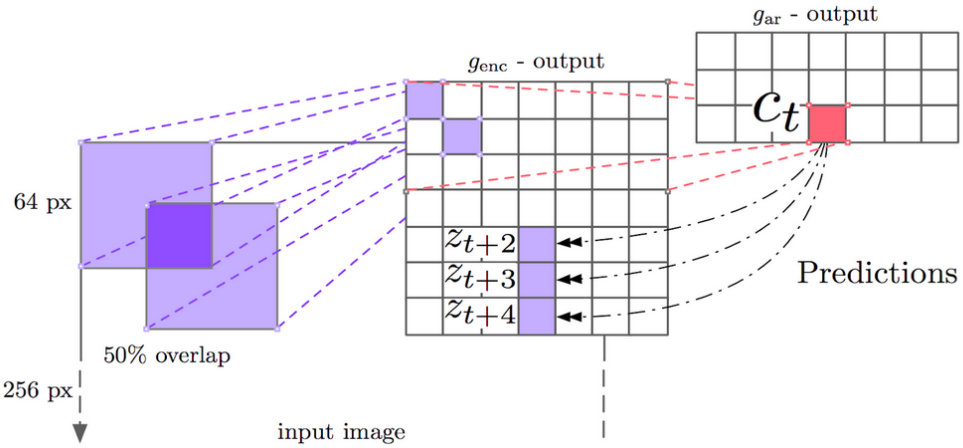


Figure 17: Visualization of CPC for Images

CPCV2 [6]: This work is very important turning point for self-supervised learning and CPC. The idea and the framework is the same. Only some changes are introduced for implementation. This model predicts the next couple of rows. However, pixel cnn sort of approach would be better and more efficient. This model computes similarity between contextual features and other latent codes. We want to compute it using some feature transformation. And we still need negative samples either from same image or from different images from the dataset. Loss is very similar to Word2Vec approach and InfoNCE(Noise Contrastive Estimation) loss is used.

Main advantage comes from the followings:

- Very large dataset is used (Unlabeled ImageNet)
- Long training time (500 epochs and 1 week)

- Larger patch size
- Effective number of negative samples
- **Augmenting every patch with a lot of spatial and color augmentations**

They can even exceed the performance of supervised baseline which is the first time for this self supervised approach.

We can again understand the importance of self-supervised learning by looking at the outcomes of this research. Because it allows more efficient way to train our classifiers as compared to the supervised approaches after the pretraining stage.

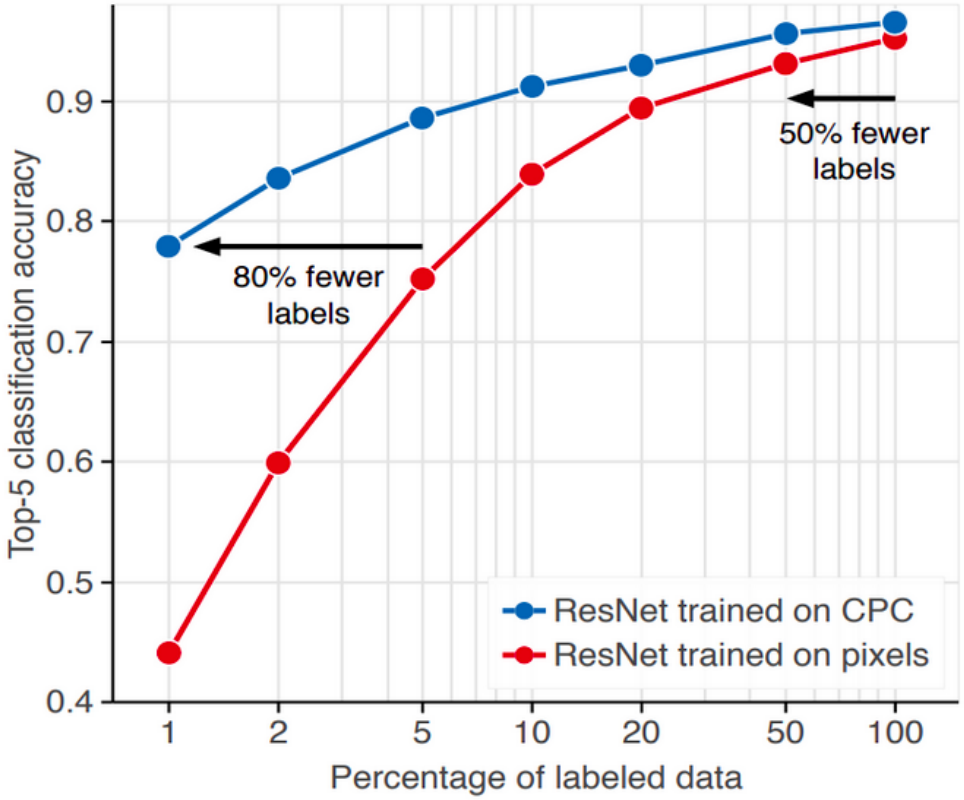


Figure 18: Data Efficiency of CPC V2

8.5 Momentum Contrast (MoCo) [5]

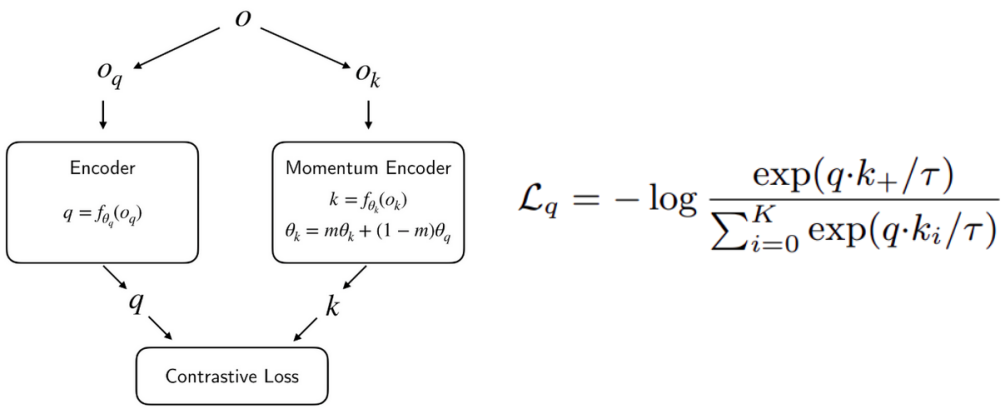


Figure 19: MoCo Overview and Loss Function

MoCo, checks whether two data instances are similar or not. Does it so by using contrastive loss. As a result, the model outputs similarity representations.

Main idea for MoCo is not using the same encoder for both parts. Thus, it introduces momentum encoder for the key part.

Critical thing for MoCo to be effective is to update weights of momentum encoder not in every step but basically update it at some certain intervals.

When update takes place, it uses momentum idea and makes use of weights of query encoder as well. This adds stability to the training.

When model capacity is maxed it even surpasses CPCv2 in terms of accuracy.

8.6 Simple Framework for Contrasive Learning of Visual Representations (SimCLR) [1]

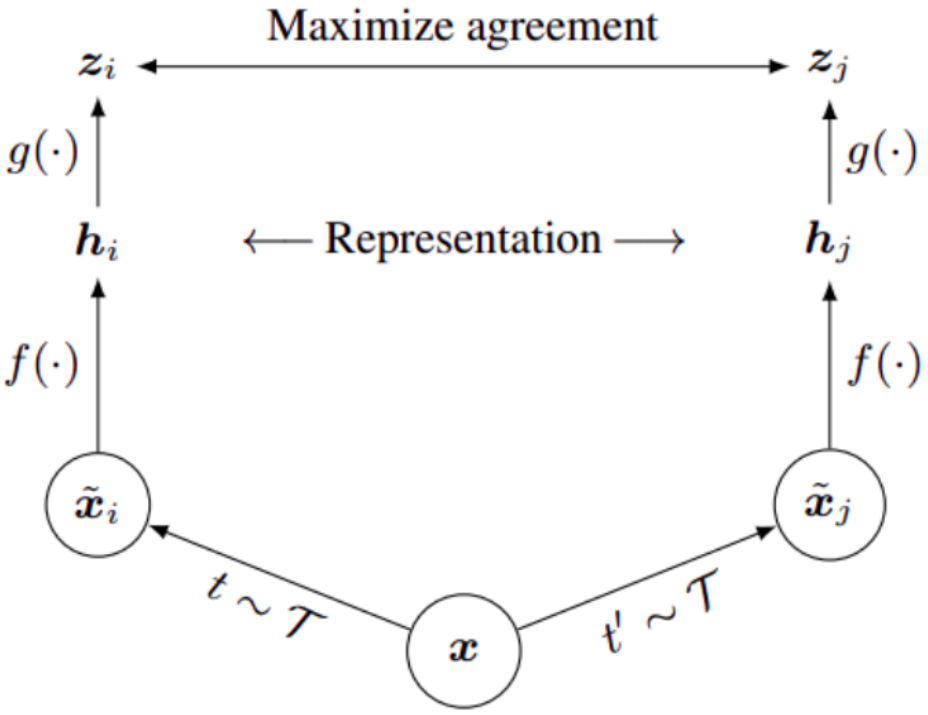


Figure 20: SimCLR Overview

Main idea of SimCLR is the use of projection head on top of encoded features. Projection head is a simple MLP. This model looks for agreement in transformed space. Data augmentation strategies are used. Distinguishes if transformed feature vectors are from the same source or not. In top-1 accuracy for ImageNet, accuracy is much better compared to MoCo and CPCv2.

Inspired from this work MoCoV2 is presented. The ideas added on top of MoCo are the followings:

- MLP Projection head
- Augmentation with extra blur
- Cosine learning rate schedule

After those improvements MoCov2 surpasses SimCLR in terms of accuracy.

8.7 Bootstrap Your Own Latent A New Approach to Self-Supervised Learning (BYOL) [4]

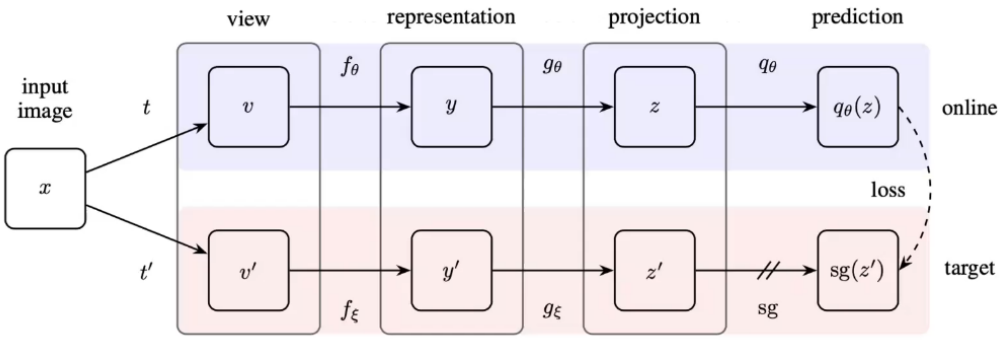


Figure 21: BYOL Architecture and Overview

BYOL borrows ideas from both MoCo and SimCLR. The general flow is as follows:

- perform some data augmentation
- generate two different views of the input Image
- Represent them using encoders
- Get projections of encodings from an MLP layer.

However, in contrast to other models BYOL does not use contrastive loss. It instead tries to predict the latent of the other data (query latent). Just as in MoCo query part of the model is not updated frequently and being updated by using exponential moving average idea plus stop gradients.

Main distinction of the BYOL from other papers is that **it does not use any negative samples at all!** and impressively performs a lot better that all the previous models that we have seen so far. In fact it even outperforms supervised peers in some datasets and tasks.

Method	Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
BYOL (ours)	**75.3**	91.3	**78.4**	**57.2**	**62.2**	**67.8**	60.6	82.5	75.5	90.4	94.2	**96.1**
SimCLR (repro)	72.8	90.5	74.4	42.4	60.6	49.3	49.8	81.4	**75.7**	84.6	89.3	92.6
SimCLR [8]	68.4	90.6	71.6	37.4	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2
Supervised-IN [8]	72.3	**93.6**	78.3	53.7	61.9	66.7	**61.0**	**82.8**	74.9	**91.5**	**94.5**	94.7
Fine-tuned:												
BYOL (ours)	**88.5**	**97.8**	86.1	**76.3**	63.7	91.6	**88.1**	**85.4**	**76.2**	91.7	**93.8**	97.0
SimCLR (repro)	87.5	97.4	85.3	75.0	63.9	91.4	87.6	84.5	75.4	89.4	91.7	96.6
SimCLR [8]	88.2	97.7	85.9	75.9	63.5	91.3	88.1	84.1	73.2	89.2	92.1	97.0
Supervised-IN [8]	88.3	97.5	**86.4**	75.8	**64.3**	**92.1**	86.0	85.0	74.6	**92.1**	93.3	**97.6**
Random init [8]	86.9	95.9	80.2	76.1	53.6	91.4	85.9	67.3	64.8	81.5	72.6	92.0

Figure 22: BYOL, transfer learning results from ImageNet(IN) with the standard ResNet-50 architecture

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[2] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction, 2016.

[3] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

[4] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[5] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[6] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

[7] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization, 2019.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[9] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting, 2016.

[10] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[11] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research (JMLR)*, 11(110):3371–3408, 2010.

[12] Richard Zhang, Phillip Isola, and Alexei A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. *CoRR*, abs/1611.09842, 2016.