Self-supervised learning in computer vision

План

- Постановка задачи
- Как начинался SSL
- Современные методы
  - Contrastive learning
  - Joint embedding architectures
  - Clusterization
  - SSL for transformers

Постановка задачи

Проблемы обучения без предобучения:

- Рандомная инициализация
- Малый размер датасетов

Постановка задачи

Большие датасеты:

+ умеем извлекать информацию из картинок

- разные домены

SSL:

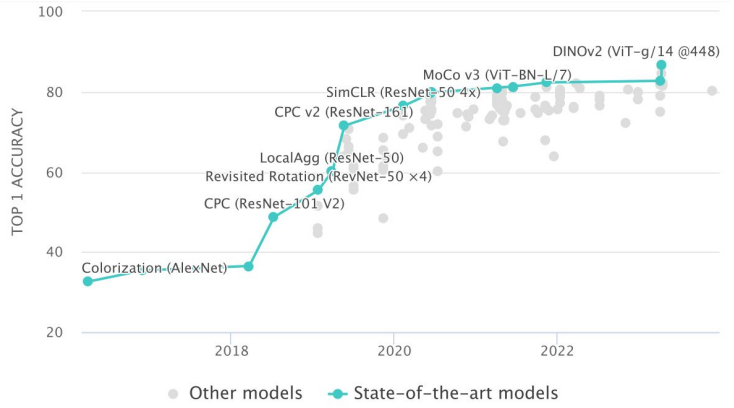
+ тот же домен
+ ещё больше данных

- больше ресурсов

Постановка задачи

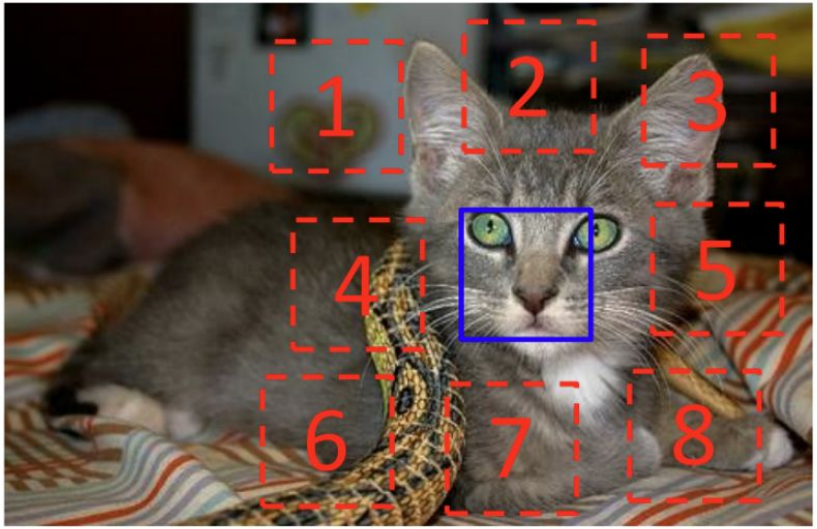
Linear Evaluation Protocol - метрика SSL

1. Предобучаем метод SSL на ImageNet
2. “Замораживаем” feature extractor
3. Обучаем линейный классификатор
4. Оцениваем качество на ImageNet



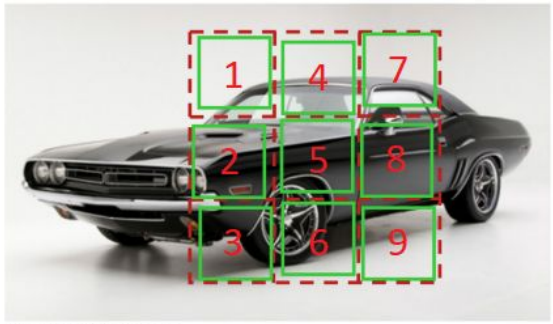
Как начинался SSL

Предсказание контекста куска изображения - 2015



Где правое ухо, если мордочка по центру?

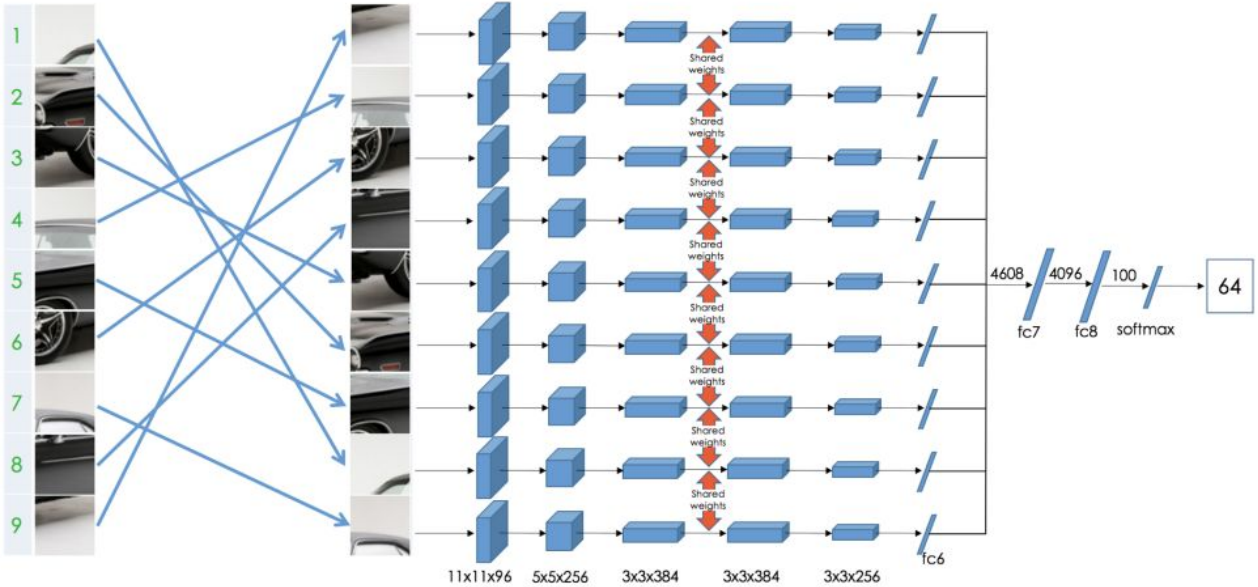
Jigsaw (пазл) - 2016



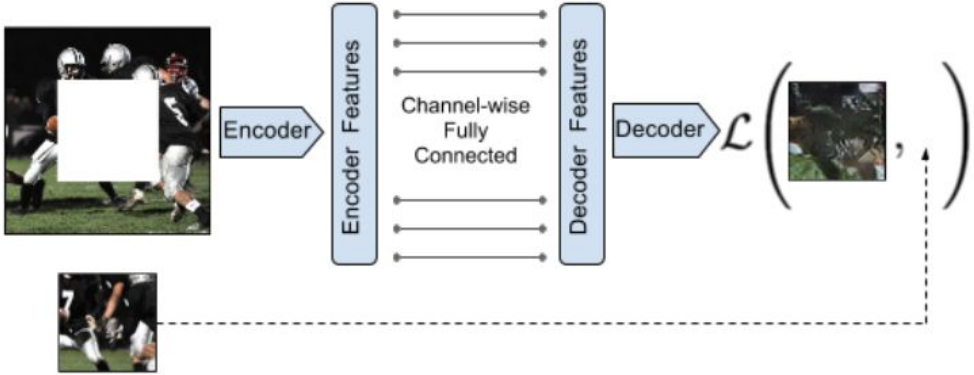
Permutation Set

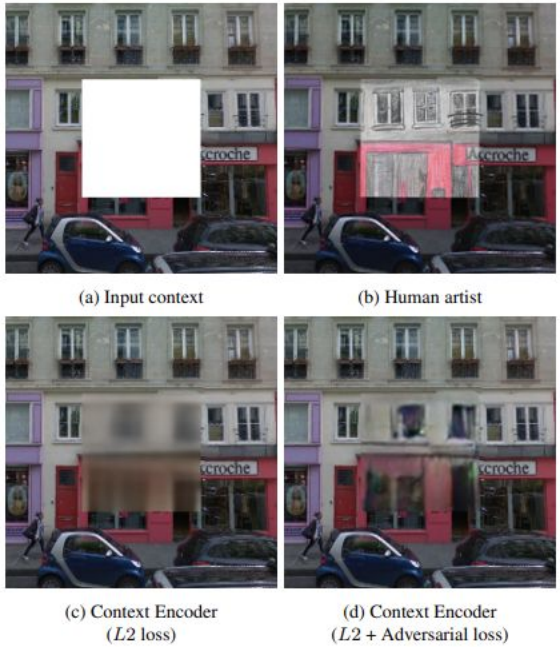
index	permutation
64	9,4,6,8,3,2,5,1,7

Reorder patches according to the selected permutation



Маскирование изображений - 2016

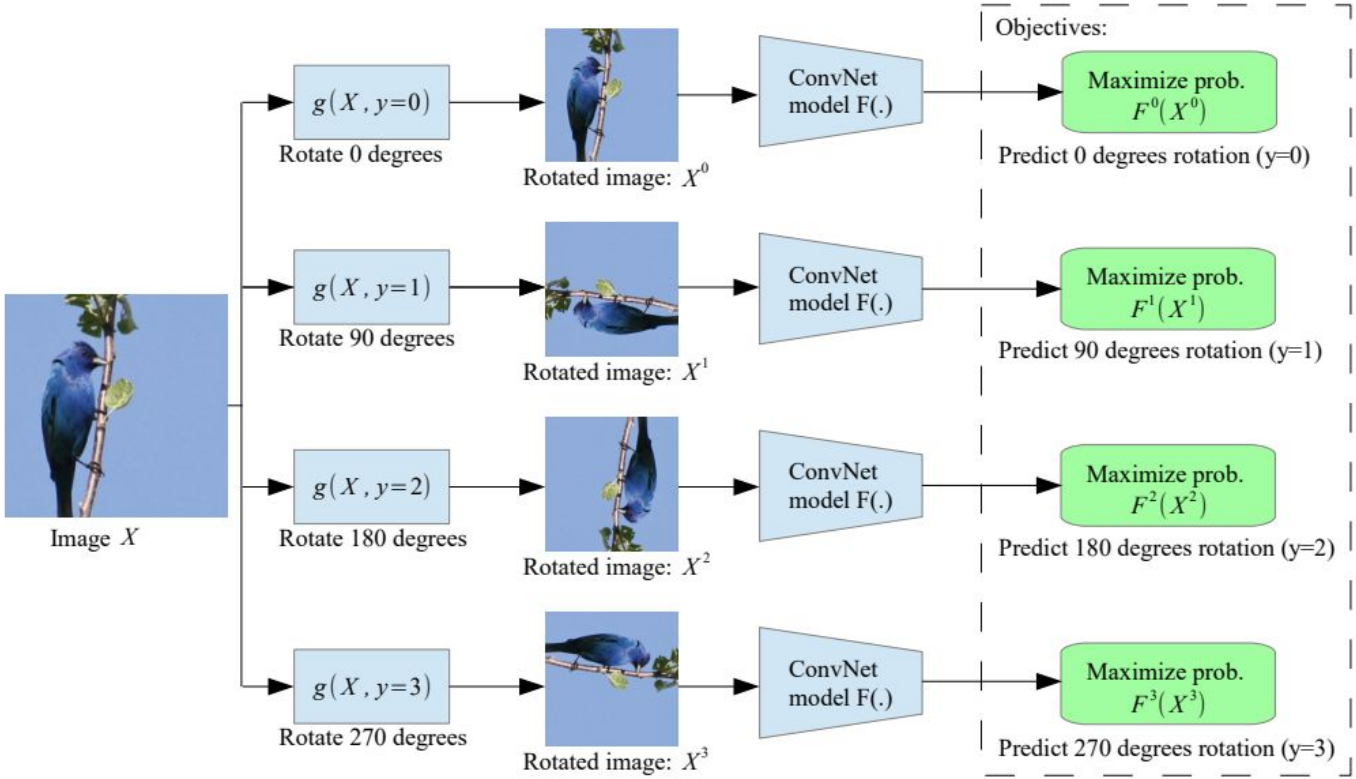




(a) Input context (b) Human artist

(c) Context Encoder ($L2$ loss) (d) Context Encoder ($L2$ + Adversarial loss)

Предсказание поворотов - 2018



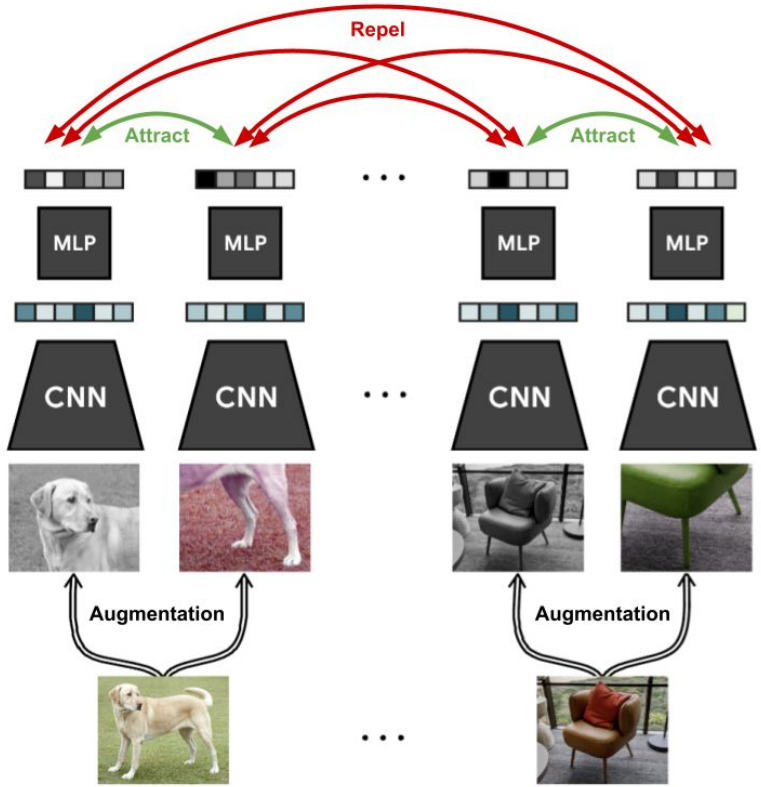
Contrastive Learning

Идея

- Одинаковые картинки “похожи”, а разные “отличаются”

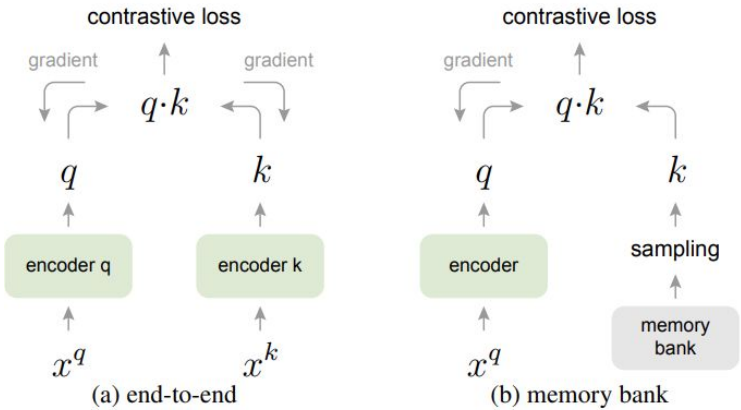
Проблема

- Картинки могут быть похожими, но будут считаться за разные



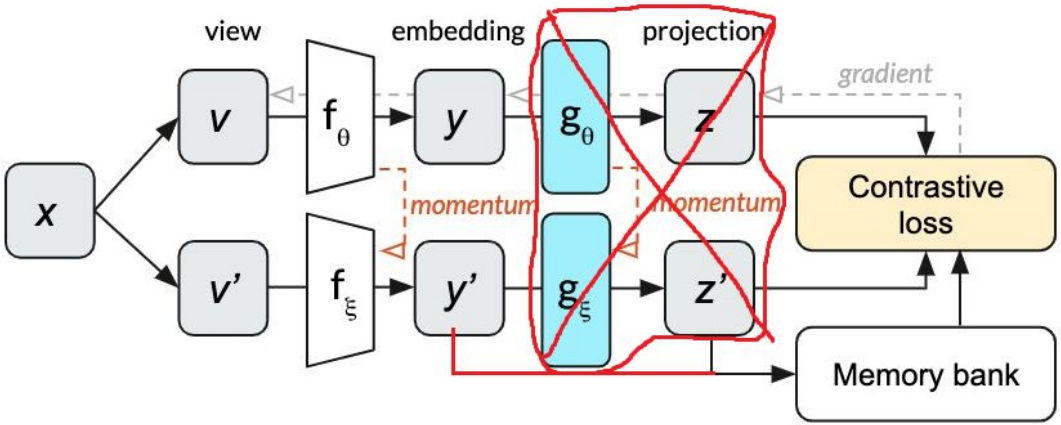
Contrastive Loss

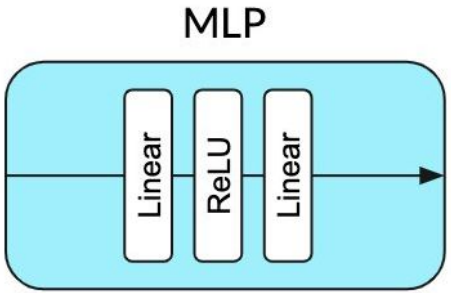
$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$



Momentum Contrast (MoCo) - 2019

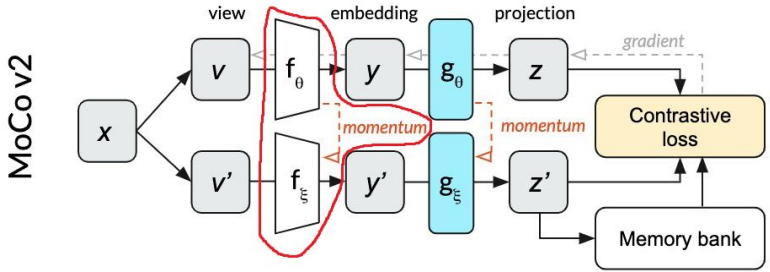
MoCo v2 ~~v1~~





MoCo - Momentum Encoder

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}$$



Зачем?

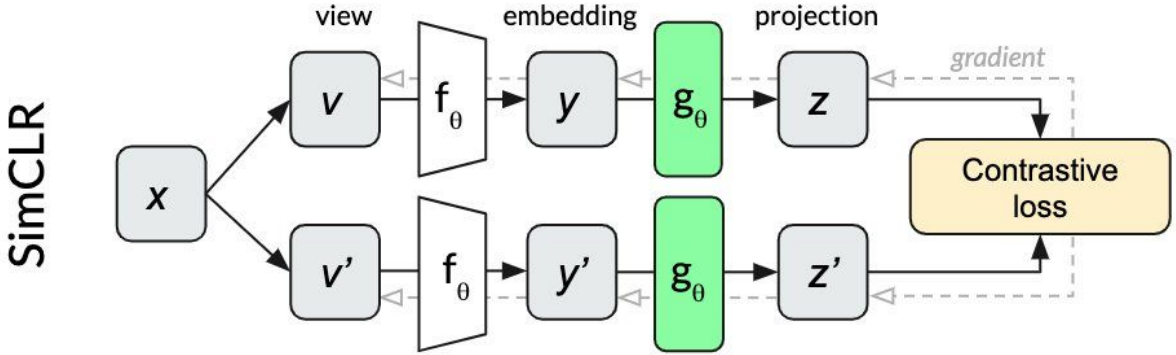
- градиент по всем значениям из memory bank - сложно
- копирование весов даёт результаты хуже

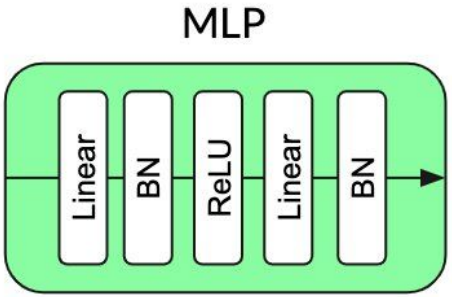
MoCo - результаты

case	unsup. pre-train MLP	aug+	cos	epochs	ImageNet acc.	VOC detection AP_{50}	AP	AP_{75}
supervised					76.5	81.3	53.5	58.8
MoCo v1				200	60.6	81.5	55.9	62.6
(a)	✓			200	66.2	82.0	56.4	62.6
(b)		✓		200	63.4	82.2	56.8	63.2
(c)	✓	✓		200	67.3	**82.5**	57.2	63.9
(d)	✓	✓	✓	200	67.5	82.4	57.0	63.6
(e)	✓	✓	✓	**800**	**71.1**	**82.5**	**57.4**	**64.0**

Table 1. **Ablation of MoCo baselines**, evaluated by ResNet-50 for (i) ImageNet linear classification, and (ii) fine-tuning VOC object detection (mean of 5 trials).

Simple Framework for Contrastive Learning of Visual Representations (SimCLR) - 2020





- нет Memory Bank, но большой Batch Size
- модели обучаются параллельно

SimCLR - результаты

Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

Table 8. Comparison of transfer learning performance of our self-supervised approach with supervised baselines across 12 natural image classification datasets, for ResNet-50 (4×) models pretrained on ImageNet.

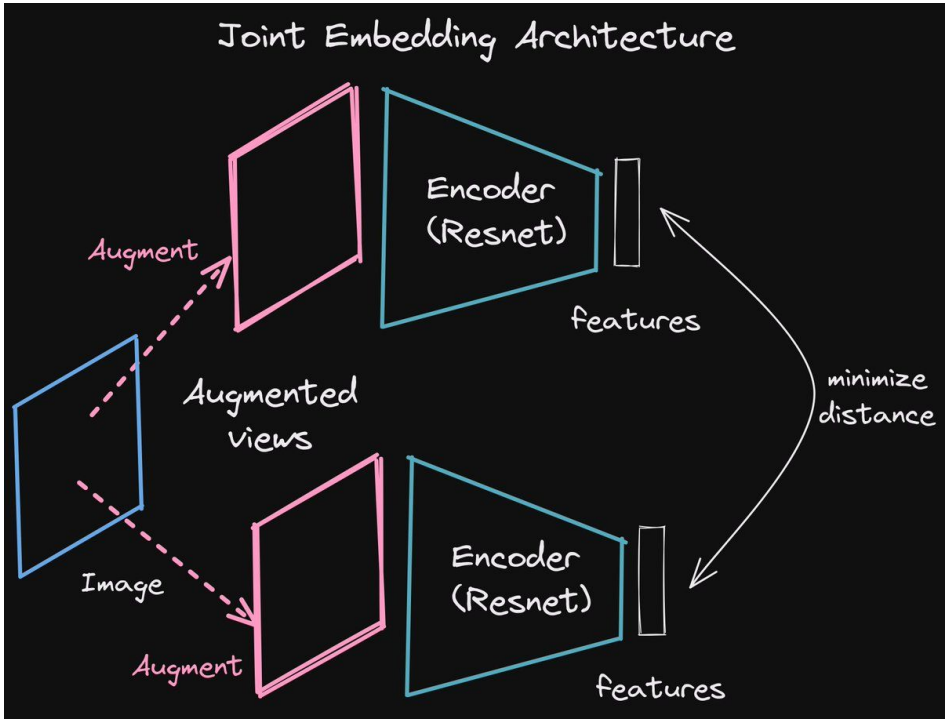
Joint embedding architectures

Идея

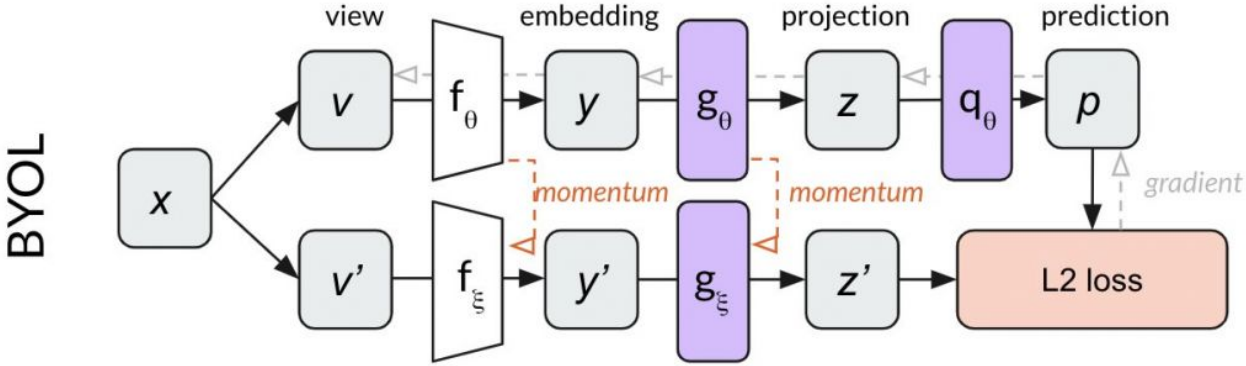
- Минимизируем расстояние между эмбеддингами аугментаций с одной картинки

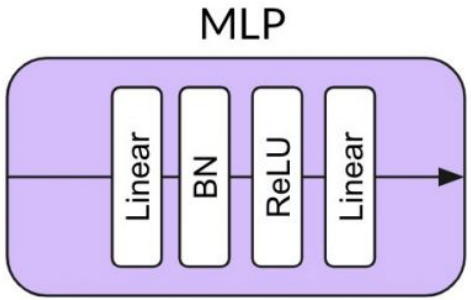
Проблема

- Есть тривиальное решение - выдавать всегда константу



Bootstrap Your Own Latent (BYOL) - 2020

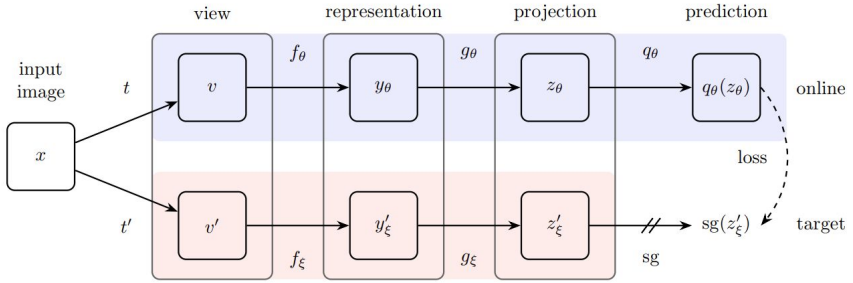




$$\mathcal{L}_{\theta,\xi} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\| q_\theta(z_\theta) \right\|_2 \cdot \left\| z'_\xi \right\|_2}$$

BYOL - почему решение не коллапсирует

- ξ двигается **не** в сторону $\nabla_{\xi}\mathcal{L}^{\texttt{BYOL}}_{\theta,\xi}$
- немного математики:



$$q^\star \triangleq \arg\min_q \mathbb{E}\left[\left\|q(z_\theta) - z'_\xi\right\|_2^2\right], \quad \text{where} \quad q^\star(z_\theta) = \mathbb{E}\left[z'_\xi | z_\theta\right]$$

$$\nabla_\theta \mathbb{E}\left[\left\|q^\star(z_\theta) - z'_\xi\right\|_2^2\right] = \nabla_\theta \mathbb{E}\left[\left\|\mathbb{E}\left[z'_\xi | z_\theta\right] - z'_\xi\right\|_2^2\right] = \nabla_\theta \mathbb{E}\left[\sum_i \mathrm{Var}(z'_{\xi,i} | z_\theta)\right]$$

Note that for any random variables X, Y, and Z, $\mathrm{Var}(X|Y,Z) \leq \mathrm{Var}(X|Y)$

$\mathrm{Var}(z'_\xi | z_\theta) \leq \mathrm{Var}(z'_\xi | c) \Rightarrow$ решение с константой неустойчиво

BYOL - результаты

Method	Top-1	Top-5
Local Agg.	60.2	-
PIRL [35]	63.6	-
CPC v2 [32]	63.8	85.3
CMC [11]	66.2	87.0
SimCLR [8]	69.3	89.0
MoCo v2 [37]	71.1	-
InfoMin Aug. [12]	73.0	91.1
BYOL (ours)	**74.3**	**91.6**

(a) ResNet-50 encoder.

Method	Architecture	Param.	Top-1	Top-5
SimCLR [8]	ResNet-50 (2×)	94M	74.2	92.0
CMC [11]	ResNet-50 (2×)	94M	70.6	89.7
BYOL (ours)	ResNet-50 (2×)	94M	77.4	93.6
CPC v2 [32]	ResNet-161	305M	71.5	90.1
MoCo [9]	ResNet-50 (4×)	375M	68.6	-
SimCLR [8]	ResNet-50 (4×)	375M	76.5	93.2
BYOL (ours)	ResNet-50 (4×)	375M	78.6	94.2
BYOL (ours)	ResNet-200 (2×)	250M	**79.6**	**94.8**

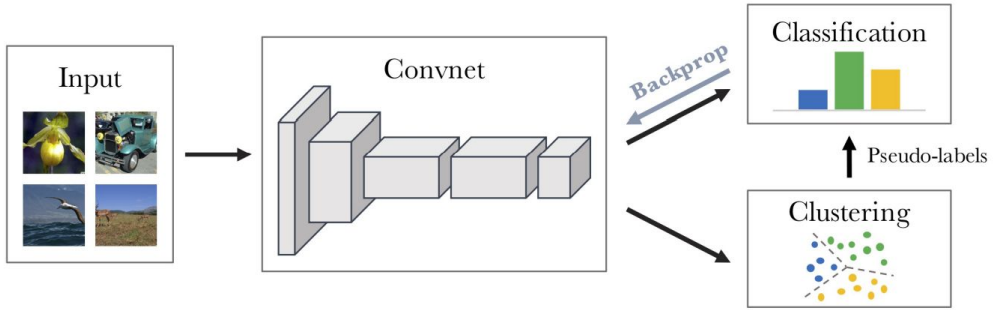
(b) Other ResNet encoder architectures.

Table 1: Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet.

Методы, основанные на кластеризации

Идея

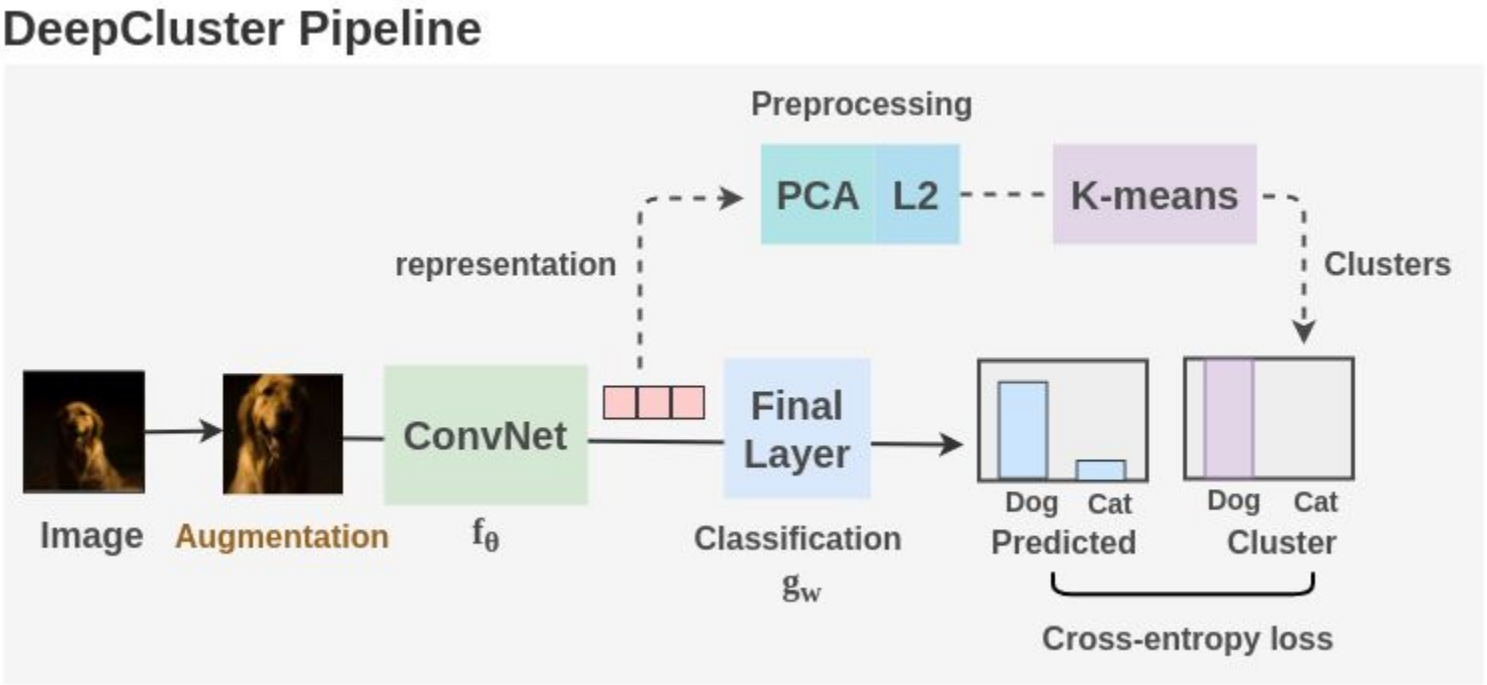
- Давайте сами себе разметим датасет



Проблема

- Тривиальное решение:
  - все данные в 1 кластере

DeepCluster - 2018



DeepCluster - избегаем тривиальных решений

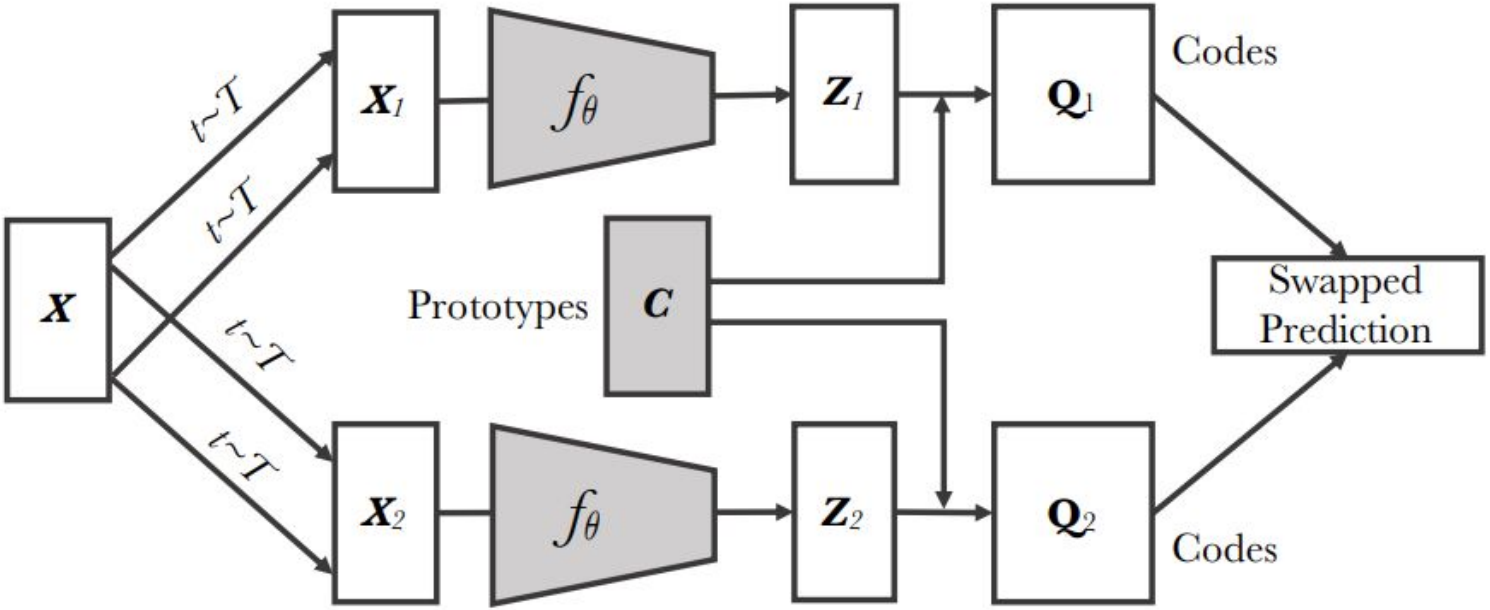
- Будем пересчитывать кластеры на каждом шаге
- Постараемся, чтобы распределение картинок в датасете было равномерным

DeepCluster - результаты

Method	ImageNet					Places				
	conv1	conv2	conv3	conv4	conv5	conv1	conv2	conv3	conv4	conv5
Places labels	–	–	–	–	–	22.1	35.1	40.2	43.3	44.6
ImageNet labels	19.3	36.3	44.2	48.3	50.5	22.7	34.8	38.4	39.4	38.7
Random	11.6	17.1	16.9	16.3	14.1	15.7	20.3	19.8	19.1	17.5
Pathak *et al.* [38]	14.1	20.7	21.0	19.8	15.5	18.2	23.2	23.4	21.9	18.4
Doersch *et al.* [25]	16.2	23.3	30.2	31.7	29.6	19.7	26.7	31.9	32.7	30.9
Zhang *et al.* [28]	12.5	24.5	30.4	31.5	30.3	16.0	25.7	29.6	30.3	29.7
Donahue *et al.* [20]	17.7	24.5	31.0	29.9	28.0	21.4	26.2	27.1	26.1	24.0
Noroozi and Favaro [26]	**18.2**	28.8	34.0	33.9	27.1	23.0	32.1	35.5	34.8	31.3
Noroozi *et al.* [45]	18.0	**30.6**	34.3	32.5	25.7	**23.3**	**33.9**	36.3	34.7	29.6
Zhang *et al.* [43]	17.7	29.3	35.4	35.2	32.8	21.3	30.7	34.0	34.1	32.5
DeepCluster	12.9	29.2	**38.2**	**39.8**	**36.1**	18.6	30.8	**37.0**	**37.5**	**33.1**

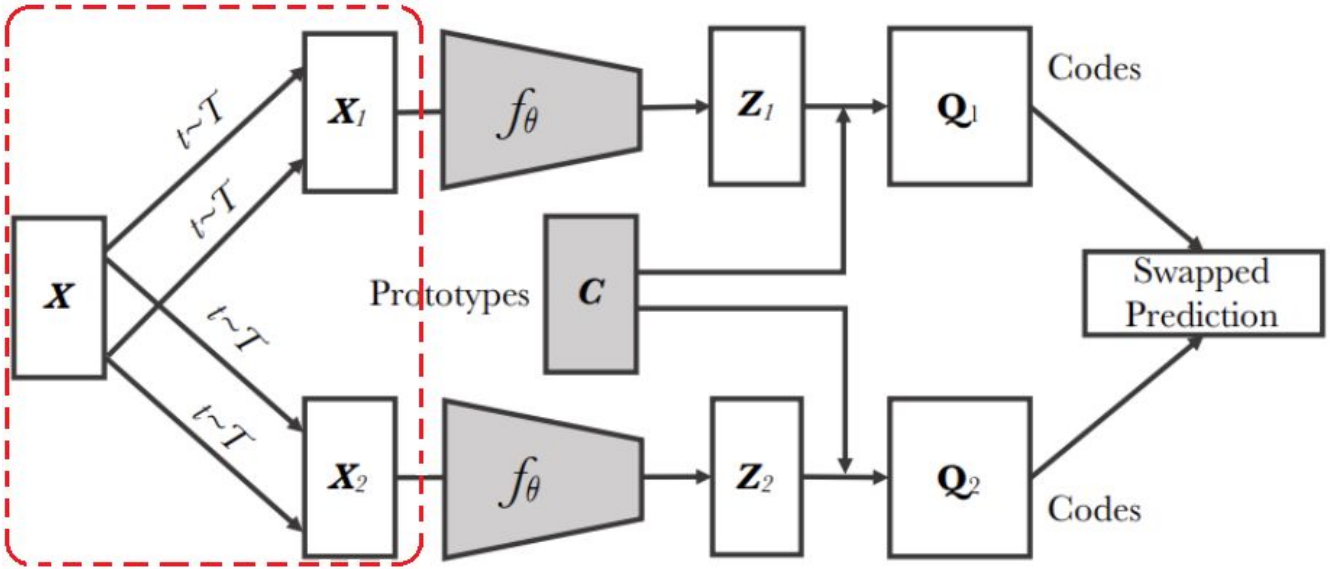
Table 1: Linear classification on ImageNet and Places using activations from the convolutional layers of an AlexNet as features.

Swapping **A**ssignments between multiple **V**iews of the same image (SwAV) - 2020



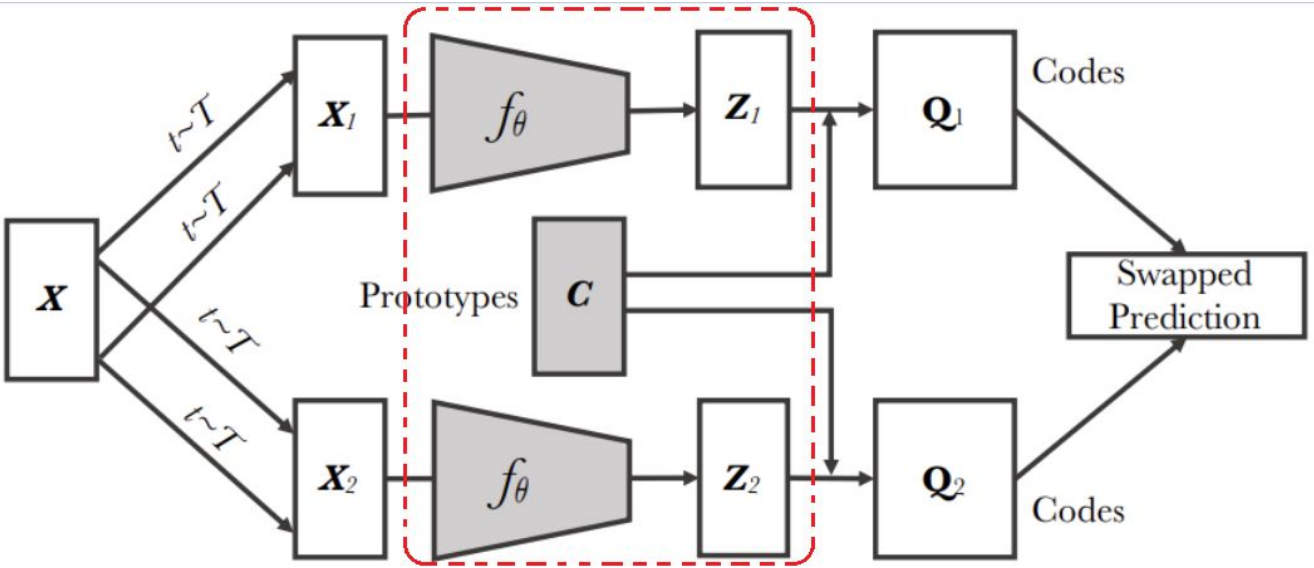
SwAV - как работает

- Аугментируем изображения батча, получаем $X_t,\ X_s$



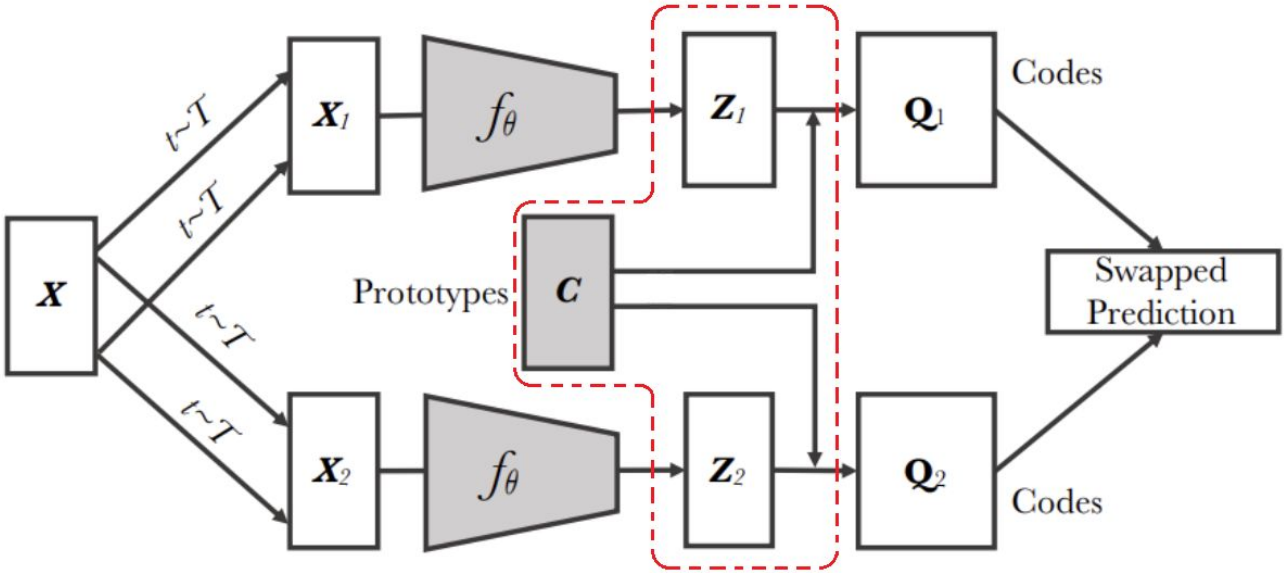
SwAV - как работает

- Прогоняем аугментации через модель f_θ, получаем эмбеддинги $Z_t,\ Z_s$



SwAV - как работает

- Запоминаем эмбеддинги, прогоняем их через C, нормализуем полученную матрицу с помощью Sinkhorn-Knopp, получаем Q_t , Q_s

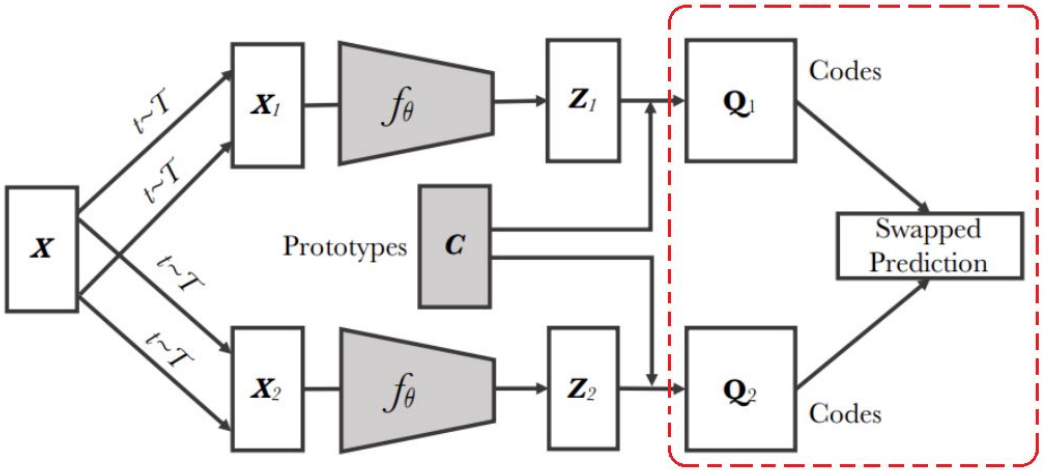


SwAV - как работает

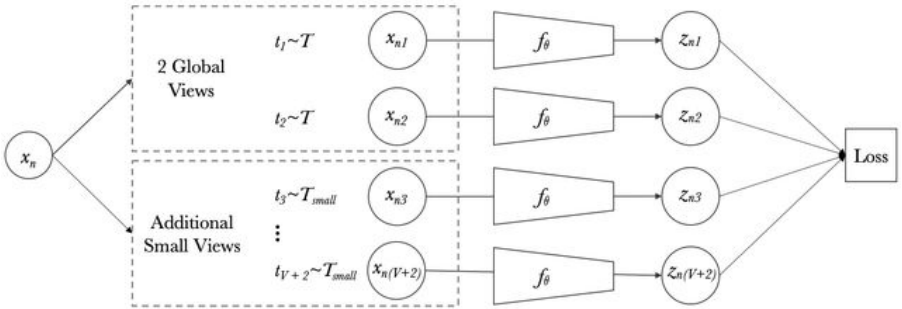
Считаем Loss:

$$L(\mathbf{z}_t, \mathbf{z}_s) = \ell(\mathbf{z}_t, \mathbf{q}_s) + \ell(\mathbf{z}_s, \mathbf{q}_t)$$

$$\ell(\mathbf{z}_t, \mathbf{q}_s) = -\sum_k \mathbf{q}_s^{(k)} \log \mathbf{p}_t^{(k)}, \quad \text{where} \quad \mathbf{p}_t^{(k)} = \frac{\exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_{k'}\right)}$$



SwAV - Multi-Crop



Method	Top-1 2x224	Top-1 2x160+4x96	Δ
Supervised	76.5	76.0	−0.5
SimCLR	68.2	70.6	+2.4
SeLa-v2	67.2	71.8	+4.6
DeepCluster-v2	70.2	74.3	+4.1
SwAV	70.1	74.1	+4.0

SwAV - результаты

Method	Arch.	Param.	Top1
Supervised	R50	24	76.5
Colorization [65]	R50	24	39.6
Jigsaw [46]	R50	24	45.7
NPID [58]	R50	24	54.0
BigBiGAN [15]	R50	24	56.6
LA [68]	R50	24	58.8
NPID++ [44]	R50	24	59.0
MoCo [24]	R50	24	60.6
SeLa [2]	R50	24	61.5
PIRL [44]	R50	24	63.6
CPC v2 [28]	R50	24	63.8
PCL [37]	R50	24	65.9
SimCLR [10]	R50	24	70.0
MoCov2 [11]	R50	24	71.1
SwAV	R50	24	**75.3**

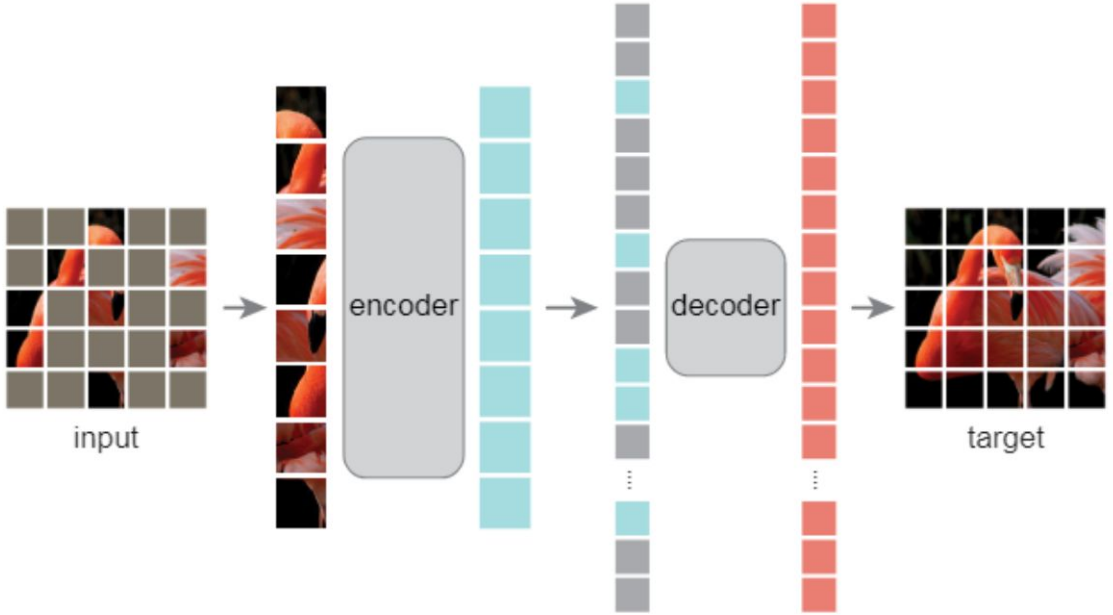
Figure 2: **Linear classification on ImageNet.**

	Linear Classification			Object Detection		
	Places205	VOC07	iNat18	VOC07+12 (Faster R-CNN R50-C4)	COCO (Mask R-CNN R50-FPN)	COCO (DETR)
Supervised	53.2	87.5	46.7	81.3	39.7	40.8
SwAV	**56.7**	**88.9**	**48.6**	**82.6**	**41.6**	**42.1**

Table 2: **Transfer learning on downstream tasks.**

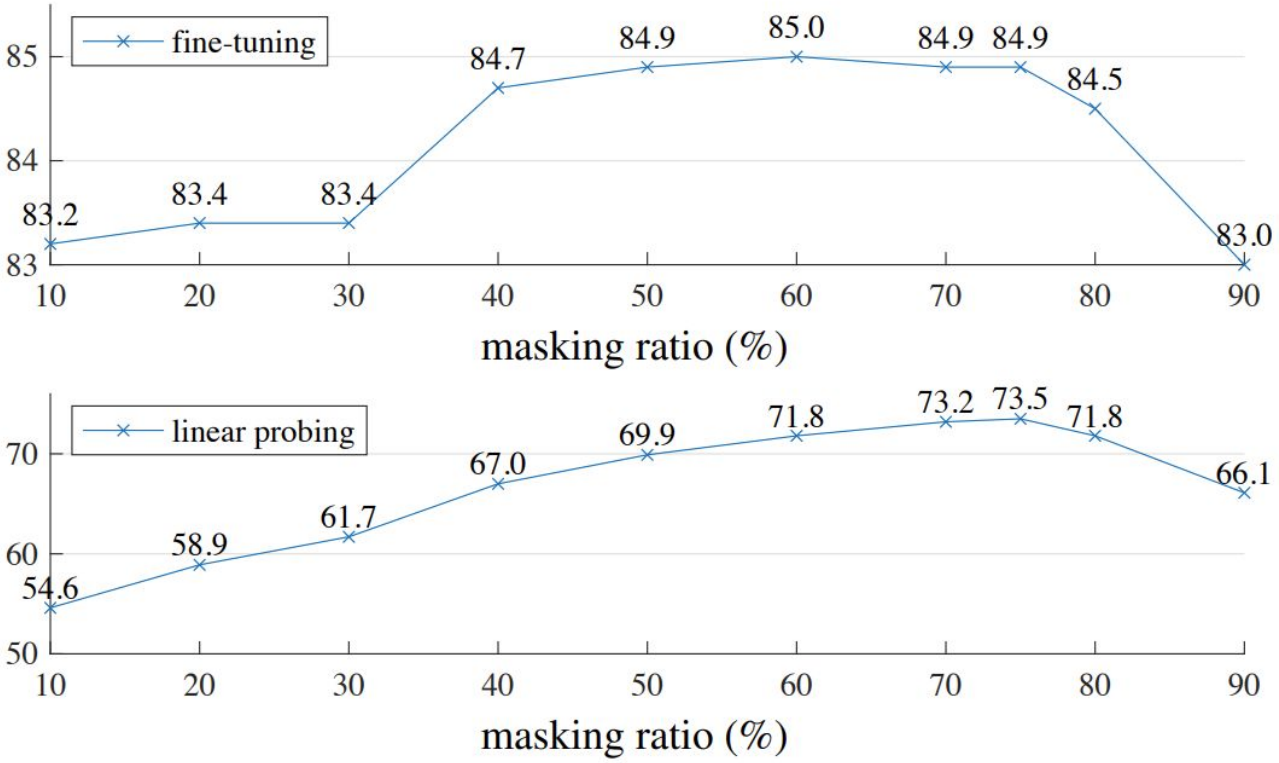
SSL for transformers

Masked AutoEncoder (MAE) - 2021

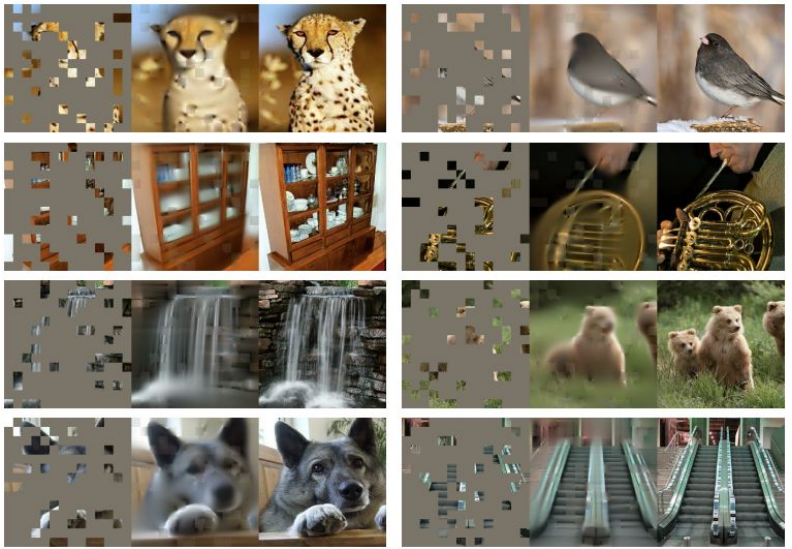


$$\mathcal{L}_\theta = MSE$$

MAE - Masking Ratio



MAE



MAE - результаты

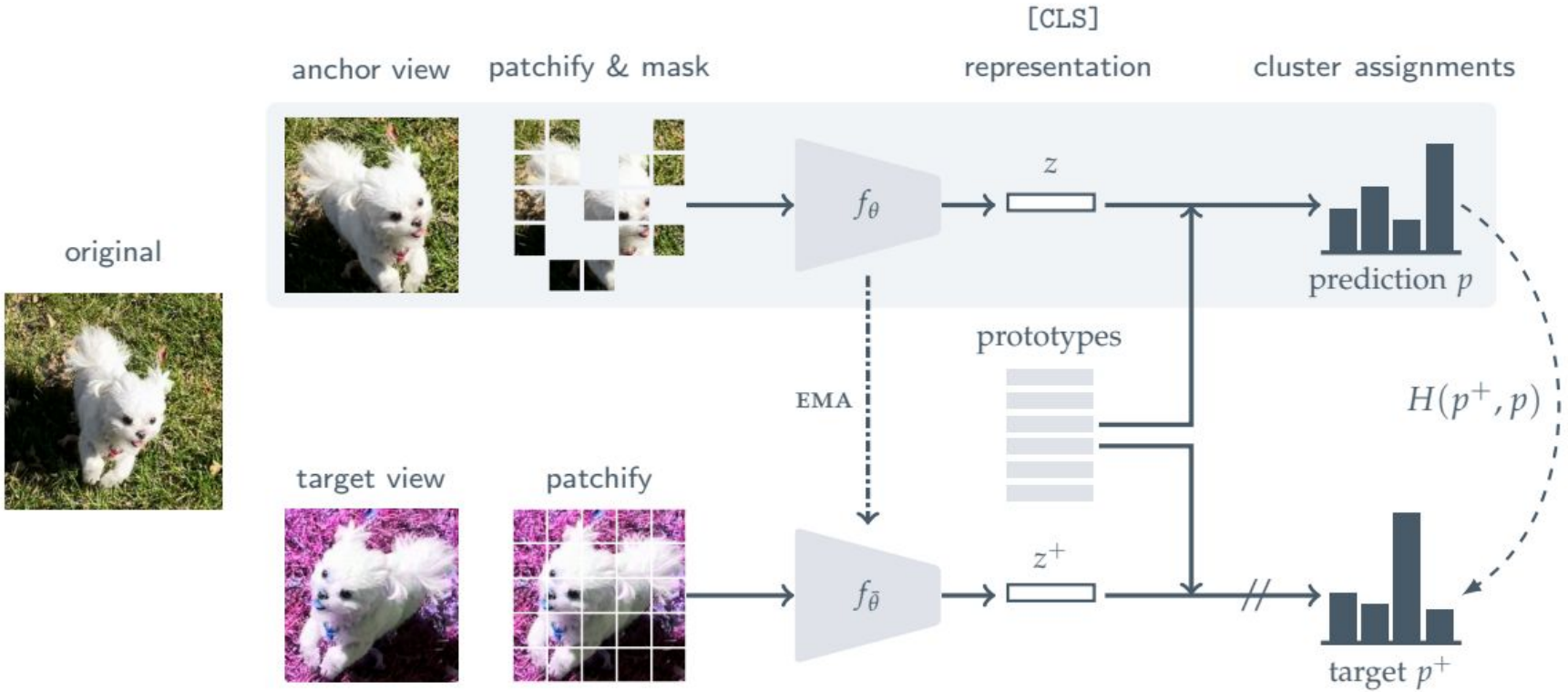
method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H_{448}
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	83.6	85.9	86.9	**87.8**

Table 3. **Comparisons with previous results on ImageNet-1K**. The pre-training data is the ImageNet-1K

		AP^{box}		AP^{mask}	
method	pre-train data	ViT-B	ViT-L	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3	42.9	43.9
MoCo v3	IN1K	47.9	49.3	42.7	44.0
BEiT	IN1K+DALLE	49.8	**53.3**	44.4	47.1
MAE	IN1K	**50.3**	**53.3**	**44.9**	**47.2**

Table 4. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline.

Masked Siamese Network (MSN) - 2022



MSN - Masks

(a) No Mask

(b) Random Mask

(c) Focal Mask

Anchor View	Top 1
No Mask	49.3
Focal Mask	39.3
Random Mask	52.3
Random Mask + Focal Mask	**59.8**

MSN - результаты

Table 3: Linear evaluation on ImageNet-1K using 100% of the labels.

Method	Architecture	Params.	Epochs	Top 1
Comparing similar architectures				
SimCLRv2 (Chen et al., 2020c)	RN50	24M	800	71.7
BYOL (Grill et al., 2020)	RN50	24M	1000	74.4
DINO (Caron et al., 2021)	ViT-S/16	22M	800	77.0
iBOT (Zhou et al., 2021)	ViT-S/16	22M	800	**77.9**
MSN	ViT-S/16	22M	600	76.9
Comparing larger architectures				
MAE (He et al., 2021)	ViT-H/14	632M	1600	76.6
BYOL (Grill et al., 2020)	RN200 (2×)	250M	800	79.6
SimCLRv2 (Chen et al., 2020c)	RN151+SK (3×)	795M	800	79.8
iBOT (Zhou et al., 2021)	ViT-B/16	86M	400	79.4
DINO (Caron et al., 2021)	ViT-B/8	86M	300	80.1
MoCov3 (Chen et al., 2021)	ViT-BN-L/7	304M	300	**81.0**
MSN	ViT-L/7	304M	200	**80.7**

Итог:

- Разобрались, какие есть плюсы и минусы у SSL
- Вспомнили ранние методы SSL
- Рассмотрели основные современные методы SSL и примеры их реализации

Источники:

- Цикл статей на хабре, откуда я брал статьи
- BatchNorm в BYOL