Uczenie typu self-supervised w przetwarzaniu obrazów

ARKADIUSZ KWASIGROCH

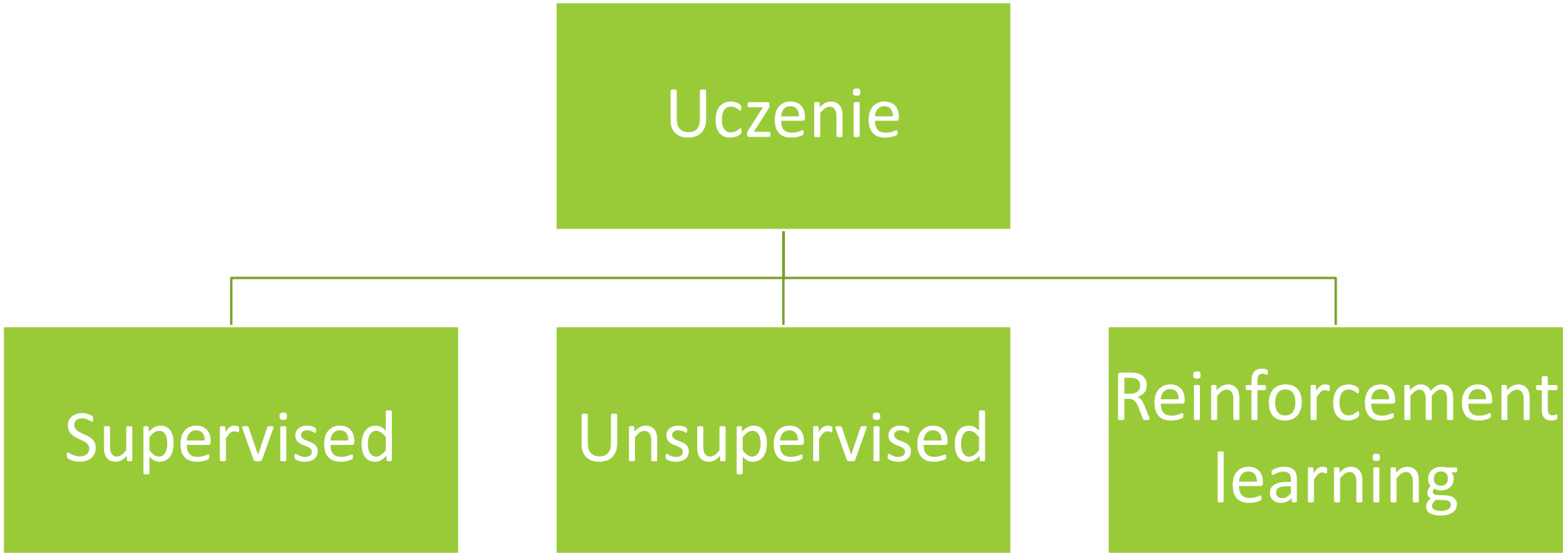
WYDZIAŁ ELEKTROTECHNIKI I AUTOMATYKI, POLITECHNIKA GDAŃSKA

akwasigroch.github.io

Plan prezentacji

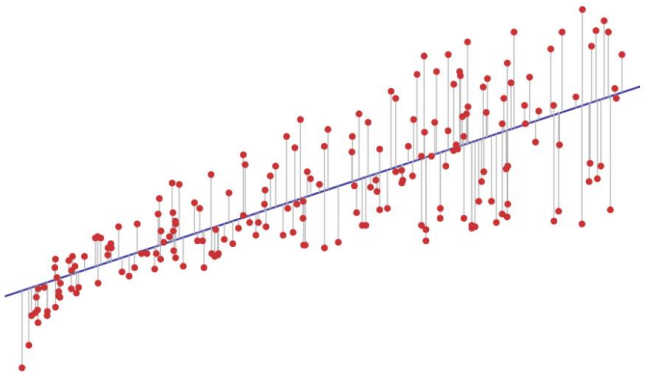
1. Typy uczenia oraz różnice pomiędzy nimi
   - Supervised
   - Unsupervised
   - Self-supervised
   - Semi-supervised
   - Weakly supervised
2. Ogólny przegląd i klasyfikacja metod typu self-supervised
3. Szczegółowy 2 metod

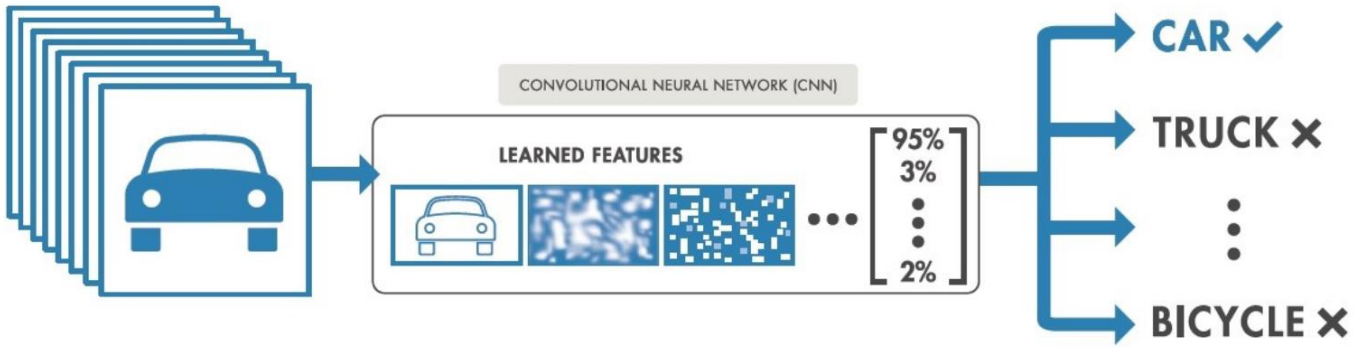
Podział algorytmów



Supervised learning

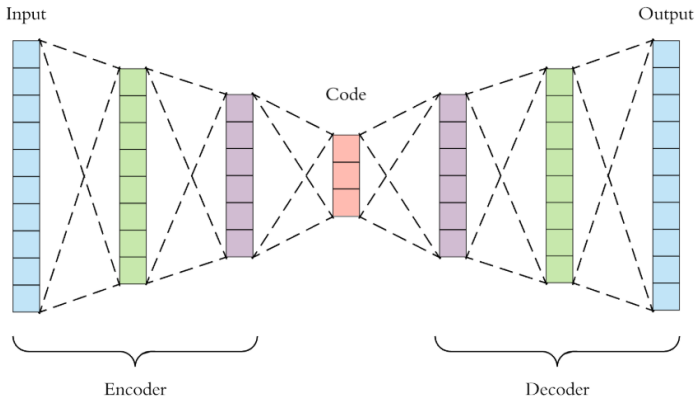
- Do każdego przykładu przypisana jest etykieta
- Przykłady: problemy klasyfikacji i regresji
- Etykiety przygotowane przez człowieka (*fine-grained, human-annotated*)
- Wadą jest duży koszt przygotowania etykiet (Amazon Mechanical Turk)

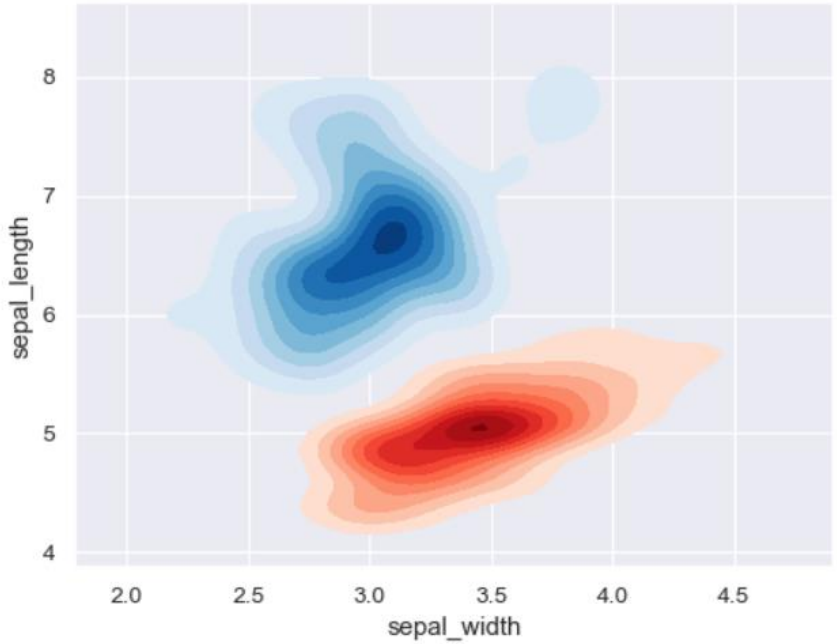


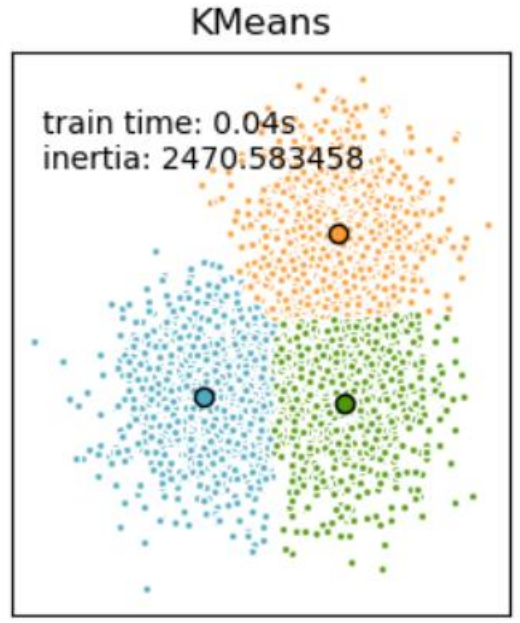


Unsupervised learning

- Algorytmy operują na danych bez etykiet
- Przykłady
  - Estymacja rozkładu
  - Klasteryzacja
  - Redukcja wymiarowości

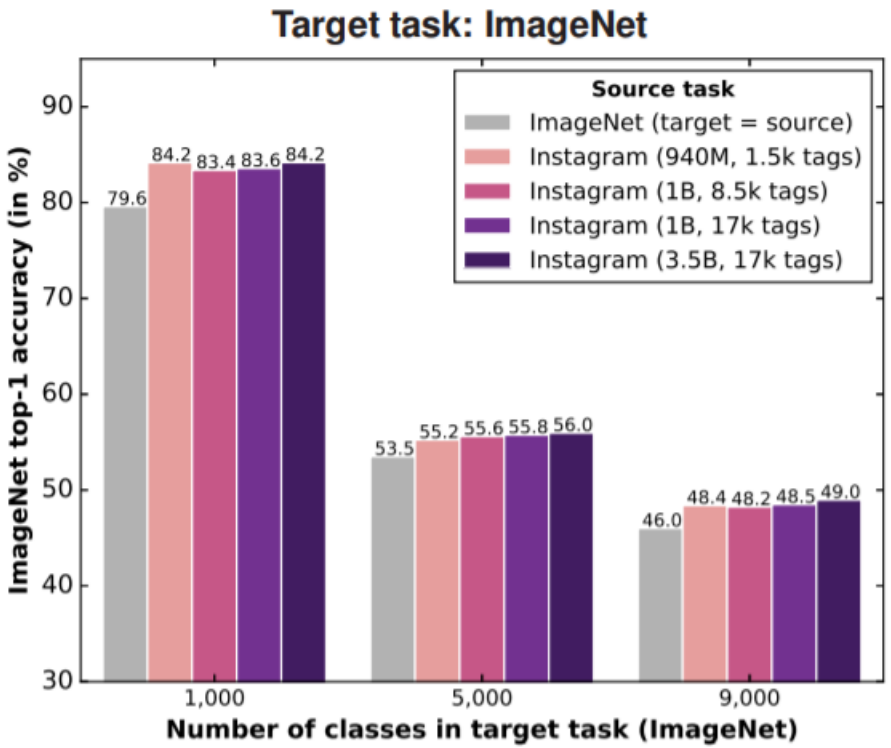




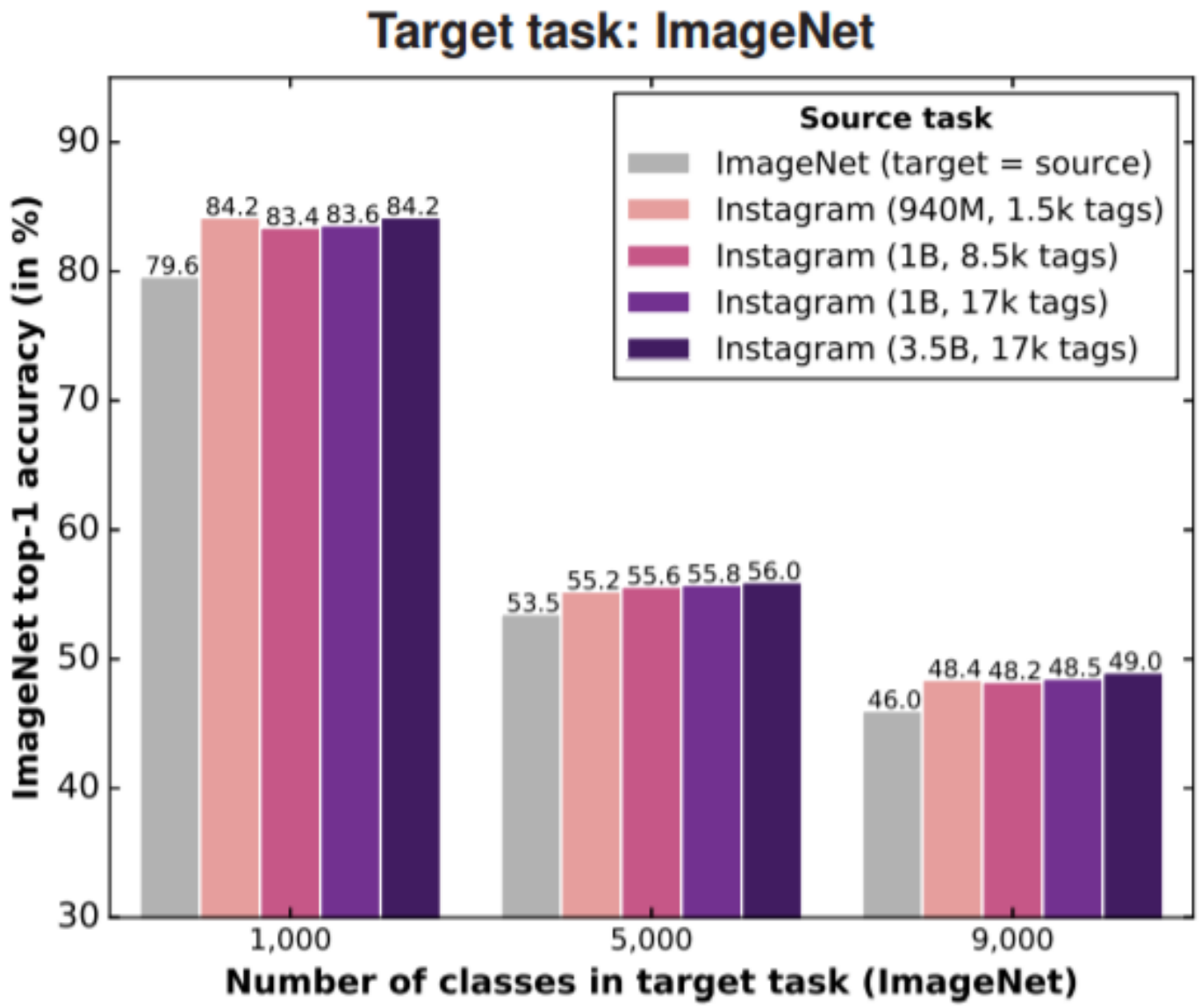


Weakly supervised

- Wykorzystanie mniej szczegółowych etykiety (coarse-grained vs fine-grained)
- Wykorzystanie etykiet niedokładnych
- Mniejszy koszt uzyskania etykiet

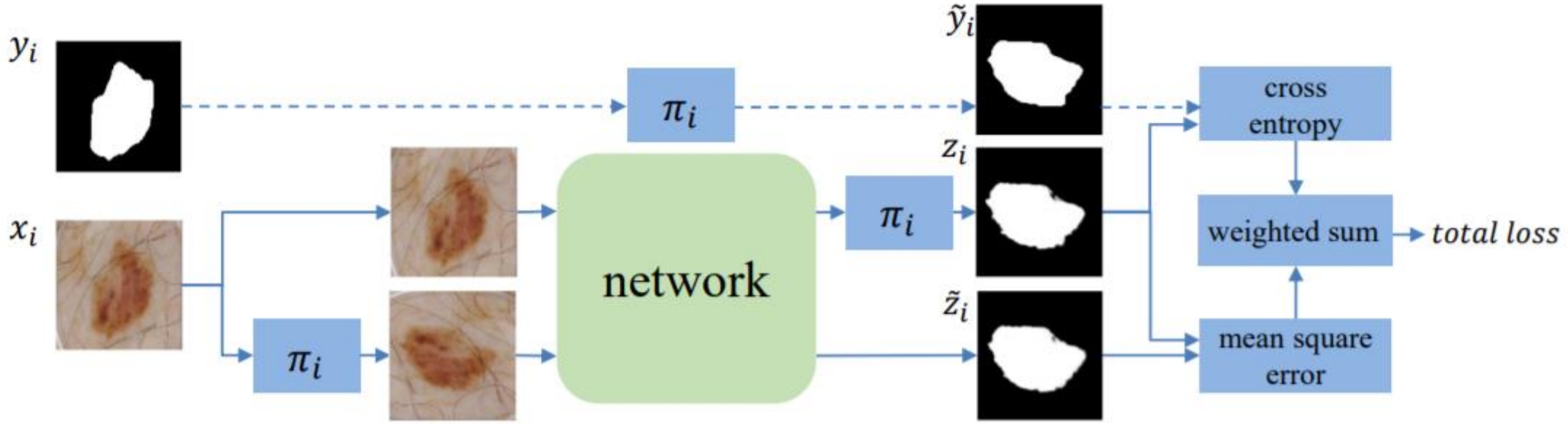


https://arxiv.org/pdf/1805.00932.pdf


Target task: ImageNet
Source task
ImageNet (target = source)
Instagram (940M, 1.5k tags)
Instagram (1B, 8.5k tags)
Instagram (1B, 17k tags)
Instagram (3.5B, 17k tags)
ImageNet top-1 accuracy (in %)
90
80
70
60
50
40
30
79.6
84.2
83.4
83.6
84.2
53.5
55.2
55.6
55.8
56.0
46.0
48.4
48.2
48.5
49.0
1,000
5,000
9,000
Number of classes in target task (ImageNet)

Semi-supervised learning

- Włączenie do uczenia przykładów oetykietowanych i nieoetykietowanych
- Funkcja celu składa się z dwóch ważonych członów: człon supervised i człon unsupervised



Model	50 labeled, 1950 unlabeled data		
	JA	DI	SE
Supervised-only	72.85	81.15	82.77
Supervised with regularization	73.25	81.60	83.30
Our Method	**75.31**	**83.79**	**86.37**
Our Method-A	74.59	83.27	82.77
Our Method-B	74.21	82.68	83.15

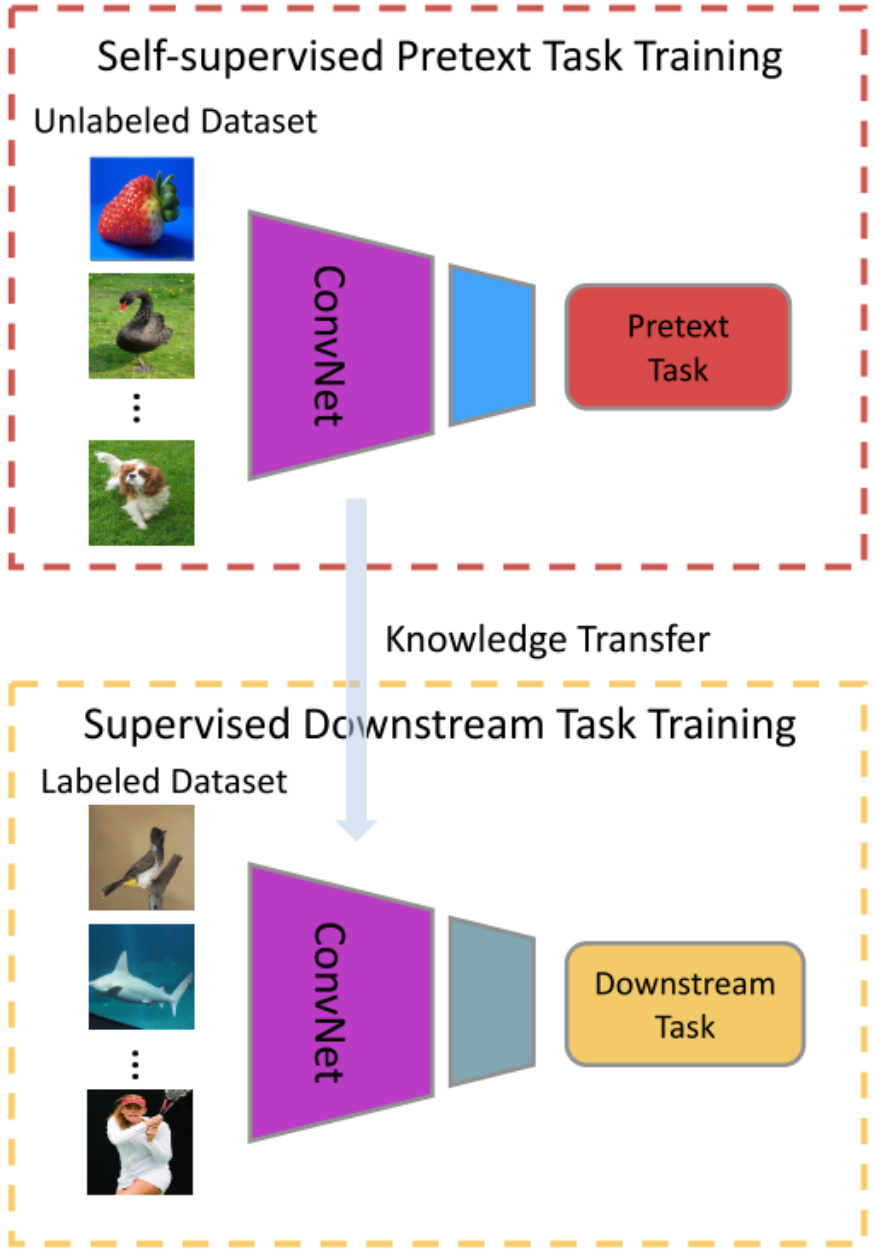
300 oetykietowanych, 1700 nieoetykietowanych
Baseline - supervised

Team	JA	DI	AC	SE	SP
Our Semi-supervised Method	**0.798**	**0.874**	**0.943**	**0.879**	0.953
Our Baseline	0.772	0.853	0.936	0.837	0.969
Yuan and Lo [28]	0.765	0.849	0.934	0.825	0.975
Berseth [1]	0.762	0.847	0.932	0.820	0.978
Bi et al. [3]	0.760	0.844	0.934	0.802	**0.985**
RECOD	0.754	0.839	0.931	0.817	0.970
Jer	0.752	0.837	0.930	0.813	0.976

http://bmvc2018.org/contents/papers/0162.pdf

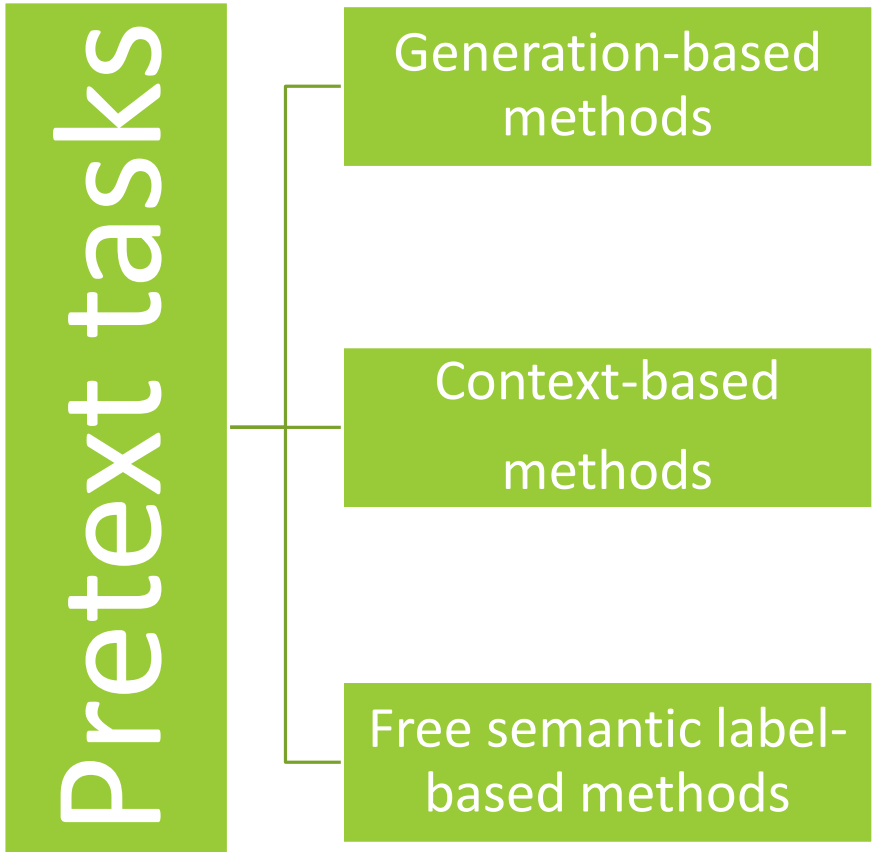
Self-supervised

- Poddziedzina unsupervised learning
- Etykiety generowane automatycznie
- Pierwszy etap – uczenie zadania pomocniczego (pretext task)
- Drugi etap – uczenie zadania docelowego (downstream task)
- Podejście podobne do transfer learning
- Zastosowanie w przetwarzaniu obrazów inspirowane rozwiązaniami NLP



Self-supervised learning – pretext tasks

- Zadanie pomocnicze (pretext task) jest zadaniem, które ma rozwiązać sieć neuronowa, w celu nauczenia odpowiedniej reprezentacji danych.
- Pseudo-etykiety generowane są automatycznie

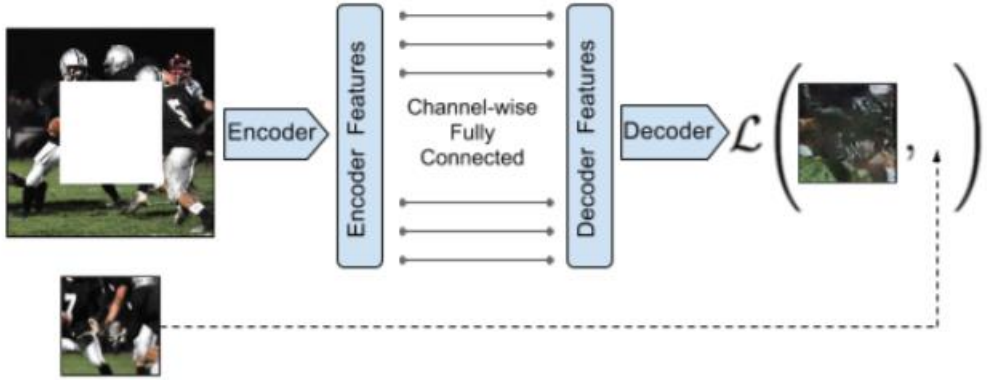


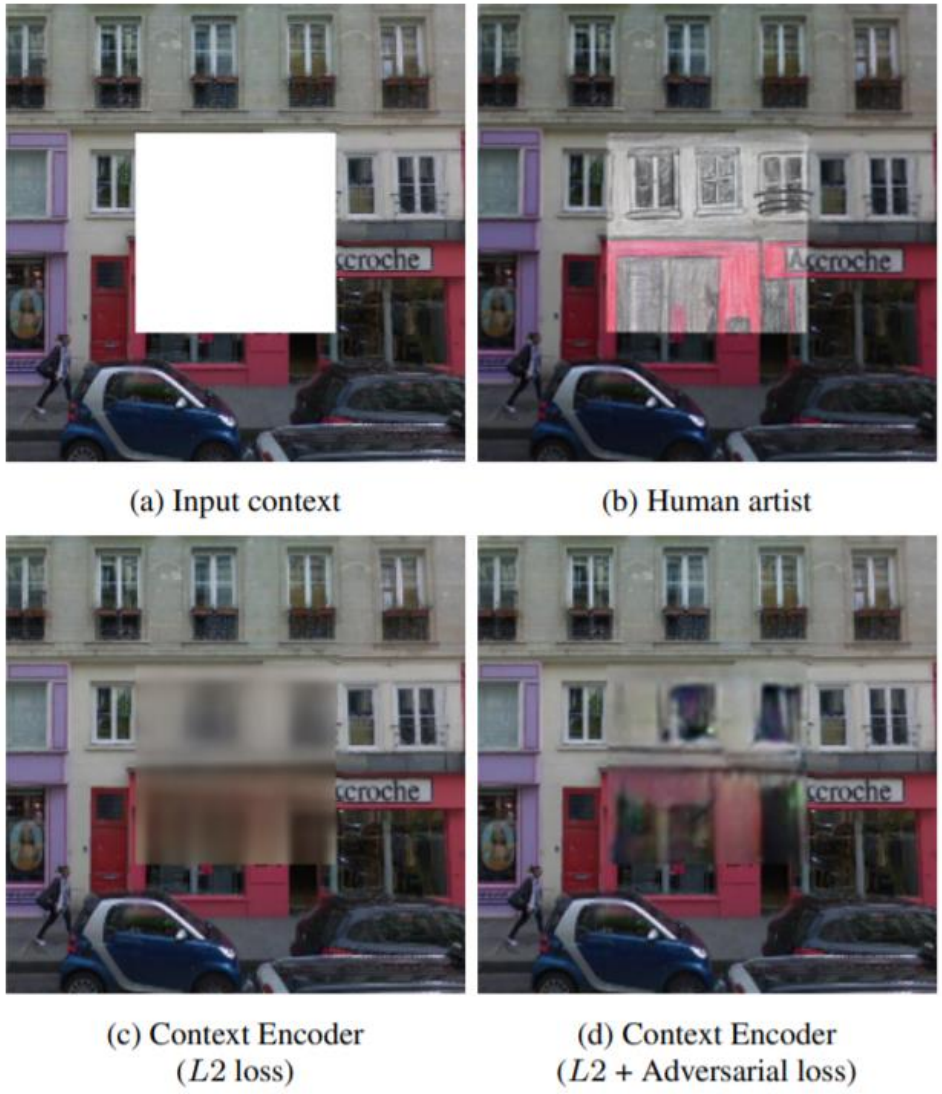
Self supervise learning – downstream task

- Zadaniem docelowym najczęściej jest klasyfikacja, detekcja lub segmentacja
- Ewaluacja przeprowadzana jest najczęściej z zamrożonymi wagami
- W przypadku klasyfikacji, wykorzystuje się klasyfikator liniowy

Generation based methods

Image inpainting





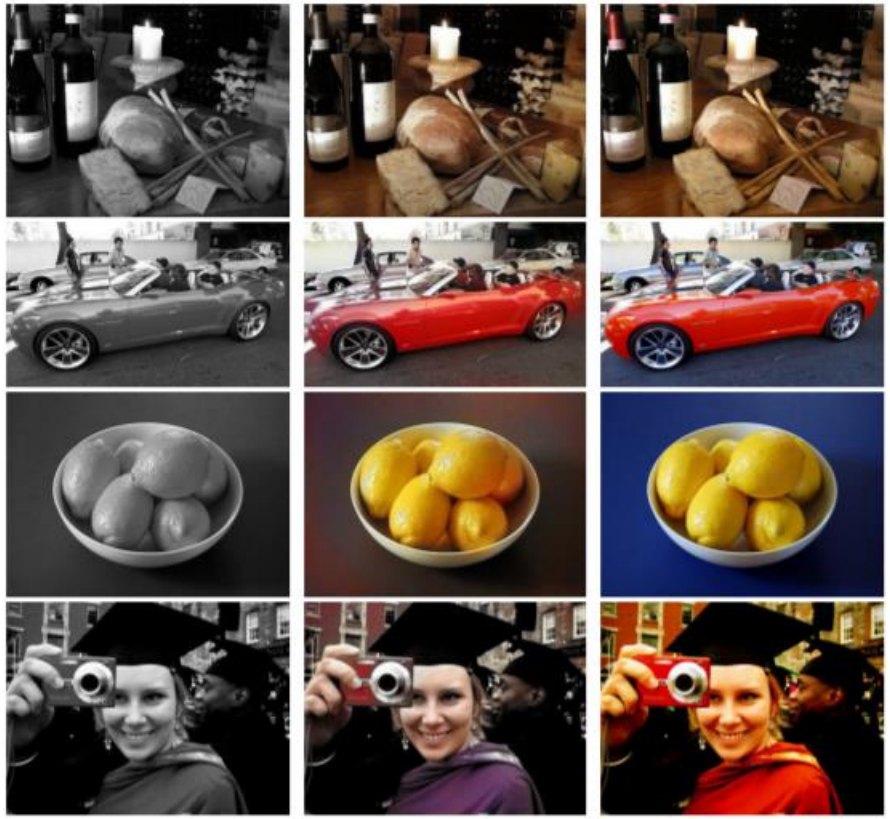
(a) Input context (b) Human artist

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

https://arxiv.org/pdf/1604.07379.pdf

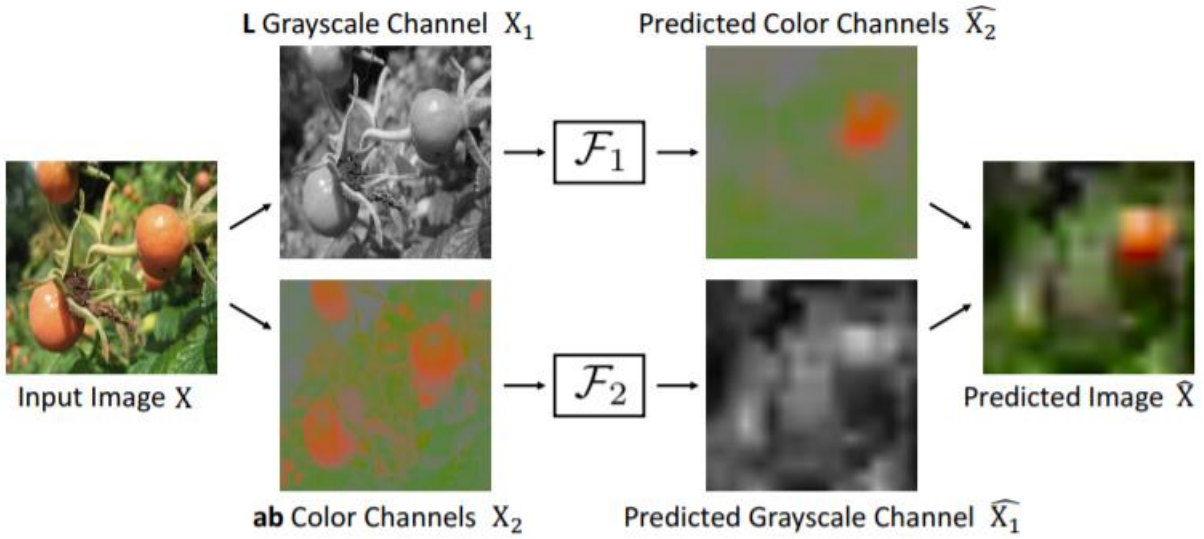
Generation based methods

Image colorization



https://arxiv.org/pdf/1603.06668.pdf

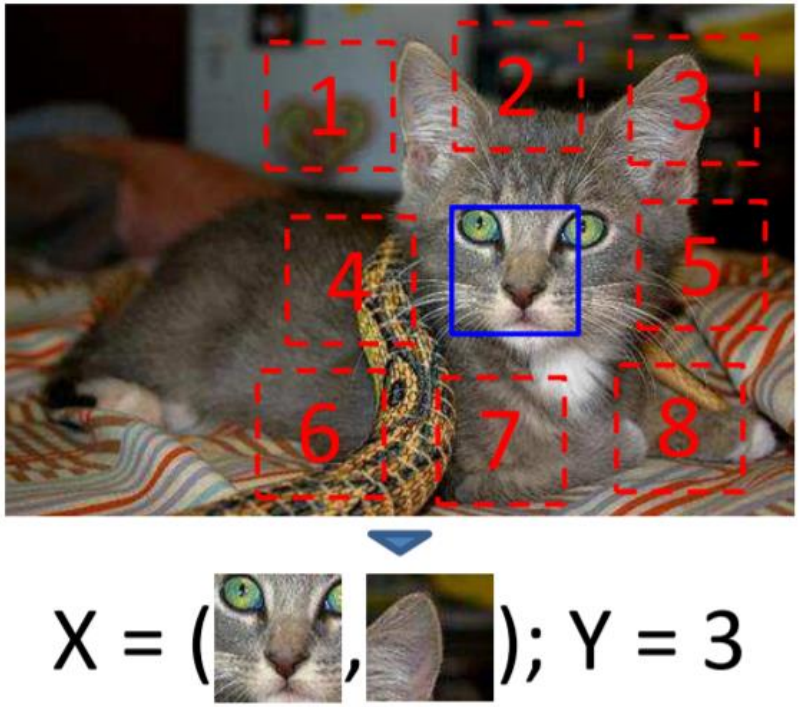
Split-Brain Autoencoders:



https://arxiv.org/pdf/1611.09842.pdf

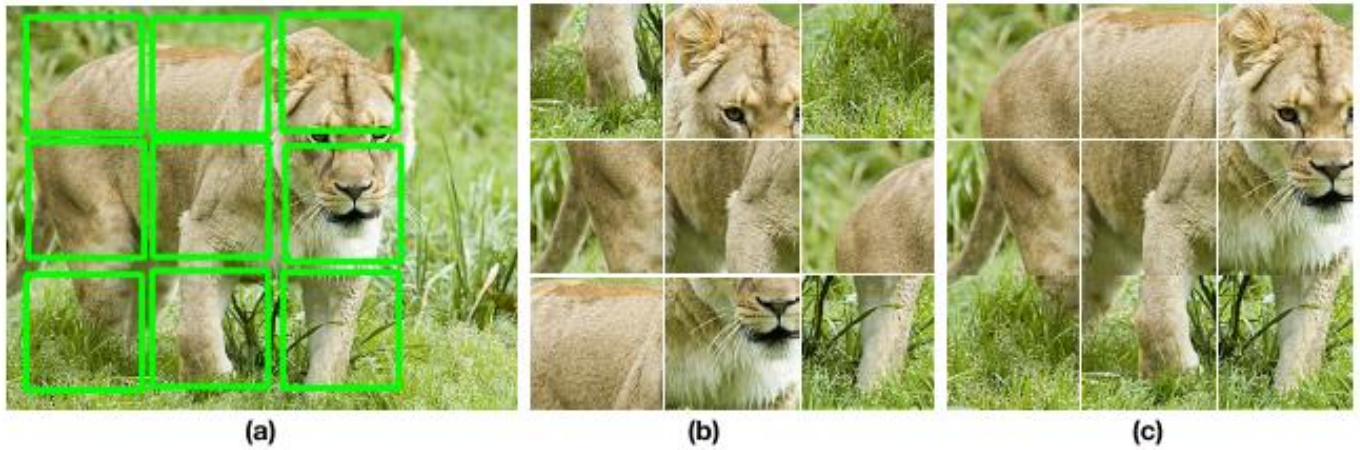
Context based methods

Context prediction



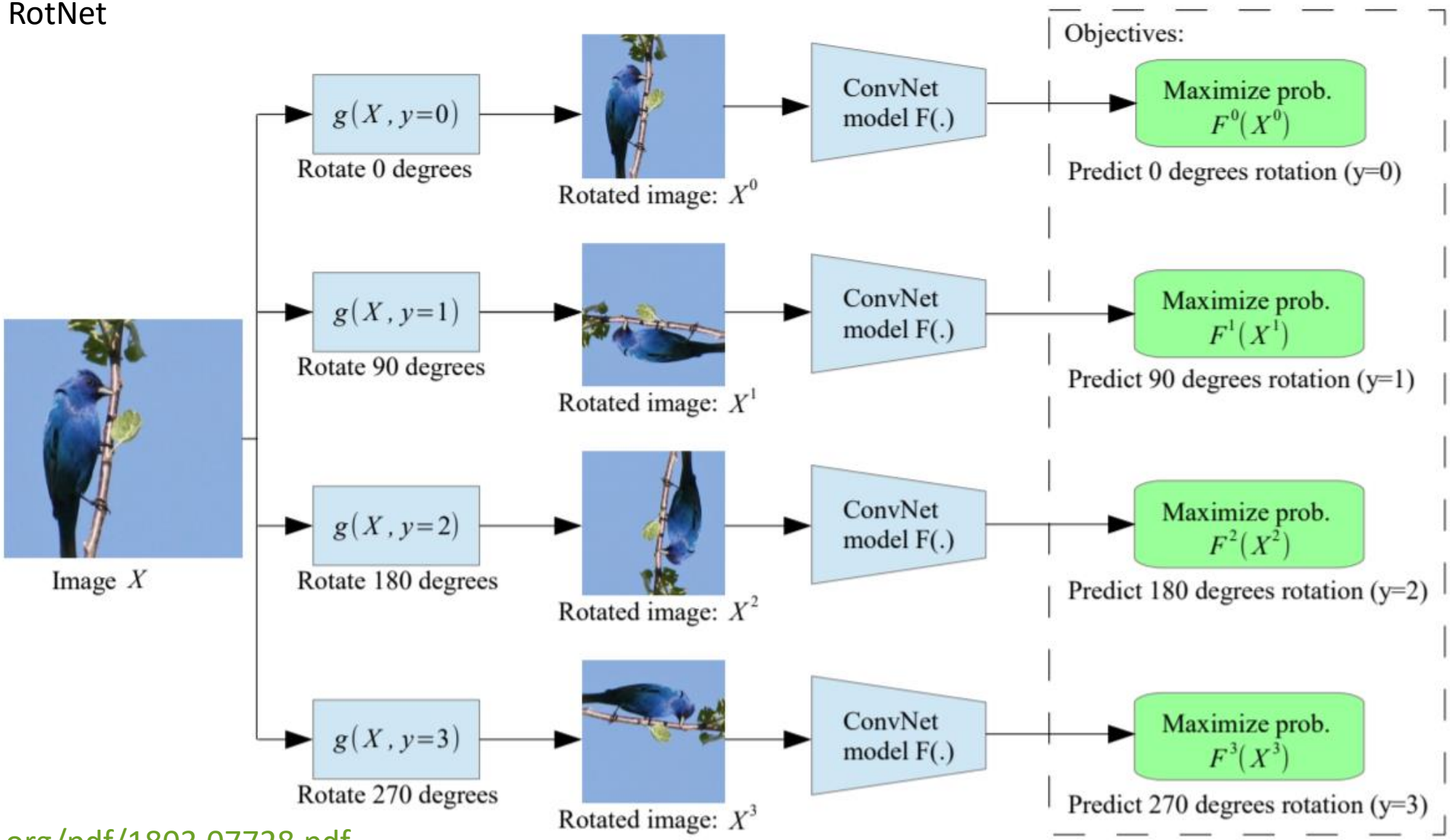
https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Doersch_Unsupervised_Visual_Representation_ICCV_2015_paper.pdf

Jigsaw puzzle solving



https://arxiv.org/pdf/1603.09246.pdf

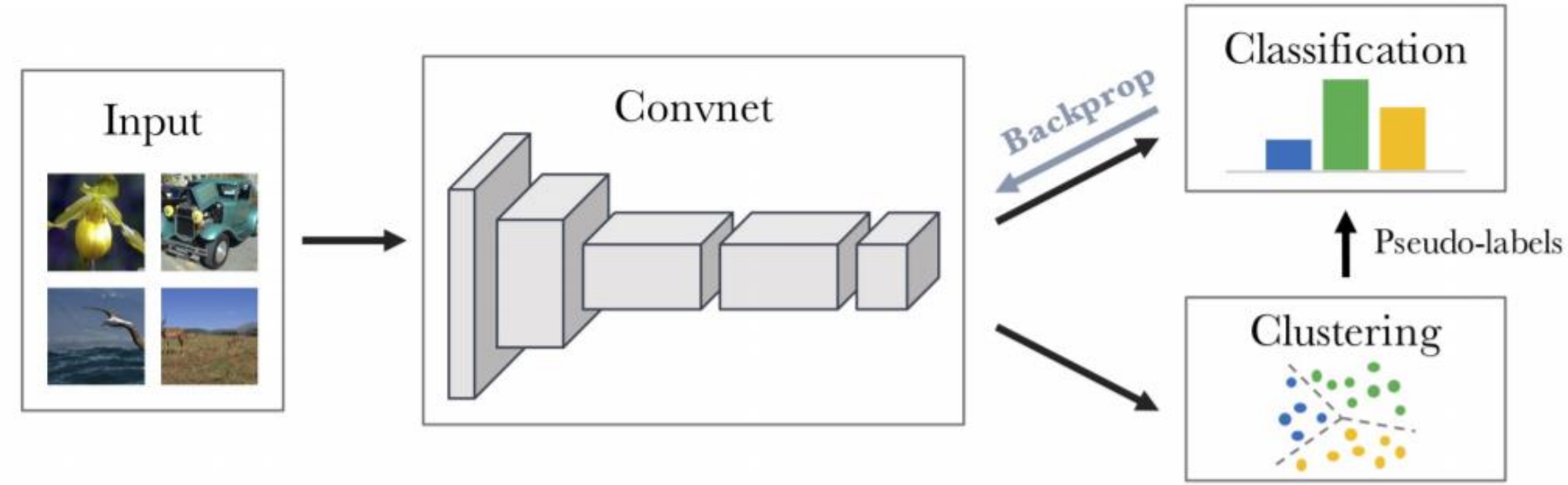
RotNet



https://arxiv.org/pdf/1803.07728.pdf

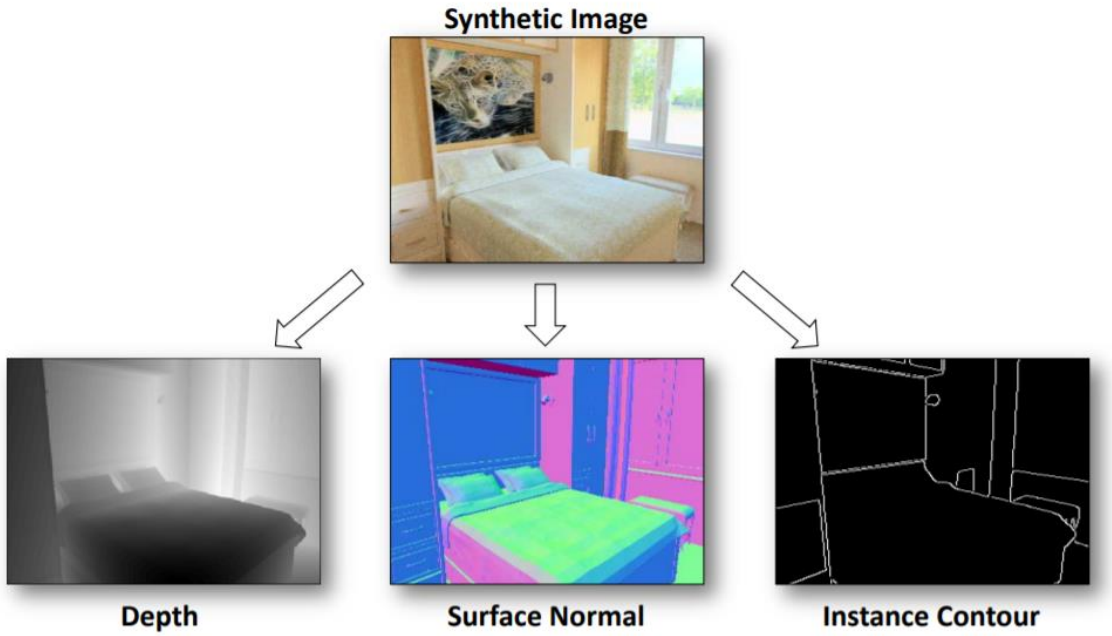
Context based methods

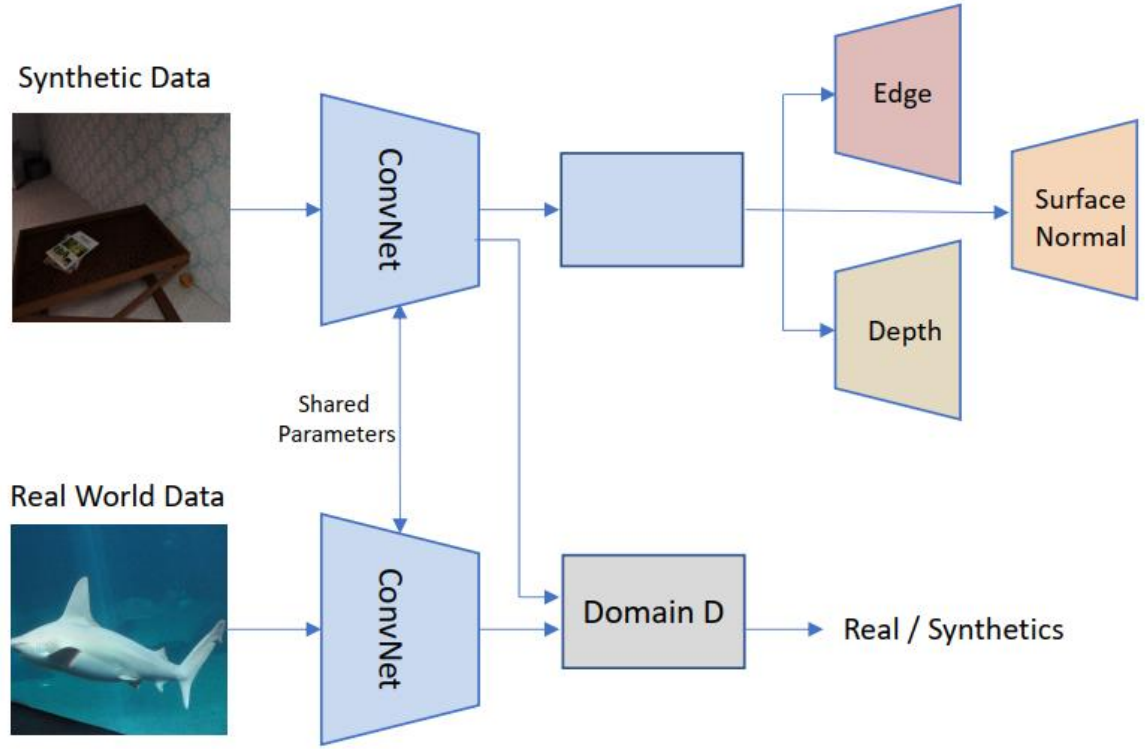
DeepCluster



https://arxiv.org/pdf/1807.05520.pdf

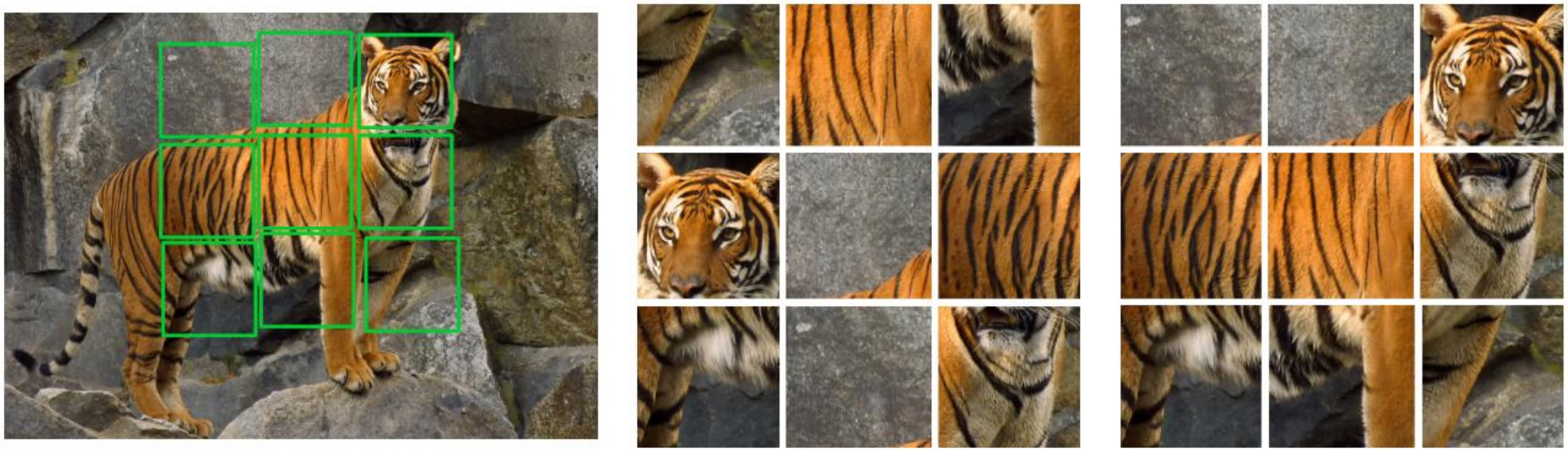
Free semantic label-based methods

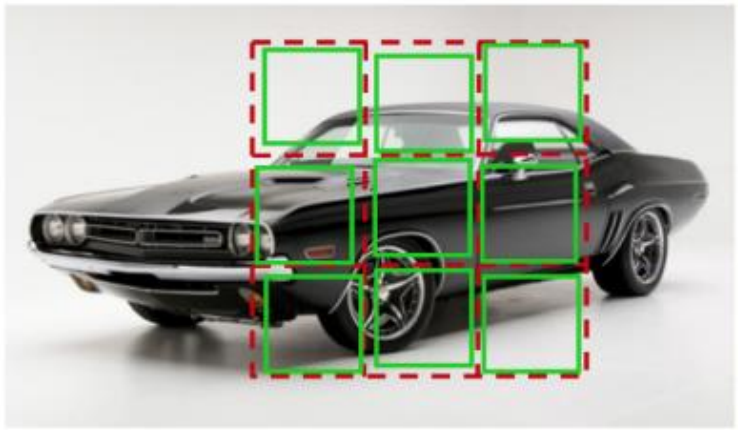




https://arxiv.org/pdf/1711.09082.pdf

Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles

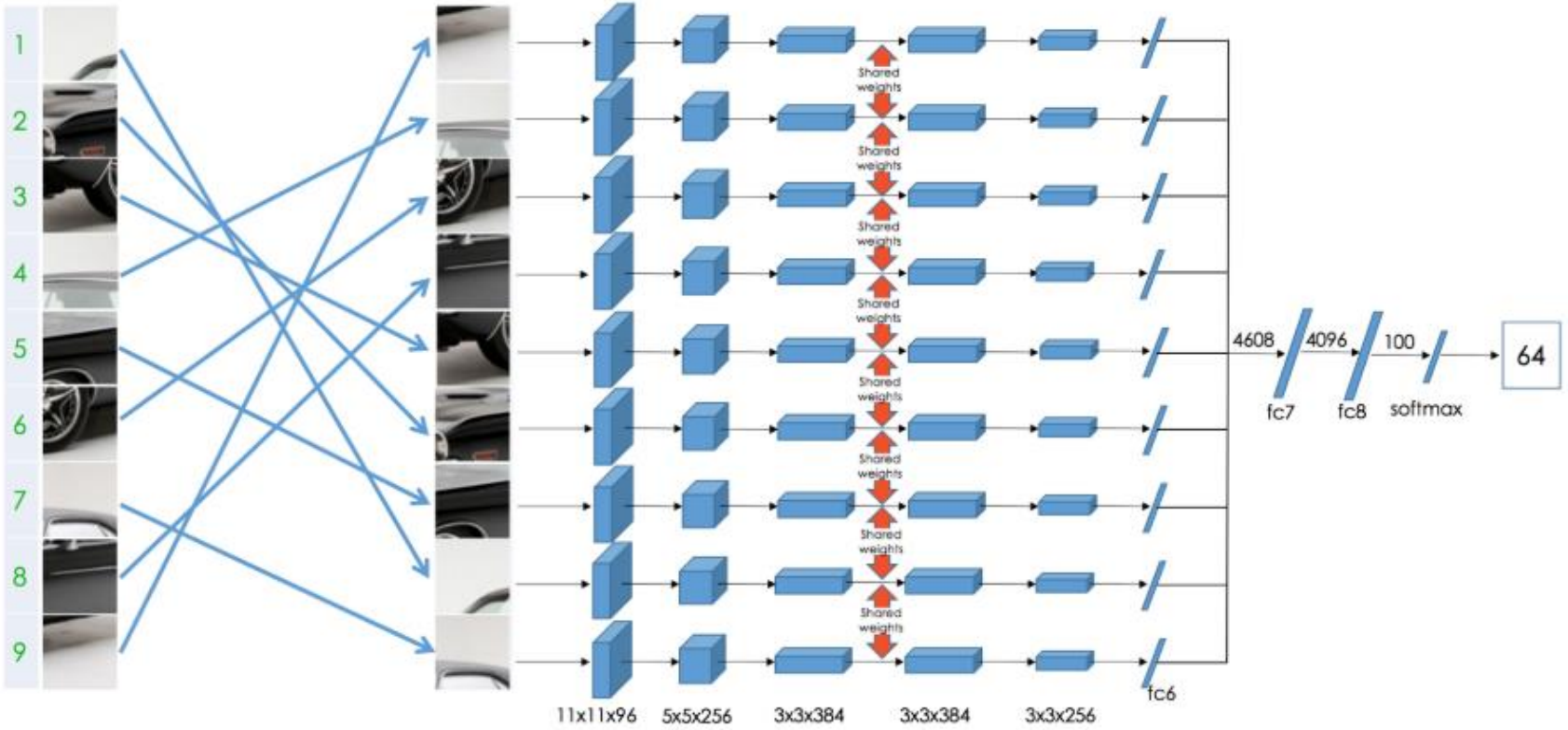




Permutation Set

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

Reorder patches according to the selected permutation



Wybór permutacji

- Liczba możliwych permutacji wynosi 9! = 362 880
- Należało ograniczyć liczbę permutacji i wybrać zbiór permutacji, które są do siebie „najbardziej odmienne”
- Miarą odmienności permutacji jest odległość Hamminga

Odległość Hamminga (ang. Hamming distance)– wprowadzona przez Richarda Hamminga miara odmienności dwóch ciągów o takiej samej długości, wyrażająca liczbę miejsc (pozycji), na których te dwa ciągi się różnią.

https://pl.wikipedia.org/wiki/Odleg%C5%82o%C5%9B%C4%87_Hamminga

Przykład:

4	2	3	1	8	6	7	5
4	3	2	1	8	7	6	5

Odległość Hamminga wynosi 4

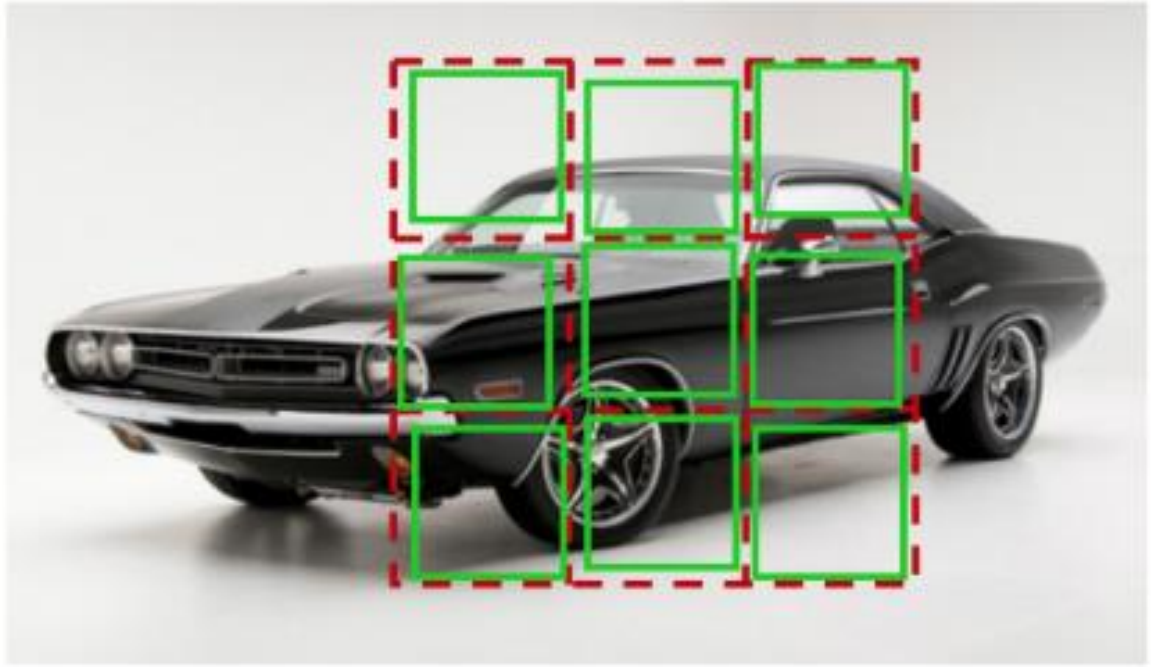
Algorithm 1. Generation of the *maximal* Hamming distance permutation set

Input: N \\ *number of permutations*
Output: P \\ *maximal permutation set*
$\bar{P} \leftarrow$ all permutations $[\bar{P}_1, \ldots, \bar{P}_{9!}]$ \\ $\bar{P}$ *is a* $9 \times 9!$ *matrix*
$P \leftarrow \emptyset$
$j \sim \mathcal{U}[1, 9!]$ \\ *uniform sample out of* 9! *permutations*
$i \leftarrow 1$
repeat
    $P \leftarrow [P\ \bar{P}_j]$ \\ *add permutation* $\bar{P}_j$ *to* P
    $\bar{P} \leftarrow [\bar{P}_1, \ldots, \bar{P}_{j-1}, \bar{P}_{j+1}, \ldots]$ \\ *remove* $\bar{P}_j$ *from* $\bar{P}$
    $D \leftarrow \text{Hamming}(P, P')$ \\ D *is an* $i \times (9! - i)$ *matrix*
    $\bar{D} \leftarrow \mathbf{1}^T D$ \\ $\bar{D}$ *is a* $1 \times (9! - i)$ *row vector*
    $j \leftarrow \arg\max_k \bar{D}_k$ \\ $\bar{D}_k$ *denotes the k-th entry of* $\bar{D}$
    $i \leftarrow i + 1$
until $i \leq N$

A good self-supervised task is neither simple nor ambiguous.

Utrudnienie zadania

- Maksymalizacja dystansu Hamminga pomiędzy permutacjami
- W celu uniknięcia uczenia na podstawie „ciągłości” pikseli elementy są od siebie oddalone (średnio o 11 pikseli)
- Każdy element normalizowany indywidualnie, wprowadzono color jittering



Wyniki

Table 1: Results on PASCAL VOC 2007 Detection and Classification. The results of the other methods are taken from Pathak *et al.* [30].

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%**

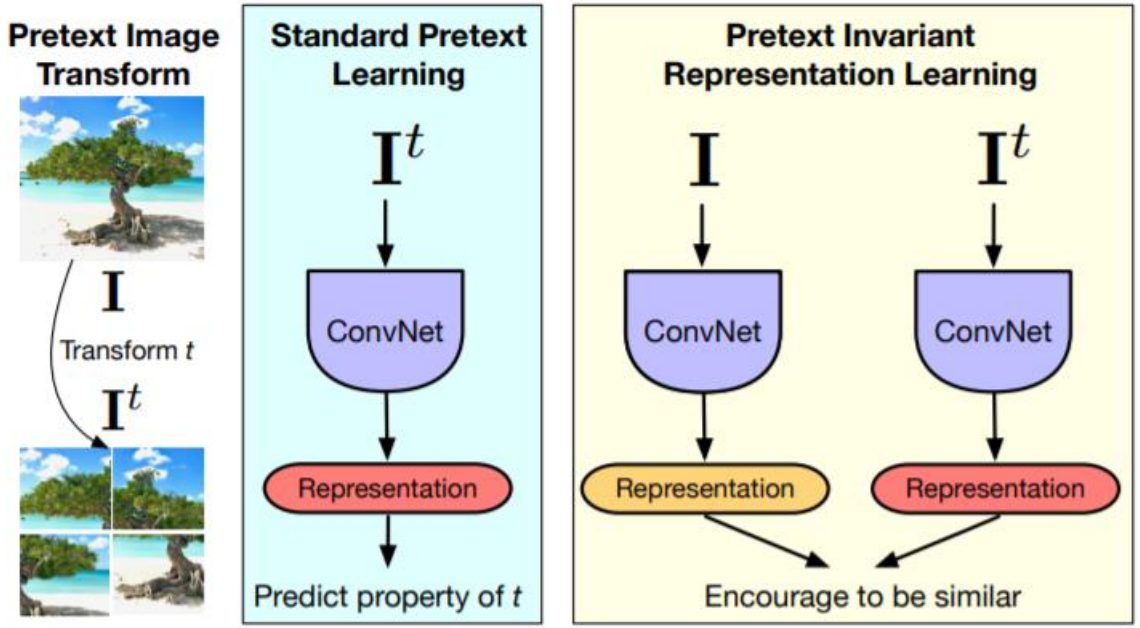
Table 4: Ablation study on the impact of the permutation set.

Number of permutations	Average hamming distance	Minimum hamming distance	Jigsaw task accuracy	Detection performance
1000	8.00	2	71	**53.2**
1000	6.35	2	62	51.3
1000	3.99	2	54	50.2
100	8.08	2	88	52.6
95	8.08	3	90	52.4
85	8.07	4	91	52.7
71	8.07	5	92	52.8
35	8.13	6	94	52.6
10	8.57	7	97	49.2
7	8.95	8	98	49.6
6	9	9	99	49.7

Table 5: Ablation study on the impact of the shortcuts.

Gap	Normalization	Color jittering	Jigsaw task accuracy	Detection performance
✗	✓	✓	98	47.7
✓	✗	✓	90	43.5
✓	✓	✗	89	51.1
✓	✓	✓	88	52.6

Self-Supervised Learning of Pretext-Invariant Representations



https://arxiv.org/pdf/1912.01991.pdf

Funkcja celu

Funkcja celu zbliża do siebie cechy obrazu oryginalnego i jego modyfikacji. Oddala cechy obraz modyfikowanego oraz cechy innych obrazów ze zbioru uczącego.

$$h(\mathbf{v_I}, \mathbf{v_{I^t}}) = \frac{\exp\left(\frac{s(\mathbf{v_I}, \mathbf{v_{I^t}})}{\tau}\right)}{\exp\left(\frac{s(\mathbf{v_I}, \mathbf{v_{I^t}})}{\tau}\right) + \sum_{\mathbf{I'} \in \mathcal{D}_N} \exp\left(\frac{s(\mathbf{v_{I^t}}, \mathbf{v_{I'}})}{\tau}\right)}. \tag{3}$$

$$\begin{aligned} L_{\mathrm{NCE}}\left(\mathbf{I}, \mathbf{I}^t\right) = & -\log\left[h\left(f(\mathbf{v_I}), g(\mathbf{v_{I^t}})\right)\right] \\ & - \sum_{\mathbf{I'} \in \mathcal{D}_N} \log\left[1 - h\left(g(\mathbf{v}_\mathbf{I}^t), f(\mathbf{v_{I'}})\right)\right] \end{aligned}$$

Bank cech

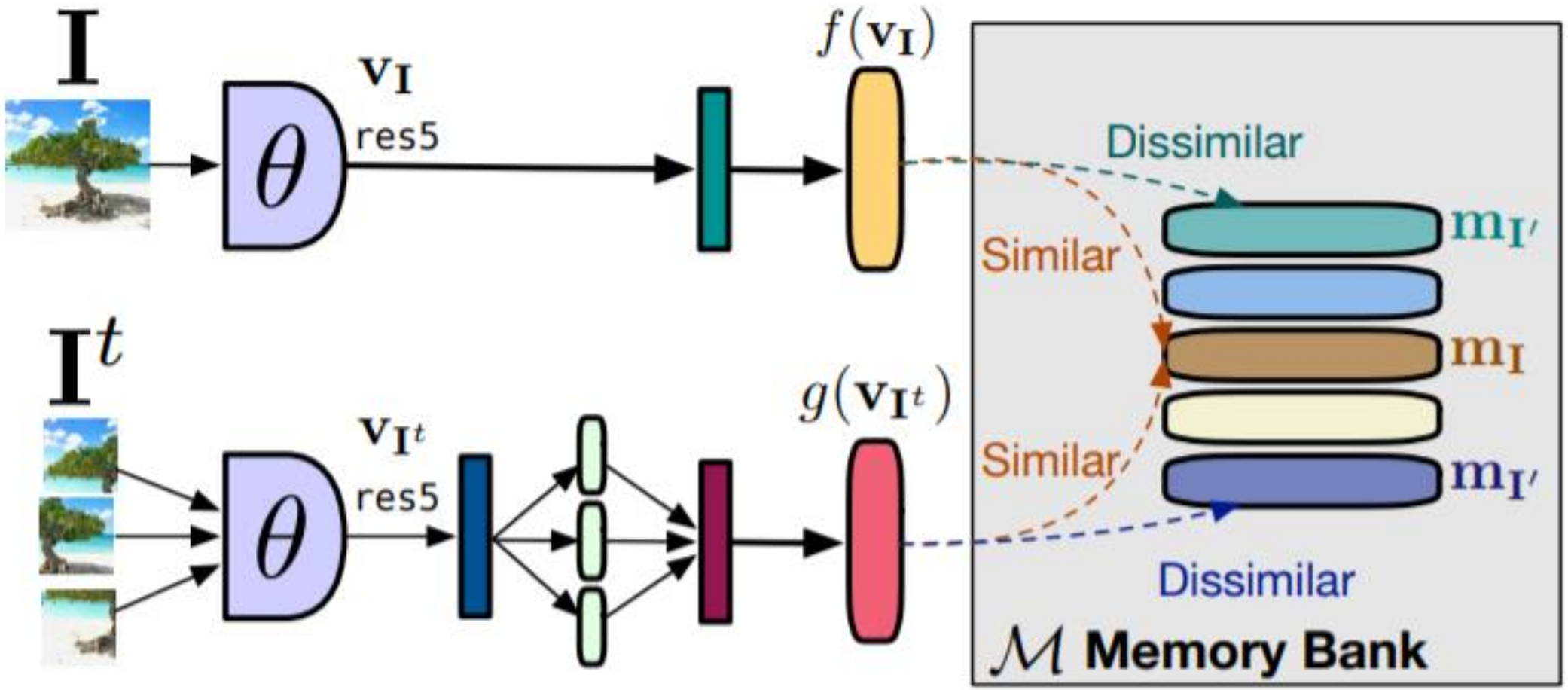
Bank cech przechowuje cechy każdego obrazu znajdującego się w zbiorze uczącym. Przechowywane reprezentacje są średnią ekspotencjalną cech oryginalnego obrazu, uzyskiwane co każdą epokę.

Nowa funkcja celu

Funkcja celu z poprzedniego slajdu nie porównywała reprezentacji obrazów oryginalnych z reprezentacją innych obrazów oryginalnych znajdujących się w bazie. W tym celu funkcję celu zmodyfikowano:

$$L\left(\mathbf{I}, \mathbf{I}^t\right) = \lambda L_{\mathrm{NCE}}(\mathbf{m}_{\mathbf{I}}, g(\mathbf{v}_{\mathbf{I}^t})) + (1-\lambda) L_{\mathrm{NCE}}(\mathbf{m}_{\mathbf{I}}, f(\mathbf{v}_{\mathbf{I}})).$$

- Pierwszy człon funkcją celu z poprzedniego slajdu. Zamiast aktualnie obliczanej reprezentacji, wykorzystuje się reprezentację przechowywaną w pamięci
- Drugi człon posiada dwie role:
  - porównywanie reprezentacji w pamięci z oryginalną reprezentacją zapobiegają gwałtownym zmianom parametrów
  - Oddala reprezentację oryginalnego obrazu od reprezentacji innych oryginalnych obrazów znajdujących się w bazie


I
θ
$\mathbf{v_I}$
res5
$f(\mathbf{v_I})$
Dissimilar
$\mathbf{m_{I'}}$
Similar
$\mathbf{m_I}$
$\mathbf{I}^t$
$\mathbf{v_{I^t}}$
res5
$g(\mathbf{v_{I^t}})$
Similar
$\mathbf{m_{I'}}$
Dissimilar
$\mathcal{M}$ Memory Bank

Szczegóły techniczne

- Sieć ResNet50 jako ekstraktor cech
- Sieć ResNet50 zwraca wektor 2048 elementowy, stosowana jest dodatkowa warstwa w pełni połączona, w celu zredukowania tego wektora do rozmiaru 128 elementów
- Zadaniem pomocniczym (pretext task) jest „układanie puzzli”, każdy element generuje wektor 128 elementowy, następnie wektory są łączone ze sobą, tworząc wektor 9* 128 elementów. Taki wektor podawany jest na warstwę w pełni połączoną w celu uzyskania wektora 128 elementowego.
- Współczynnik funkcji celu 0.5

Wyniki - detekcja

Detekcja obrazu na bazie VOC07+12, pretraining na bazie Imagenet

Method	Network	AP^{all}	AP^{50}	AP^{75}	ΔAP^{75}
Supervised	R-50	52.6	**81.1**	57.4	=0.0
Jigsaw [19]	R-50	48.9	75.1	52.9	-4.5
Rotation [19]	R-50	46.3	72.5	49.3	-8.1
NPID++ [72]	R-50	52.3	79.1	56.9	-0.5
PIRL (**ours**)	R-50	**54.0**	80.7	**59.7**	**+2.3**

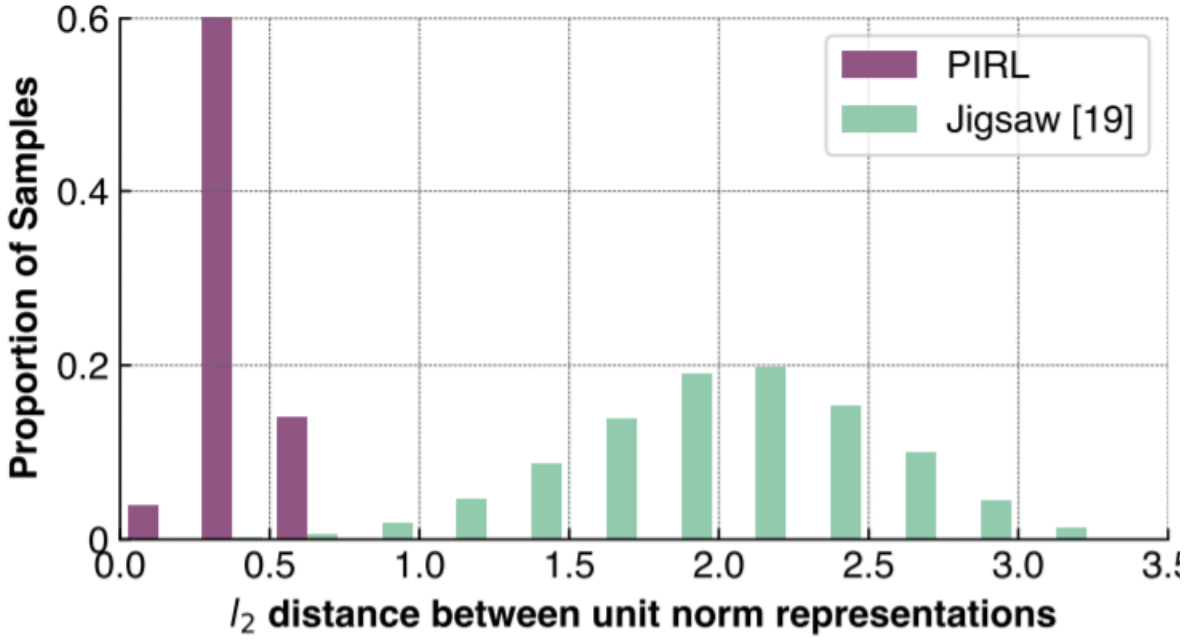
Wyniki – klasyfikacja liniowa

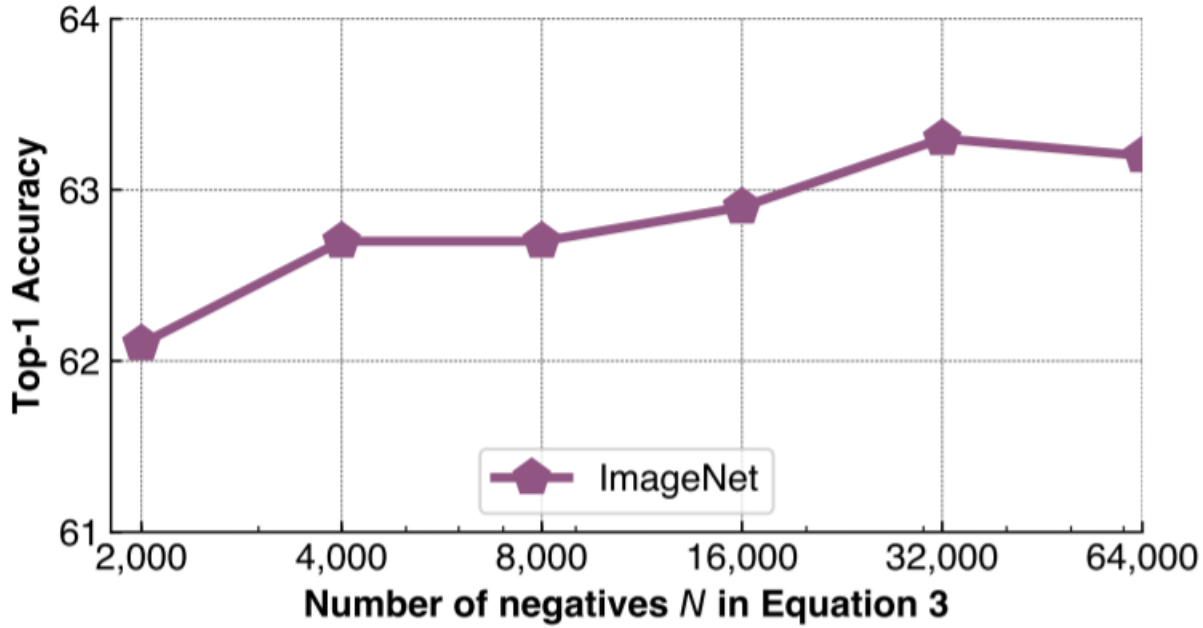
Method	Parameters	Transfer Dataset			
		ImageNet	VOC07	Places205	iNat.
ResNet-50 using evaluation setup of [19]					
Supervised	25.6M	75.9	87.5	51.5	45.4
Colorization [19]	25.6M	39.6	55.6	37.5	–
Rotation [18]	25.6M	48.9	63.9	41.4	23.0
NPID++ [72]	25.6M	59.0	76.6	46.4	32.4
MoCo [24]	25.6M	60.6	–	–	–
Jigsaw [19]	25.6M	45.7	64.5	41.2	21.3
PIRL (**ours**)	25.6M	**63.6**	**81.1**	**49.8**	**34.1**

Wyniki – klasyfikacja przy małych zbiorach uczących

Method	Data fraction → Backbone	1%	10%
		Top-5 Accuracy	
Random initialization [72]	R-50	22.0	59.0
NPID [72]	R-50	39.2	77.4
Jigsaw [19]	R-50	45.3	79.3
NPID++ [72]	R-50	52.6	81.5
VAT + Ent Min. [20, 45]	R-50v2	47.0	83.4
S^4L Exemplar [75]	R-50v2	47.0	83.7
S^4L Rotation [75]	R-50v2	53.4	**83.8**
PIRL (**ours**)	R-50	**57.2**	**83.8**

Wyniki


PIRL
Jigsaw [19]
Proportion of Samples
0.6
0.4
0.2
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
l_2 distance between unit norm representations


ImageNet
Top-1 Accuracy
64
63
62
61
2,000
4,000
8,000
16,000
32,000
64,000
Number of negatives N in Equation 3

Wyniki – inne zadania pomocnicze

Method	Params	Transfer Dataset			
		ImageNet	VOC07	Places205	iNat.
Rotation [18]	25.6M	48.9	63.9	41.4	23.0
PIRL (Rotation; **ours**)	25.6M	60.2	77.1	47.6	31.2
Δ of PIRL	-	**+11.3**	**+13.2**	**+6.2**	**+8.2**
Combining pretext tasks using PIRL					
PIRL (Jigsaw; **ours**)	25.6M	62.2	79.8	48.5	31.2
PIRL (Rotation + Jigsaw; **ours**)	25.6M	**63.1**	**80.3**	**49.7**	**33.6**

Prezentacja przygotowana na podstawie przeglądu:

https://arxiv.org/pdf/1902.06162.pdf