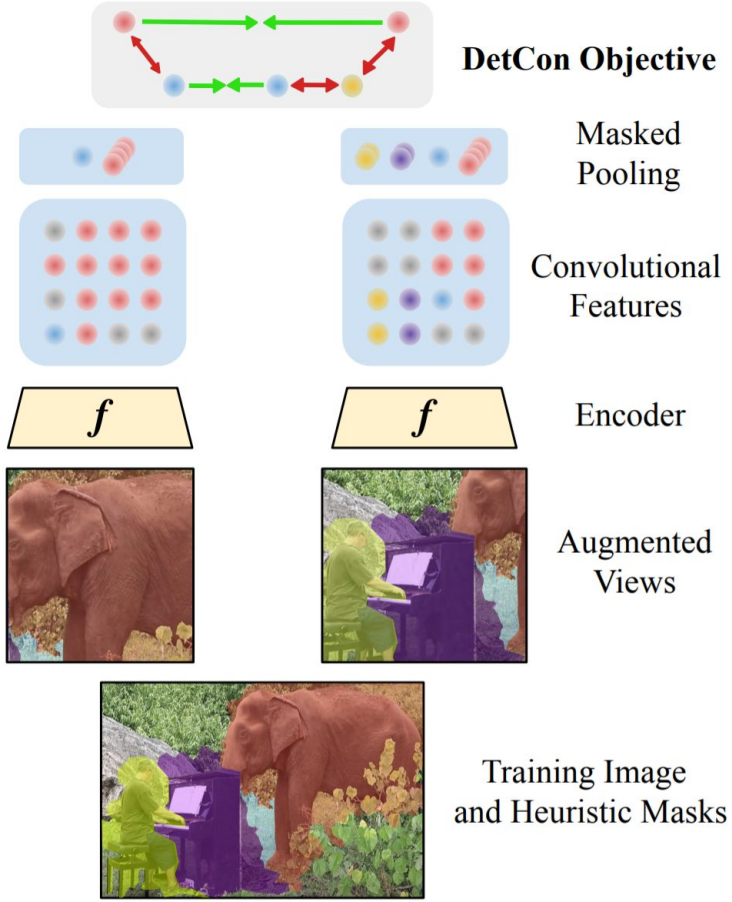
Efficient Visual Pretraining with Contrastive Detection



Visual self-supervised pretraining

Задача: предобучение сети на датасете без разметки

dataset with labels

unlabeled dataset

Self-supervised model

Supervised fine tuning

Trained model

Известные методы

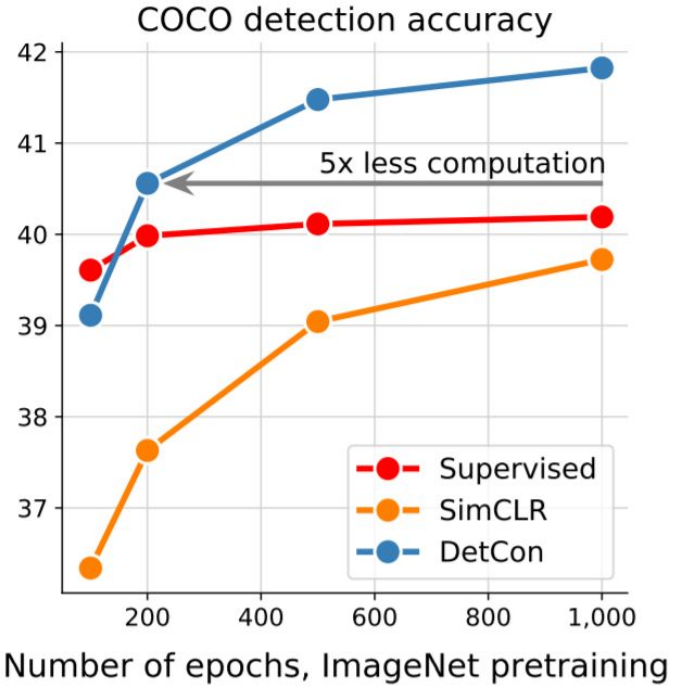
- BYOL
- SimCLR
- SwAV
- …

Чем этот метод лучше?

Известные методы

- BYOL
- SimCLR
- SwAV
- …

Чем этот метод лучше?



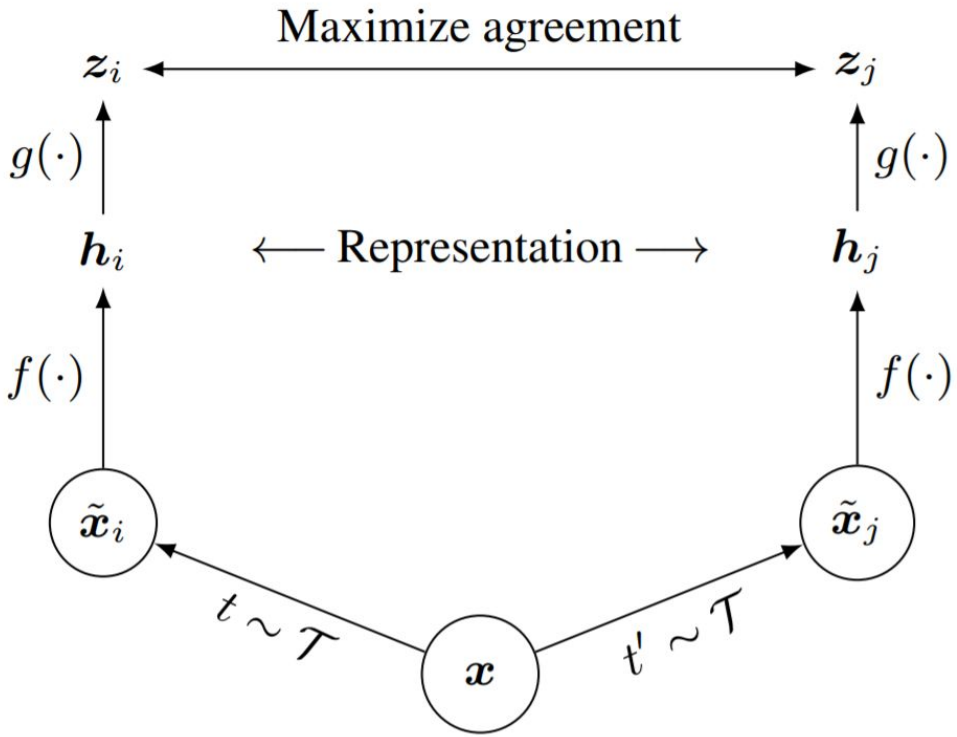
Предобучение значительно эффективнее

Описание метода

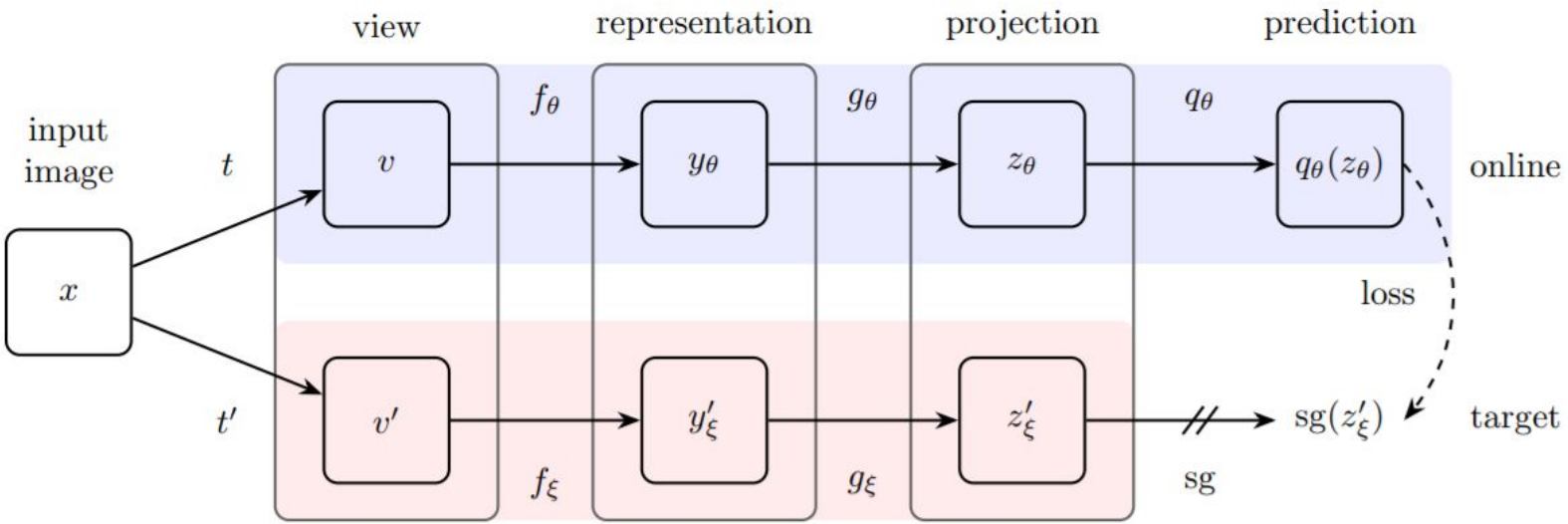
Две версии метода:

- основанная на SimCLR – DetCon_S
- основанная на BYOL – DetCon_B

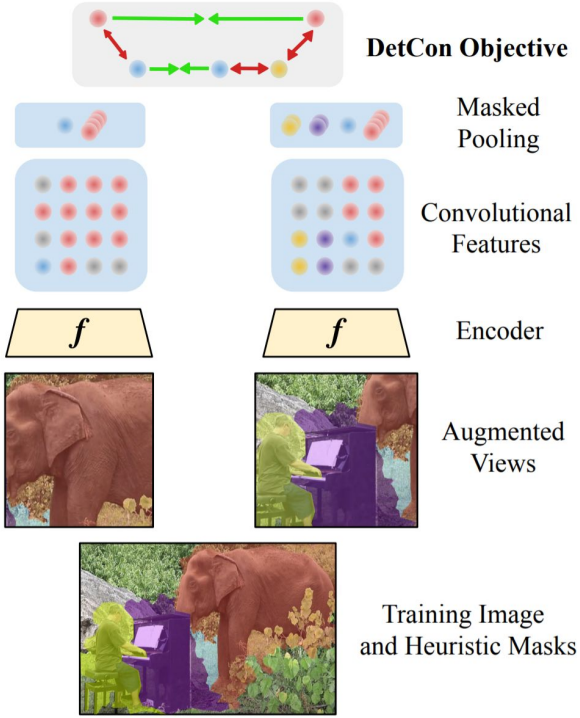
Напоминание. SimCLR



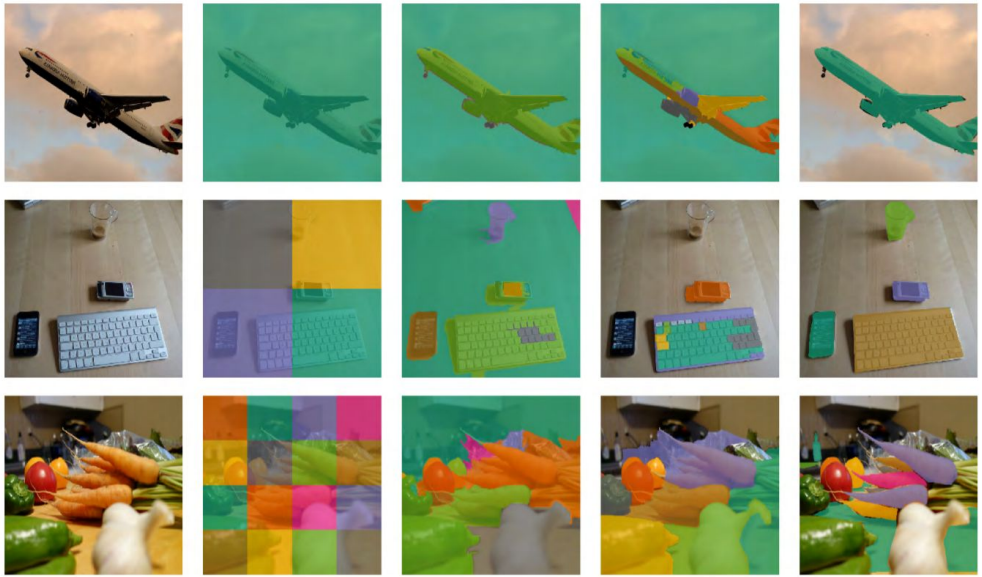
Напоминание. BYOL

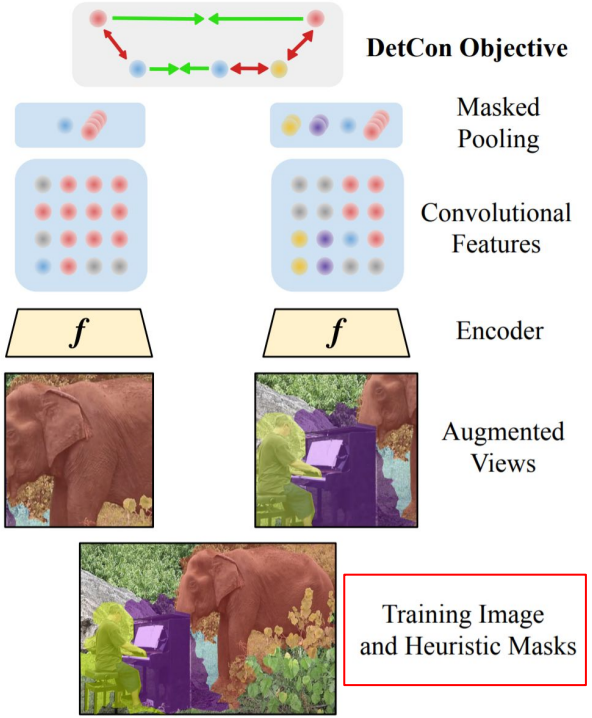


Архитектура DetCon_S и DetCon_B



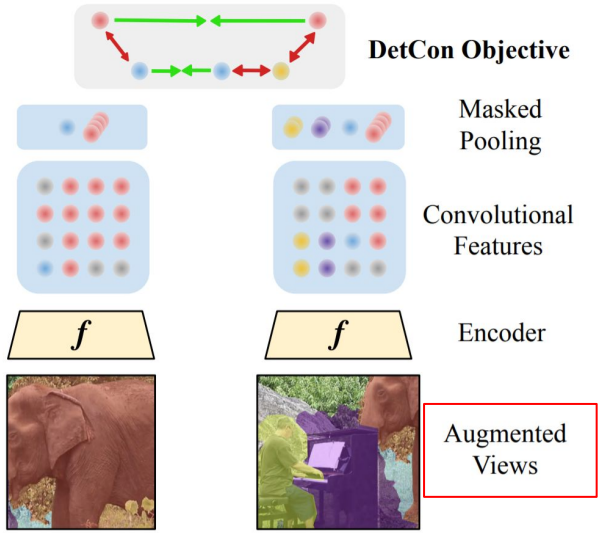
Архитектура DetCon_S и DetCon_B





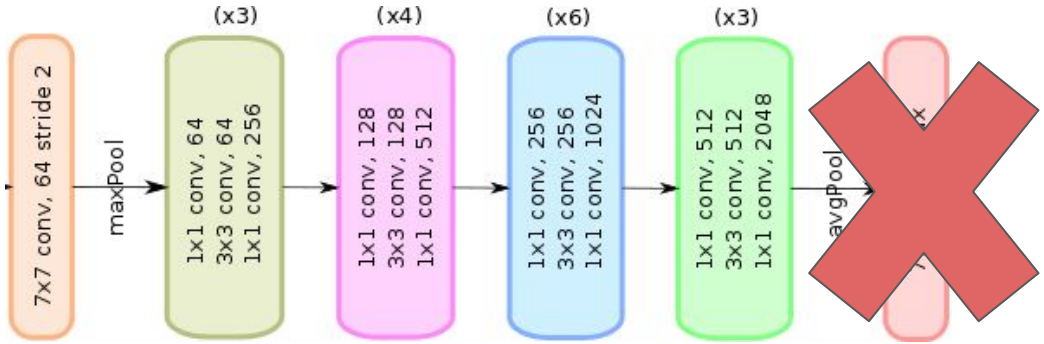
Архитектура DetCon_S и DetCon_B

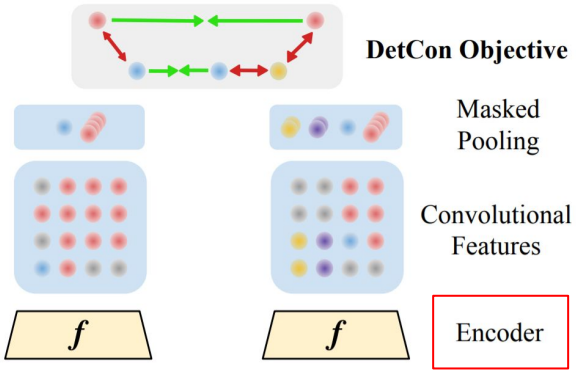
- Random cropping
- Horizontal flipping
- Color jittering
- Color dropping (to grey-scale)
- Blurring
- Solarization ($x \cdot \mathbb{1}_{x<0.5} + (1-x) \cdot \mathbb{1}_{x \geq 0.5}$)



Архитектура $DetCon_S$ и $DetCon_B$

ResNet-50 без последнего average pool слоя



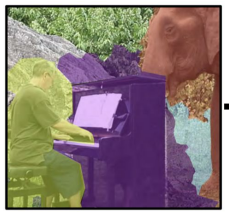


Архитектура DetCon_S и DetCon_B

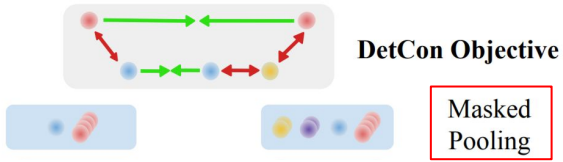
$$\boldsymbol{h_m} = \frac{1}{\sum_{i,j} m_{i,j}} \sum_{i,j} m_{i,j}\, \boldsymbol{h}[i,j]$$

$\boldsymbol{h}[i,j]$ $\quad$ $m_{i,j}$

Convolutional Features



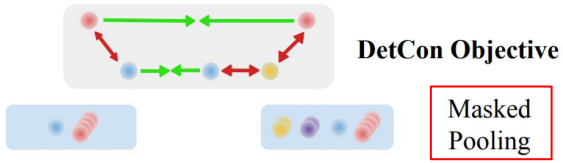
Average pooling



Архитектура DetCon_S и DetCon_B

DetCon_S $\quad \boldsymbol{v_m} = g_\theta(\boldsymbol{h_m}), \quad \boldsymbol{v'_{m'}} = g_\theta(\boldsymbol{h'_{m'}})$

DetCon_B $\quad \boldsymbol{v_m} = q_\theta \circ g_\theta(\boldsymbol{h_m}), \quad \boldsymbol{v'_{m'}} = g_\xi(\boldsymbol{h'_{m'}})$



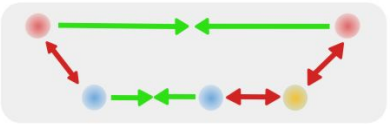
Архитектура DetCon_S и DetCon_B

$$\ell_{m,m'} = -\log \frac{\exp(\boldsymbol{v}_m \cdot \boldsymbol{v}'_{m'})}{\exp(\boldsymbol{v}_m \cdot \boldsymbol{v}'_{m'}) + \sum_n \exp(\boldsymbol{v}_m \cdot \boldsymbol{v}_n)}$$

$$\mathcal{L} = \sum_m \sum_{m'} \mathbb{1}_{m,m'} \ell_{m,m'}$$

$\{\boldsymbol{v}_n\}$ – negative samples

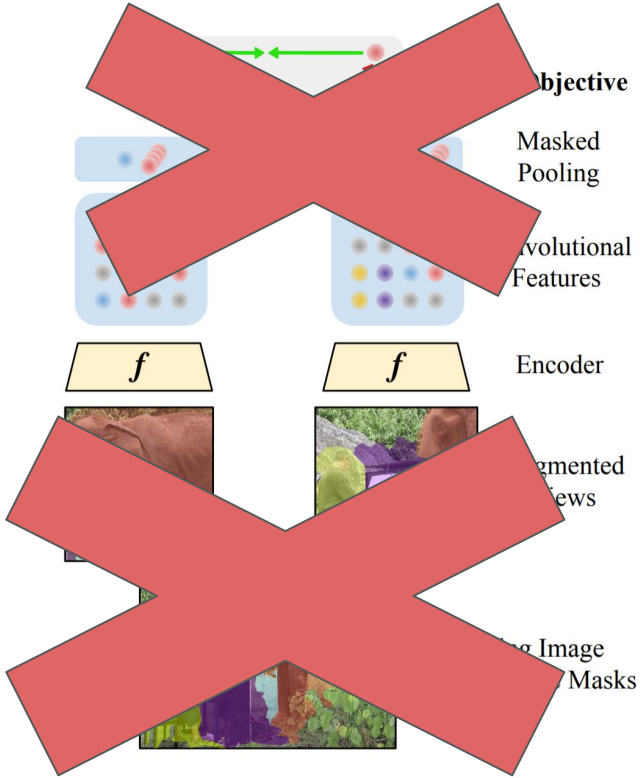
$\mathbb{1}_{m,m'}$ – индикатор того, что маски соответствуют одному региону



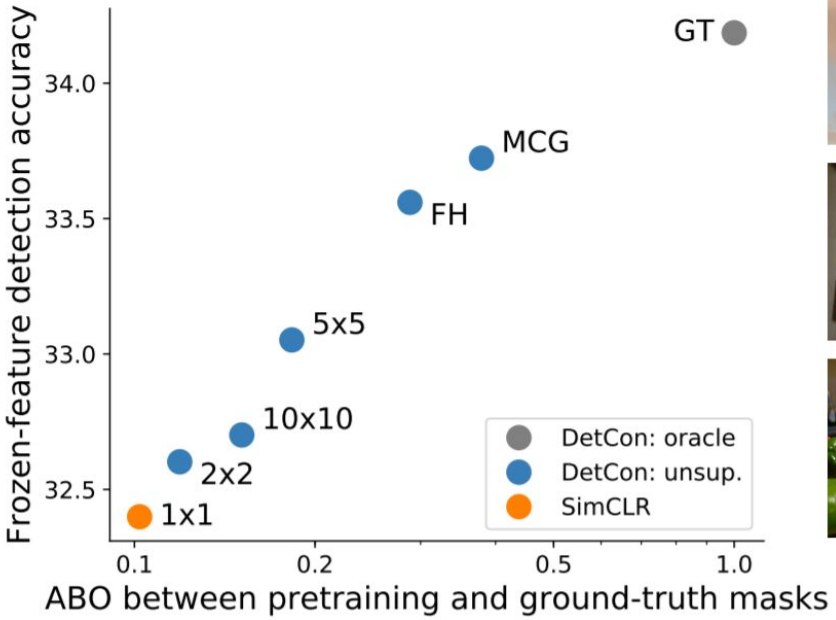
DetCon Objective

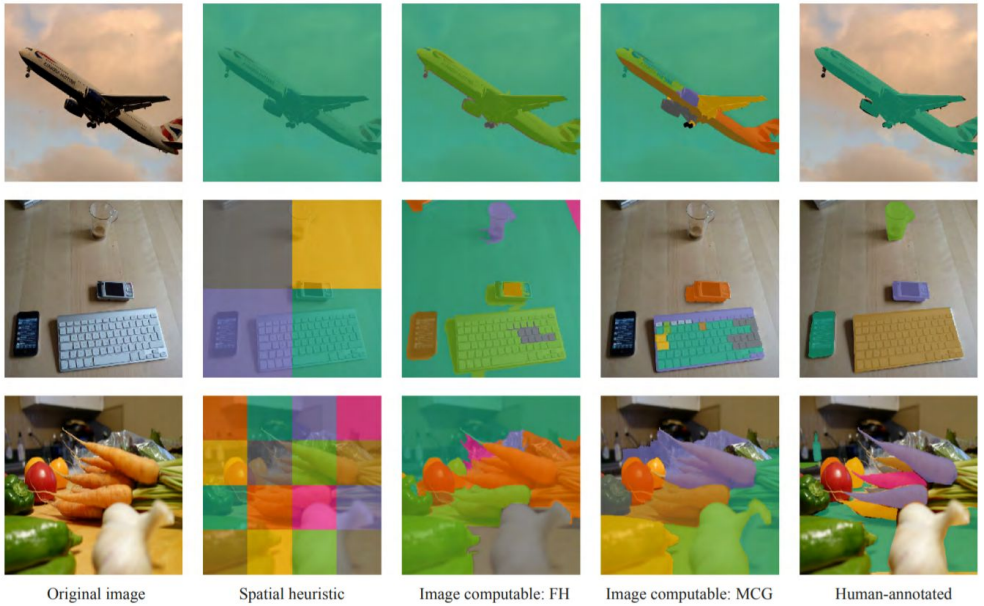
Fine tuning

Оставляем
только encoder

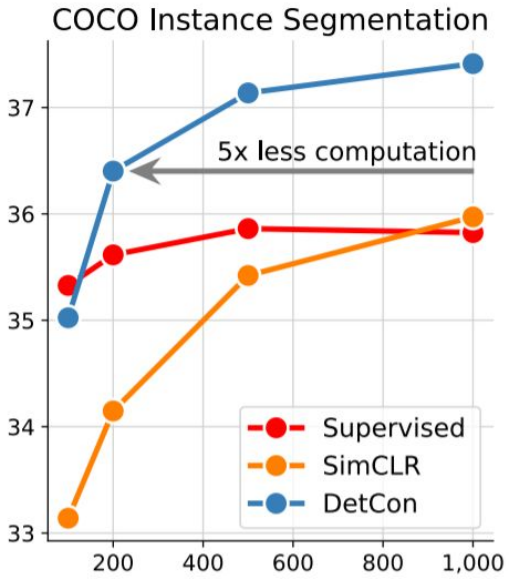


Результаты. Зависимость от алгоритма сегментации





Результаты. Сравнение с SimCLR и BYOL на COCO



	Instance Segmentation COCO	
Pretrain epochs	300	1000
BYOL	37.1	37.2
DetCon$_B$	37.8	38.2
Efficiency Gain	$> 3\times$	

ImageNet pretraining

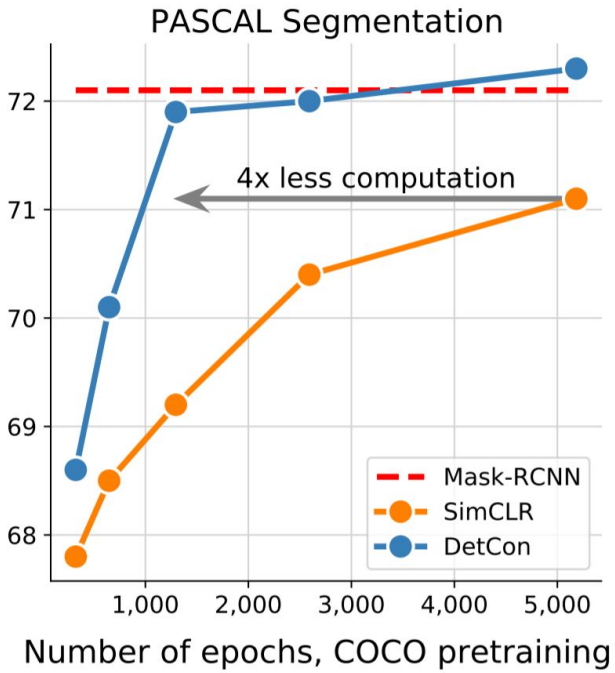
Результаты. Сравнение с другими методами на COCO

bb – bounding box
mk – mask

method	Fine-tune 1×		Fine-tune 2×	
	AP^{bb}	AP^{mk}	AP^{bb}	AP^{mk}
Supervised	39.6	35.6	41.6	37.6
VADeR	39.2	35.6	-	-
MoCo	39.4	35.6	41.7	37.5
SimCLR	39.7	35.8	41.6	37.4
MoCo v2	40.1	36.3	41.7	37.6
InfoMin	40.6	36.7	42.5	38.4
PixPro	41.4	-	-	-
BYOL	41.6	37.2	42.4	38.0
SwAV	41.6	37.8	-	-
DetCon$_S$	41.8	37.4	42.9	38.1
DetCon$_B$	**42.7**	**38.2**	**43.4**	**38.7**

ImageNet pretraining

Результаты. Сравнение с supervised методом



Результаты. Нужны ли положительные пары и большой батч?

model	all neg	two views	Masks FH	Masks GT
DetCon	✓	✓	**33.6**	37.0
(a)		✓	32.2	38.5
(b)			27.7	**38.8**

all neg – в 128 раз больше отрицательных примеров
two views – contrastive loss между признаками из разных аугментаций

Рецензент

Плюсы

- актуальная область применения
- сравнивается с лучшими на тот момент подходами и показывает более высокий результат
- алгоритм понятный и хорошо масштабируемый на новые подходы
- эксперименты на больших известных датасетах из открытого доступа
- наглядные графики для всех экспериментов и детальный ablation
- код с четкой инструкцией для воспроизведения
- ощутимый прирост в скорости обучения

Минусы

- нет обзора принципов генерации масок (возможно есть лучше)
- странно, что не используется для задачи классификации

Рецензент

Воспринимаемость
Хорошее изложение, все дополнительные детали расписаны и проиллюстрированы в аппендиксе

Вопросы

- как себя покажет в сравнении с новыми моделями (DINO)?
- как улучшить качество генерируемых масок?
- как показывает себя на задаче классификации?

Оценка: 8 (идея достойная, но так ли она сильна на текущий момент?)
Уверенность: 4 (для 100%-й надо еще потрогать код)

Практик-исследователь

Информация о статье
Препринт появился в марте 2021, осенью статью приняли на ICCV2021

Информация об авторах
Статья написана шестью исследователями из DeepMind (британская компания, занимающаяся искусственным интеллектом. В 2014 году была приобретена Google)

Работы, основанные на статье
Статья довольно новая и у нее всего 20 цитирований, в том числе она цитируется в статье, которую написали два из шести авторов оригинальной статьи в этом же году: Divide and Contrast: Self-supervised Learning from Uncurated Data и которую тоже приняли на ICCV2021

Практик-исследователь

Предыдущие работы авторов
Двое из авторов (Olivier J. He´naff и Aaron van den Oord) уже занимались self-supervised learning и в 2020м году выпустили статью Data-Efficient Image Recognition with Contrastive Predictive Coding. В статье для обучения используются представления для небольших патчей с изображения и контрастив лосс, возможно DetCon это продолжение идей этой работы.

Связанные работы
Авторы в своей работе при выборе архитектуры моделей и пайплайнов аугментаций опираются на SimCLR и BYOL

Практик-исследователь

Конкурирующие статьи

Сами авторы в статье отмечают что их подход очень похож на Unsupervised Semantic Segmentation by Contrasting Object Mask Proposals и Self-Supervised Visual Representation Learning from Hierarchical Grouping. Обе эти статьи также используют сегментацию изображений для self-supervised learning, отличия заключаются в том, что в них учится backbone предназначенный для решения задач сегментации и авторы этих статей не проводят эксперименты по pretraining efficiency (важная тема в DetCon).

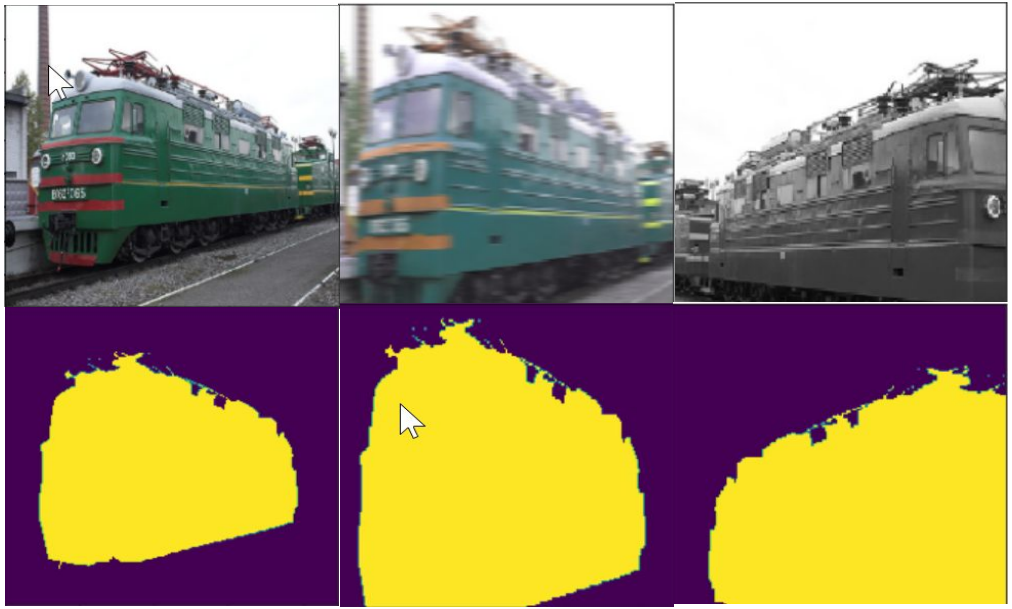
Что можно улучшить

Для работы алгоритма необходимо разбиение изображения на маски, авторы предлагают несколько способов это делать (различные эвристики, ручная разметка). Логичным продолжением кажется попытаться сделать весь алгоритм end-to-end, то есть предложить какую-либо дифференцируемую сегментацию изображений.

Хакер

Официальная имплементация (TF): https://github.com/deepmind/detcon
Имплементация на PyTorch: https://github.com/isaaccorley/detcon-pytorch

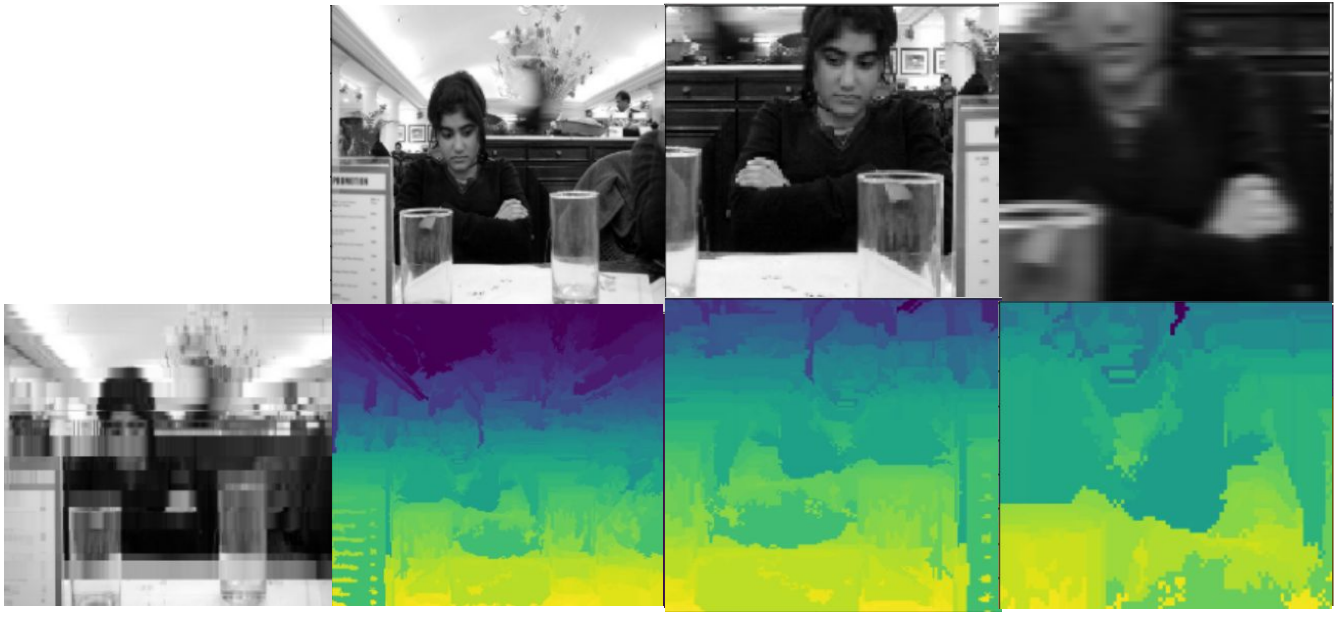
Методом, предложенным в статье было предобучено 2 сети (PASCAL VOC dataset): одна обучалась с ground truth масками, вторая с масками, сгенерированными с помощью алгоритма Felzenszwalb-Huttenlocher



Хакер

Официальная имплементация (TF): https://github.com/deepmind/detcon
Имплементация на PyTorch: https://github.com/isaaccorley/detcon-pytorch

Методом, предложенным в статье было предобучено 2 сети (PASCAL VOC dataset): одна обучалась с ground truth масками, вторая с масками, сгенерированными с помощью алгоритма Felzenszwalb-Huttenlocher



Хакер. Эксперимент

Зафиксируем обученные encoderы. И будем обучать только FC-голову (Pooling + Linear) для задачи классификации на датасете CIFAR10.

Для сравнения также возьмем предобученный (на ImageNet) encoder и также будем обучать только голову.

	ImNet	DetCon_gt	DetCon_gen
Train acc	0.722	0.263	0.192
Test acc	**0.690**	0.260	0.195

Результаты классификации на датасете CIFAR10.