Михаил Катунькин,
БПМИ-192

https://arxiv.org/abs/2106.08254

https://arxiv.org/pdf/2112.10740.pdf

Are Large-scale Datasets Necessary for Self-Supervised Pre-training?

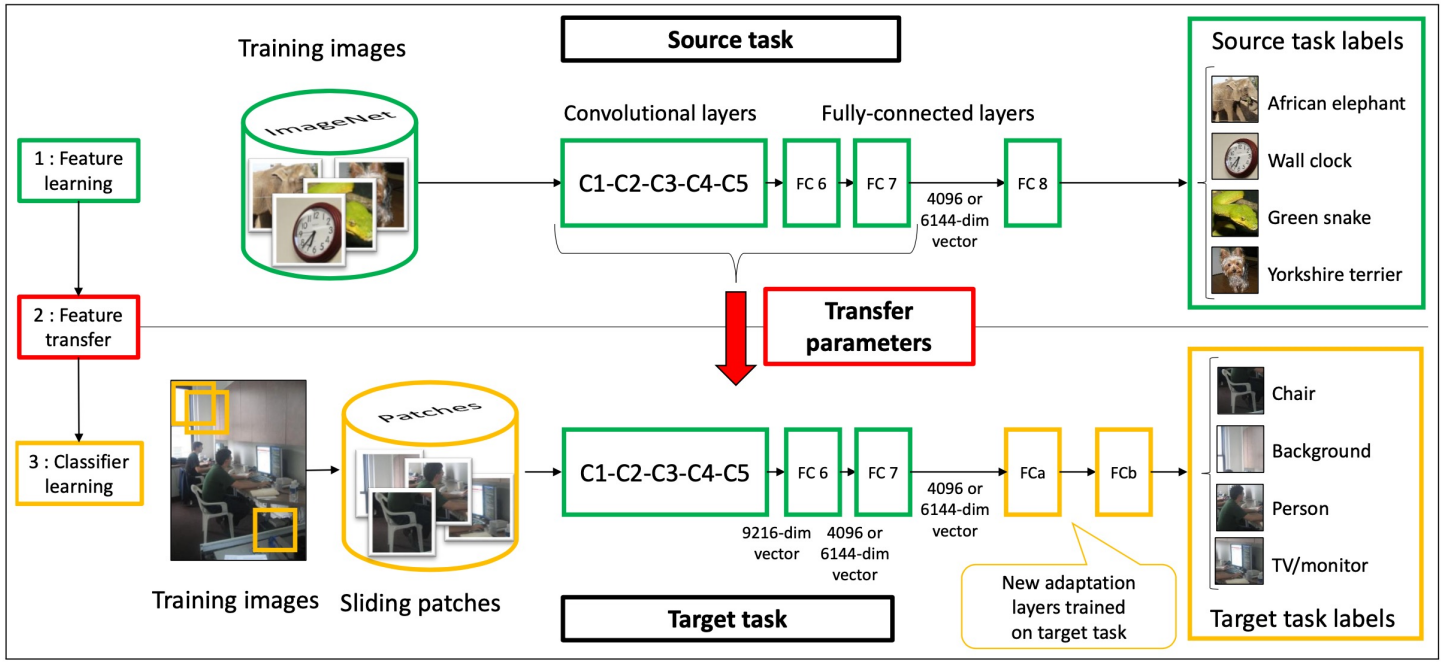
Нужны ли большие датасеты для предобучения моделей обработки изображений?

Что делают, если данных мало?

- Обучают модель на большом датасете типа ImageNet
- Используют веса этой модели для решения целевой задачи с небольшим числом данных

Supervised pre-training

- Берем сеть, которая обучалась классифицировать картинки из ImageNet
- Отрываем голову, дообучаем на нашей задаче



https://www.microsoft.com/en-us/research/wp-content/uploads/2014/06/paper.pdf

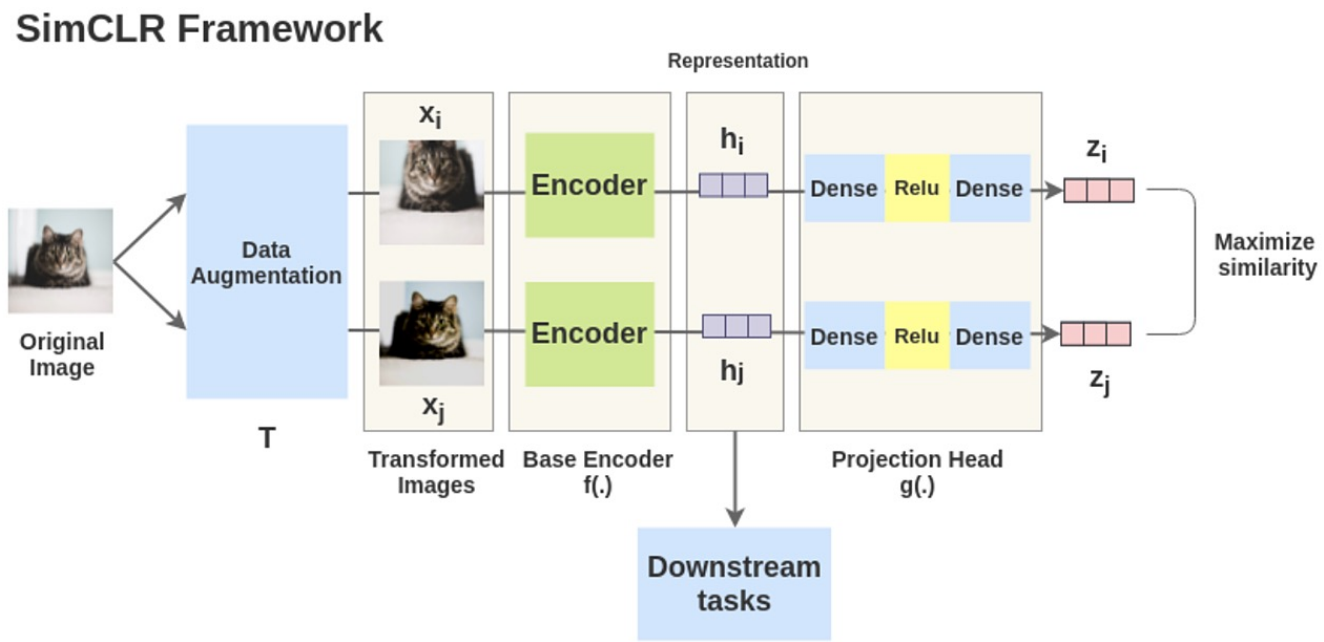
Проблемы supervised pre-training

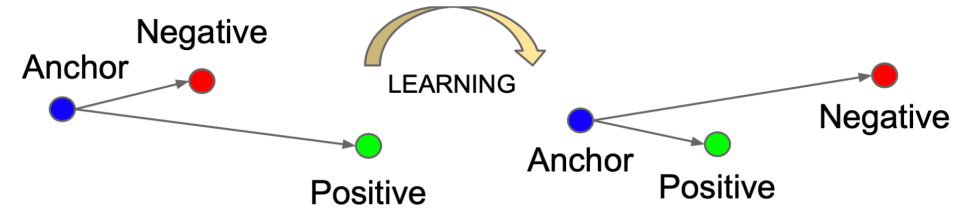
1. Данные целевой задачи из другого распределения нежели при предобучении *(Domain Shift)*
2. Модель учится соответствовать меткам и отбрасывает важную информацию *(Supervision Collapse)*

Self-supervised pre-training

Contrastive learning

- Pretext task – получить близкие эмбеддинги для аугментаций картинки и далекие для разных картинок

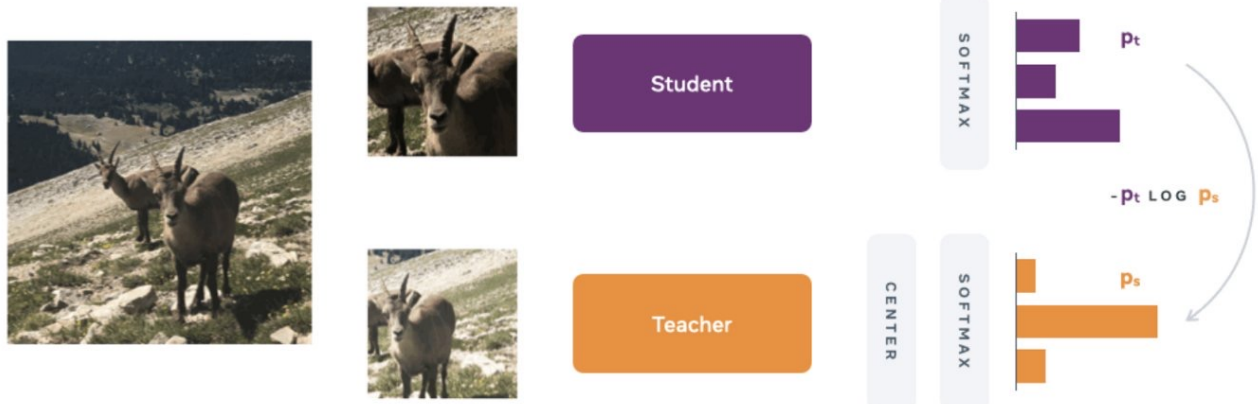




https://medium.com/geekculture/understanding-contrastive-learning-and-moco-efe491e4eed9

Self-supervised pre-training

DINO



Joint embeddings

- Есть сети ученик и учитель
- Веса ученика обновляют веса учителя с моментом
- Ученику на вход подается деталь изображения, учителю – все изображение
- Ученик должен выдать эмбеддинг похожий на учителя
- Т.е. по локальному куску восстановить глобальный контекст

$$\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$$

https://ai.facebook.com/blog/dino-paws-computer-vision-with-self-supervised-transformers-and-10x-more-efficient-training/ *** Признана в РФ экстремистской организацией ***

Проблемы классического self-supervised

1. Данные целевой задачи из другого распределения нежели при предобучении *(Domain Shift)*
2. Аугментации типа random-crop подразумевают, что объект в центре
3. Нужны большие датасеты
4. Подходы подгоняются под ImageNet

Хочется

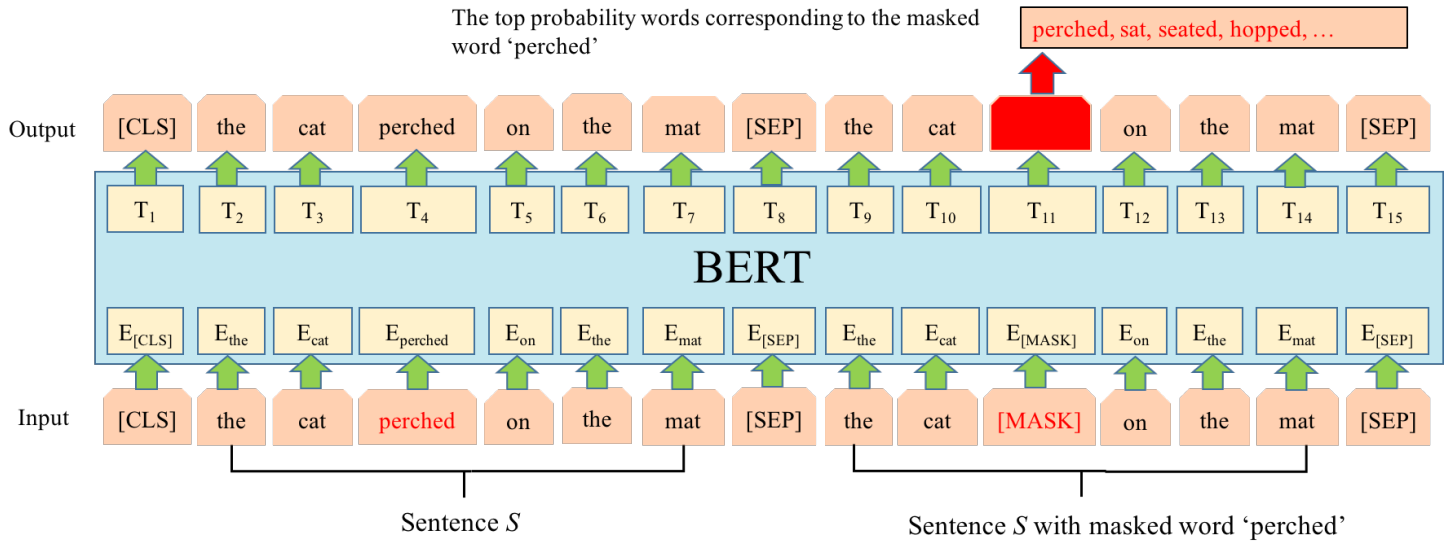
Уметь обучать модель на данных из целевого распределения (или хотя бы похожих)

Но тогда нужно обучаться на датасетах:

- небольшого размера
- с неотцентрованными объектами

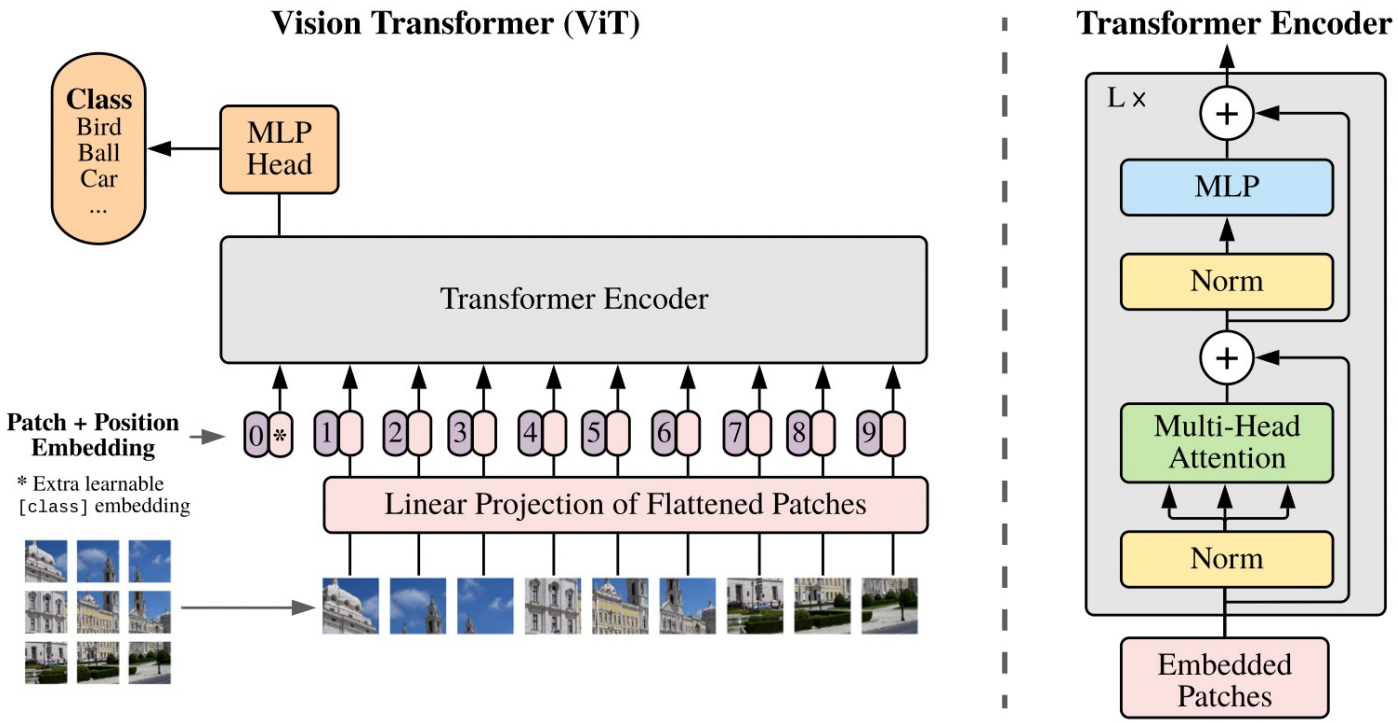
Видели такое в NLP

- Берут трансформеры
- Обучают на задаче Masked-Language-Modelling
- Тексты для pre-train – какие дадите



Vision Transformers (ViT)

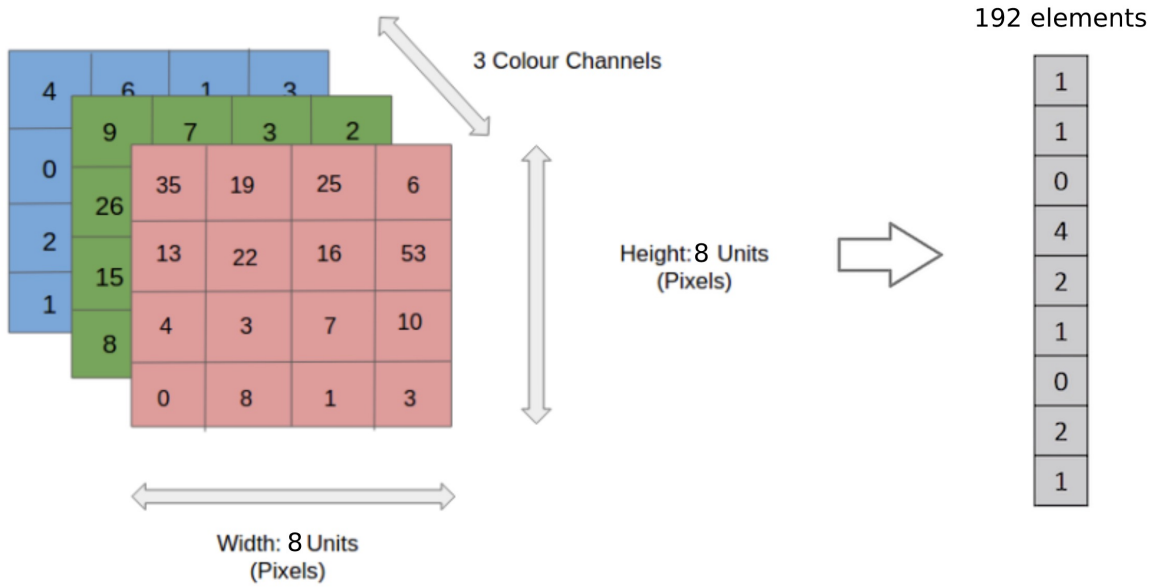
- Изображение бьется на патчи 16x16
- Патчу сопоставляется некий токен
- Добавляется позиционный эмбеддинг
- Последовательность подается на вход трансформеру



https://towardsdatascience.com/implementing-visualttransformer-in-pytorch-184f9f16f632

Как сопоставлять токен?

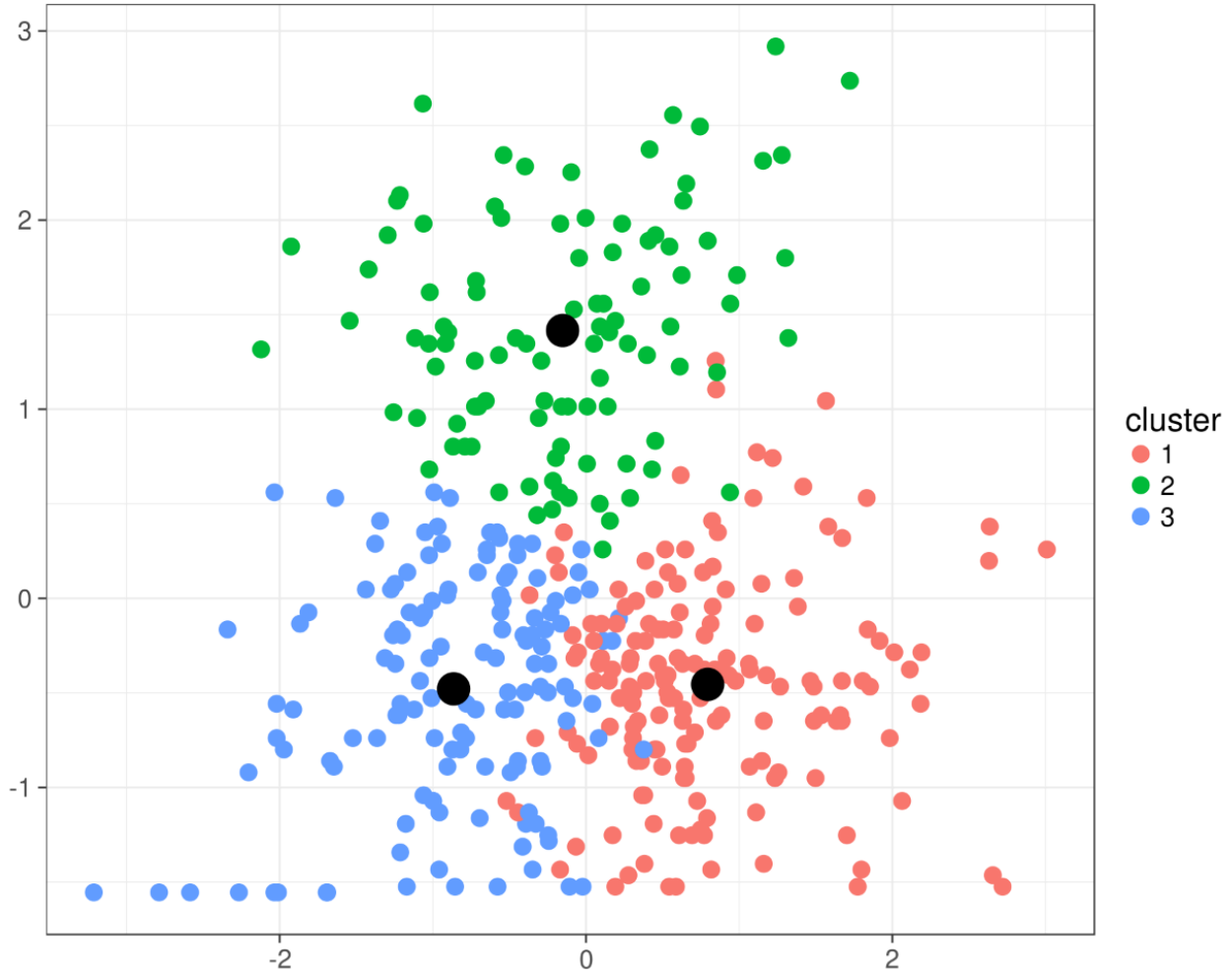
- Патч вытягивается в вектор
- В пространстве R^d задается алфавит (d = 192) из 8192 токенов
- Патчу сопоставляется токен с которым у него максимальное косинусное расстояние в пиксельном пространстве



$$t = \text{argmax}_{i \in \{1,\ldots,V\}} \mathbf{x}^\top \mathbf{e}_i.$$

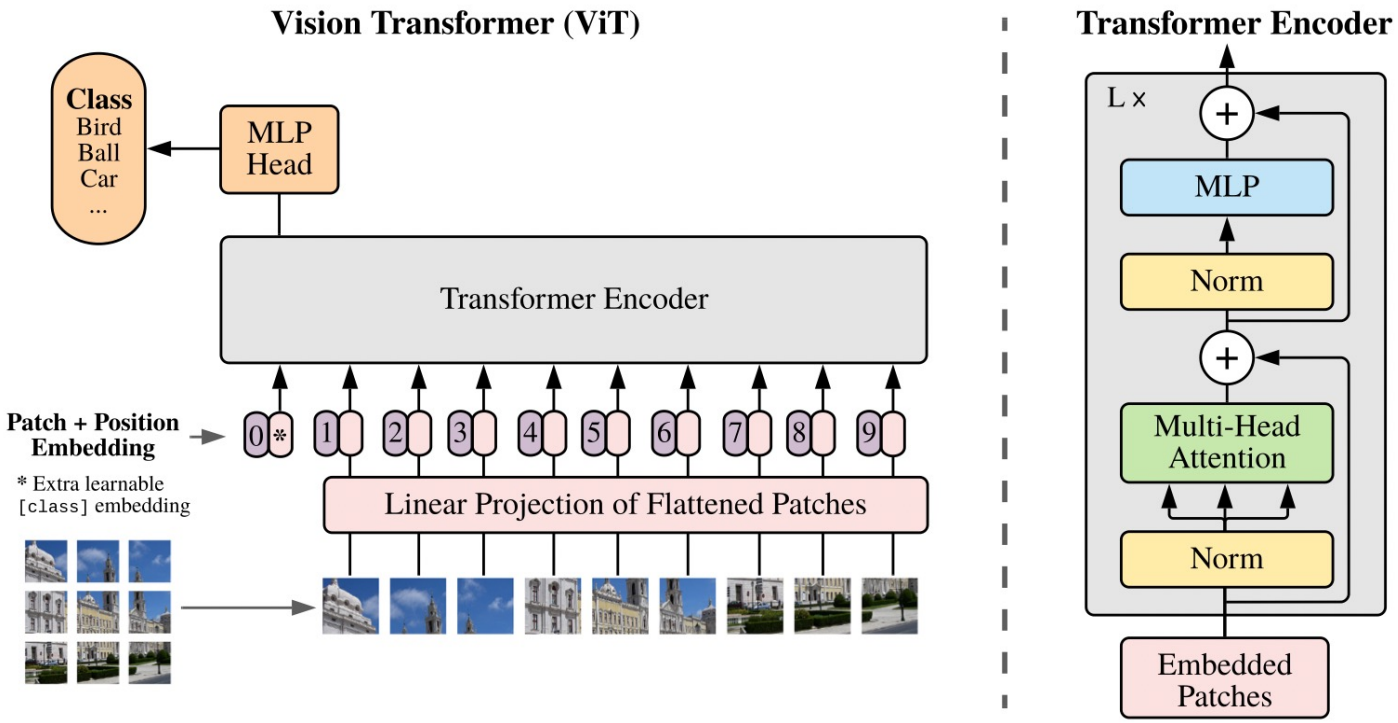
Как задавать алфавит?

- Обучить модель-токенайзер
- Сгенерировать вектора, где каждая координата из равномерного распределения
- Выбрать случайные вектора из патчей в датасете
- **Кластеризовать патчи при помощи k-means, выбрать центроиды**



Vision Transformers (ViT)

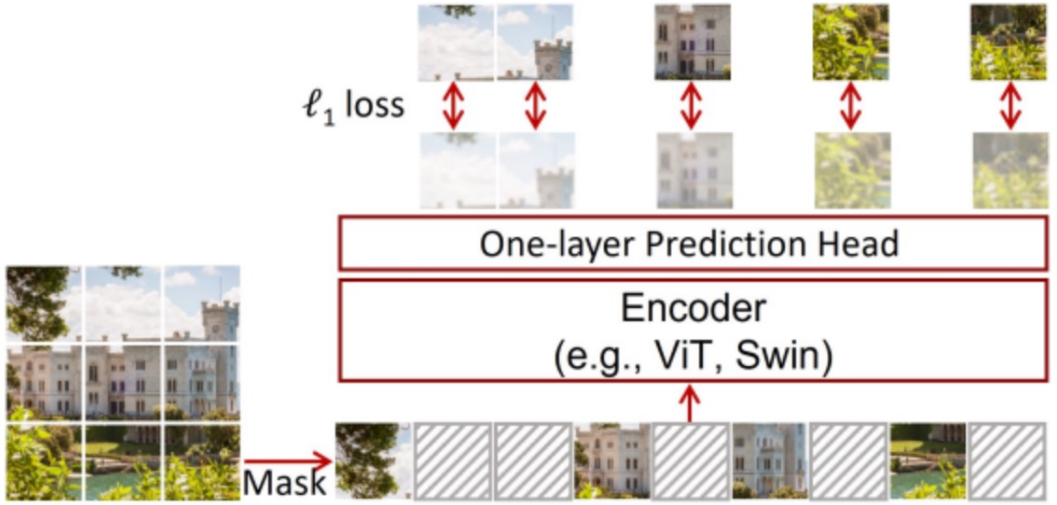
- Изображение бьется на патчи 16x16
- Патчу сопоставляется некий токен
- Добавляется позиционный эмбеддинг
- Последовательность подается на вход трансформеру



https://towardsdatascience.com/implementing-visualttransformer-in-pytorch-184f9f16f632

Masked Image Modelling

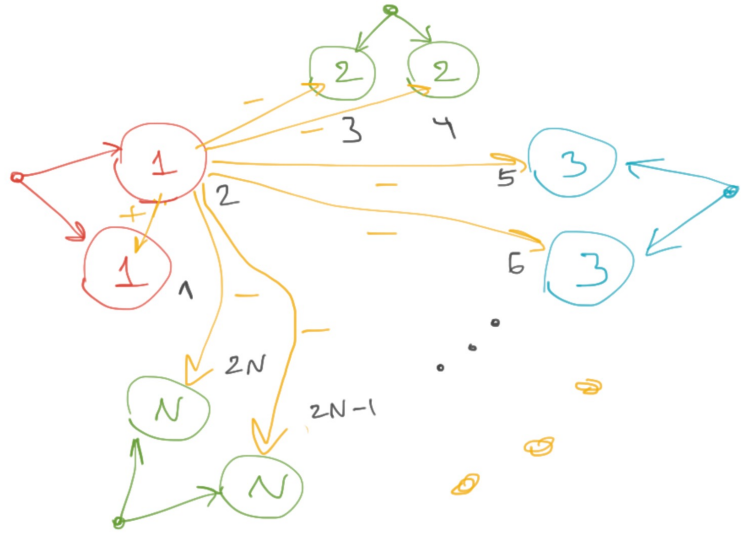
- Очень похоже на обучение BERT в NLP
- Некоторые токены закрываются маской, пропускаются через энкодер
- Слой-пресказатель про представлениям угадывает скрытые токены
- По предсказанным токенам считается loss
- В дискретном случае – кросс-энтропия



https://analyticsindiamag.com/an-illustrative-guide-to-masked-image-modelling/

SplitMask

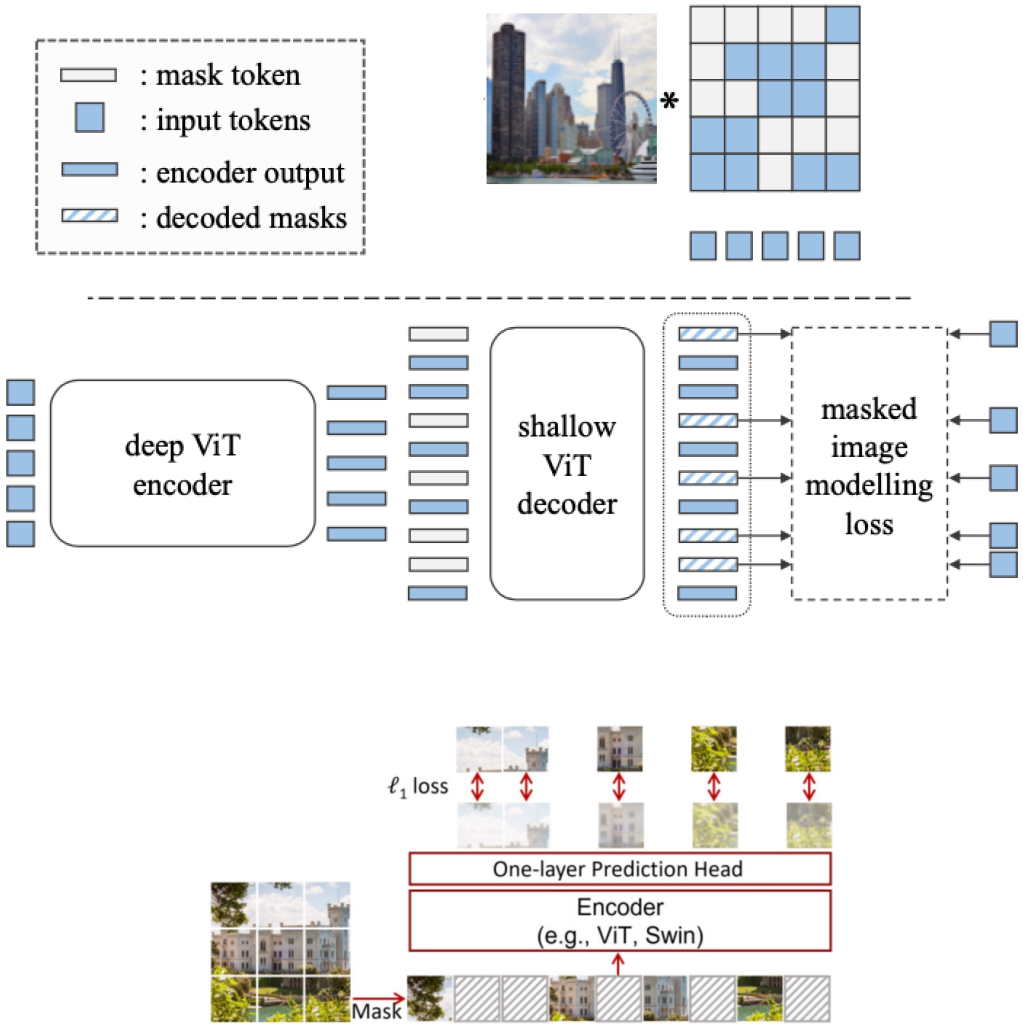
- На уровне батча картинок – хотим что-то вроде SimCLR
- Берем картинку, получаем две ее аугментации
- По аугментациям получаем два эмбеддинга
- Эмбеддинги аугментаций считаются одним классом
- Эмбеддинги остальных картинок из батча – разными
- В качестве лосса – InfoNCE



$$\ell(\mathbf{x}_a) = \frac{\exp(\mathbf{x}_a^\top \mathbf{x}_b/\tau)}{\sum_{\mathbf{y}\in\{\mathbf{x}_b\}\cup\mathcal{N}} \exp(\mathbf{x}_a^\top \mathbf{y}/\tau)},$$

SplitMask

- На уровне работы с отдельной картинкой – ViT + MIM
- Открытые токены пропускаются через энкодер ViT
- На место скрытых токенов добавляется эмбеддинг-маска
- Эмбеддинги пропускаются через неглубокий декодер ViT
- К эмбеддингам на выходе из декодера применяется софтмакс-классификатор, чтобы предсказать пропущенные токены
- По предсказанным пропускам считается Masked Image Modelling loss (кросс-энтропия)

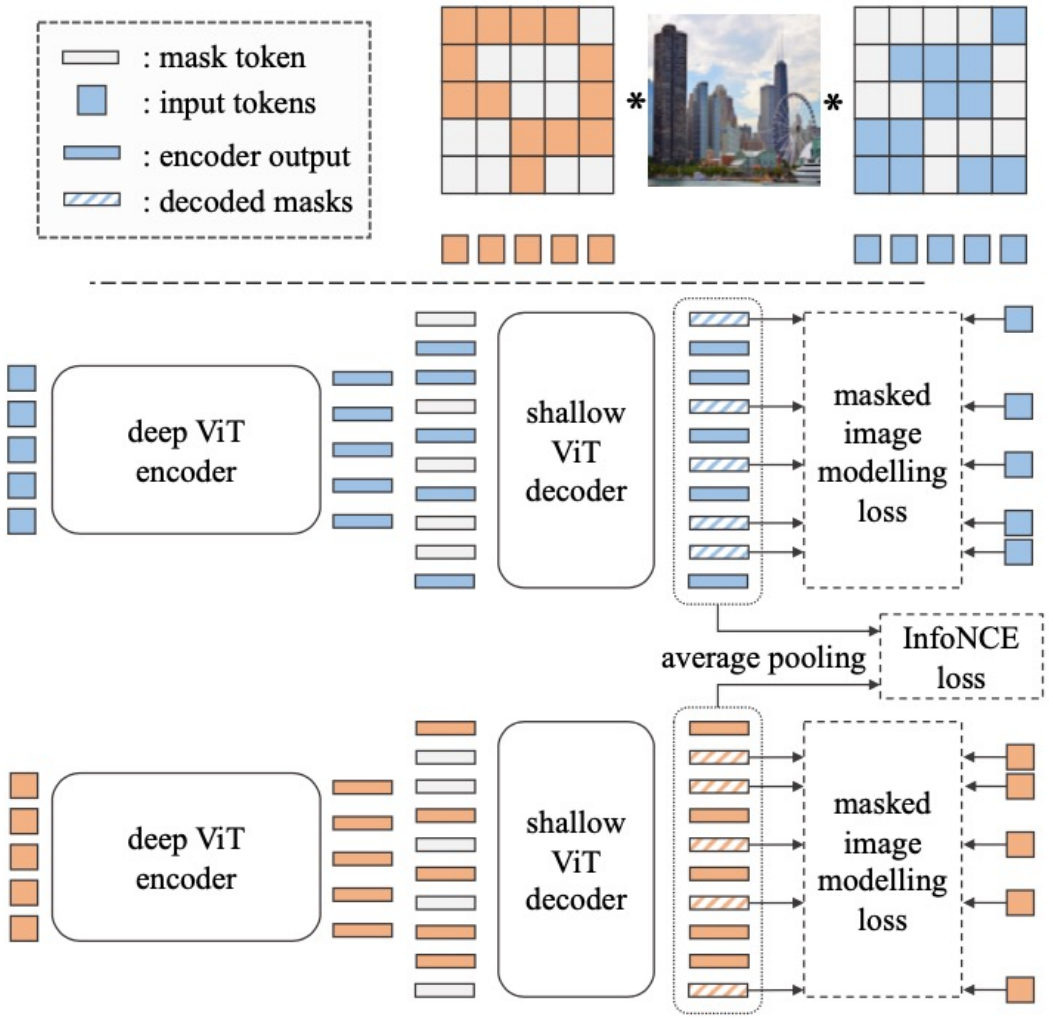


Visual Transformer (ViT) + Masked Image Modelling (MIM) + contrastive learning

SplitMask

- На самом деле – разделяем патчи (16x16) на две группы
- Получаем два изображения – в одном скрыты патчи одной группы, в другом – другой
- Пропускаем их через два MIM-пайплайна с разделяемыми весами
- Усредняем представления на выходе из декодера – это глобальный дескриптор изображения, по которому считаем InfoNCE
- Добавляем MIM loss к InfoNCE

$$\ell(\mathbf{x}_a) = \frac{\exp(\mathbf{x}_a^\top \mathbf{x}_b/\tau)}{\sum_{\mathbf{y}\in\{\mathbf{x}_b\}\cup\mathcal{N}} \exp(\mathbf{x}_a^\top \mathbf{y}/\tau)},$$



Как это потом применять?

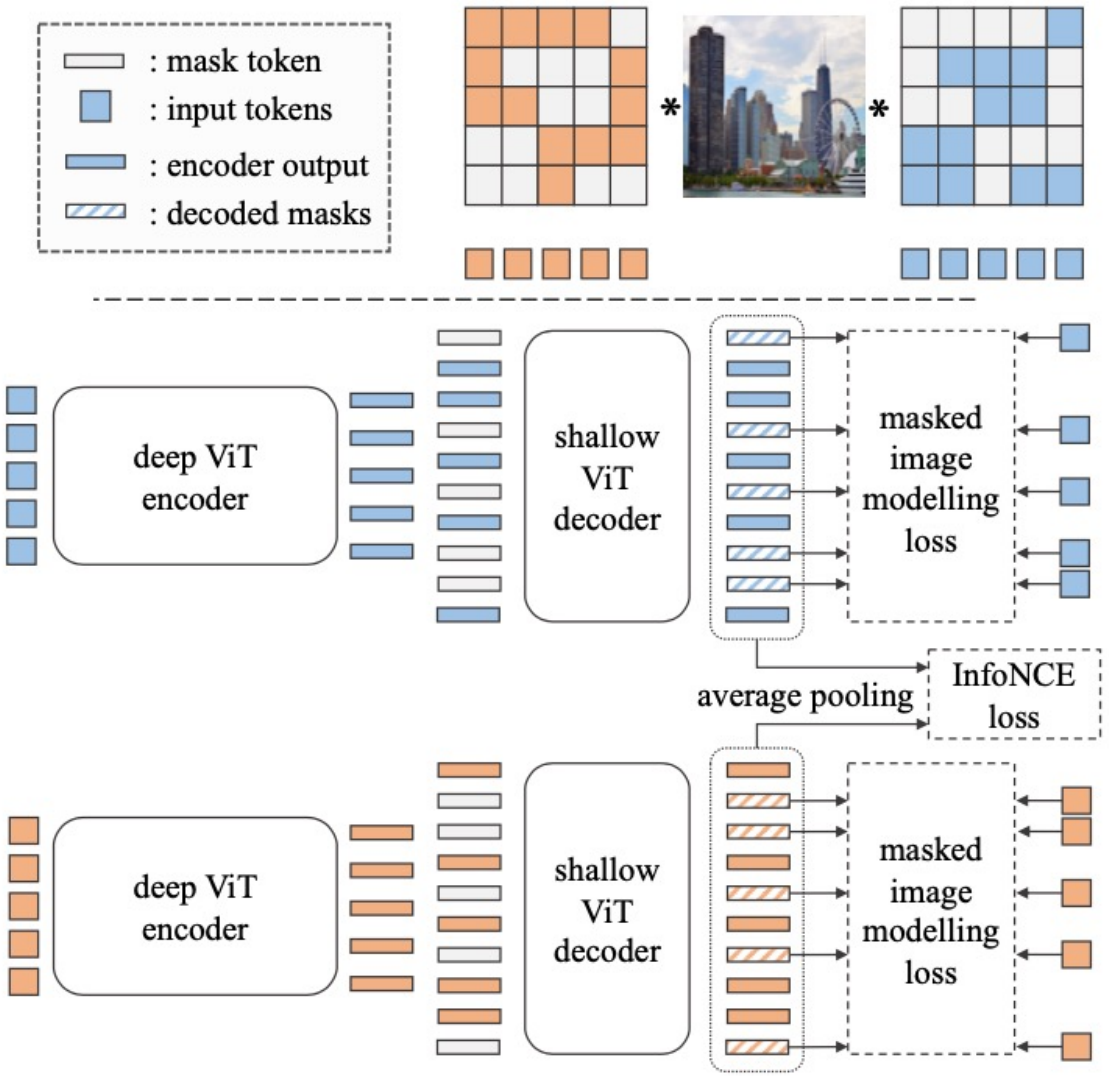
Классификация:

- берем выходы энкодера
- усредняем
- применяем софтмакс-классификатор

$$\mathrm{softmax}(\mathrm{avg}(\{\boldsymbol{h}_i^L\}_{i=1}^N \boldsymbol{W}_c))$$

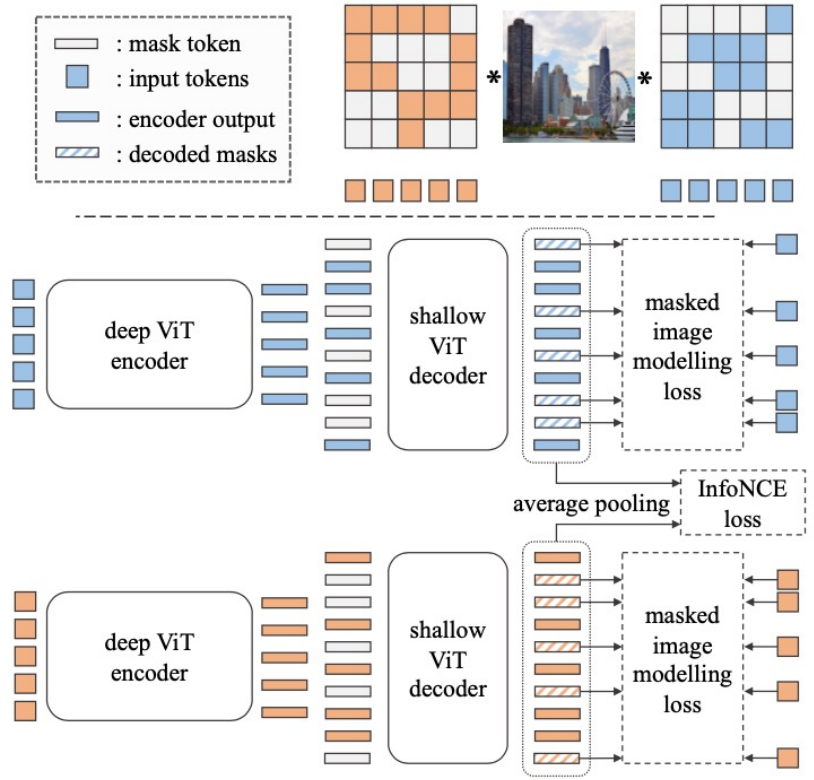
Сегментация:

- к выходам энкодера применяется несколько deconvolution-слоев, генерирующих сегментацию



Зачем так?

- MIM – локальные признаки
- Contrastive loss – глобальные
- Неглубокий декодер нужен, чтобы сэкономить на прогонянии токенов-масок через глубокий энкодер
- Благодаря декодеру из энкодера убирается функция специфичная для pretext-task – более общие признаки
- Маскирование – общий случай random crop. Поэтому хорошо работает на объектах в произвольном месте кадра



А оно работает вообще?

На ImageNet работает!

Method	Backbone	Epochs	Top-1
MocoV3 [68]	ViT-S	300	81.4
DINO [18]		300	**81.5**
BEiT [24]		300	81.3
SplitMask		300	**81.5**
MocoV3 [68]	ViT-B	300	83.2
DINO [18]		400	**83.6**
BEiT [24]		300	82.8
BEiT [24]		800	83.2
SplitMask		300	**83.6**

- На классификации картинок из ImageNet SplitMask показал State-Of-The-Art результат
- Хотя SplitMask гораздо легковеснее DINO

Хотели

Уметь обучать модель на данных из целевого распределения (или хотя бы похожих)

Но тогда нужно обучаться на датасетах:

- небольшого размера
- с неотцентрованными объектами

Sample Efficiency

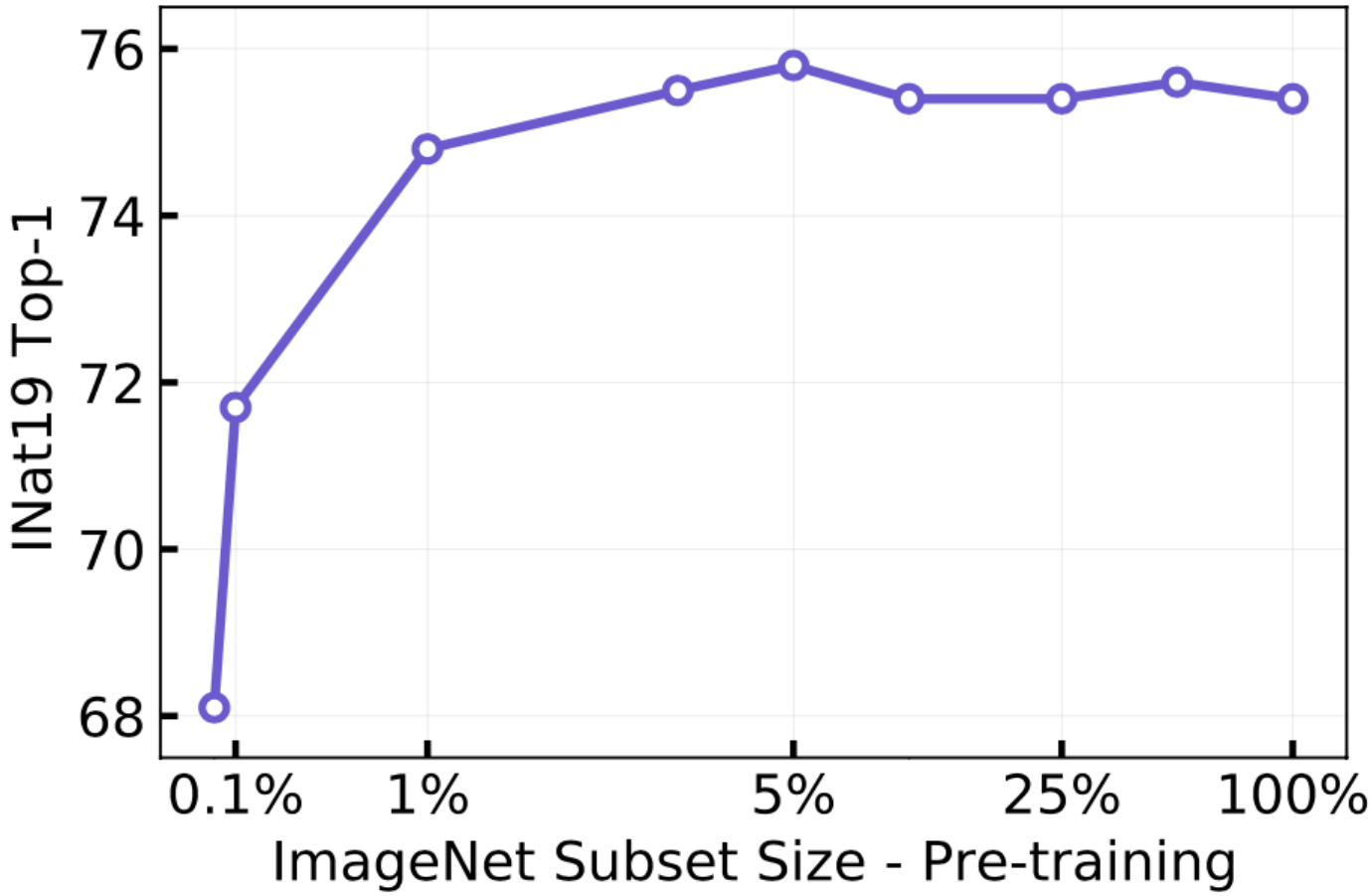
- Пробовали обучать на какой-то доле ImageNet для предсказания на датасете iNaturalist-2019
- При этом, увеличивали число эпох, чтобы число итераций не менялось
- Качество Supervised/DINO падает
- Качество denoising autoencoders (BEiT, SplitMask) почти неизменно
- На COCO (маленький, объекты не в центре) – выбили лучший результат

Method	IMNet 1% *epochs: 30k*	IMNet 10% *epochs: 3k*	IMNet Full *epochs: 300*	COCO *epochs: 3k*
Supervised	71.6	75.0	75.8	-
DINO [18]	70.1	73.1	**78.4**	71.9
BEiT [24]	74.1	74.5	75.2	74.4
SplitMask	**74.8**	**75.4**	75.4	**76.3**

BEiT – SplitMask без contrastive loss

Sample Efficiency

- Обучались на доле ImageNet
- Применяли к датасету iNaturalist- 2019
- Уже на 5% выжали максимум
- Даже по 1 картинке на класс (0.1%) дало +4% по сравнению со случайной инициализацией



Можно ли использовать целевой датасет для предобучения?

- SplitMask либо выбивает лучший результат, либо дает качество, сравнимое с предобучением на ImageNet
- Там, где SplitMask проигрывает, датасеты очень малы

Method	Backbone	Supervised pre-training	Data Used		iNat-18	iNat-19	Food 101	Cars	Clipart	Painting	Sketch
			IMNet	Target	437k	265k	75k	8k	34k	52k	49k
Liu et al. [67]∓	CVT-13	✗	✗	✓	-	-	-	-	60.6	55.2	57.6
	ResNet-50	✗	✗	✓	-	-	-	-	63.9	53.5	59.6
Random Init.	ViT-S	✗	✗	✓	59.6	67.5	84.7	35.3	41.0	38.4	37.2
DeiT [50]		✓	✓	✓	69.9	75.8	**91.5**	92.2	**79.6**	**74.2**	**72.5**
BEiT [24]		✗	✓	✓	68.1	75.2	90.5	92.4	75.3	68.7	68.5
BEiT		✗	✗	✓	68.8	76.1	90.7	92.7	-	69.0	-
SplitMask		✗	✗	✓	**70.1**	**76.3**	**91.5**	**92.8**	78.3	69.2	70.7
Random Init.	ViT-B	✗	✗	✓	59.6	68.1	83.3	36.9	41.9	37.6	34.9
DeiT [50]		✓	✓	✓	73.2	77.7	**91.9**	92.1	**80.0**	**73.8**	**72.6**
BEiT [24]		✗	✓	✓	71.6	78.6	91.0	**93.9**	78.0	71.5	71.4
BEiT		✗	✗	✓	72.4	79.3	91.7	92.7	-	70.7	-
SplitMask		✗	✗	✓	**74.6**	**80.4**	91.2	93.1	79.3	72.0	72.1

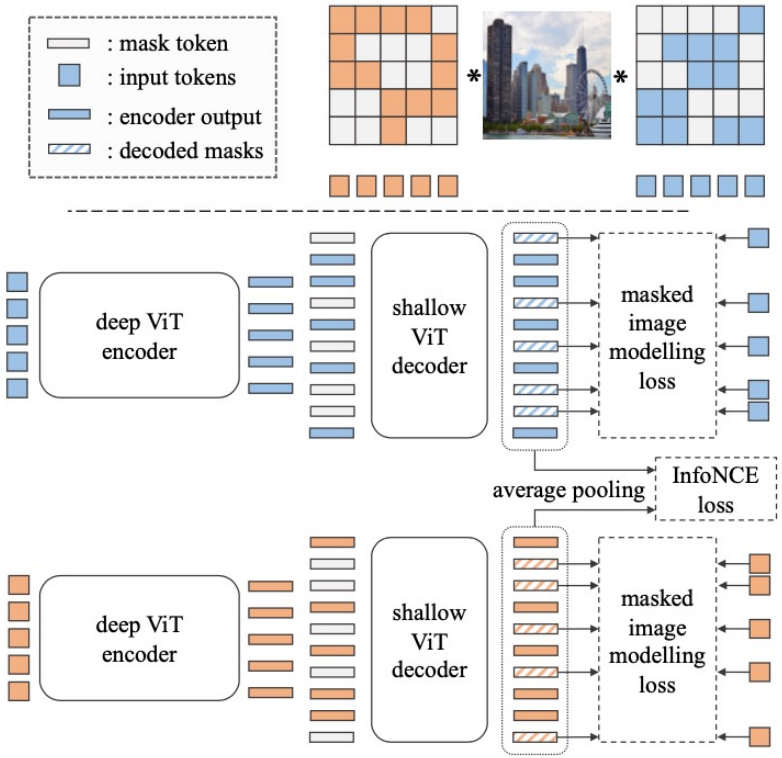
Сегментация на датасете COCO

- Использовался пайплайн Mask R-CNN
- BEiT, предобученный на COCO, показал результат лучше предобучения на ImageNet
- SplitMask превзошел BEiT

Method	Backbone	Pre-training			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
		Supervised	IMNet	COCO						
Random Initialization	ViT-S	✗	✗	✗	38.3	60.1	41.4	35.6	57.1	37.7
Random Initialization†		✗	✗	✗	42.8	64.5	45.6	39.1	61.5	41.7
DeiT [50]		✓	✓	✗	44.2	66.6	47.9	40.1	63.2	42.7
BEiT [24]		✗	✓	✗	44.5	66.2	48.8	40.3	63.2	43.1
DINO [18]		✗	✗	✓	43.7	65.5	47.7	39.6	62.3	42.3
BEiT		✗	✗	✓	44.7	66.3	48.8	40.2	63.1	43.2
SplitMask		✗	✗	✓	**45.3**	**66.9**	**49.4**	**40.6**	**63.6**	**43.5**
Random Initialization	ViT-B	✗	✗	✗	40.7	62.7	44.2	37.1	59.1	39.4
Random Initialization†		✗	✗	✗	43.0	64.2	46.9	38.8	61.3	41.6
DeiT [50]		✓	✓	✗	45.5	**67.9**	49.2	41.0	64.6	43.8
BEiT [24]		✗	✓	✗	46.3	67.6	50.6	41.6	64.5	44.9
DINO [18]		✗	✗	✓	43.1	64.4	46.9	38.9	61.4	41.4
BEiT		✗	✗	✓	46.7	67.7	51.2	41.8	65.0	44.6
SplitMask		✗	✗	✓	**46.8**	**67.9**	**51.5**	**42.1**	**65.3**	**45.1**

Влияние компонент на ошибку

- Смотрели на качество классификации на ImageNet
- Только contrastive loss не дает достаточно информации, чтобы хорошо обучить модель
- Но становится хорошим дополнением к MIM



Method	Split	Inpaint	Match	Finetune	Lin.	Hours
BEiT [24]	✗	✓	✗	82.8	41.0	32.5
SplitMask	✓	✓	✗	83.3	46.4	**31.0**
	✓	✗	✓	79.3	4.0	32.5
	✓	✓	✓	**83.6**	**46.5**	34.0

Выводы

Можно предобучать модель:

- на датасетах небольшого размера
- с объектами не по центру
- на целевом датасете

Датасеты, вроде ImageNet, не обязательны для предобучения!