

# Основы глубинного обучения

Лекция 8

Идентификация объектов. Обучение без учителя.

Евгений Соколов

[esokolov@hse.ru](mailto:esokolov@hse.ru)

НИУ ВШЭ, 2020

# Идентификация объектов

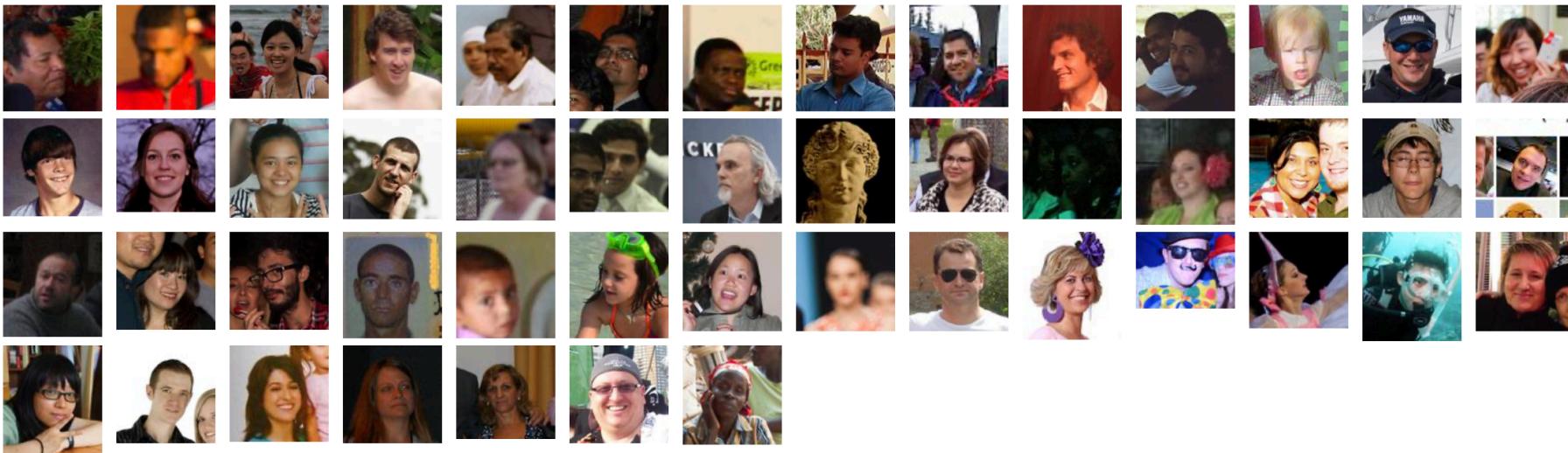
# Labeled Faces in the Wild

- Около 13 тысяч фотографий
- Около 6 тысяч человек



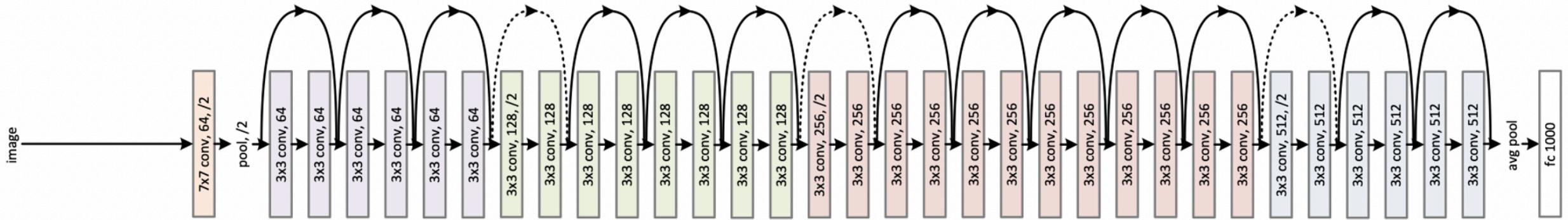
# MegaFace

- 4.7 миллионов фотографий
  - Около 700 тысяч человек
  - В среднем 7 фото на человека



<http://megaface.cs.washington.edu>

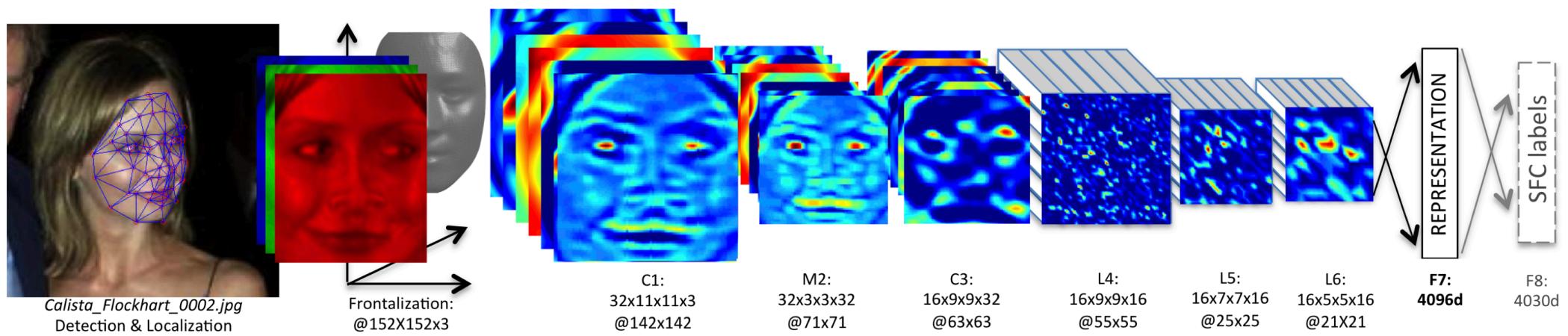
# Дообучение



Если данных совсем мало:

- Берём модель из другой задачи
- Заменяем последний слой на слой с нужным числом выходов
- Обучаем только его
- По сути, это обучение линейной модели

# DeepFace



# DeepFace

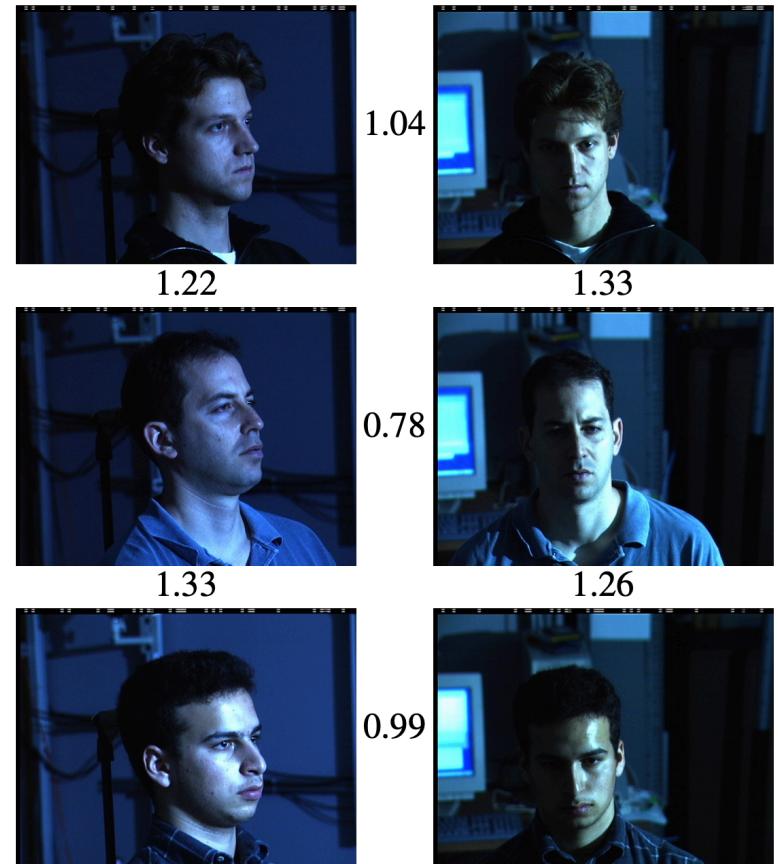
- Обучаем некоторую архитектуру для классификации (число классов = число людей в данных)
- Используем выходы предпоследнего слоя как признаковое описание изображения
- Признаки нормализуются (чтобы норма была единичной)
- Считаем близость векторов по какой-нибудь метрике

# DeepFace

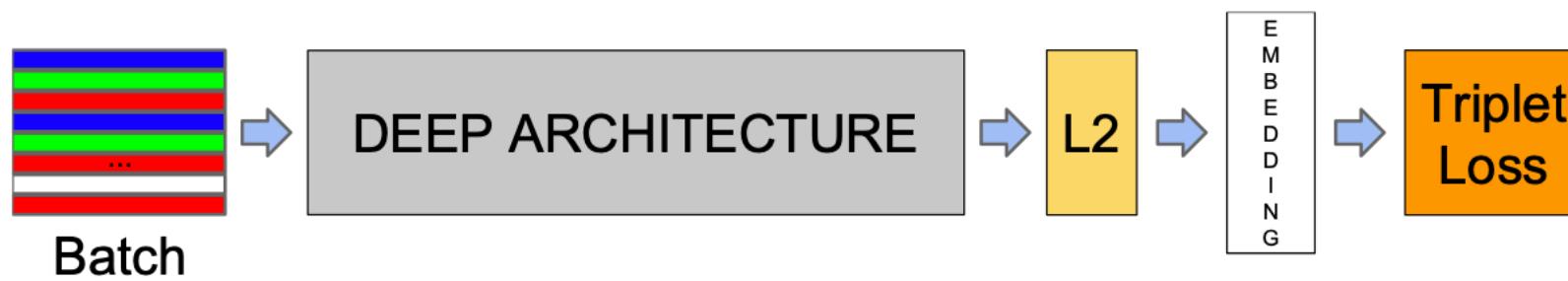
- Можно сравнить расстояние с порогом, чтобы идентифицировать человека

# FaceNet

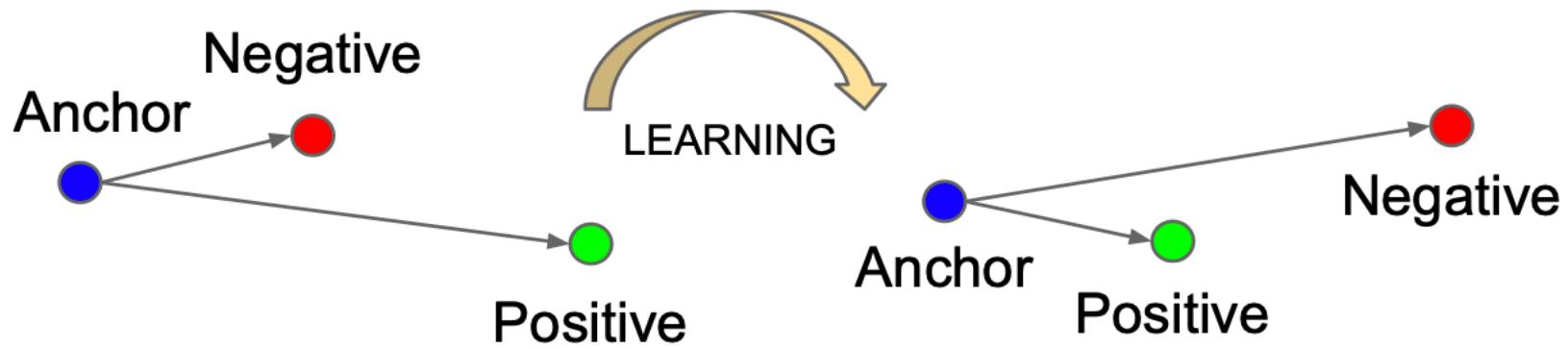
- Почему бы в явном виде не обучать представления изображений так, чтобы фотографии одного человека имели близкие представления?



# FaceNet



# FaceNet



$$\sum_i^N \left[ \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

# FaceNet

- Важно правильно выбирать триплеты
- Обычно: выбираем positive и ищем semi-hard negatives

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$$

# Триплетная и попарная ошибки

- Попарная ошибка:

$$\sum_{(i,j) \in R} [a(x_i) - a(x_j) < 0] \rightarrow \min$$

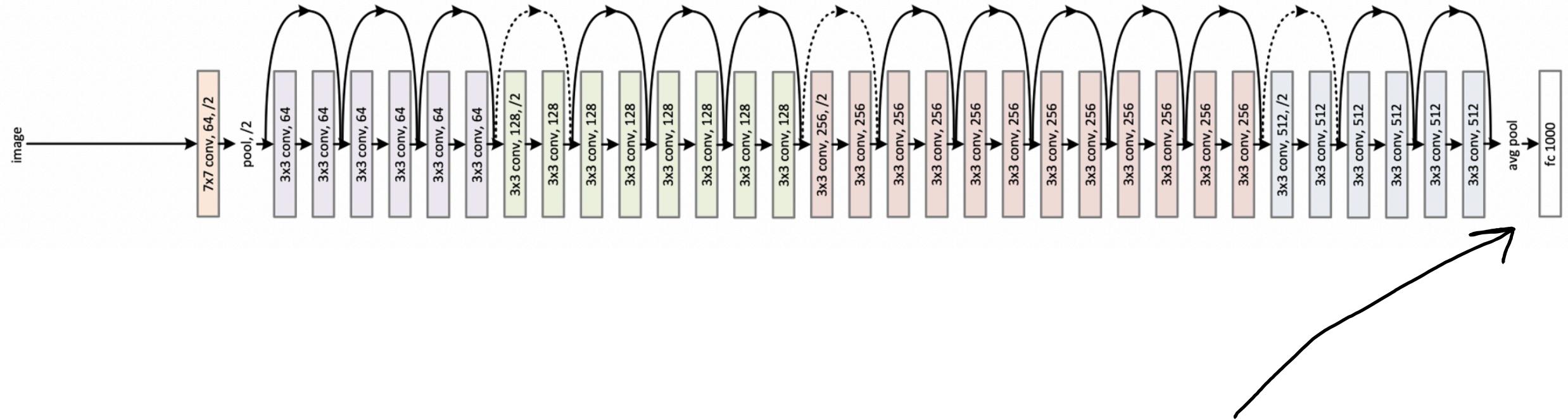
- Не совсем про обучение расстояния

# FaceNet

- Точность на LFW: 99.63%

Обучение без учителя

# Представления изображений

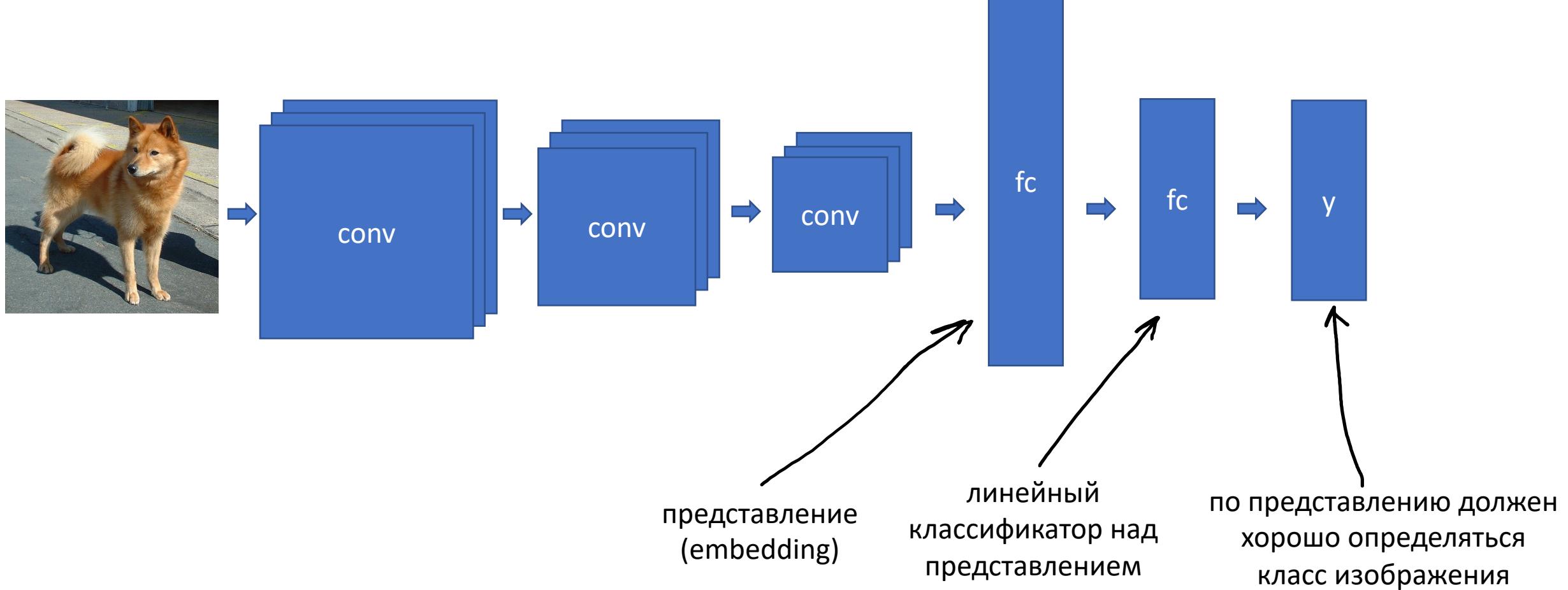


Выход предпоследнего полносвязного  
слоя — хорошее представление картинки

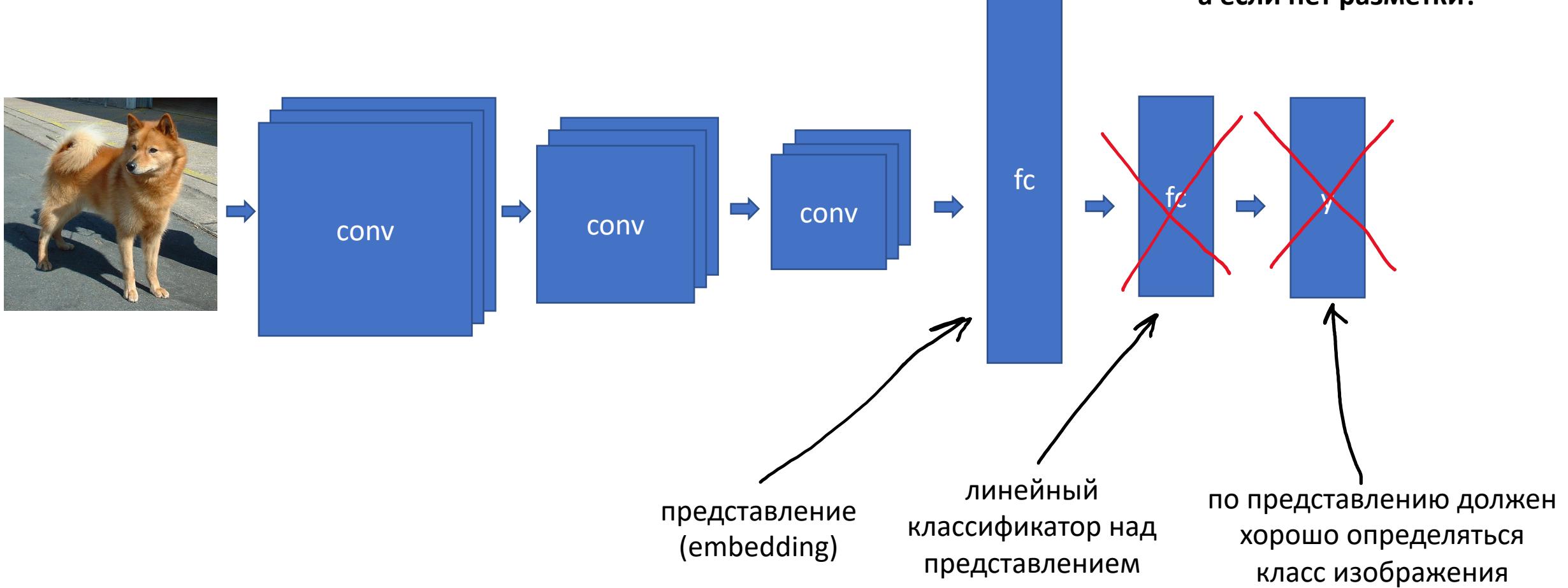
# Представления изображений

- Выход предпоследнего полносвязного слоя — хорошее представления картинки
- Но для его обучения нужны изображения с разметкой
- Может, получится строить такие представления и без разметки?

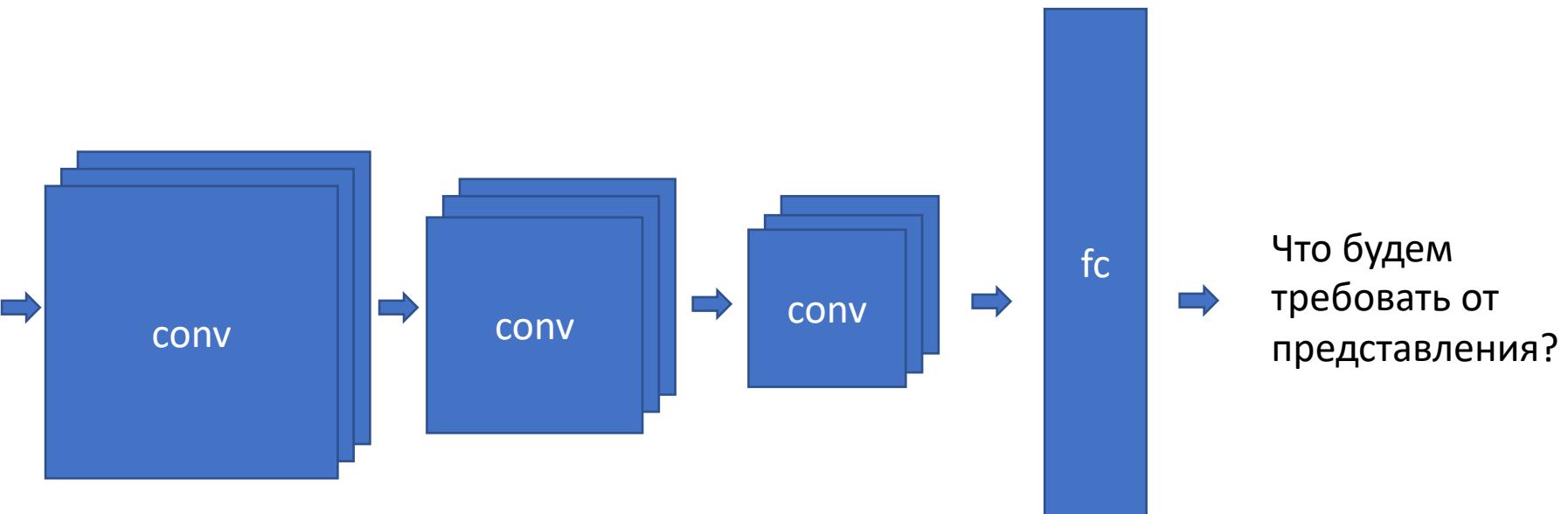
# Supervised embeddings



# Supervised embeddings

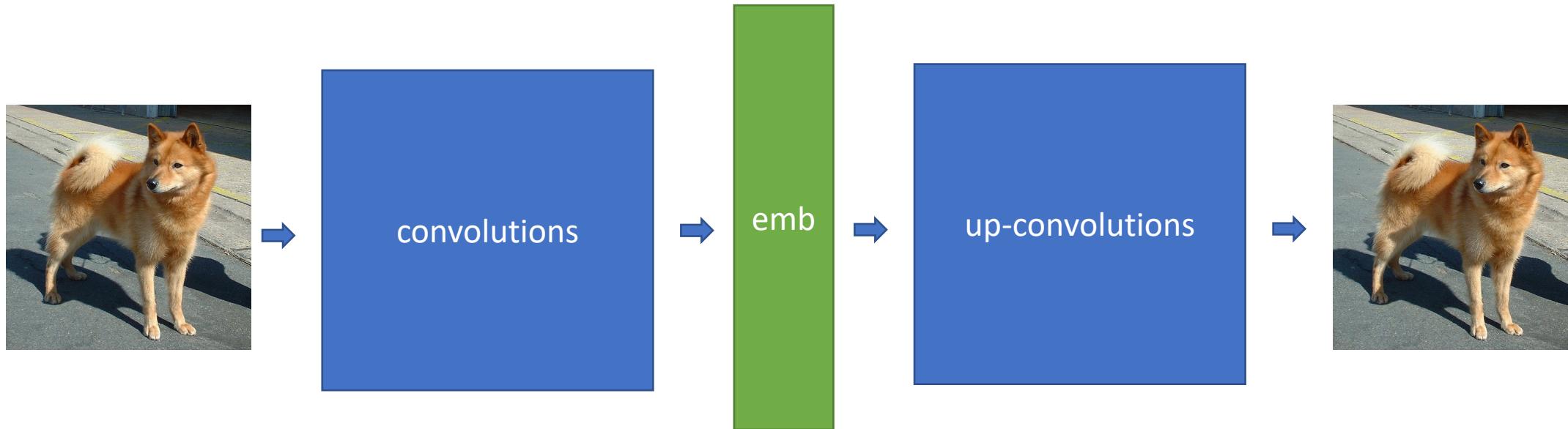


# Supervised embeddings



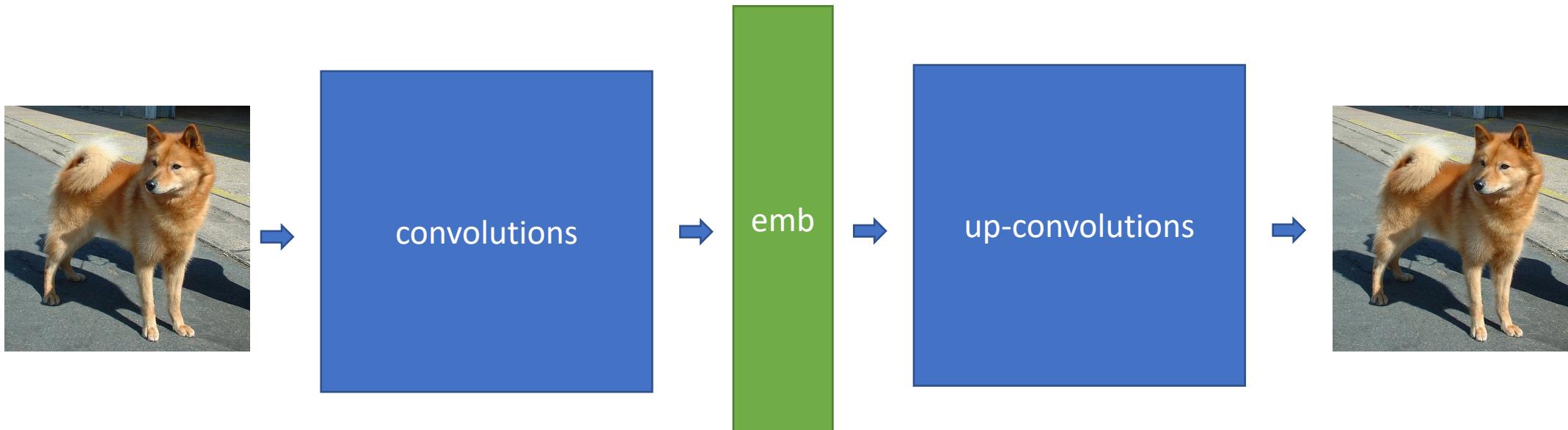
Что будем  
требовать от  
представления?

# Авто кодировщики



обойдёмся без полно связных слоёв

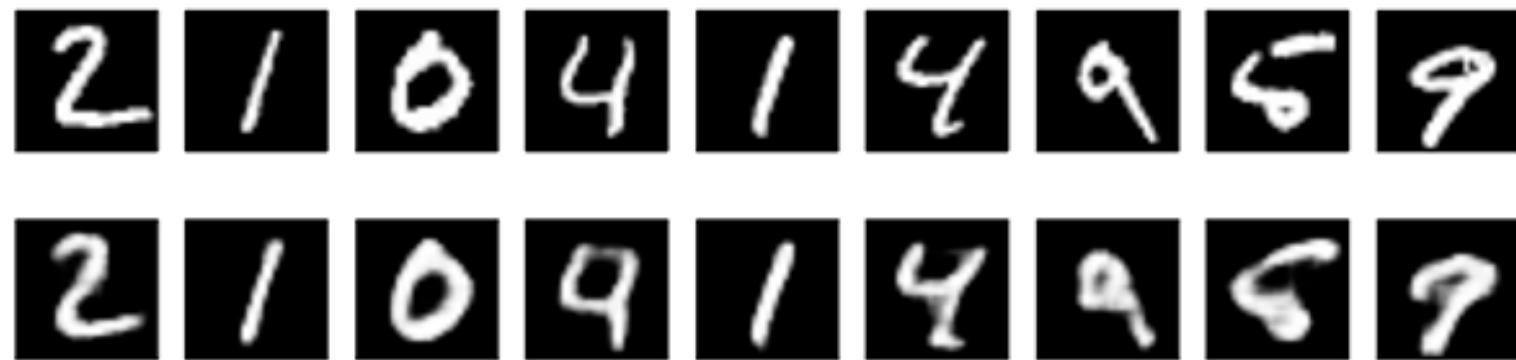
# Авто кодировщики



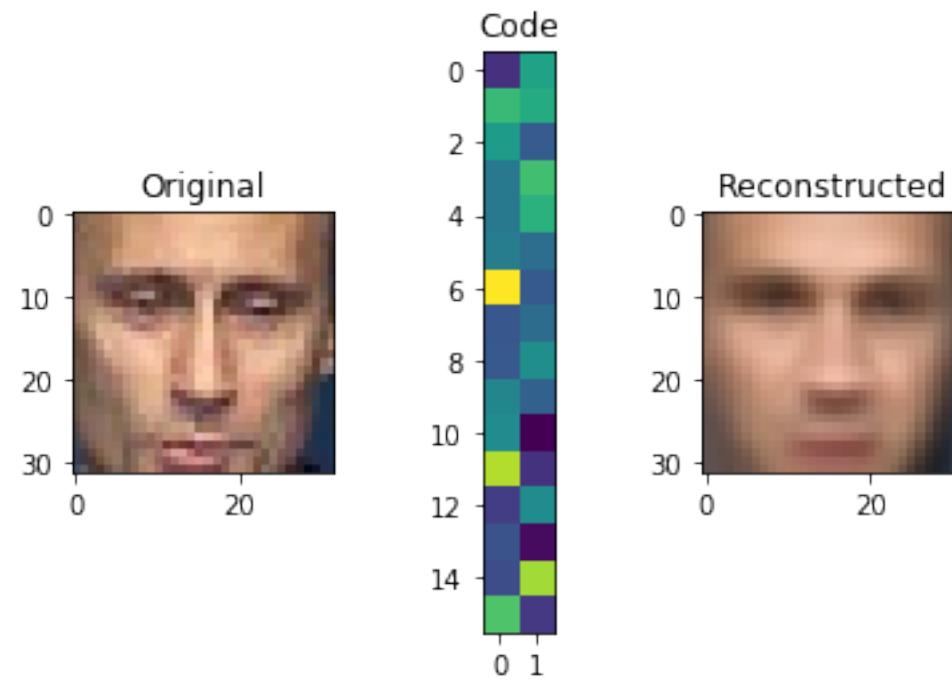
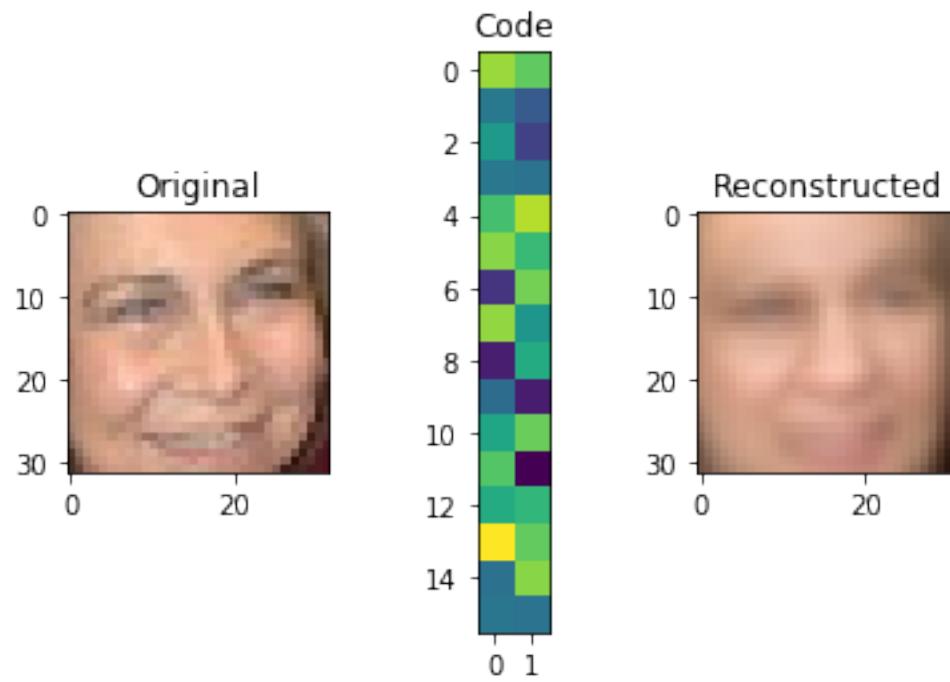
$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(x_i, g(f(x_i))) \rightarrow \min$$

$x_i$  — изображение  
 $f(x)$  — кодировщик (encoder)  
 $g(z)$  — декодировщик (decoder)  
 $L(x, \hat{x})$  — расстояние между изображениями (например, евклидово)

# Авто кодировщики



# Авто кодировщики



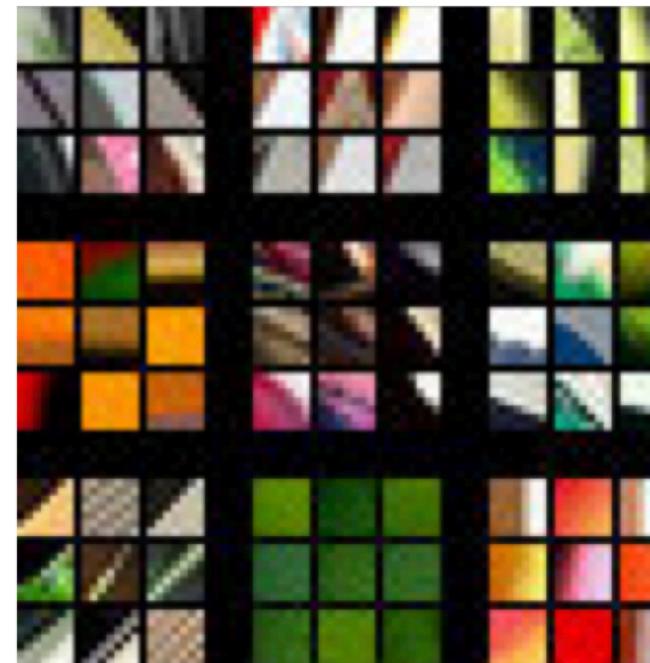
# Автокодировщики

- Восстанавливают изображения с потерями (но это логично)
- Но при этом переобучаются
- Нужно как-то регуляризовать

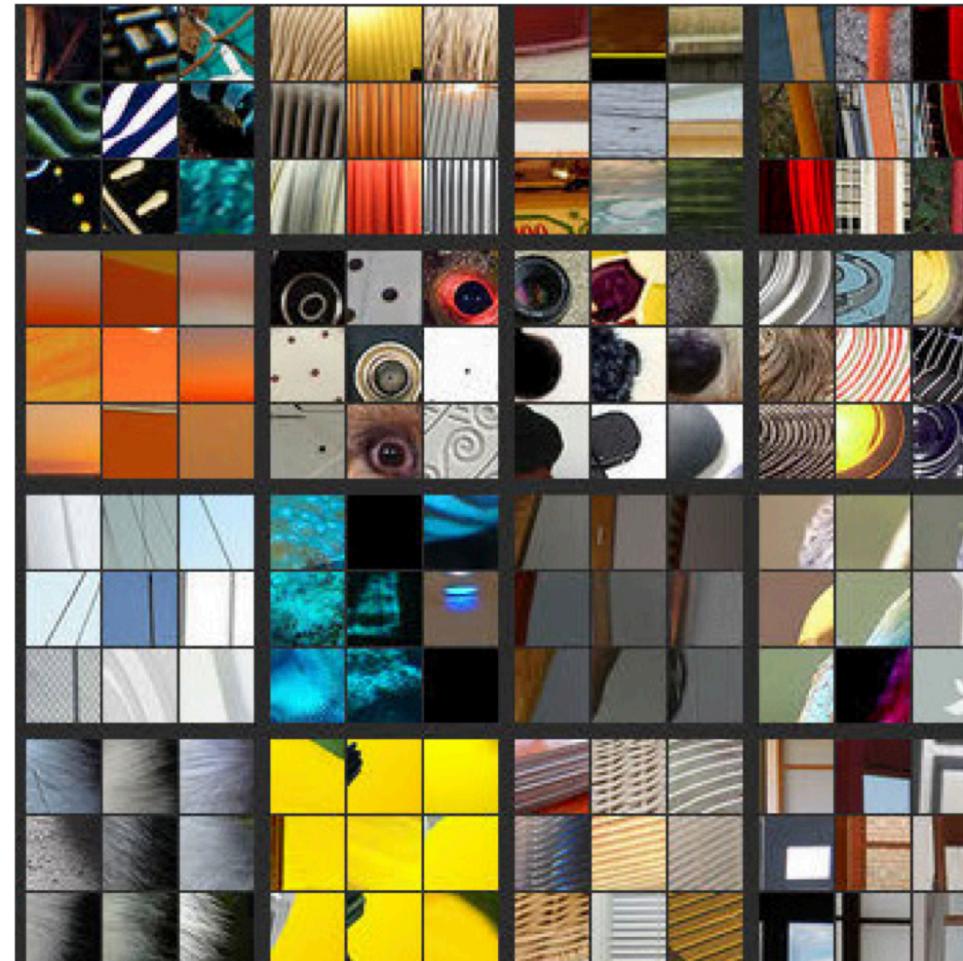
# Как ещё измерять сходство картинок?

- Нам важно, чтобы сохранялся смысл, а не в точности восстанавливались пиксели

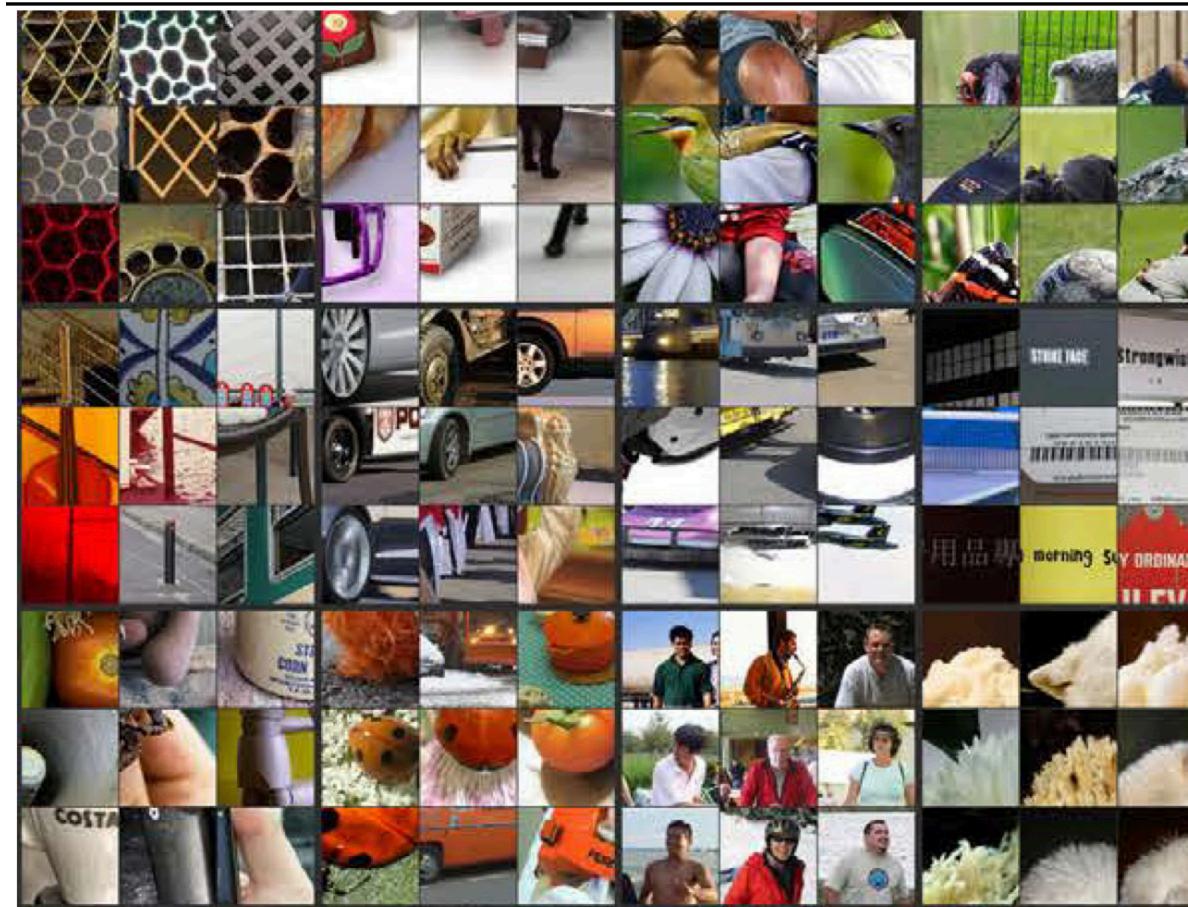
# Слой 1



# Слой 2



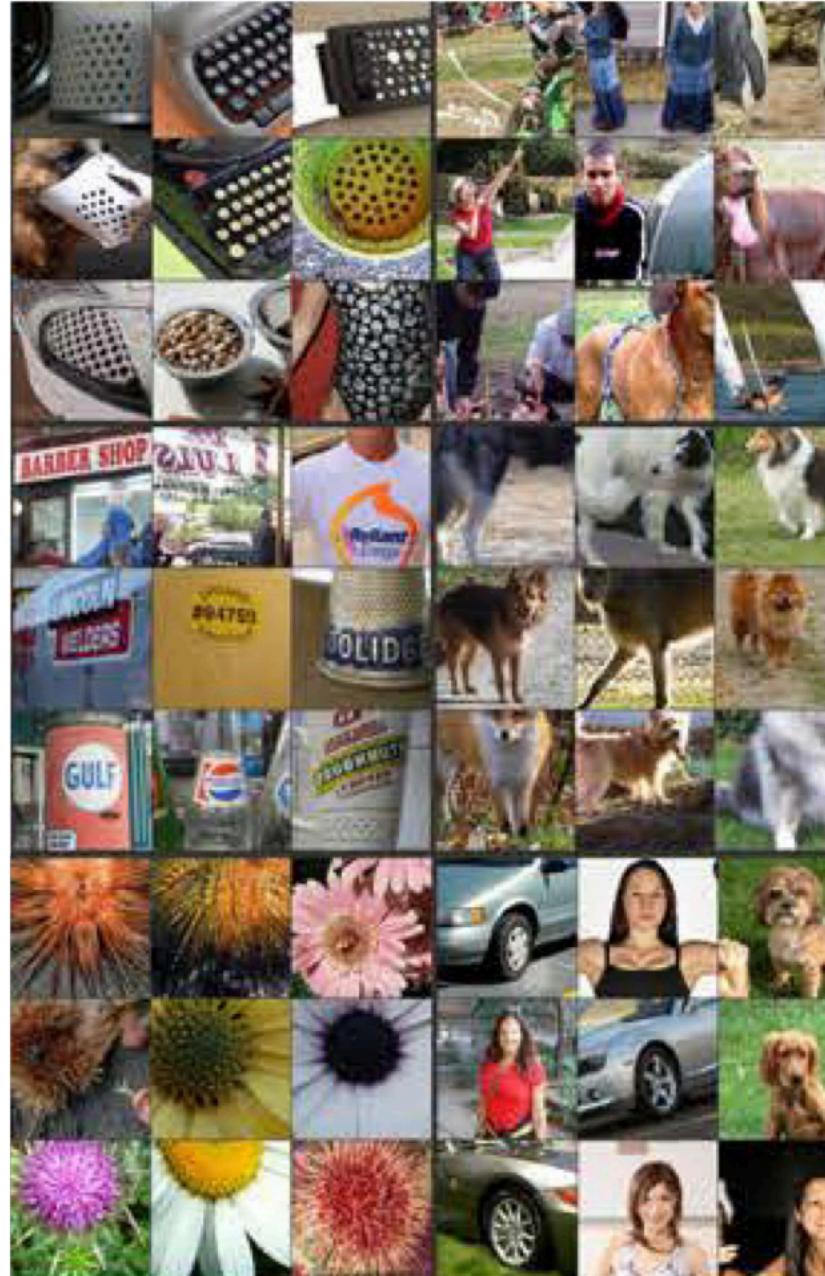
# Слой 3



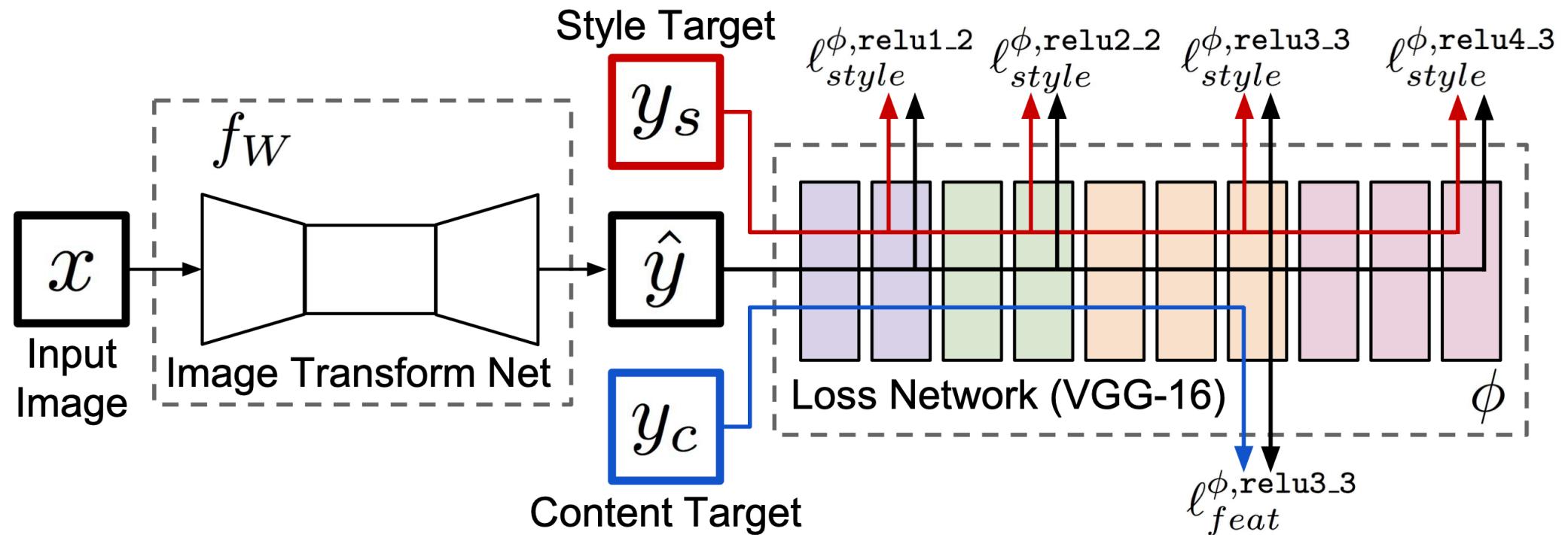
# Слой 4



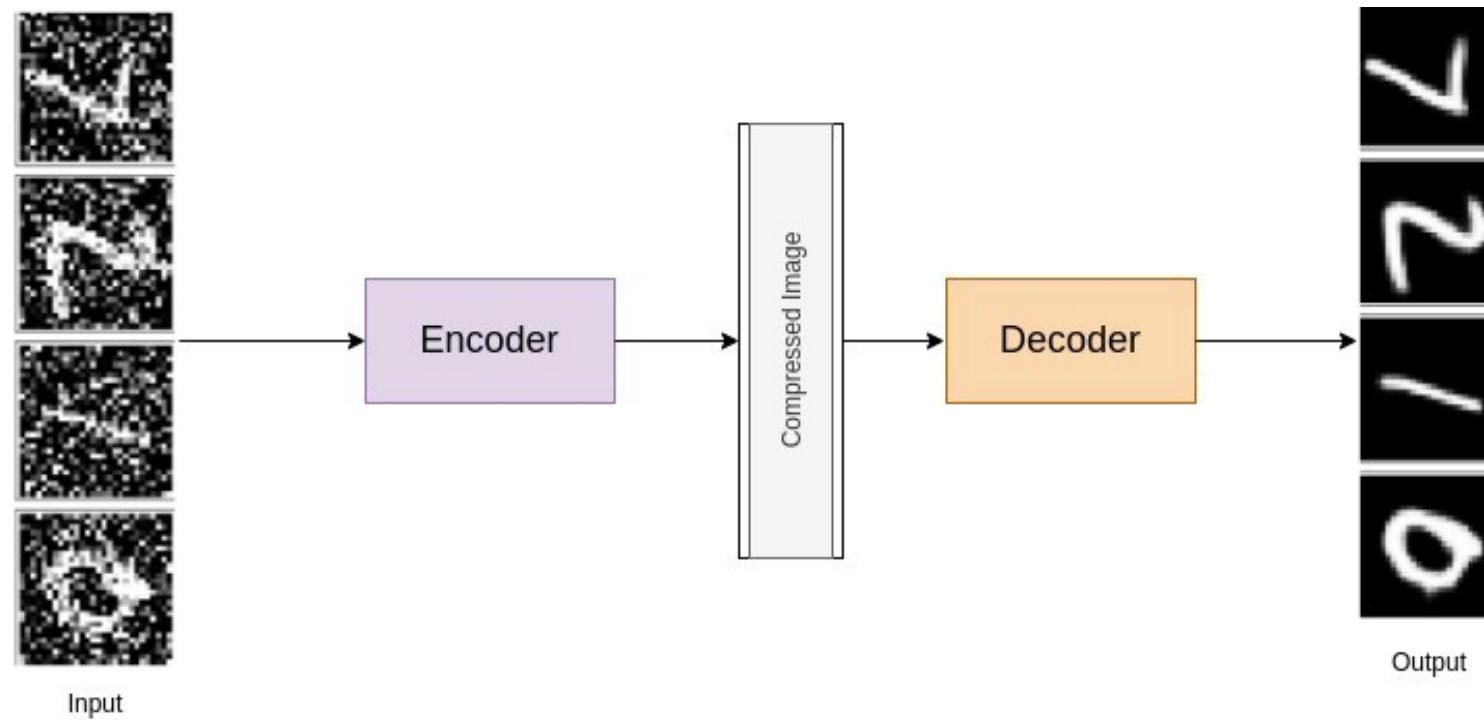
# Слой 5



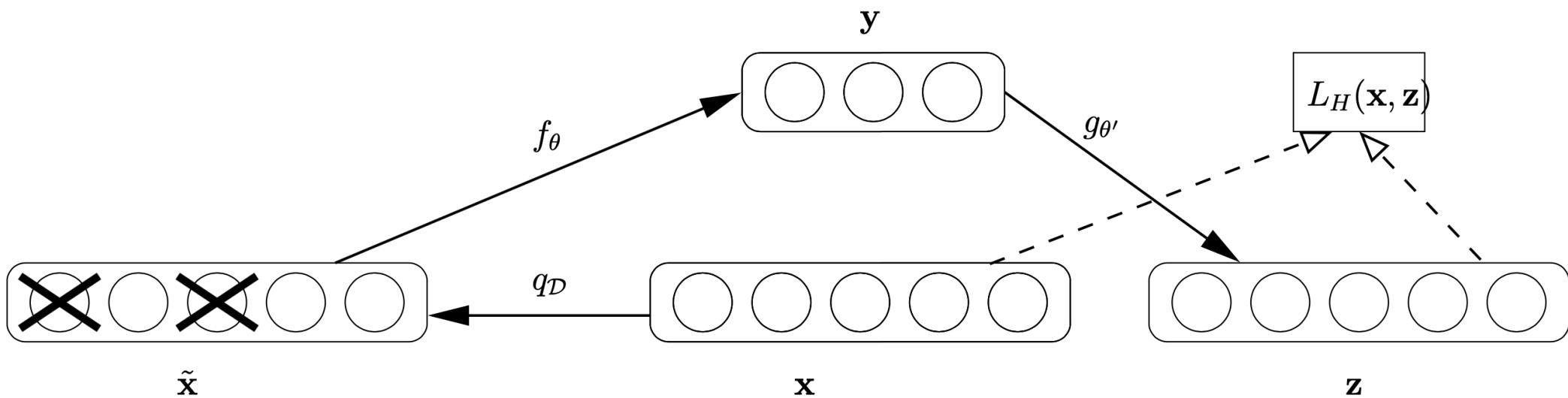
# Perceptual loss



# Denoising autoencoder



# Denoising autoencoder



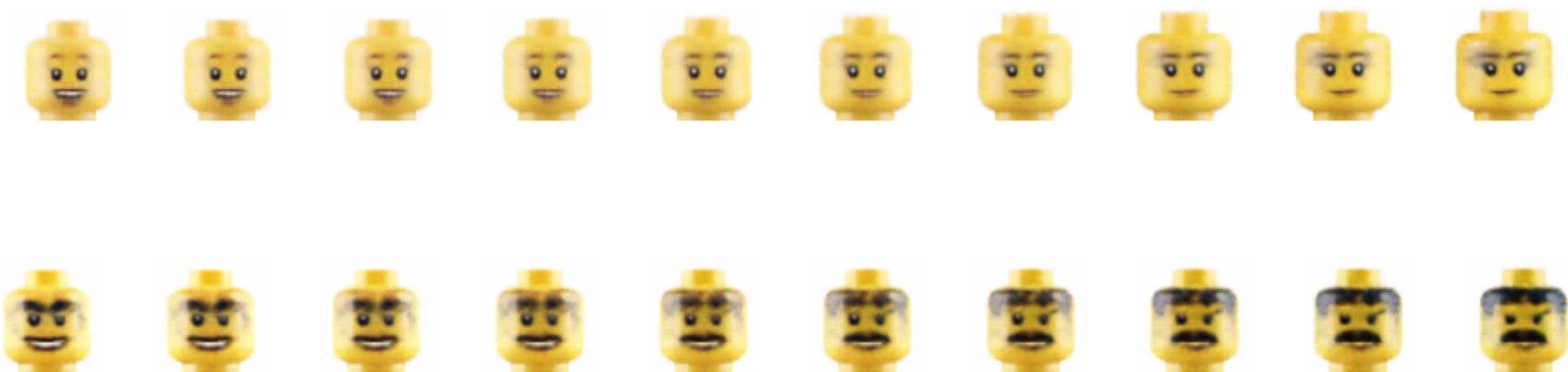
# Зачем это всё?

- Сжатие данных (нелинейных аналог PCA)
- Поиск похожих изображений
- Трансформация изображений
- Генерация изображений

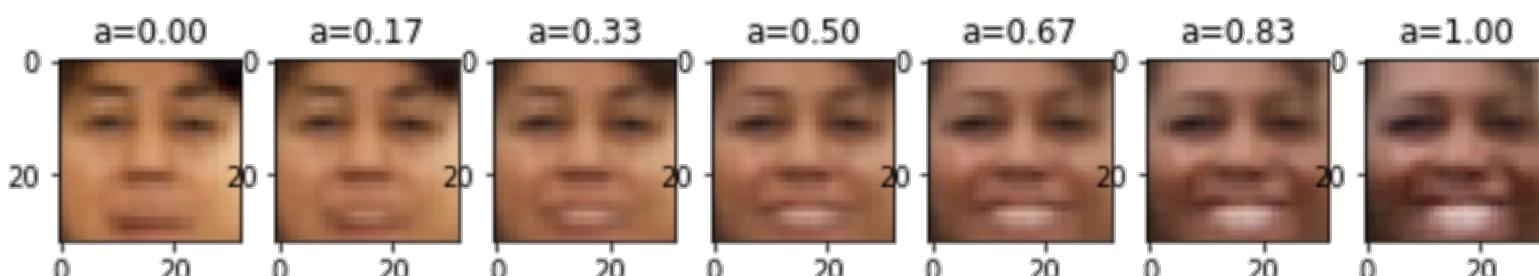
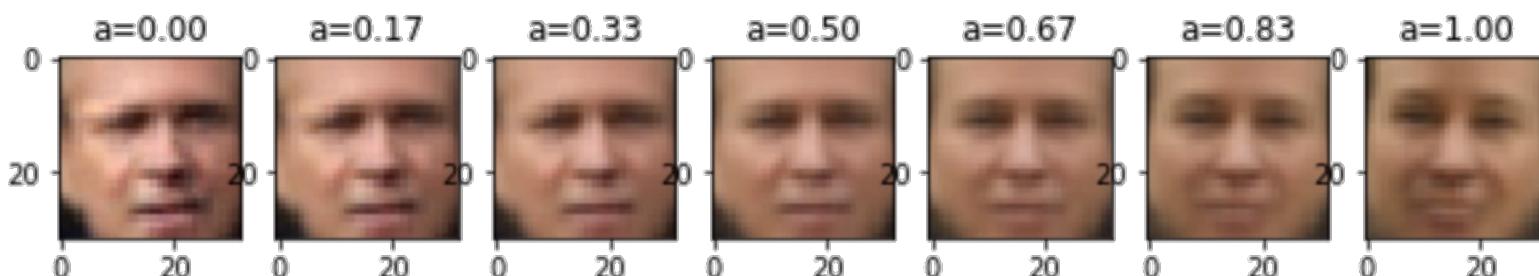
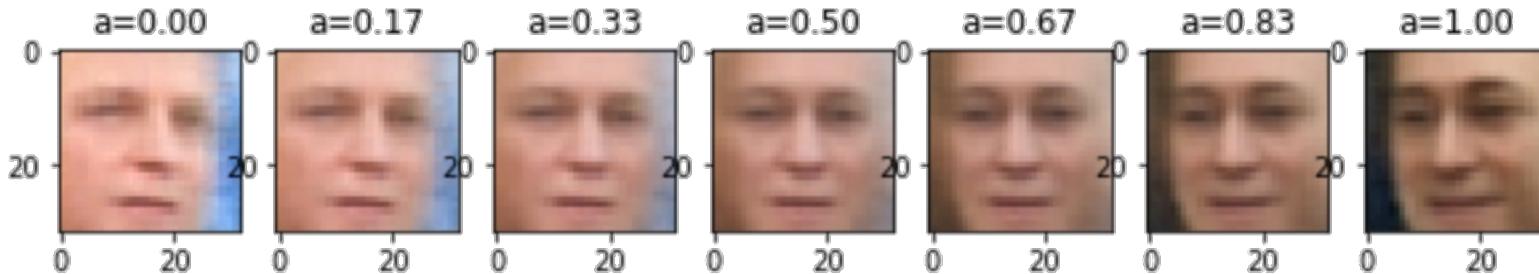
# Morphing faces



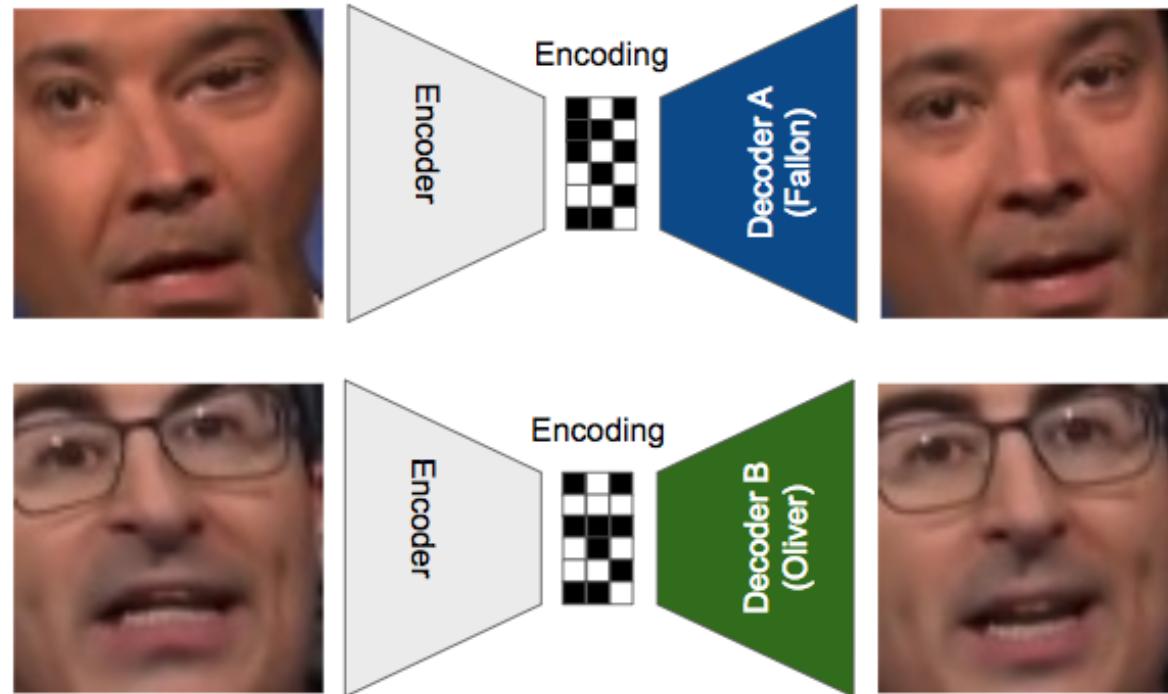
# Morphing faces



# Morping faces



# DeepFake



Важно: во время обучения изображения деформируются (warping)

# Ограничения



Модель хорошо преобразует только те ракурсы, которые были в выборке

# DeepFake



# DeepFake



# Представления для текстов

# Bag-of-words

- Заводим словарь, состоящий из всех слов в выборке
- Делаем признак-индикатор для каждого слова из словаря
- Можно добавлять n-граммы

# Bag-of-words

- Слишком много признаков
- Не учитываем смыслы слов
- Семантически похожие тексты могут иметь очень разные представления

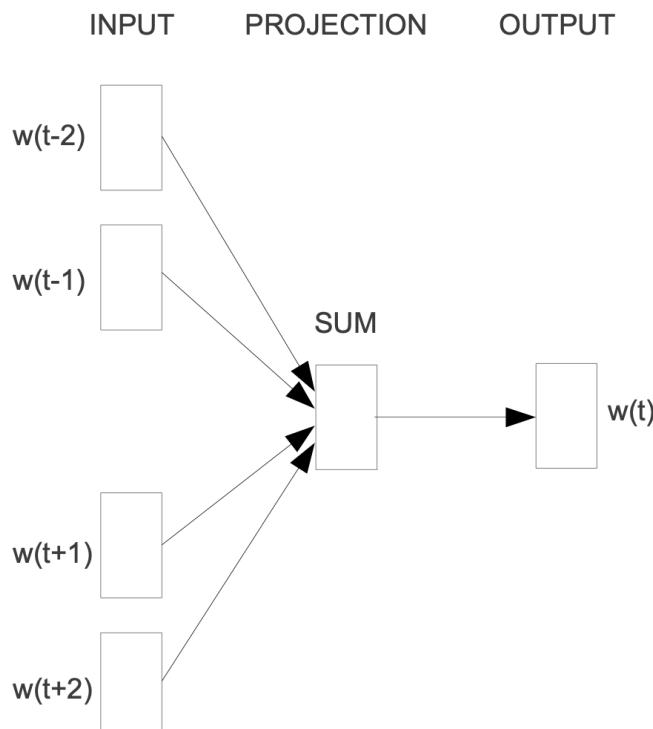
# word2vec

- Попробуем обучить вектор-представление для каждого слова
- Что потребовать от такого представления?

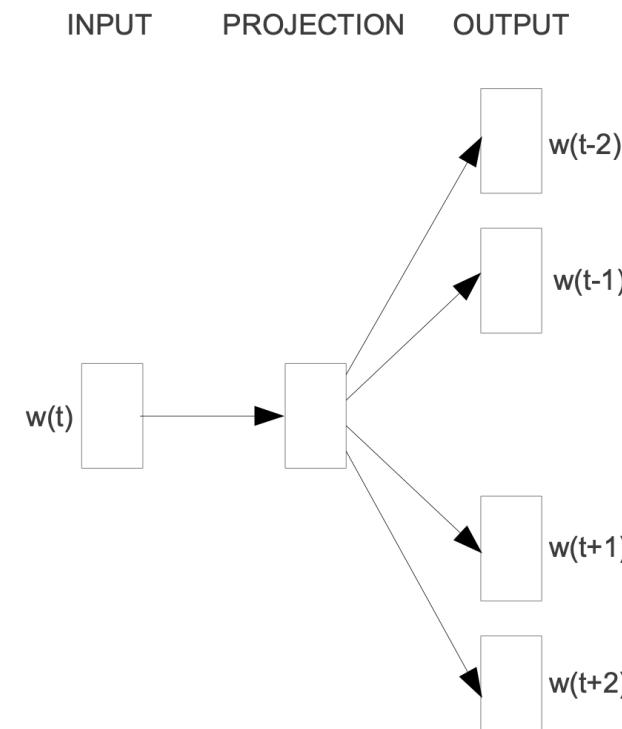
# word2vec

- Попробуем обучить вектор-представление для каждого слова
  - Что потребовать от такого представления?
- 
- Важная идея: если выкинуть слово, то оно должно хорошо восстанавливаться по представлениям соседних слов
  - Может применять и при работе с изображениями

# word2vec



**CBOW**



**Skip-gram**

# word2vec: особенности обучения

- Положительные примеры — слова, стоящие рядом
- Отрицательные примеры: подбираем к слову «шум», то есть другое слово, которое не находится рядом
- Важно семплировать в SGD слова с учётом их популярности — иначе будем обучаться только на самые частые слова

# Как это использовать?

- Можно искать похожие слова
- Можно менять формы слов
- Можно искать определённые отношения
- Можно использовать как признаки для моделей

# word2vec

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De