

# Few-shot Adversarial Learning of Realistic Neural Talking Head Models

Основная задача: по нескольким кадрам видео или набору фото этого человека, сгенерировать его изображение с заданным положением головы и выражением лица (в дальнейшем разметка).



Результат синтеза изображения головы зависят от разметки, взятого из таргет-кадра (не входит в тренировочный набор), а кадр-источник взят из тренировочного набора.

Слева: используя разметку, полученную из разных видео того же человека

Обучено восьмикадровым способом

Справа: используя разметку из видео другого человека

Обучено однокадровым способом

# Пути решения:

## 1. Деформация изображения (Warping method)

Минусы:

- Количество движений и повороты головы без видимых неполадок ограничены

## 2. Прямой синтез кадров (Direct (warping-free) method) использует сверточные сети.

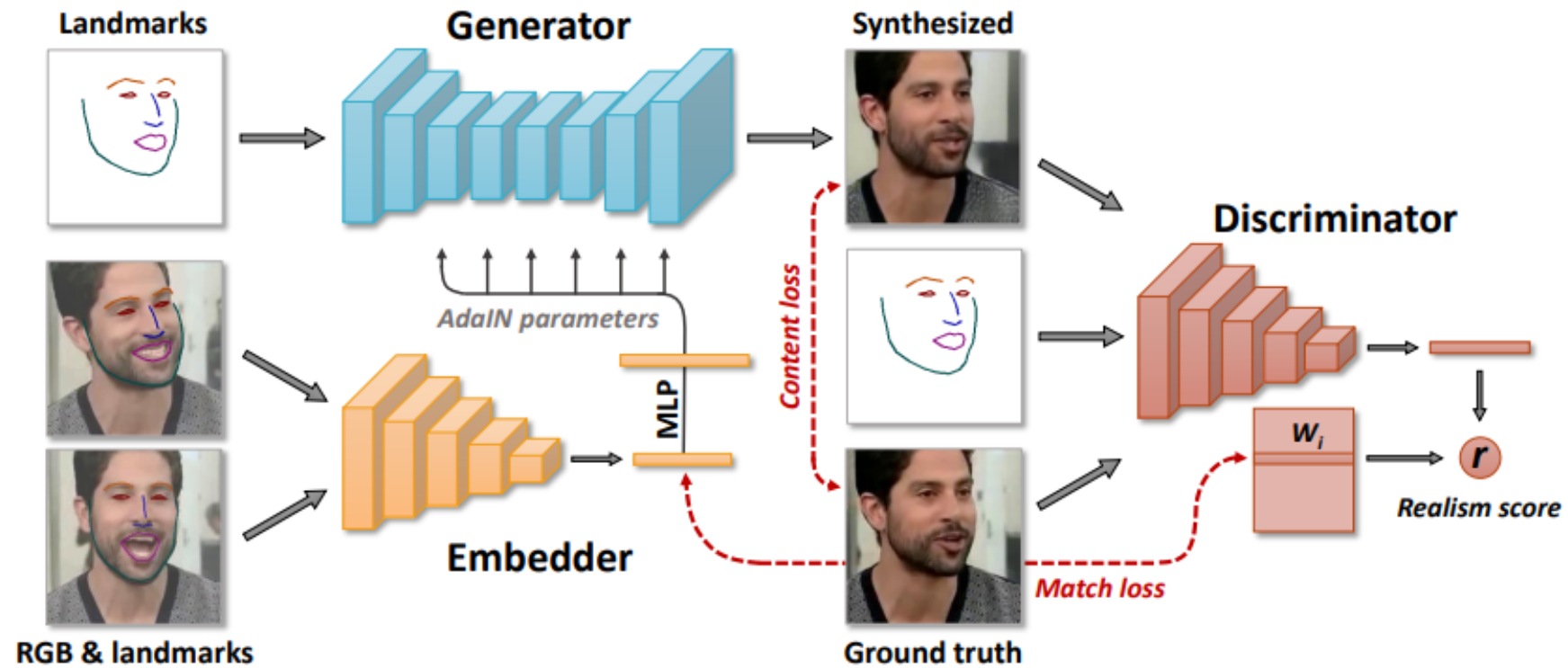
Минусы:

- Слишком большая сеть (десятки миллионов параметров в генераторе и дискриминаторе)
- Для хорошего результата требует продолжительное видео (несколько минут) или большое количество фотографий

# Few-shot learning

- Может сгенерировать нужное изображение по одному кадру или малому количеству
- Добавление кадров улучшает качество персонализации
- Работает за приемлемое время
- Поддерживает большое количество положений головы
- Работает в два этапа:
  1. Meta-learning: предобучение на большом количестве видео с разными людьми с разной внешностью (учим систему превращать landmark в приемлемое фотореалистичное изображение человека, на котором обучаемся)
  2. Fine-tuning stage: обучение на нескольких или одном кадре человека, фото которого мы должны сгенерировать, для лучшей персонализации картинки (иначе лицо будет не очень узнаваемым)

# Meta-learning stage



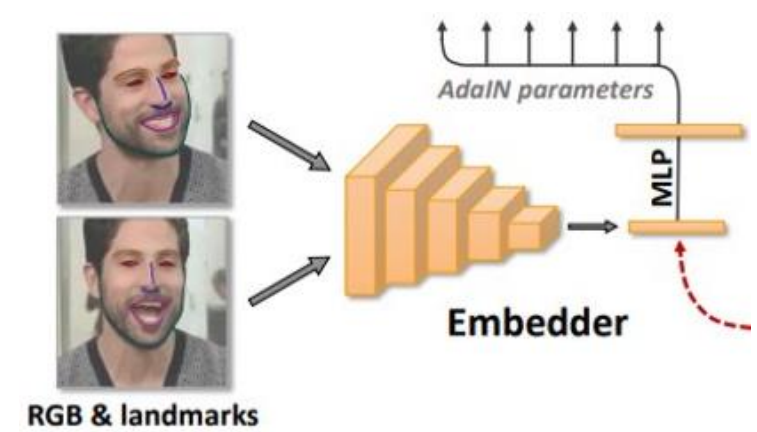
**Embedder**  $E(x_i(s), y_i(s); \phi) \rightarrow \hat{e}_i(s)$

$x_i(s)$  – кадр  $s$  из  $i$ -того видео

$y_i(s)$  – соответствующая разметка

$\phi$  – обучаемые на стадии meta-learning параметры нейросети

$\hat{e}_i(s)$  – вектор размерности  $N$ , содержащий информацию о позе и мимике человека в кадре



**Generator**  $G(y_i(t), \hat{e}_i; \psi, P) \rightarrow \hat{x}_i(t)$

$y_i(t)$  – разметка изображения, которая не проходила через embedder

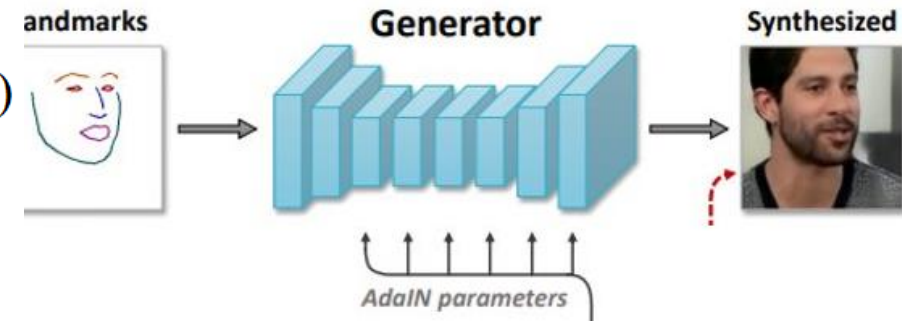
$\hat{e}_i$  – предсказанная embedder'ом сжатая информация (среднее  $\hat{e}_i(s)$  по  $s$ )

$\psi$  – параметры, отвечающие за общие качества человека

$\hat{\psi}_i$  – параметры, отвечающие за личные качества человека

$P : \hat{\psi}_i = P \hat{e}_i$

$\hat{x}_i(t)$  – сгенерированный видеокадр

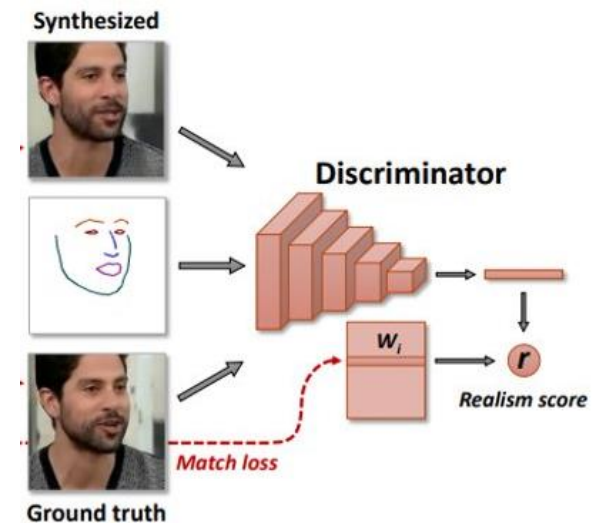


**Discriminator**  $D(x_i(t), y_i(t), i; \theta, \underset{\text{параметры сети}}{\mathbf{W}}, \mathbf{w}_0, b) = r =$

$$= V(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t); \theta)^T (\mathbf{W}_i + \mathbf{w}_0) + b$$

$V(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t); \theta)$  - ConvNet, которая превращает входные данные в вектор размерности  $N$

$r$  - realism score предсказывает, является ли  $\mathbf{x}_i(t)$  реальным кадром и соответствует ли он разметке из входных данных



# Процесс обучения

Embedder and generator losses

perceptual similarity measure  
between ground truth and  
reconstruction

L1-diff between weights of VGG-19 and VGGFace  
Trained for classification and ace recognition

$$\mathcal{L}(\phi, \psi, P, \theta, W, w_0, b) = \mathcal{L}_{\text{CNT}}(\phi, \psi, P) + \mathcal{L}_{\text{ADV}}(\phi, \psi, P, \theta, W, w_0, b) + \mathcal{L}_{\text{MCH}}(\phi, W)$$

the similarity of the two types of embeddings by penalizing  
the  $L_1$ -difference between  $E(\mathbf{x}_i(s_k), \mathbf{y}_i(s_k); \phi)$  and  $\mathbf{W}_i$ .

$$\mathcal{L}_{\text{ADV}}(\phi, \psi, P, \theta, W, w_0, b) = -r + \mathcal{L}_{\text{FM}} =$$

$$- D(\hat{x}_i(t), y_i(t), i; \theta, W, w_0, b) + \mathcal{L}_{\text{FM}}$$

a feature matching term , which es-  
sentially is a perceptual similarity measure, computed using  
discriminator (it helps with the stability of the training):

Discriminator loss

$$\mathcal{L}_{\text{DSC}}(\phi, \psi, P, \theta, W, w_0, b) =$$

$$+ \max(0, 1 + D(\hat{x}_i(t), y_i(t), i; \phi, \psi, \theta, W, w_0, b))$$

$$+ \max(0, 1 - D(x_i(t), y_i(t), i; \theta, W, w_0, b))$$



# Зачем нужна стадия fine-tuning?

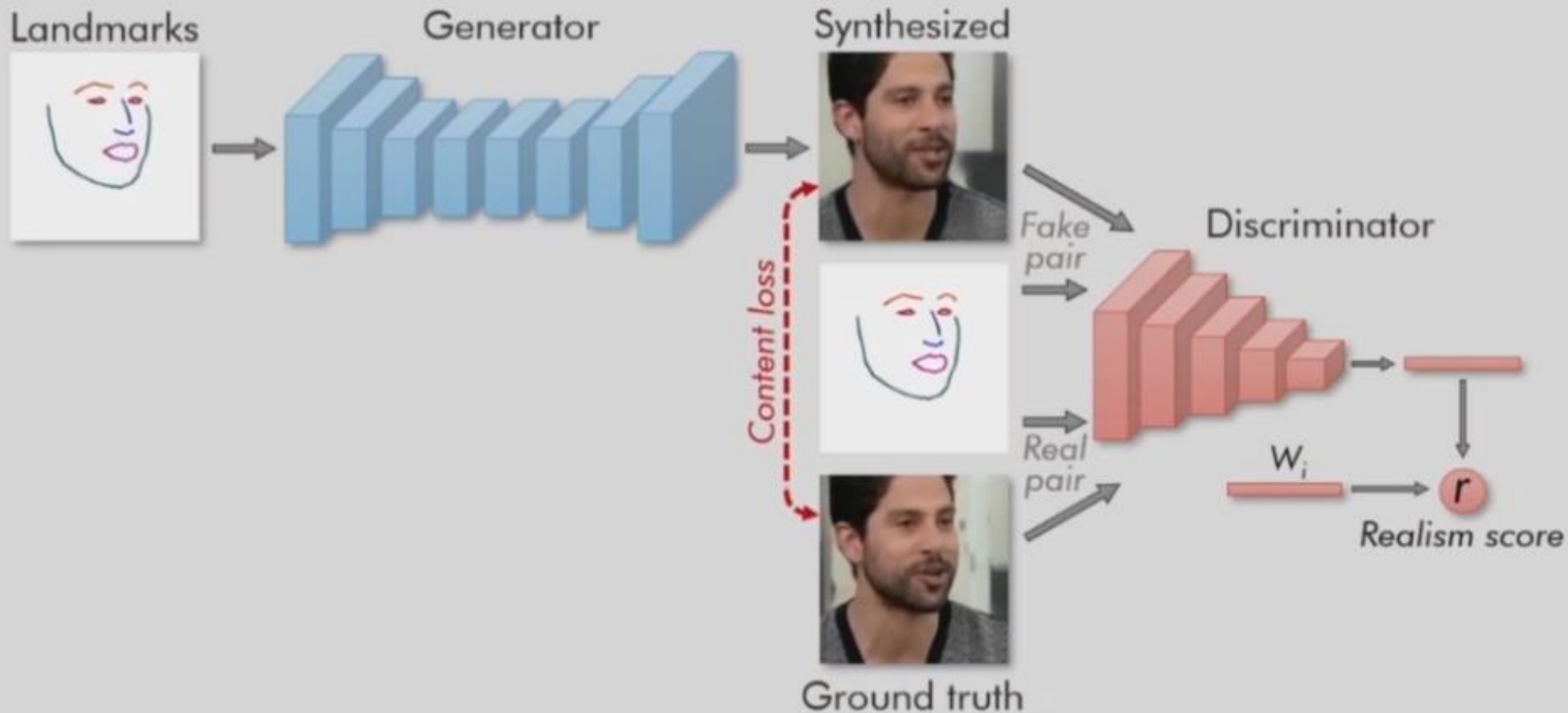
$$\hat{\mathbf{e}}_{\text{NEW}} = \frac{1}{T} \sum_{t=1}^T E(\mathbf{x}(t), \mathbf{y}(t); \phi)$$

Таким образом, чтобы получить лицо нужного нам человека с нужным выражением лица мы можем взять видео с ним и его разметку, прогнать через обученный на стадии meta-learning эмбеддер и вставить его в так же обученный генератор. Картинка получается качественной и реалистичной, НО в таком случае возникает проблема.

На сгенерированной картинке утеряна персонализация человека (иными словами видны черты другого человека, а не таргета)




# Few-shot fine-tuning



# Few-shot fine-tuning

На этом этапе берем ту же сеть, что и на прошлом, исключаем из обучения эмбеддер, меняем генератор, дискриминатор и лосс-функции

Generator:  $G'(y(t); \phi, \psi')$ , где  $\psi' = P\hat{e}_{\text{NEW}}$   т.к. эмбеддер уже предобучен

Discriminator:  $D'(x(t), y(t); \theta, w', b)$ , где  $w' = w_0 + \hat{e}_{\text{NEW}}$   из-за  $\mathcal{L}_{\text{MCH}}$  на прошлом этапе

$$D'(\hat{x}(t), y(t); \theta, w', b) = V(\hat{x}(t), y(t), \theta)^T w' + b$$

$$\mathcal{L}'(\psi, \psi', \theta, w', b) = \mathcal{L}'_{\text{CNT}}(\psi, \psi') + \mathcal{L}'_{\text{ADV}}(\psi, \psi', \theta, w', b)$$

$$\mathcal{L}'_{\text{DSC}}(\psi, \psi', \theta, w', b) = \max(0, 1 + D(\hat{\mathbf{x}}(t), \mathbf{y}(t); \psi, \psi', \theta, w', b)) + \max(0, 1 - D(\mathbf{x}(t), \mathbf{y}(t); \theta, w', b))$$

# Сравнение

Frechet-inception distance (FID), mostly measuring perceptual realism

Structured similarity (SSIM), measuring low-level similarity to the ground truth images

cosine similarity (CSIM) between embedding vectors of the state-of-the-art face recognition network for measuring identity mismatch

Method (T)	FID↓	SSIM↑	CSIM↑	USER↓
VoxCeleb1				
X2Face (1)	45.8	<b>0.68</b>	<b>0.16</b>	0.82
Pix2pixHD (1)	<b>42.7</b>	0.56	0.09	0.82
Ours (1)	43.0	0.67	0.15	<b>0.62</b>
X2Face (8)	51.5	<b>0.73</b>	<b>0.17</b>	0.83
Pix2pixHD (8)	<b>35.1</b>	0.64	0.12	0.79
Ours (8)	38.0	0.71	<b>0.17</b>	<b>0.62</b>
X2Face (32)	56.5	<b>0.75</b>	0.18	0.85
Pix2pixHD (32)	<b>24.0</b>	0.70	0.16	0.71
Ours (32)	29.5	0.74	<b>0.19</b>	<b>0.61</b>

VoxCeleb2				
Ours-FF (1)	<b>46.1</b>	0.61	<b>0.42</b>	<b>0.43</b>
Ours-FT (1)	48.5	<b>0.64</b>	0.35	0.46
Ours-FF (8)	42.2	0.64	<b>0.47</b>	0.40
Ours-FT (8)	<b>42.2</b>	<b>0.68</b>	0.42	<b>0.39</b>
Ours-FF (32)	40.4	0.65	<b>0.48</b>	0.38
Ours-FT (32)	<b>30.6</b>	<b>0.72</b>	0.45	<b>0.33</b>

Method (T)	Time, s
Few-shot learning	
X2Face (1)	0.236
Pix2pixHD (1)	33.92
Ours (1)	43.84
Ours-FF (1)	<b>0.061</b>
X2Face (8)	1.176
Pix2pixHD (8)	52.40
Ours (8)	85.48
Ours-FF (8)	<b>0.138</b>
X2Face (32)	7.542
Pix2pixHD (32)	122.6
Ours (32)	258.0
Ours-FF (32)	<b>0.221</b>

Inference	
X2Face	0.110
Pix2pixHD	<b>0.034</b>
Ours	0.139



- The FT variant is trained for half as much (75 epochs) but with  $L_{MCH}$ , which allows fine-tuning
- FF (feed-forward) variant is trained for 150 epochs without the embedding matching loss  $L_{MCH}$  and, therefore, we only use it without fine-tuning (by simply predicting adaptive parameters  $\psi_0$  via the projection of the embedding  $e^{NEW}$ ).



# VoxCeleb2 dataset training.

Увеличенный размер картинок. Сравнение между собой.



One-shot models.  
Puppeteering.

