



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

ПРИМЕНЕНИЕ АВТОЭНКОДЕРОВ И ГЕНЕРАТИВНО- СОСТАЗАТЕЛЬНЫХ СЕТЕЙ В ЗАДАЧЕ ЗАМЕНЫ ЧЕЛОВЕЧЕСКОГО ЛИЦА В ВИДЕО

Программный проект. Выполнен студентами 191 группы ПМИ ФКН
Даниилом Волгиным и Иваном Фридманом
Научный руководитель Симагин Денис

Москва, 2020



Deepfake - реалистичная манипуляция аудиофайлами или видеоматериалами с помощью искусственного интеллекта



Основные термины:

1. Нейронная сеть — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.



Основные термины:

1. Нейронная сеть — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.
2. Генеративно-состязательная сеть — алгоритм машинного обучения без учителя, построенный на комбинации из двух нейронных сетей, одна из которых (сеть G) генерирует образцы а другая (сеть D) старается отличить правильные («подлинные») образцы от неправильных. Так как сети G и D имеют противоположные цели — создать образцы и отбраковать образцы — между ними возникает Антагонистическая игра.



Основные термины:

3. Свёрточная нейронная сеть — специальная архитектура искусственных нейронных сетей, нацеленная на эффективное распознавание образов, входит в состав технологий глубокого обучения. Идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв (англ. convolution layers) и субдискретизирующих слоёв (англ. subsampling layers или англ. pooling layers, слоёв подвыборки).



Основные термины:

3. Свёрточная нейронная сеть — специальная архитектура искусственных нейронных сетей, нацеленная на эффективное распознавание образов, входит в состав технологий глубокого обучения. Идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв (англ. convolution layers) и субдискретизирующих слоёв (англ. subsampling layers или англ. pooling layers, слоёв подвыборки).
4. Глубокое обучение — совокупность методов машинного обучения (с учителем, с частичным привлечением учителя, без учителя, с подкреплением), основанных на обучении представлениям (англ. feature/representation learning), а не специализированным алгоритмам под конкретные задачи.



Цель:

Основной целью нашего проекта было создание программы, решающей задачу замены человеческого лица на видео, специализированной под видео определенного формата (формата нашего датасета).

Задачи:

1. Ознакомление с принципами, терминами и алгоритмами, необходимыми для понимания научных статей по глубокому обучению. Этот этап включал в себя выполнение обучающих заданий. Одним из таких заданий была классификация иероглифов.
2. Изучение существующих методов решения задачи замены человеческого лица в видео.
3. Выбор одного из методов и его реализация.
4. Сбор собственного датасета для тестирования алгоритма.
5. Улучшение результатов работы алгоритма, используя особенности собранного нами датасета. Анализ слабых сторон выбранного алгоритма и минимизация потери качества полученного видео.



Описание аналогов

1. Faceswap. Бесплатное и открытое кроссплатформенное программное обеспечение.
Используются две пары энкодер-декодер (технология автоэнкодинга).

2. Faceswap-GAN. Продолжение предыдущего решения. К автоэнкодером добавлены технологии: генеративно-состязательные сети и архитектура VGGFace.

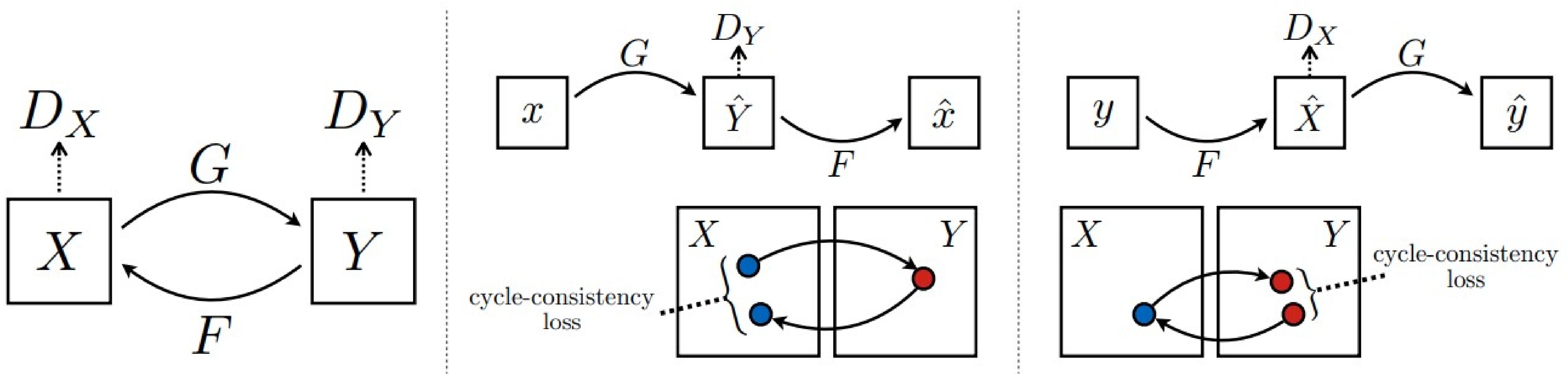
3. Zao – простое в использовании приложение для создания дипфейков. Позволяет получить результат за считанные секунды, однако возможностей меньше, чем у DeepFaceLab.

4. DeepFaceLab. Основано на проекте Faceswap, но использованы новые архитектуры.
Добавлены новые методы обнаружения и извлечения лиц.

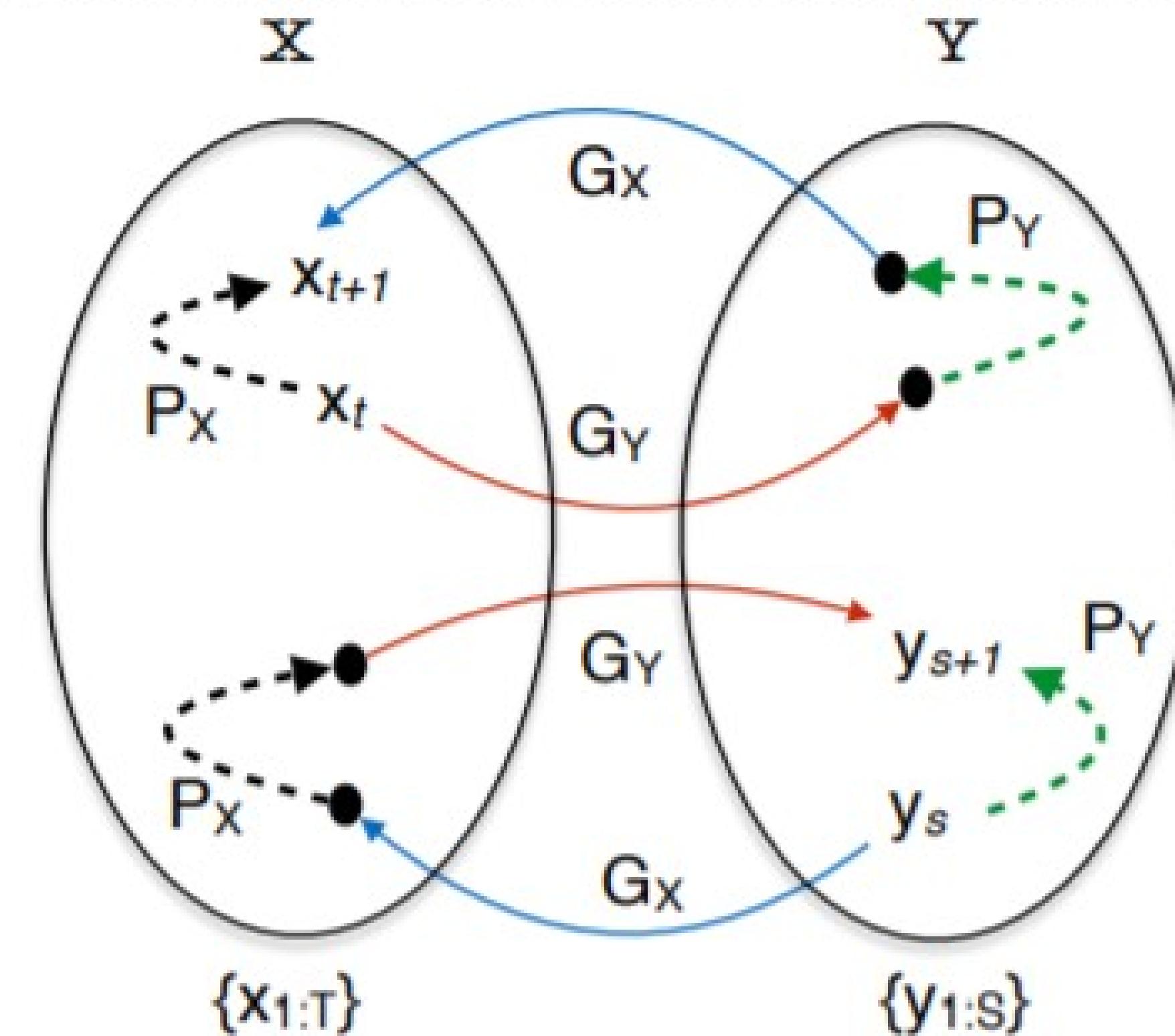


Алгоритм работы программы. Принцип работы CycleGAN

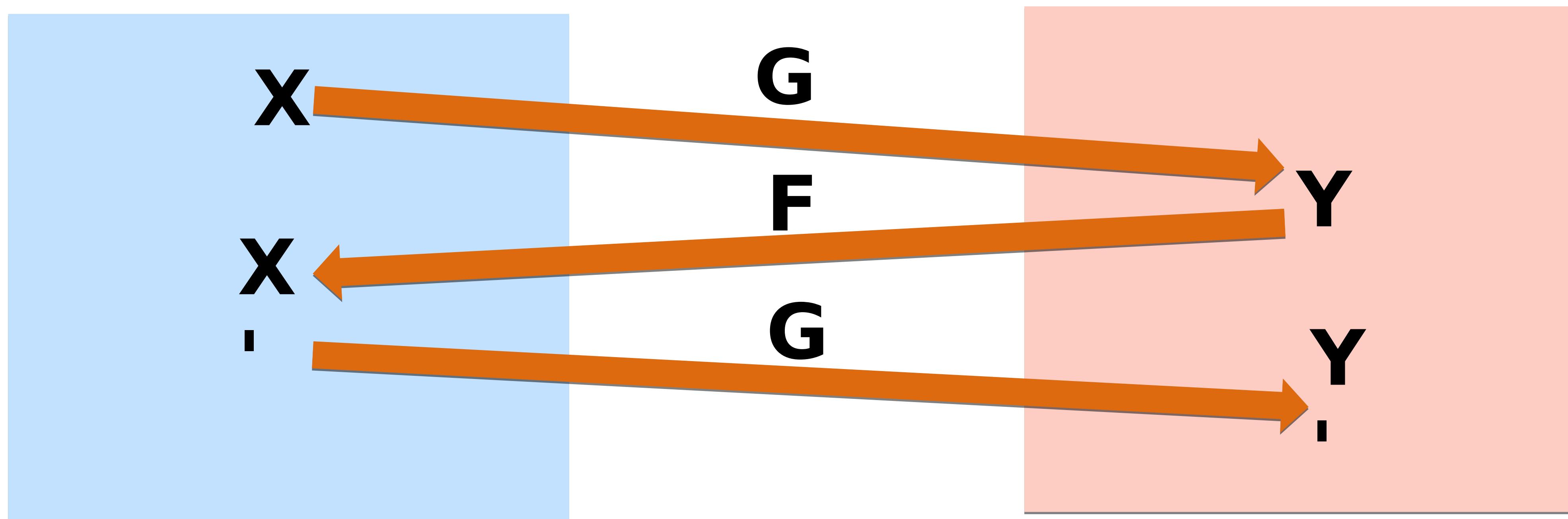
1. Два генератора и два дискриминатора
2. Прямая cycle-consistency ошибка
3. Обратная cycle-consistency ошибки



Алгоритм работы программы. Принцип работы RecycleGAN



Алгоритм работы программы. Обобщение идеи CycleGAN





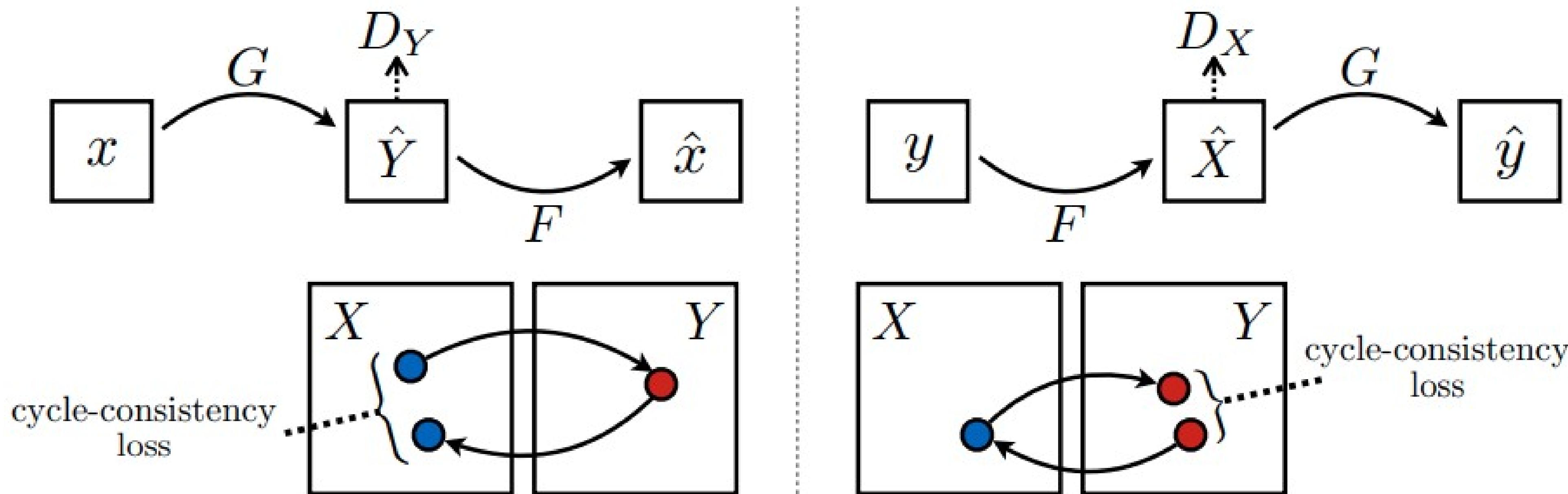
Face Parsing. BiSeNet



Сторонняя предобученная нейросеть BiSeNet для получения маски частей лица

Алгоритм работы программы. Attention Loss

Аналогично cycle-consistency ошибке считаем Average L1, но по выбранным частям лица





Dropout

Трансформируем изображение после прямого преобразования, чтобы усложнить задачу генератору обратного преобразования.

Таким образом генераторам будет сложнее построить общую стратегию кодирования/декодирования.

Применяемая трансформация - зануляем какую-то часть пикселей.





Конечная формула ошибки. Базовые функции ошибки

$$\text{CycleLoss}_A(X) = \text{L}_1(X; \text{G}_{B \rightarrow A}(\text{G}_{A \rightarrow B}(X)))$$

$$\text{GANLoss}_A(X) = \text{L}_1(\text{D}_A(X); \text{I}[X \in A])$$

$$\text{AttentionLoss}_A(X) = \text{CycleLoss}_A[\text{mask}](X)$$

$$\text{MultiCycleLoss}_A(X) = \text{CycleLoss}_B(\text{G}_{A \rightarrow B}(X))$$

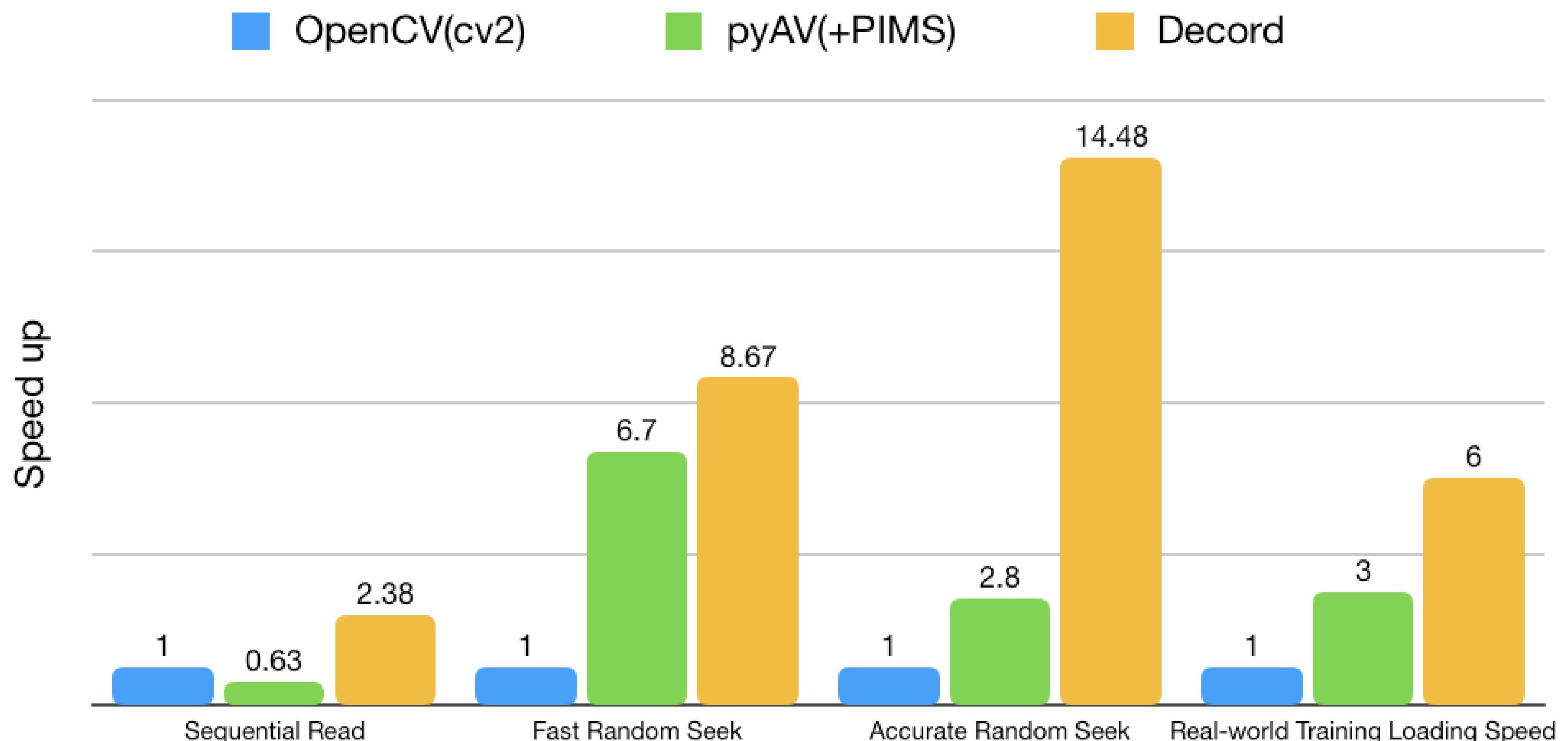


Конечная формула ошибки. Полная функция ошибки

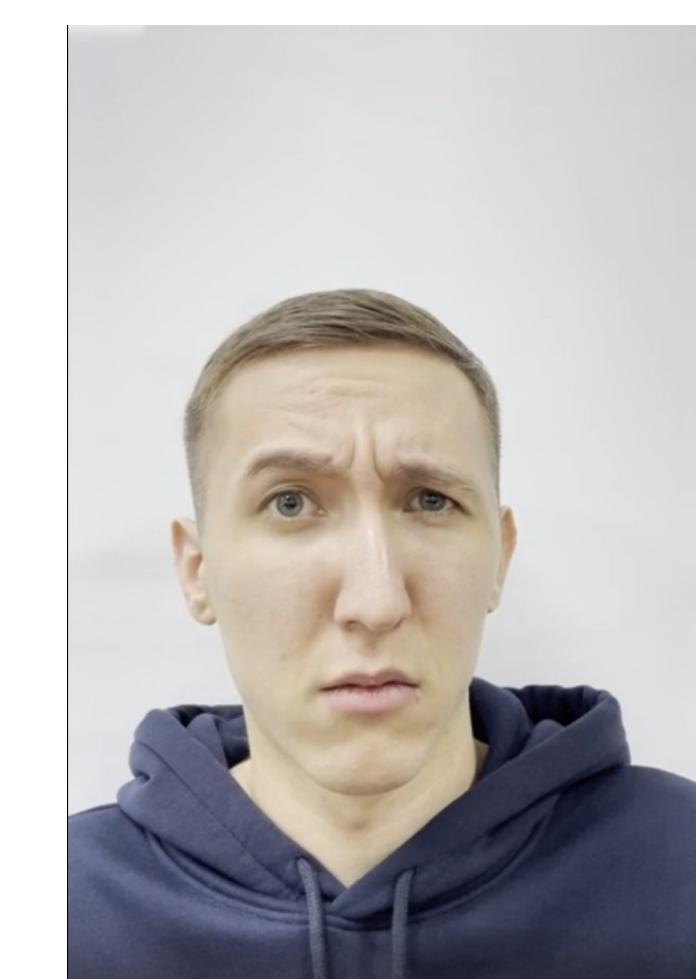
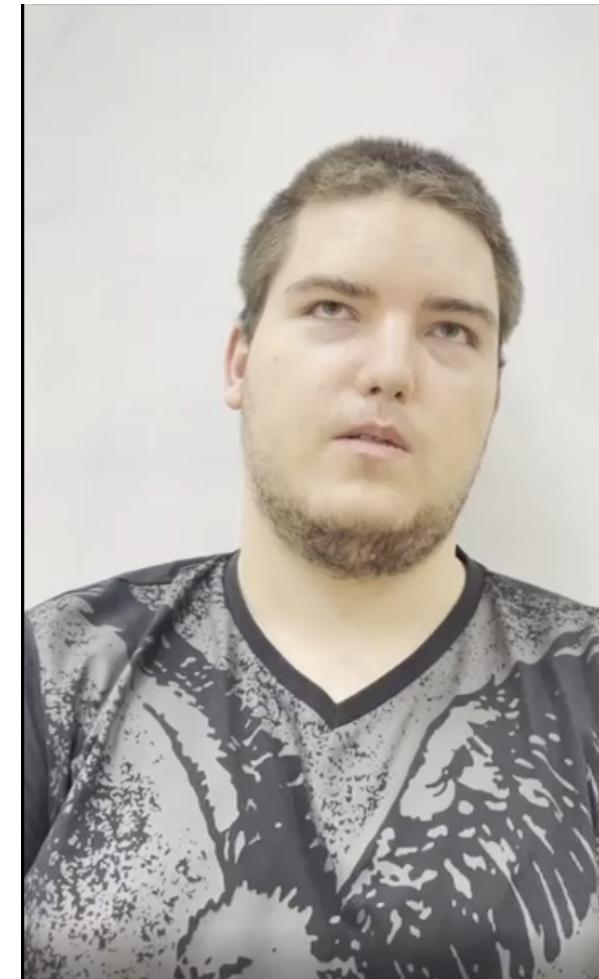
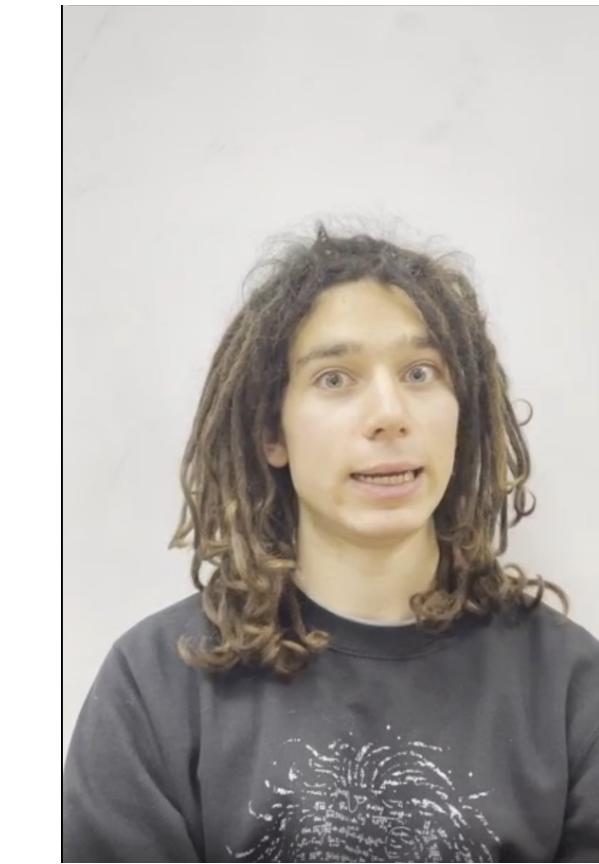
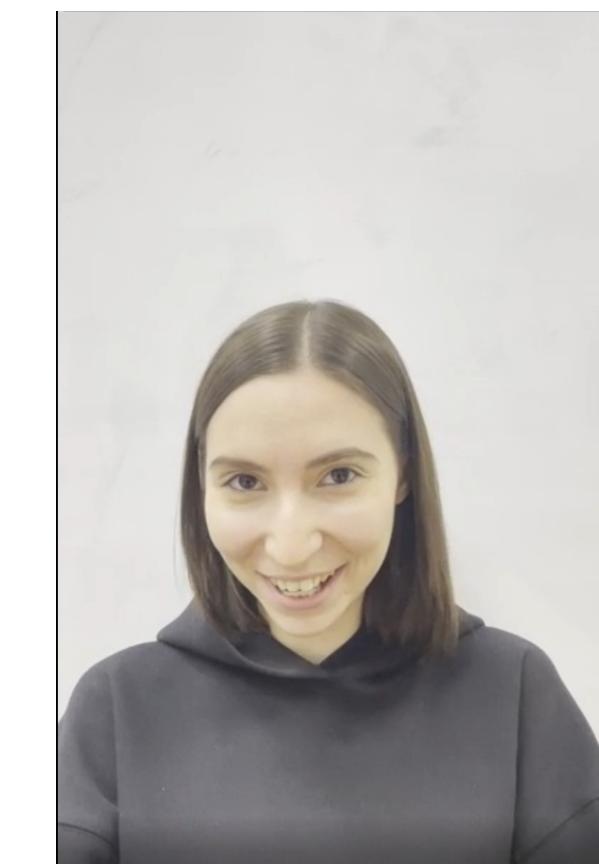
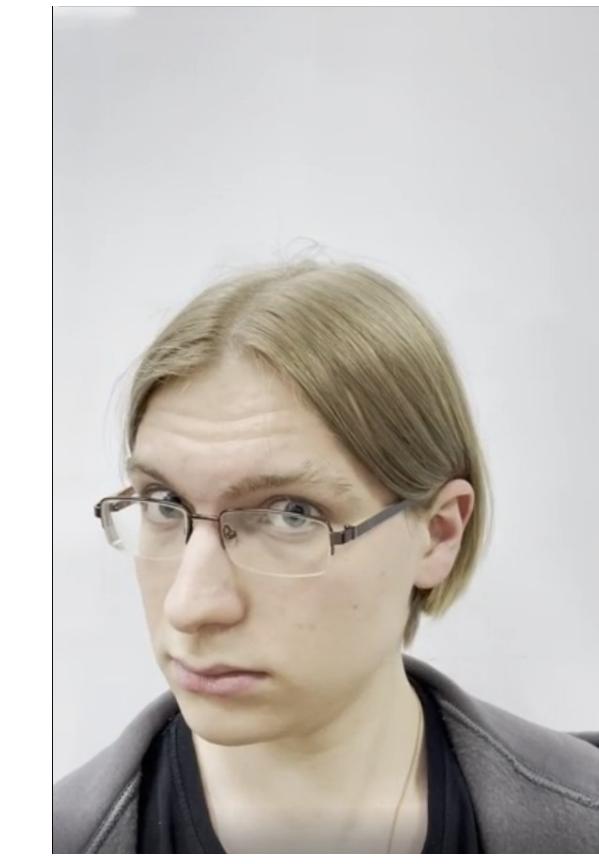
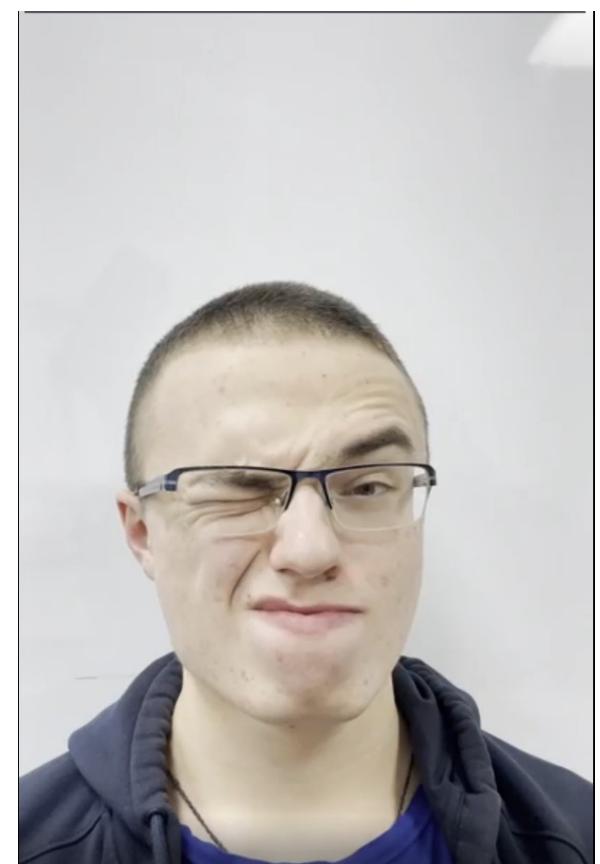
$$\begin{aligned} \text{TotalLoss}(X, Y) = & c_{cycle} \cdot \text{CycleLoss}_A(X) + c_{cycle} \cdot \text{CycleLoss}_B(Y) + \\ & + c_{gan} \cdot \text{GANLoss}_A(\text{G}_{B \rightarrow A}(Y)) + c_{gan} \cdot \text{GANLoss}_B(\text{G}_{A \rightarrow B}(X)) + \\ & + c_{attention} \cdot \text{AttentionLoss}_A(X) + c_{attention} \cdot \text{AttentionLoss}_B(Y) + \\ & + c_{1.5cycle} \cdot \text{MultiCycleLoss}_A(X) + c_{1.5cycle} \cdot \text{MultiCycleLoss}_B(Y) + \\ & + c_{gan} \cdot \text{GANLoss}_B(\text{G}_{A \rightarrow B}(\text{G}_{B \rightarrow A}(\text{G}_{A \rightarrow B}(X)))) + \\ & + c_{gan} \cdot \text{GANLoss}_A(\text{G}_{B \rightarrow A}(\text{G}_{A \rightarrow B}(\text{G}_{B \rightarrow A}(Y)))) \end{aligned}$$

Ускорение обучения. Библиотека Decord

Для ускорения прохода по датасету и быстрого доступа к случайному кадру мы воспользовались python-обёрткой над hardware декодерами (FFMPEG и Nvidia Codecs)



СБОР ДАТАСЕТА





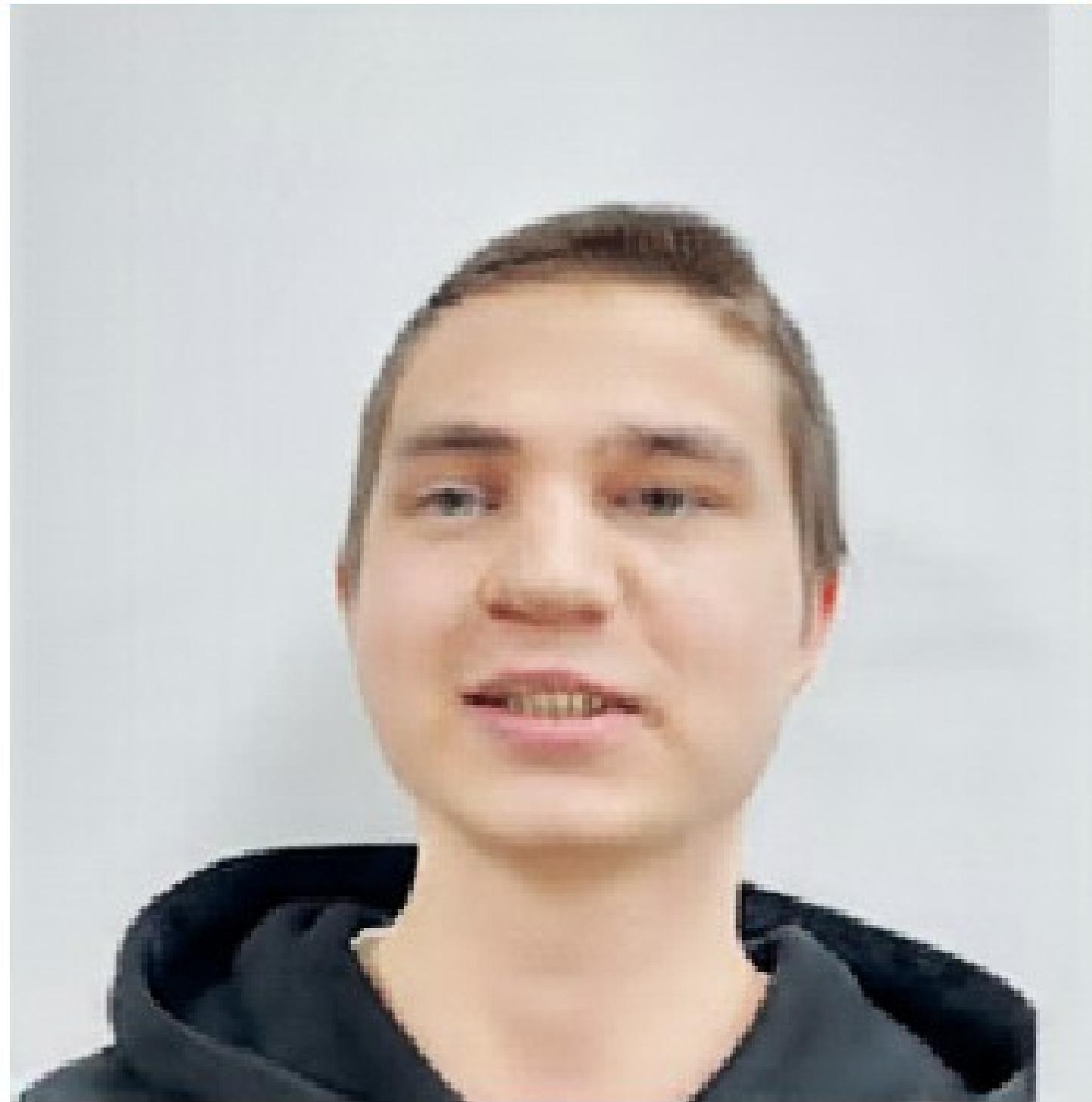
TensorBoard



TensorBoard

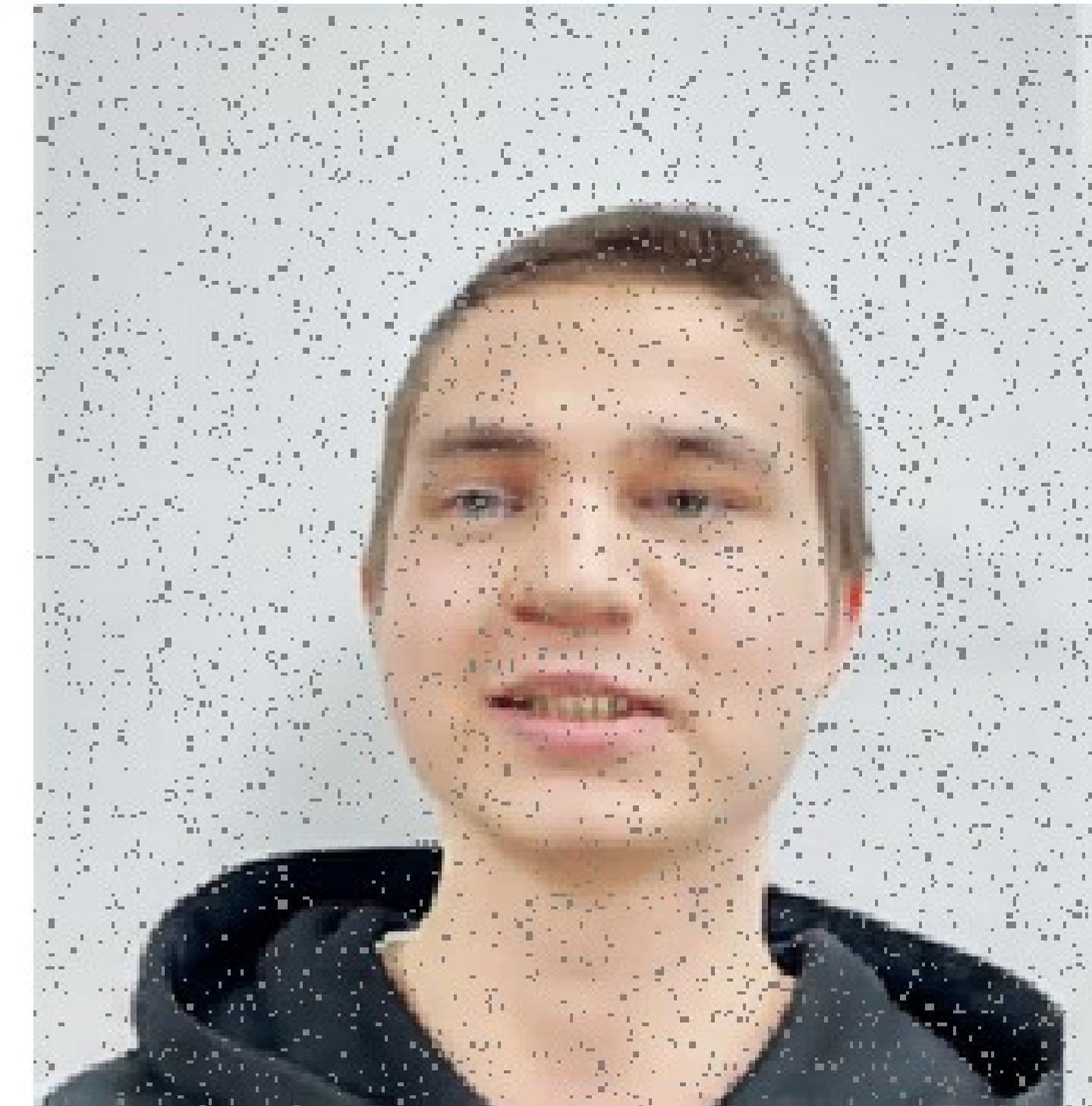
TrainImages/A/fake
tag: TrainImages/A/fake
step 19

Tue Jun 01 2021 02:40:43 GMT+0300 (Moscow Standard Time)



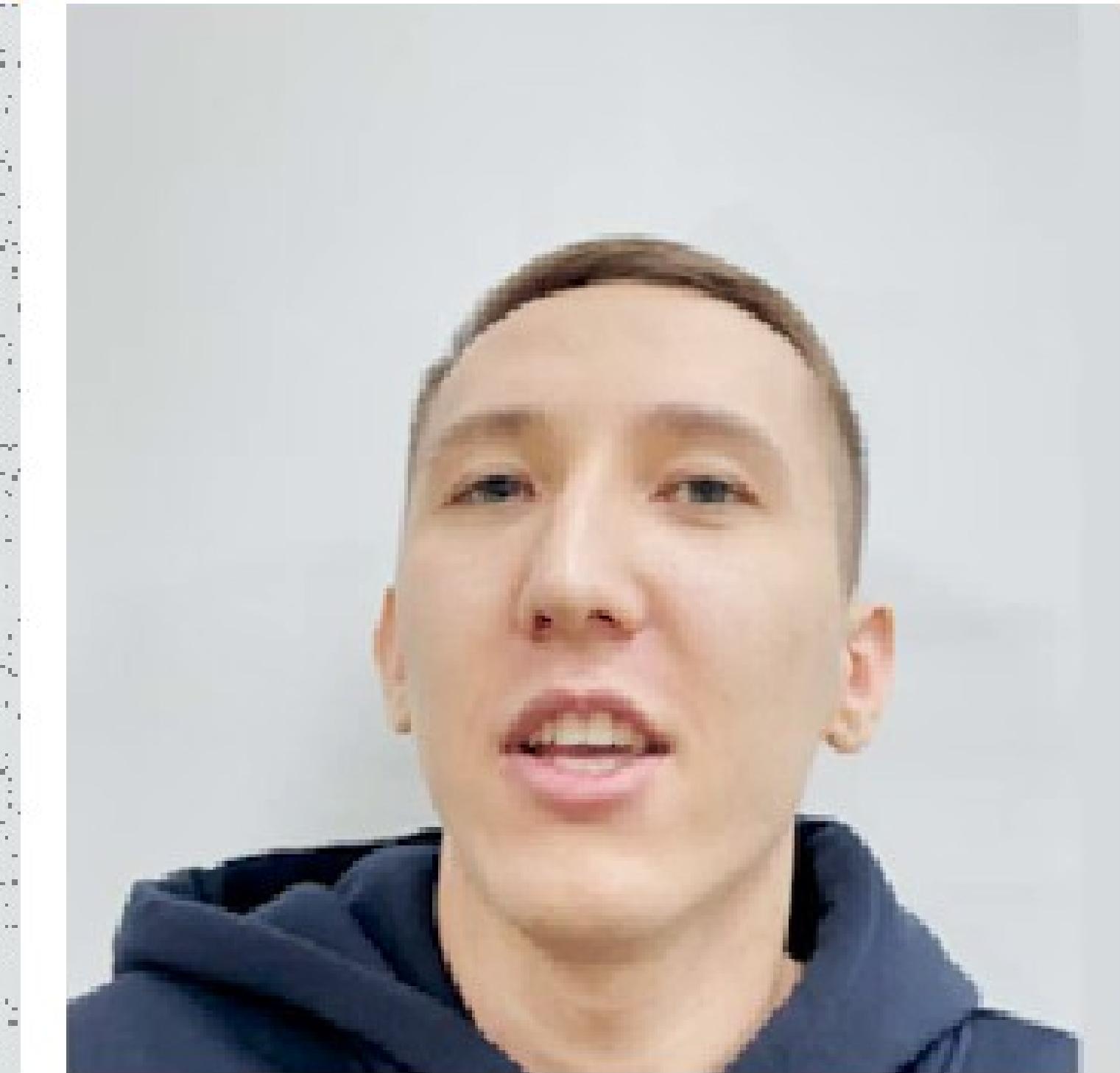
TrainImages/A/mess_up_image
tag: TrainImages/A/mess_up_image
step 19

Tue Jun 01 2021 02:40:43 GMT+0300 (Moscow Standard Time)



TrainImages/A/real
tag: TrainImages/A/real
step 19

Tue Jun 01 2021 02:40:43 GMT+0300 (Moscow Standard Time)





Yandex Cloud

Облачные сервисы Yandex Cloud использовались для обучения моделей и сохранения контрольных точек.

Виртуальные машины Yandex Compute Cloud для запуска скриптов для обучения и применения модели.

Объектное S3 хранилище Yandex Object Storage для сохранения и загрузки контрольных точек

Yandex Datasphere - достаточно новый проект, работает нестабильно и имеет множество багов. Отдали предпочтение Compute Cloud

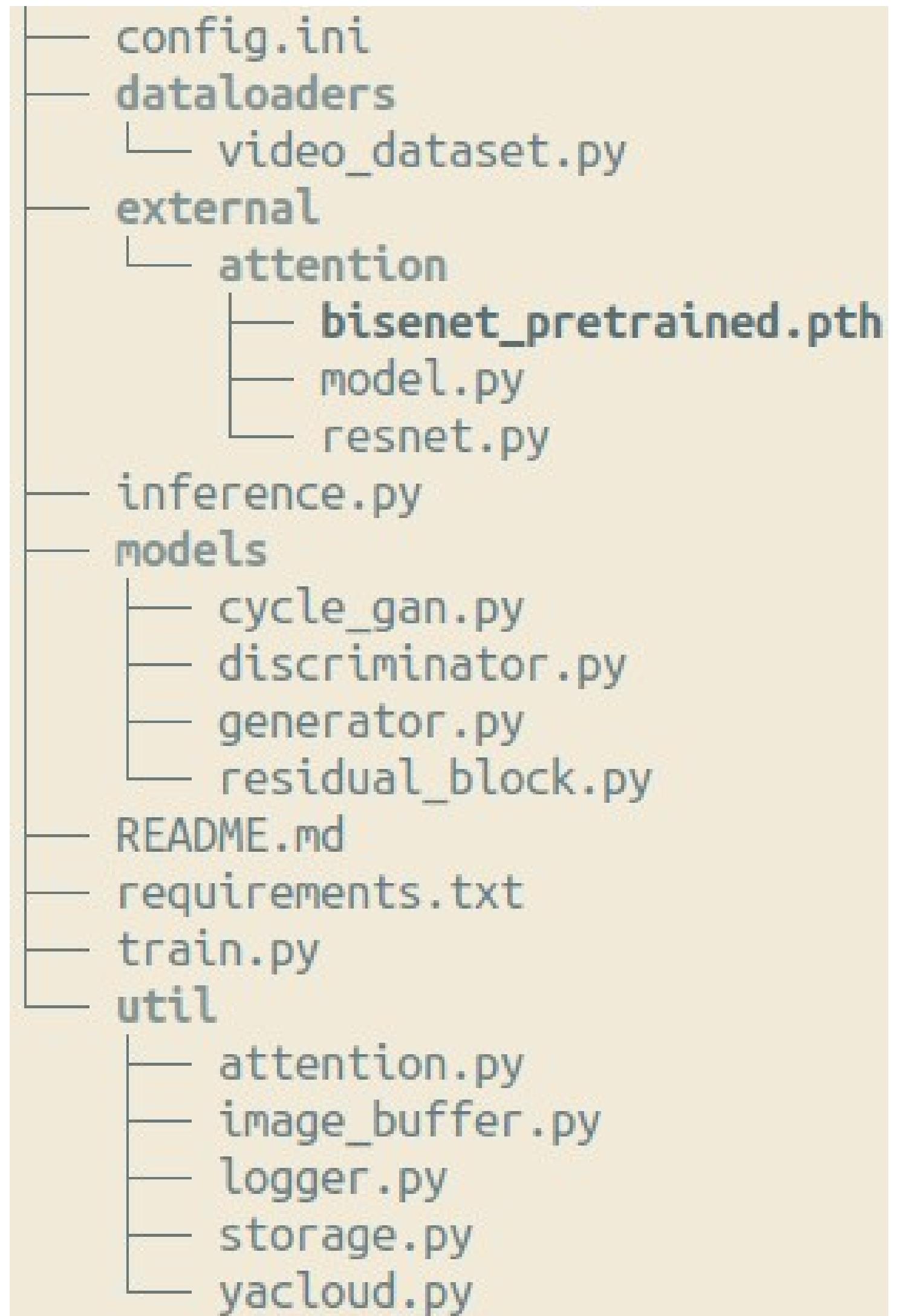
The screenshot shows the Yandex Object Storage interface. On the left, a sidebar lists various configuration options: Объекты (Objects), Веб-сайт (Website), Жизненный цикл (Lifecycle), CORS, Политика доступа (Access Policy), HTTPS, Версионирование (Versioning), Шифрование (Encryption), Операции (Operations), and Настройки (Settings). The main area displays a list of objects in the 'deepfake-checkpoints/experiment-yas-6' folder. The table has columns for Name, Size, and Storage Class. All files listed are 10.55 MB or 43.42 MB in size and are categorized under the 'Холодное' (Cold) storage class. The objects are named epoch0-netD_A.pth, epoch0-netD_B.pth, epoch0-netG_A2B.pth, epoch0-netG_B2A.pth, epoch1-netD_A.pth, epoch1-netD_B.pth, epoch1-netG_A2B.pth, and epoch1-netG_B2A.pth.

<input type="checkbox"/> Имя	Размер	Класс хранилища
epoch0-netD_A.pth	10.55 МБ	Холодное
epoch0-netD_B.pth	10.55 МБ	Холодное
epoch0-netG_A2B.pth	43.42 МБ	Холодное
epoch0-netG_B2A.pth	43.42 МБ	Холодное
epoch1-netD_A.pth	10.55 МБ	Холодное
epoch1-netD_B.pth	10.55 МБ	Холодное
epoch1-netG_A2B.pth	43.42 МБ	Холодное
epoch1-netG_B2A.pth	43.42 МБ	Холодное



Структура проекта

1. Директория `models` – код классов нейронных сетей, основанные на `nn.Module` библиотеки `pytorch`, а так же обертка для набора сетей `CycleGAN`, содержащая два дискриминатора и два генератора и реализующая методы для удобного использования в обучении и применении.
2. Директория `dataloaders` – классы, отвечающие за чтение видео и формирующие из него объекты датасета.
3. Директория `utils` – вспомогательные функции и классы, реализующие доступ к ресурсам `Yandex Cloud`, логирование обучения в `Tensorboard`, буферы изображений для обучения дискриминаторов, обертки для внешних зависимостей.
4. Директория `external` – реализации внешних моделей: сеть `BiSeNet` для получения маски классов частей лица.
5. Скрипты `train.py` и `inference.py` отвечают за процессы обучения и применения обученной модели
6. Конфиг-файл `config.ini`
7. Список зависимостей `requirements.txt`





Заключение

Основные результаты

- Реализована программа, решающая задачу замены человеческого лица в видео
- Собран датасет для тестирования программы
- Придуманы методы улучшения качества работы алгоритма Cycle_GAN на собранном датасете



Заключение

Перспективы дальнейшей работы

- Протестировать, насколько наши методы модернизации Cycle_GAN'а хорошо работают на других датасетах
- Дообучить Recycle_GAN и посмотреть на разницу в результатах
- Реализовать использование нескольких генераторов в качестве G_A.
В теории это должно воспрепятствовать построению общей стратегии кодирования/декодирования генераторов.



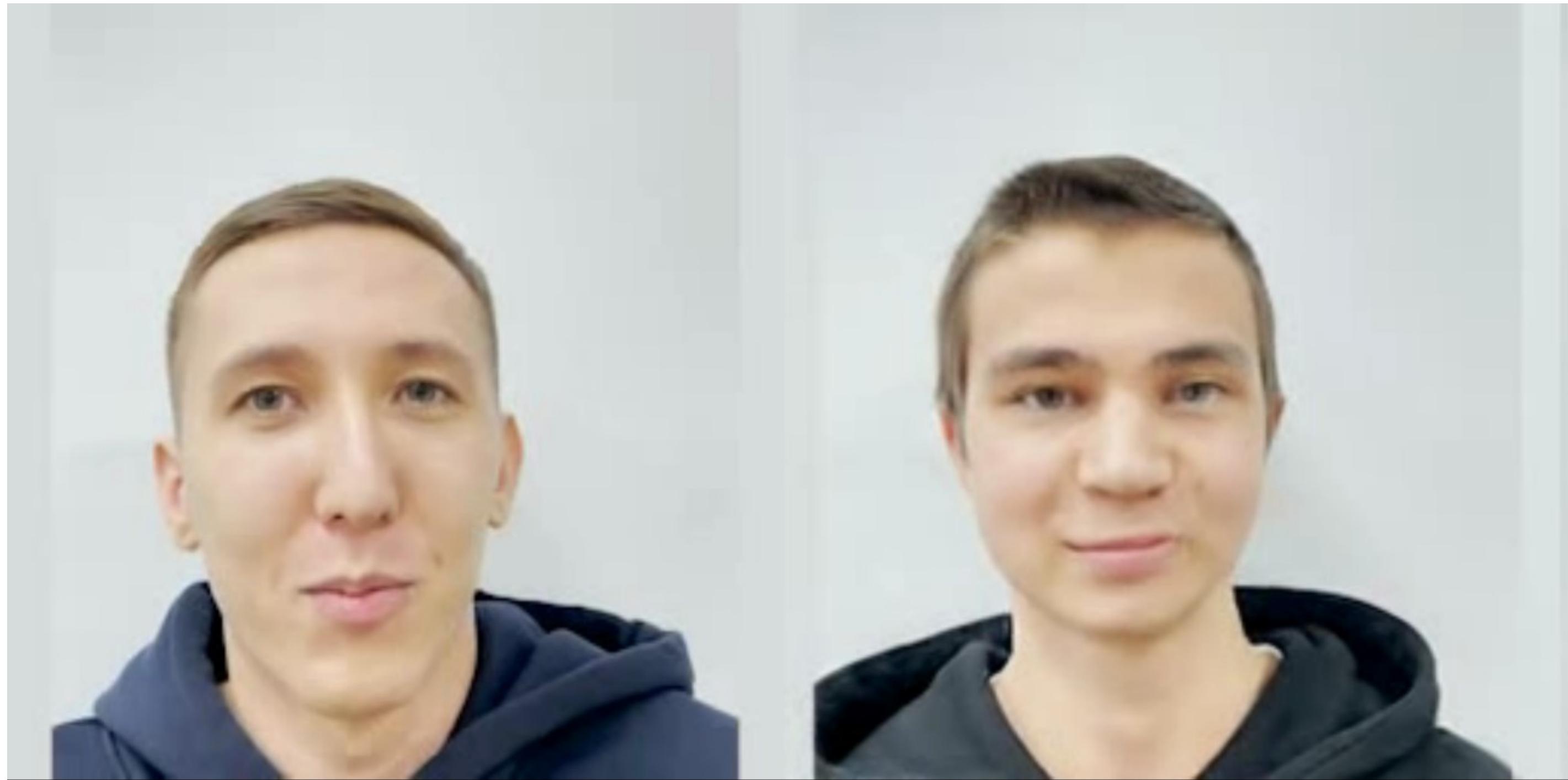
АНАЛИЗ РЕЗУЛЬТАТОВ РАБОТЫ

Замена лица на отдельных кадрах работает неплохо, зачастую даже человеческим глазом нельзя отличить преобразованную картинку от кадра из настоящего видео.

Однако, на некоторых кадрах при небольшом количестве эпох появляются плохие кадры при сложных эмоциях и движениях головы.



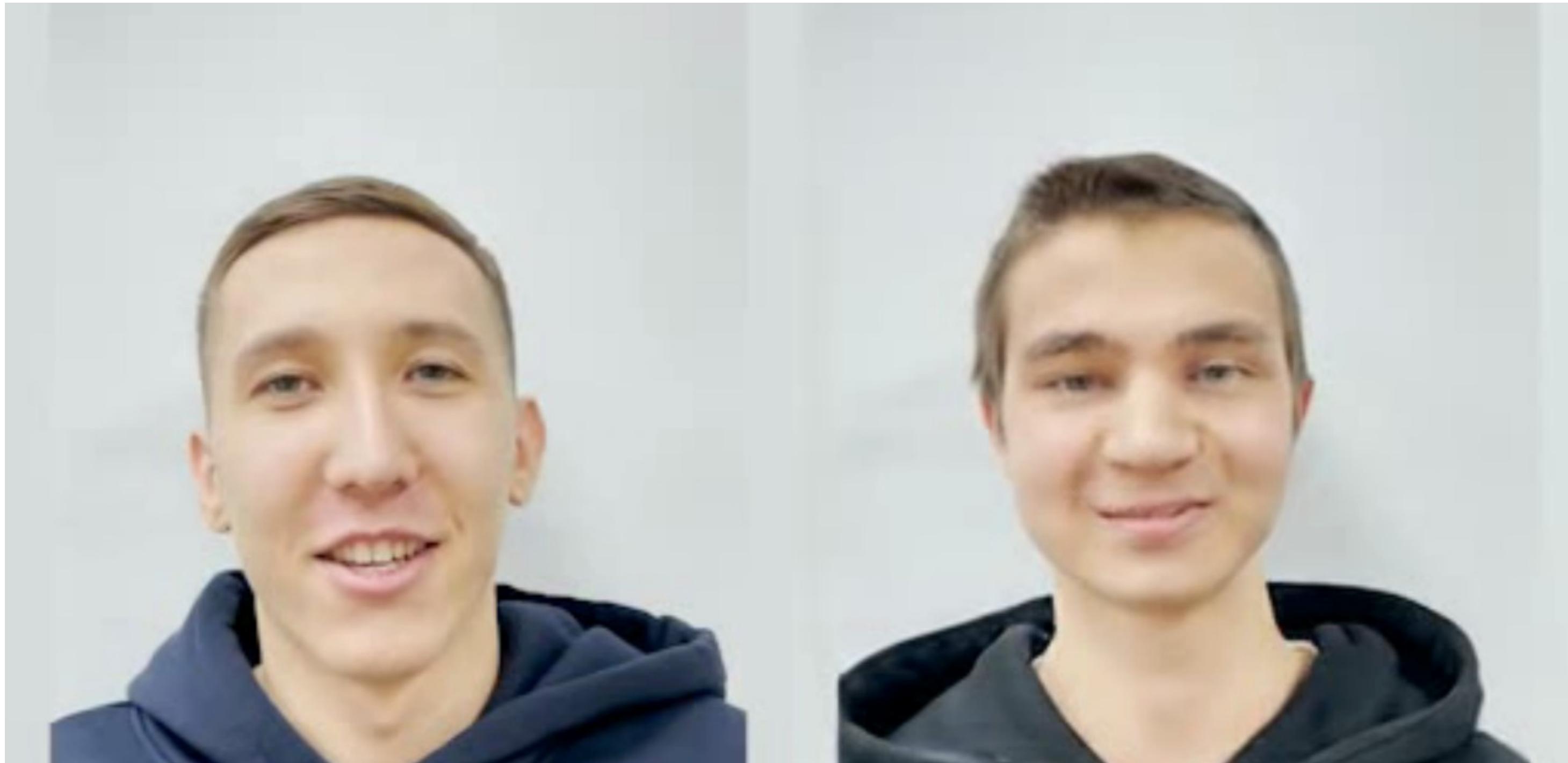
ПРИМЕРЫ ЗАМЕНЫ ЛИЦА НА ОТДЕЛЬНЫХ КАДРАХ



Пример с закрытым ртом (во всех примерах слева оригинал, справа – сгенерированное изображение)



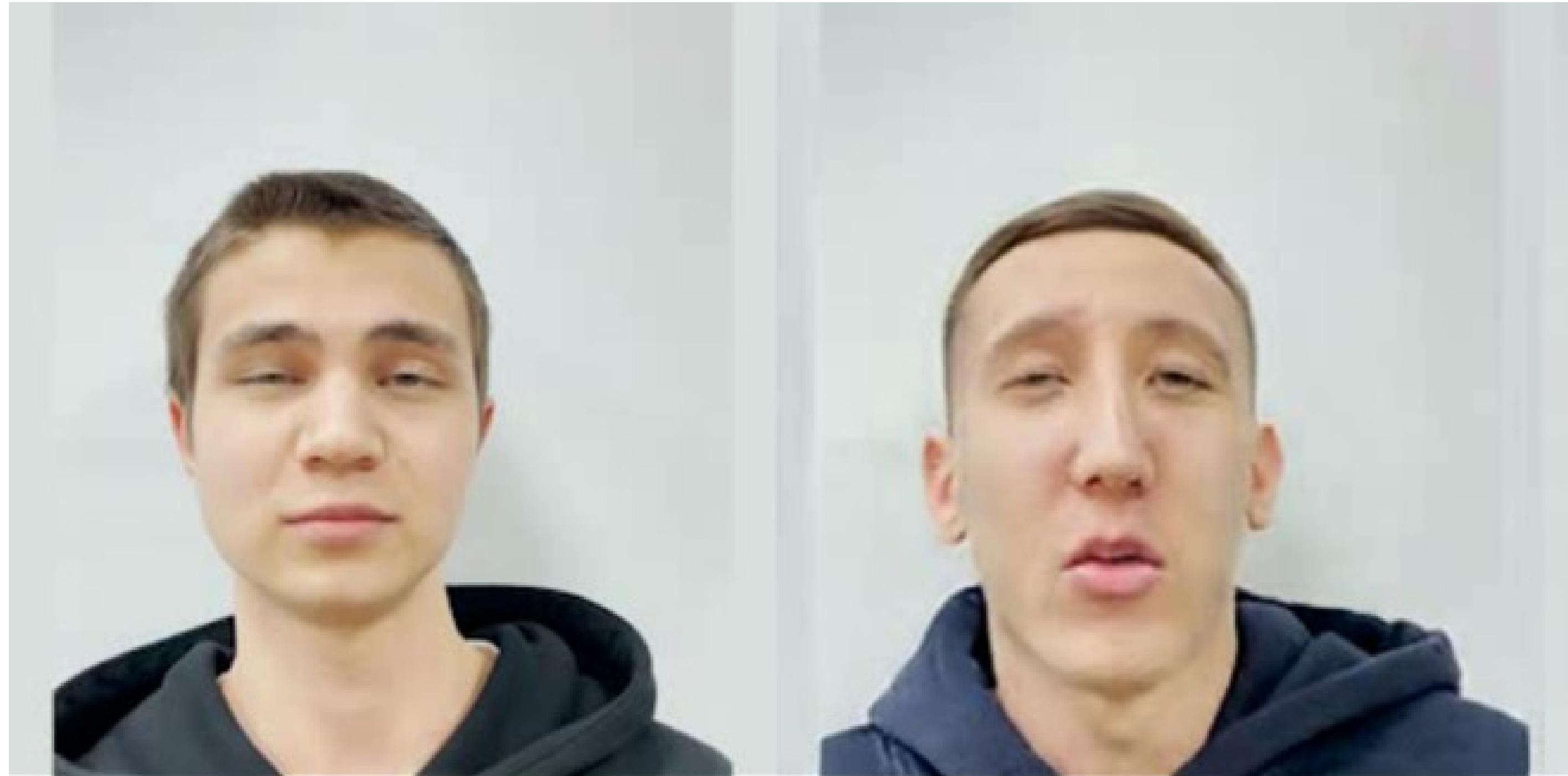
ПРИМЕРЫ ЗАМЕНЫ ЛИЦА НА ОТДЕЛЬНЫХ КАДРАХ



Пример с улыбкой



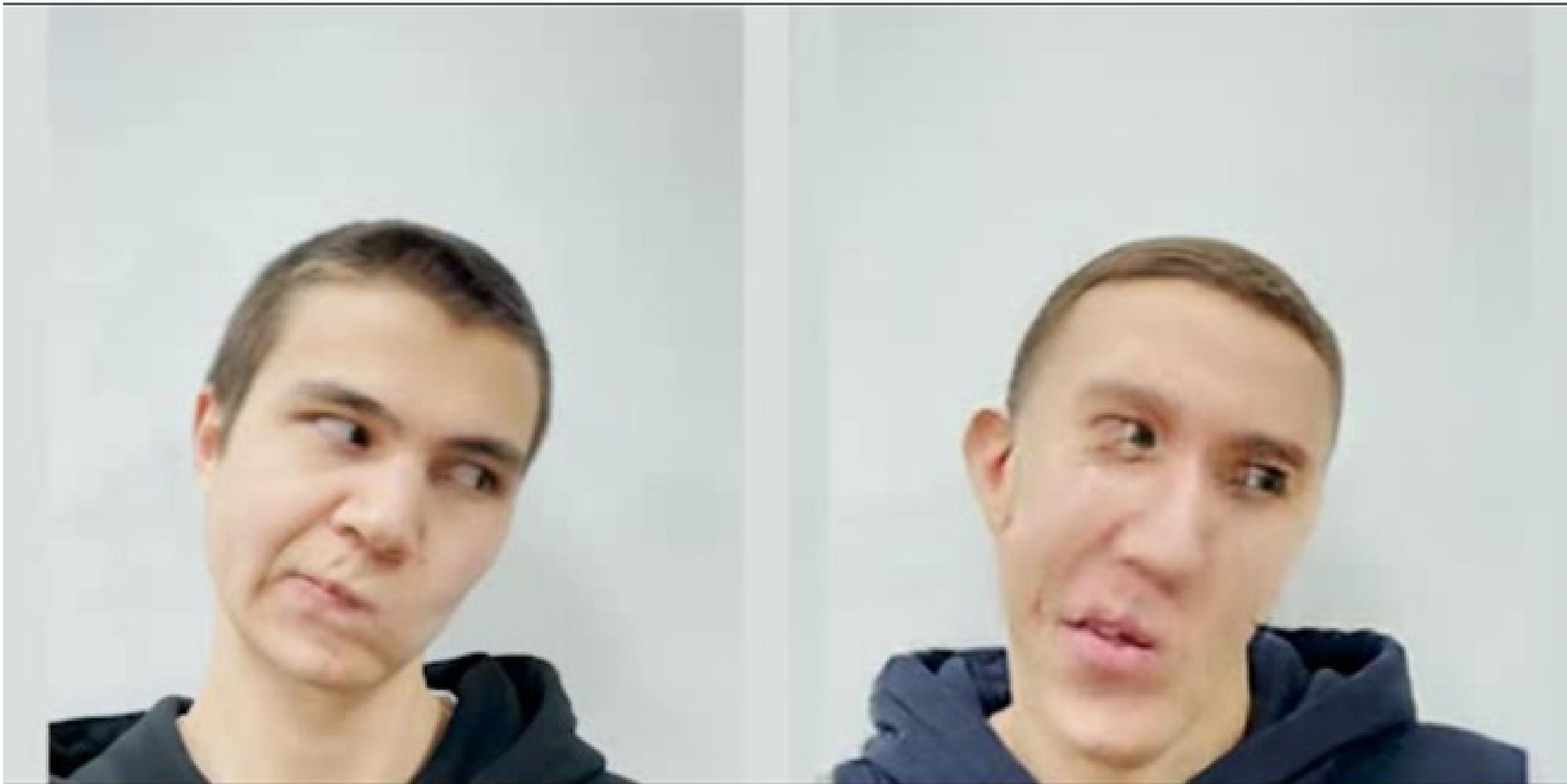
ПРИМЕРЫ ЗАМЕНЫ ЛИЦА НА ОТДЕЛЬНЫХ КАДРАХ



Пример с нестандартной эмоцией



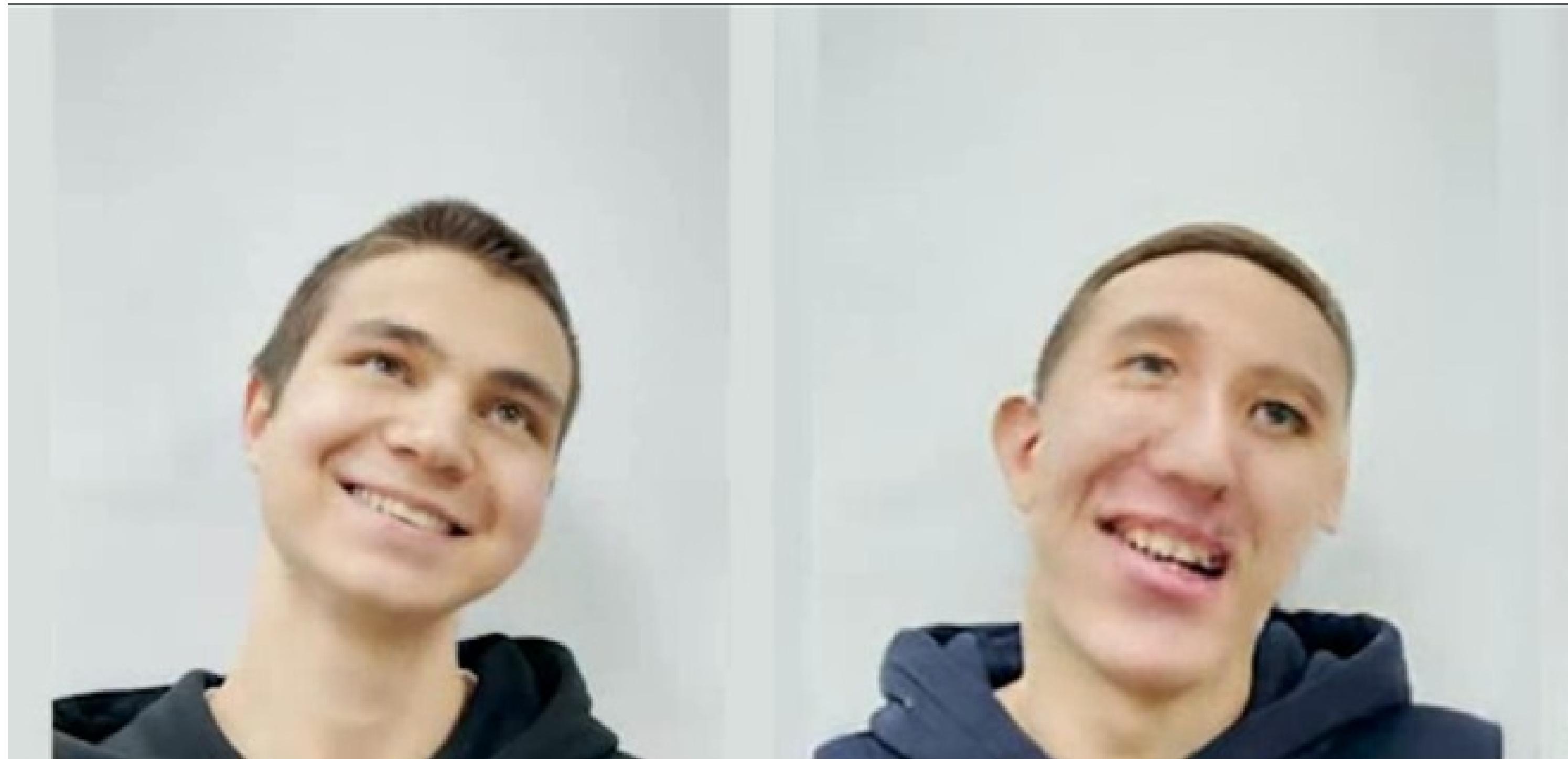
ПРИМЕРЫ ЗАМЕНЫ ЛИЦА НА ОТДЕЛЬНЫХ КАДРАХ



Сложная эмоция и движение головы



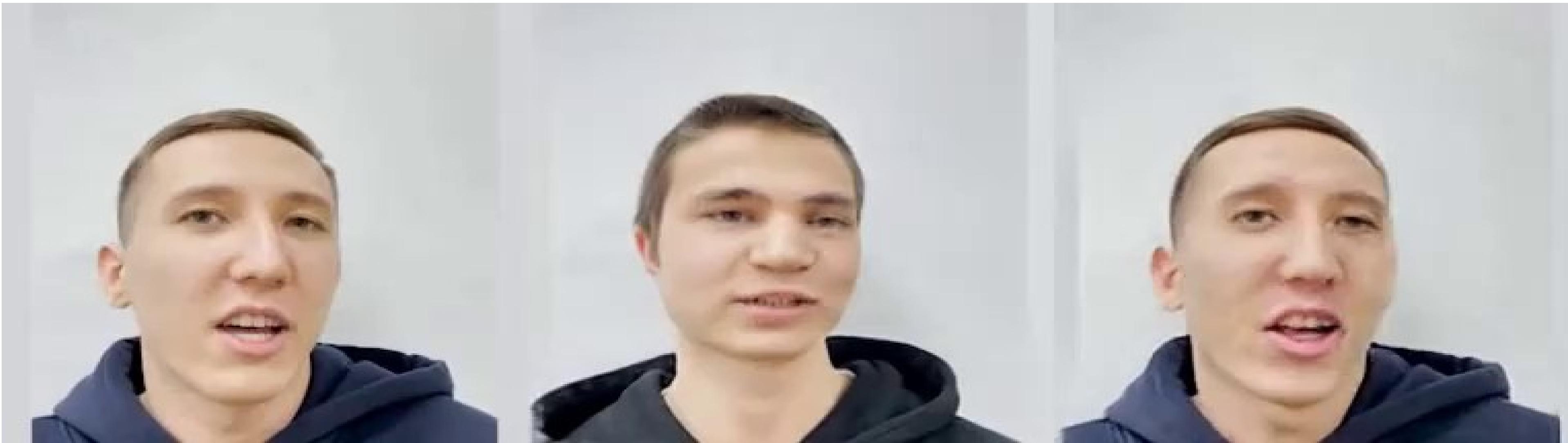
ПРИМЕРЫ ЗАМЕНЫ ЛИЦА НА ОТДЕЛЬНЫХ КАДРАХ



Не очень сложная эмоция, но движение головы все испортило



ПЕРВАЯ ПОЛОВИНА ВИДЕО: РАЗГОВОР





ВТОРАЯ ПОЛОВИНА ВИДЕО МИМИКА И ВРАЩЕНИЕ ГОЛОВЫ



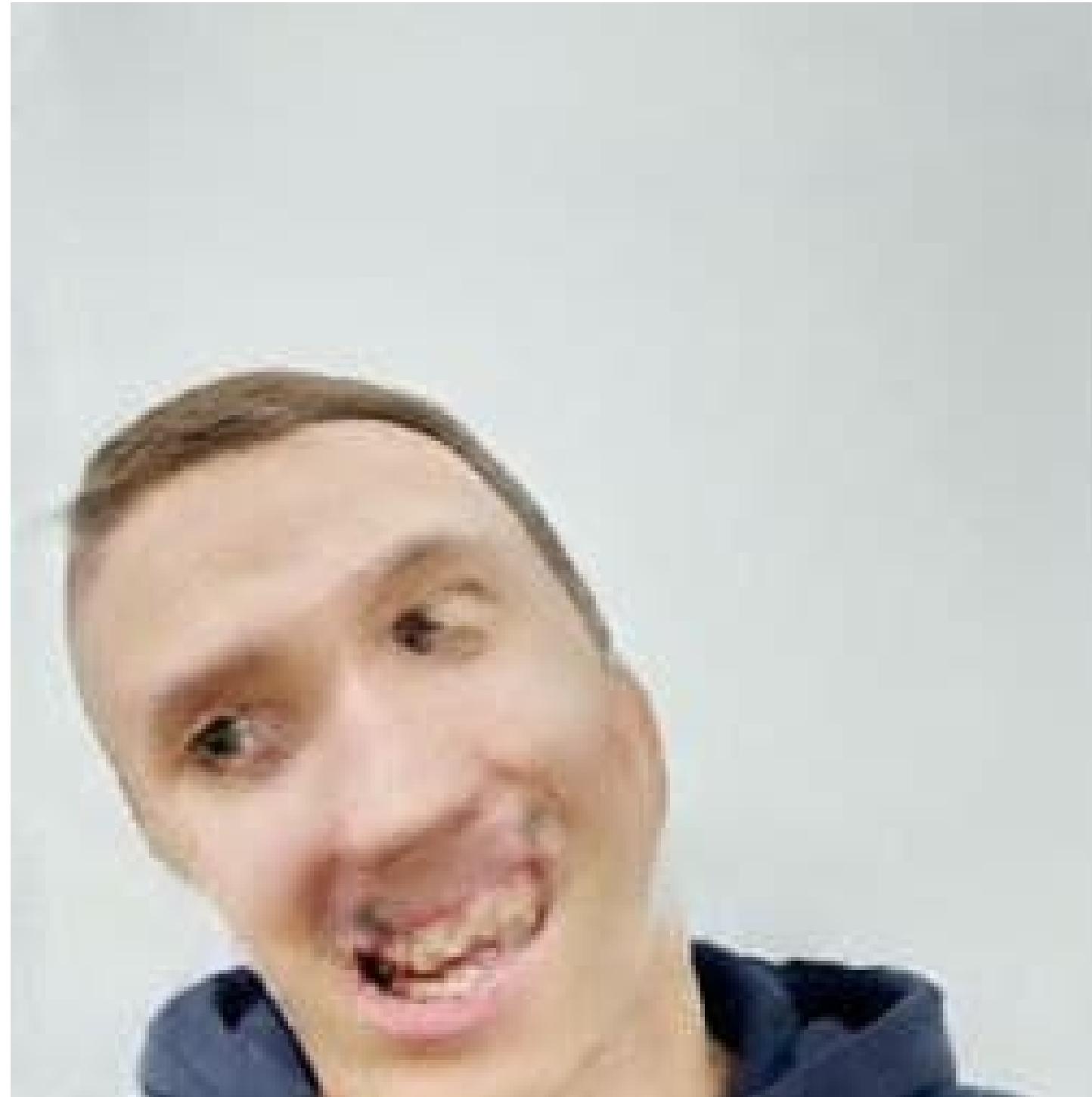


ССЫЛКИ И ИСТОЧНИКИ

1. Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Электронный ресурс] / arXiv. Режим доступа: <https://arxiv.org/abs/1703.10593>, свободный. (дата обращения: 08.02.21)
2. face-parsing.PyTorch [Электронный ресурс] / GitHub. Режим доступа: <https://github.com/zllrunning/face-parsing.PyTorch>, свободный. (дата обращения: 03.06.21)
3. Recycle-GAN: Unsupervised Video Retargeting [Электронный ресурс] / Carnegie Mellon University. Режим доступа: <http://www.cs.cmu.edu/~aayushb/Recycle-GAN>, свободный. (дата обращения: 03.06.21)
4. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition [Электронный ресурс] / arXiv. Режим доступа: <https://arxiv.org/abs/1512.03385>, свободный. (дата обращения: 08.02.21)
5. Decord [Электронный ресурс] / GitHub. Режим доступа: <https://github.com/dmlc/decord>, свободный. (дата обращения: 03.06.21)



СПАСИБО ЗА ВНИМАНИЕ!





НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ