

РАЗРАБОТКА АЛГОРИТМА ГЕНЕРАТИВНОЙ НЕЙРОННОЙ СЕТИ ДЛЯ СОЗДАНИЯ УПРАВЛЯЕМЫХ КАДРОВ ИЗОБРАЖЕНИЙ ЛИЦ

*А.А. Друки, к.т.н., доц. ОИТ ИШИТР,
П.Р. Маслов, студент гр.8В7Б.
Томский политехнический университет
E-mail: prml@tpu.ru*

Введение

На сегодняшний день становится все популярнее подход генеративно-сопоставительных сетей, позволяющий добиваться ранее недостижимых результатов во многих сферах применения ИНС, в том числе и при обработке изображений. Генеративно-сопоставительная сеть (Generative adversarial network) – алгоритм машинного обучения без учителя, включающий две связанные нейронные сети: первая из них – «генератор», создает образцы, вторая – «дискриминатор», пытается уличить генератор в неправдоподобности его результатов по сравнению с реальными примерами [1]. Так как генератор и дискриминатор имеют противоположные цели — создать образцы и отбраковать образцы — между ними возникает состязание. Применение этого подхода позволяет, в частности, генерировать фотографии, которые человеком могут быть восприняты как натуральные изображения.

Поэтому в данной работе исследуется область генерации человеческих лиц с применением GAN. Более конкретно, рассматривается область управляемой генерации – создания новых кадров с опорой на характерные особенности других уже имеющихся кадров, что позволяет добиться анимации лица любого человека, имея лишь его фотографии.

Описание алгоритма

Для решения задачи генерации управляемых изображений был использован фреймворк машинного обучения Torch [2] в версии для языка программирования Python. Обучающие данные представлены набором VoxCeleb, состоящим из разреженных кадров видеointервью с различными знаменитостями.

Существует несколько реализаций алгоритмов генеративных сетей, решающих текущую задачу, но имеющие свойственные всем нейронным сетям достоинства и недостатки, связанные с вычислительной сложностью (временем работы) и качеством результата работы.

Наиболее отличительные из таких алгоритмов:

- MonkeyNet [3]
- First Order Motion Model [4]
- X2Face [5]
- Few-Shot Adversarial Model [6]

Общим в приведенных архитектурах является использование комплекса сетей для отображения входных данных в n -мерное признаковое пространство, взятия из этого пространства определенного вектора признаков с обратным отображением в пространство входных данных (генерации выходных данных) и реализации самоконтроля на этапе обучения.

Так как наиболее прогрессивными (имеющими значительный прирост в качестве при сравнимо малом увеличении вычислительной сложности) являются две последних архитектуры в приведенном списке, то решено основываться на алгоритмических конструкциях, наиболее приближенных к входящим в них.

Обобщенная схема разработанного алгоритма приведена на рисунке 1. Она включает себя три сети: первая отображает исходное изображение лица в вектор n -мерного пространства, который передается во вторую сеть-генератор (стоит отметить, что целью первой сети является выделение индивидуальных признаков лица, которые не зависят от его положения в пространстве, т.е. являются инвариантными – признаки внешности), вторая сеть, используя полученный вектор, а также входное управляющее изображение лица, генерирует новое лицо (переходит в обобщенное пространство признаков исходного и управляющего изображений и выбирает наиболее подходящий вектор). Третья сеть-дискриминатор имеет важное значение при обучении и выполняет роль адаптивной (обучаемой) функции потерь, точнее сразу двух функций, штрафующих как сеть выделения инвариантных признаков (расстояние между исходным изображением и сгенерированным), так и генератор (расстояние между управляющим изображением и сгенерированным), тем самым представляя звено самоконтроля общего алгоритма сети.

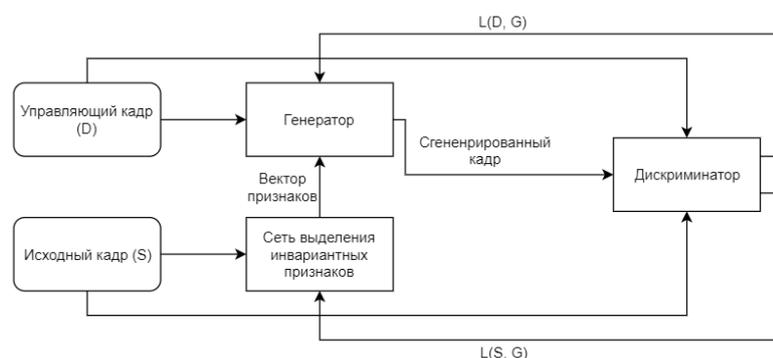


Рис. 1. Обобщенная схема архитектуры сети

Тестирование алгоритма

На рисунке 2 приведены результаты работы алгоритма на текущей стадии разработки. Стоит отметить, что повторение положения головы осуществляется достаточно выразительно, однако заметно, что кадр при этом теряет в реалистичности (размытости, артефакты), что связано как с недостатком информации в исходном кадре (часть лица на исходном изображении не попадает на двухмерную проекцию), так и недостаточно эффективной работы дискриминатора: его минимизации расстояния инвариантных признаков.



Рис. 2. Визуальное отображение работы алгоритма

Заключение

В результате проделанной работы реализован алгоритм генеративной нейронной сети, способный по управляющим и исходным кадрам генерировать новые, несущие в себе пространственные признаки первых и признаки внешности вторых.

На данном этапе алгоритм способен выполнять поставленную задачу, но имеющиеся качественные недочеты требуют дальнейших действий по изменению отдельных элементов алгоритма и подходов к его обучению, что в перспективе приведет к более стабильной работе, а также позволит сделать акцент либо на увеличении разрешения изображений (повышении качества), либо на скорости выполнения, что важно в условиях обработки в реальном времени.

Список использованных источников

1. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio: « Generative Adversarial Networks», 2014
2. Документация PyTorch. [Электронный ресурс]. – URL: <https://pytorch.org/> (дата обращения: 04.03.2021).
3. Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, Nicu Sebe: «Animating Arbitrary Objects via Deep Motion Transfer», IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019
4. Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, Nicu Sebe: « First Order Motion Model for Image Animation», The Conference and Workshop on Neural Information Processing Systems, 2019
5. Olivia Wiles, A. Sophia Koepke, Andrew Zisserman: « X2Face: A network for controlling face generation by using images, audio, and pose codes», European Conference on Computer Vision, 2018
6. Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, Victor Lempitsky: « Few-Shot Adversarial Learning of Realistic Neural Talking Head Models», Conference on Computer Vision and Pattern Recognition, 2019