

Санкт–Петербургский государственный университет

*МИРОШНИЧЕНКО Александр Сергеевич*

Отчет по научно-исследовательской работе

*Улучшение качества изображений с помехами с использованием  
генеративных состязательных сетей*

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2017 «Прикладная математика, фундаментальная информатика и программирование»

Профиль «Исследование и проектирование систем управления  
и обработки сигналов»

Научный руководитель:  
доцент, кафедра теории систем управления электрофизической аппаратурой,  
к.ф. - м.н. Козынченко Владимир Александрович

Санкт-Петербург  
2020 г.

# Содержание

Введение . . . . .	3
Постановка задачи . . . . .	3
Метод обучения . . . . .	3
Подготовка данных для обучения . . . . .	3
Процесс обучения . . . . .	4
Анализ полученных результатов . . . . .	4
Заключение . . . . .	4
Список литературы . . . . .	4
Приложение . . . . .	4

## Введение

При съемке фотографии могут возникнуть различные искажения. Например фотография может получиться размытой, матрицы фотоаппарата неисправна, или если объектив камеры движется слишком быстро, и не успевает сфокусироваться. В таких случаях можно повторить снимок, но какие-то объекты, которые были засняты в первый раз, уже могут и не появиться в кадре. Иногда фотографию и вовсе нельзя пере-снять.

Встает вопрос о восстановлении качества изображения с помехами.

При этом надо понимать, что, конечно, мы не сможем восстановить именно ту фотографию, которая могла получиться, не будь обстоятельств, смазавших кадр. Ведь по сути дела можно представить искажение четкой фотографии как некоторое отображение из множества четких фото в множество размытых. И такое отображение с большой долей вероятности не будет биективным. То есть наверняка существует много различных четких фотографий, которые при обратном отображении в множество размытых дают один и тот же результат.

Однако можно попытаться сказать, каким мог бы быть один из возможных вариантов четких изображений.

Для такой проблемы можно использовать нейронные сети, так как их решают задачу перевода одного тензора данных в другой.

## Постановка задачи

Реализовать программу, имплементирующую алгоритм улучшения качества размытого изображения при помощи нейронной сети.

## Метод обучения

Среди существующих видов сетей можно отметить GAN - генеративные состязательные сети. Модели GAN обучаются отображению случайного вектора шума  $z$  в изображение  $y$ ,  $G : z \rightarrow y$ . Условные же GAN учатся преобразованию изображения  $x$  и случайного вектора шума  $z$  в изображение  $y$ ,  $G : \{x, z\} \rightarrow y$ . Генератор  $G$  обучается генерировать изображения, которые не может отличить от «реальных» дискриминатор  $D$ , который, в свою очередь, обучается распознавать «подделки» генератора.

Целевая функция условной GAN может быть выражена как:

$$loss_{GAN} = E_{x,y} [\log D(x, y)] + E_{x,z} [\log (1 - D(x, G(x, z)))],$$

где  $G$  пытается минимизировать  $loss$ , в то время как соревнующийся с ним  $D$  пытается максимизировать ее. Другими словами  $G^* = \arg \min_G \max_D loss_{GAN}(G, D)$

В статье о сети pix2pix было проведено исследование, в котором функция L1 в сочетании с традиционной  $loss_{GAN}$  приводила к тому, что работа дискриминатора  $D$  не изменялась, а вот генератор  $G$  обучался не только обманывать дискриминатор, но и генерировать изображения, близкие к истинным.

$$L1(G) = E_{x,y,z} [\|y - G(x, z)\|]$$

В итоге, наша функция выглядит как:

$$G^* = \arg \min_G \max_D loss_{GAN}(G, D) + \lambda L1(G)$$

Значение  $\lambda = 100$ , так как именно это значение предлагают авторы вышеупомянутой статьи.

Алгоритм обучения представлен на рисунке [1]. Датасет изначально перемешан и разбит на батчи. В данном случае размер батча равен одному. Прогон каждого батча и изменение весов сети обозначены одним шагом обучения.

## Подготовка данных для обучения

Для того, чтобы обучить нейронную сеть, необходимо иметь как размытые изображения  $x$ , так и четкие изображения  $y$ . Четкие изображения  $y$  было решено <sup>3</sup>зять из открытого датасета places365, состоящего из

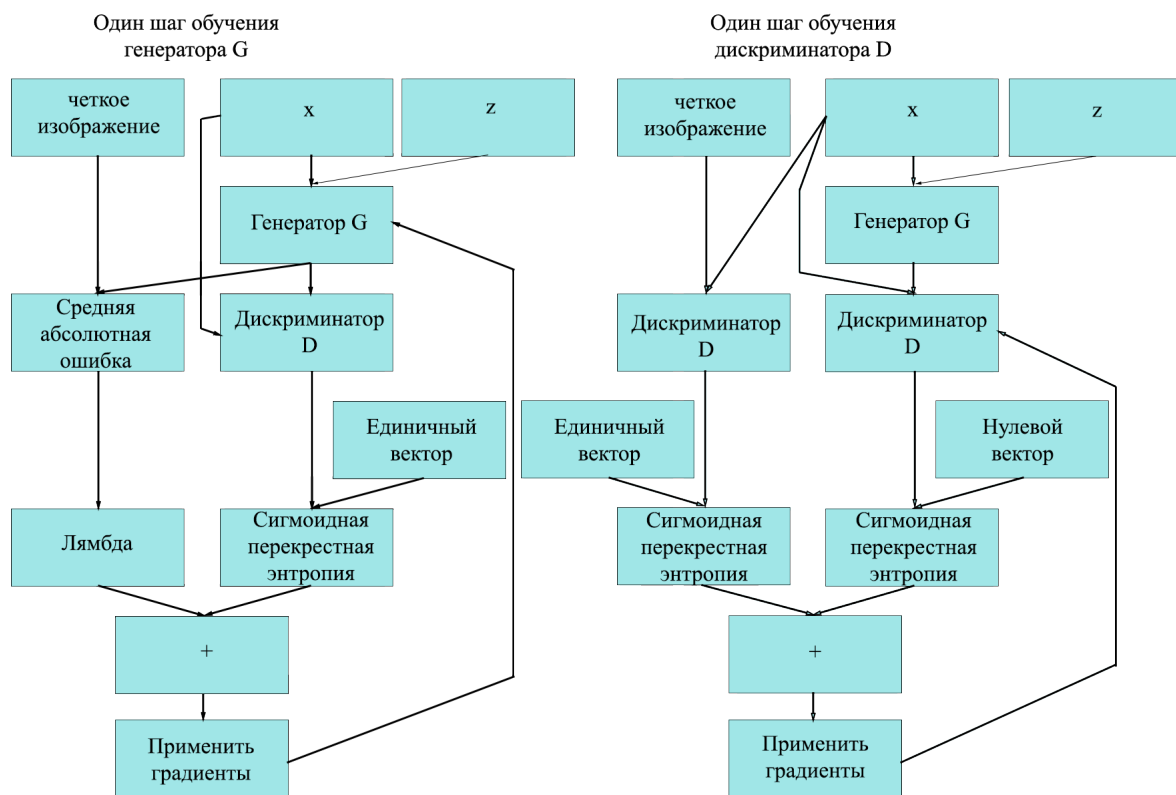


Рис. 1: Шаг обучения для генератора и дискриминатора

365 категорий четких фотографий, собранных из интернета. Из этих категорий были выбраны следующие 4: замки, леса, горы и небоскребы. Для тренировочного датасета из каждой категории было выбрано по 250 фотографий, а для тестового - 150. На каждую фотографию был наложен эффект размытия при помощи функции `cv2.blur()` с ядром (14, 14) из библиотеки `OpenCV`. Это имитация реальных помех, которые могут произойти при съемке фотографий. Всего получилось 2000 изображений (1000 четких и 1000 размытых) для тренировочного датасета и 1200 изображений (600 четких и 600 размытых) для тестового датасета. После обработки датасеты были загружены на облако Google Drive.

## Процесс обучения

## Анализ полученных результатов

## Заключение

## Список литературы

## Приложение

Ссылка на репозиторий с кодом: