

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В.ЛОМОНОСОВА»

ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И ИНФОРМАТИКИ

Нейросетевой синтез текстур с трендами

Выполнил студент

435 группы:

Будакян Я. С.

Научный руководитель:

к.т.н., доц. Грачев Е. А.

Москва

2017

1 Введение

Цель работы состоит в построении процедуры синтеза изображений среды, которые будут содержать в себе тренд. Под текстурой с трендом понимается изображение, в котором есть изменение некоторой статистической характеристики вдоль одного из направлений. Такими характеристиками, например, могут быть изменение интенсивности появления частиц среды или изменение пористости среды.

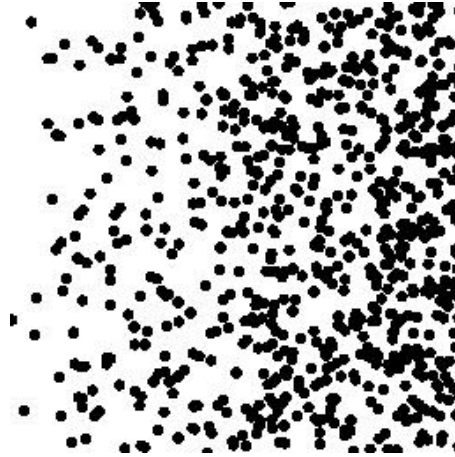


Рис. 1: Пример текстуры с трендом интенсивности частиц.

2 Математическая постановка

Рассмотрим гипотетическое многомерное пространство, в котором находятся все изображения. Тогда обучающая выборка изображений с трендами задает в этом пространстве вероятностное распределение P_X , в котором точки, соответствующие изображениям из выборки, имеют высокую вероятность, а остальные - очень низкую. Это так называемая вероятностная постановка задачи обучения [1, 2]. Тогда с математической точки зрения задача синтеза текстуры с трендом сводится к синтезу случайного изображения X' , принадлежащего распределению, близкому к задаваемому обучающей выборкой:

$$P_{X'} \approx P_X,$$

где P_X - распределение изображений с трендами, удовлетворяющих следующим ограничениям (для упрощения задачи):

- Это монохромные изображения 256 x 256 пикселей
- Изменяющимся свойством является интенсивность появления частиц λ
- Тренд является линейным и направлен вдоль оси x: $\lambda = \lambda_0 + kx$

3 Существующие подходы к решению задачи

Есть несколько подходов к решению задач подобного рода:

- 'Классический' статистический подход
- Базовый нейросетевой подход
- Генеративные состязательные сети (GAN)

3.1 'Классический' статистический подход

- Вводится параметризованное семейство распределений вероятности $P_\theta(x)$
- Параметры θ находятся из обучающей выборки:

$$\mathcal{L}_\theta(D) = \prod_{x \in D} P_\theta(x)$$

$$\theta^* = \arg \max_{\theta} \mathcal{L}_\theta(D)$$

- Генерируется текстура из распределения P_{θ^*}

Этот подход приводит к проблемам:

- Пространство параметров θ может быть огромной размерности
- Известной параметрической модели распределения может вообще не существовать

Простой пример - синтез человеческих лиц: с помощью классического подхода эта задача не была решена с хорошим качеством.

3.2 Базовый нейросетевой подход

- Вводится параметризованное семейство распределений вероятности $P_\theta(x)$
 - Вводятся скрытые переменные V и функция(нейросеть) для получения x из V (фактически, классификация, развернутая в другую сторону)
- Определяются параметры распределения (т.е. обучение нейросети)
- Генерируется выборка из P_{θ^*}

Этот подход возможен, однако на практике трудноосуществим [3].

3.3 GAN - генеративные состязательные сети

Вернемся к изначальной задаче: найти процедуру генерирования X' так, чтобы $P_{X'} \approx P_X$. Переформулируем:

$$\rho(P_{X'}, P_X) \longrightarrow \min_{P_{X'}}$$

Введем некоторые скрытые переменные с фиксированным распределением, например

$$V \sim U^n[-1, 1],$$

и параметризованную процедуру генерации:

$$X' = g_\theta(V)$$

Переформулируем:

$$\rho(P_{X'}, P_X) \longrightarrow \min_{P_{X'}}$$

$$\rho(g_\theta(V), P_X) \longrightarrow \min_{g_\theta(V)}$$

$$\rho(g_\theta(V), P_X) \longrightarrow \min_{\theta}$$

Возникает вопрос: что использовать в качестве метрики похожести двух распределений ρ , где одно из распределений задано обучающей выборкой. В качестве такой метрики можно использовать функцию потерь обученного классификатора. Тогда задача примет вид:

$$\rho(P_{X'}, P_X) \longrightarrow \min \Leftrightarrow L \longrightarrow \max,$$

где L - функция потерь обученного классификатора. Соответственно, можно ввести две нейросети:

- $d_\zeta(x)$ - классификатор для измерения расстояния, **'дискриминатор'**
- $g_\theta(x)$ - сеть, трансформирующая V в X' , **'генератор'**

Суть использования двух сетей состоит в том, что они обучаются совместно, конкурируя друг с другом: генератор пытается имитировать целевое распределение, а дискриминатор пытается классифицировать поступающие от генератора и из обучающей выборки изображения на 2 класса: реальные (из изначального распределения P_X) и ложные (из P'_X , т.е. произведенные генератором). Для дальнейшего рассмотрения введем функцию потерь дискриминатора (например, logloss):

$$\begin{aligned} L(X, X') &= \frac{1}{2} \mathbb{E}_X l(d_\zeta(x), 1) + \frac{1}{2} \mathbb{E}_{X'} l(d_\zeta(x'), 0) = \\ &= -\frac{1}{2} (\mathbb{E}_X \log d_\zeta(x) + \mathbb{E}_{X'} \log(1 - d_\zeta(x'))) = \\ &= -\frac{1}{2} (\mathbb{E}_X \log d_\zeta(x) + \mathbb{E}_V \log(1 - d_\zeta(g_\theta(v)))) = \\ &= L(\zeta, \theta). \end{aligned}$$

$l(d_\zeta(x), 1)$ и $l(d_\zeta(x'), 0)$ здесь - это потери при неправильной классификации реального и ложного изображений соответственно. Функция потерь обученного классификатора:

$$L^*(\theta) = \min_{\zeta} L(\zeta, \theta)$$

Соответственно,

$$\begin{aligned} \min_{\zeta} L(\zeta, \theta) &\longrightarrow \max_{\theta} \\ \theta^* &= \arg \max_{\theta} \left[\min_{\zeta} L(\zeta, \theta) \right] \end{aligned}$$

Определим оптимальный дискриминатор:

$$\begin{aligned} d_{\theta}^* &= d_{\zeta^*(\theta)} \\ \zeta^*(\theta) &= \arg \min_{\zeta} L(\zeta, \theta) \end{aligned}$$

4 Обучение GAN

Итак, задача обучения GAN свелась к нахождению

$$\theta^* = \arg \max_{\theta} \left[\min_{\zeta} L(\zeta, \theta) \right]$$

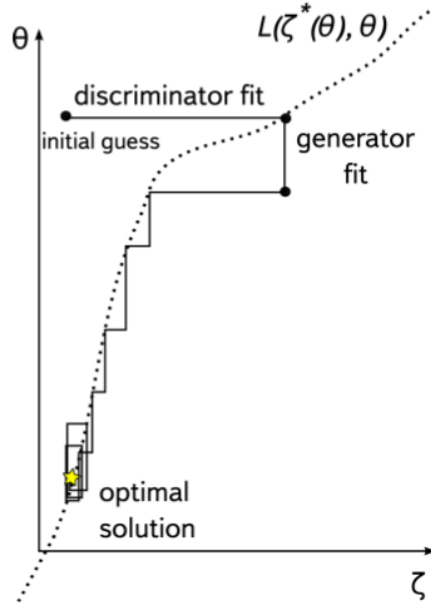


Рис. 2: Схематическое изображение процесса обучения GAN

Решить ее можно, например, методом стохастического градиентного спуска:

$$\Delta\theta \sim \nabla L(\zeta^*(\theta), \theta)$$

Для малых изменений $\Delta\theta$:

$$\nabla L(\zeta^*(\theta), \theta) \approx \nabla L(\zeta^*(\theta), \theta + \Delta\theta)$$

В итоге, процесс обучения принимает следующий вид:

- Обучаем дискриминатор при фиксированном генераторе
- Обучаем генератор при фиксированном дискриминаторе
- Повторяем до сходимости параметров обеих моделей

5 pix2pix GAN

Для решения задачи было попробовано применить модификацию GAN-сети по названием "pix2pix GAN". Для нее функционал потерь выглядит следующим образом:

$$L(G, D) = L_{adv}(G, D) + \eta \mathbb{E}_{p_{data}(s_1, s_2, r)}(\| r - G(s_1, s_2) \|_1)$$

$$L_{adv}(G, D) = \mathbb{E}_{p_{data}(s_1, s_2, r)} \log D(s_1, s_2, r) + \mathbb{E}_{p_{data}(s_1, s_2)} \log(1 - D(s_1, s_2, G(s_1, s_2)))$$

где G , D - генератор и дискриминатор, (s_1, s_2, r) - тройка изображений (интенсивность слева, справа и реальное изображение с трендом), $\mathbb{E}_{p_{data}(s_1, s_2, r)}$ - мат. ожидание логарифмического правдоподобия того, что тройка изображений (s_1, s_2, r) принадлежит вероятностному распределению реальных троек $p_{data}(s_1, s_2, r)$, а $p_{data}(s_1, s_2)$ соответствует распределению реальных изображений s_1, s_2 .

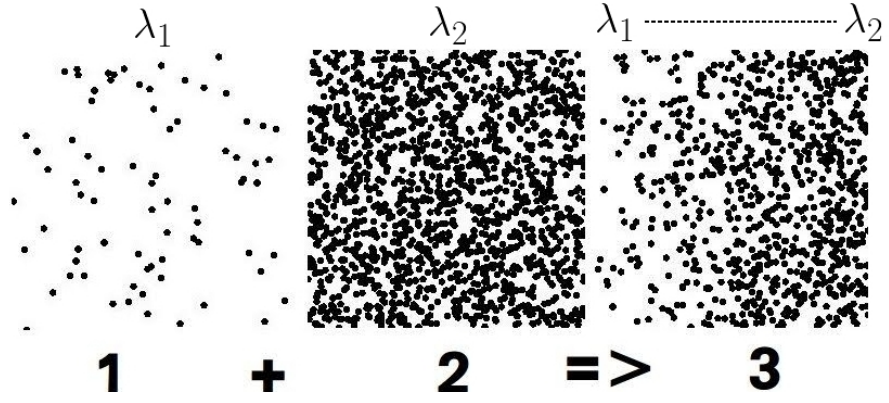


Рис. 3: Вход и желаемый выход нейросети

6 Критерий качества

После обучения генератора, необходимо проверить, что сгенерированные им изображения действительно имеют искомые характеристики. Для этого нужно ввести специальную метрику, которая будет учитывать наличие в изображении тренда интенсивности частиц. Было решено использовать среднюю плотность черных пикселей в некотором окне, и проходить этим окном по изображению (Рис. 4):

$$\xi_k = \frac{1}{Hw} \sum_{i=k}^{k+w} \sum_{j=0}^H \left| \frac{x(i, j) - 255}{255} \right|,$$

$$k = \overline{1, W - w + 1}$$

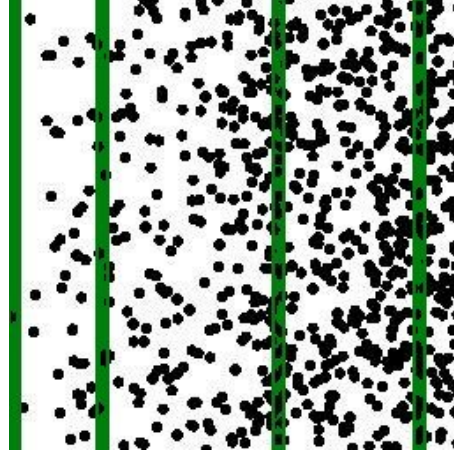


Рис. 4: Прохождение окном, W , H - размеры изображения, w - ширина окна

Построив график $\xi(k)$ можно увидеть, как меняется плотность пикселей и прослеживается ли тренд. В качестве метрики можно взять среднеквадратичную ошибку:

$$\xi = \frac{1}{W - w} \sum_{k=1}^{W-w+1} (\xi_k - \xi_{0k})^2,$$

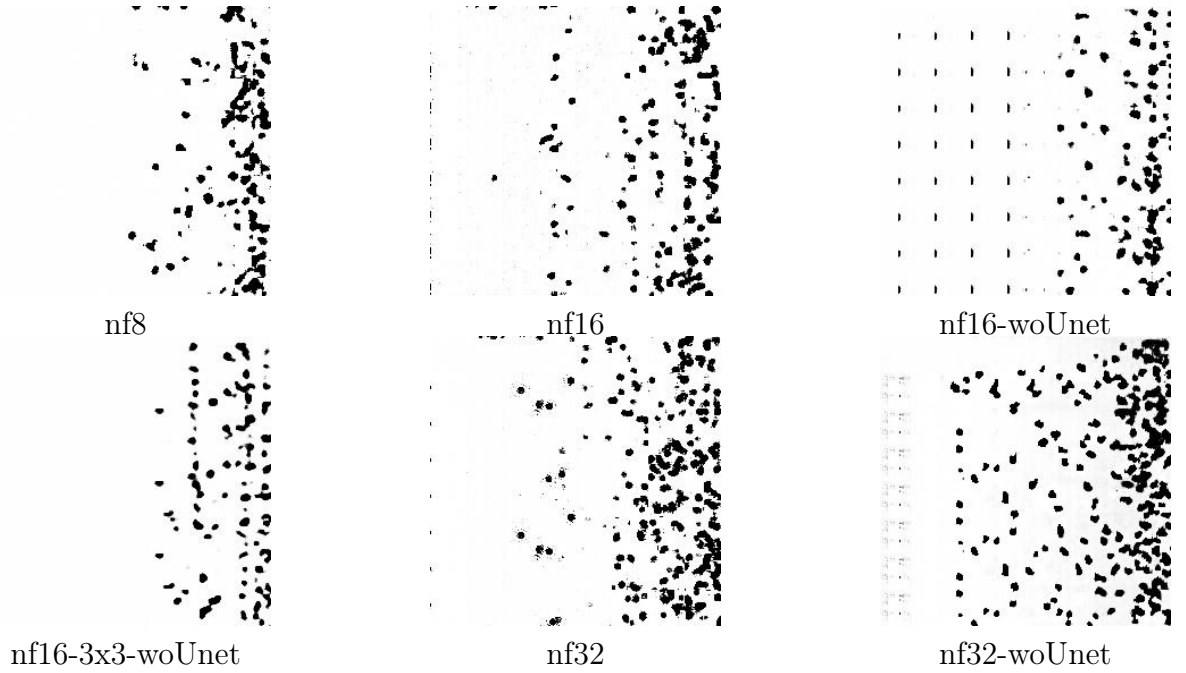


Рис. 5: Примеры сгенерированных текстур

где ξ_{0k} - это ξ_k , усредненное по примерам из обучающей выборки. Соответственно, чем меньше значение метрики, тем лучше тренд, присутствующий на сгенерированном изображении, приближает искомый.

7 Результаты

Было проведено обучение нейросети описанной архитектуры при различных гиперпараметрах. Обучающей выборкой был массив из 3500 троек изображений. Примеры сгенерированных текстур приведены на (Рис. 5). Была подсчитана введенная метрика для сгенерированных наборов текстур. Графики плотности черных пикселей в зависимости от сдвига окна, квадратичного отклонения от тренда и сами значения метрики приведены на (Рис. 6, 7, 8, 9) и в (Таб. 1, 2)

Сеть	Метрика
nf8	0.00825
nf32	0.00549
nf32-woUnet	0.00688

Таблица 1: Значения введенной метрики для разных сетей

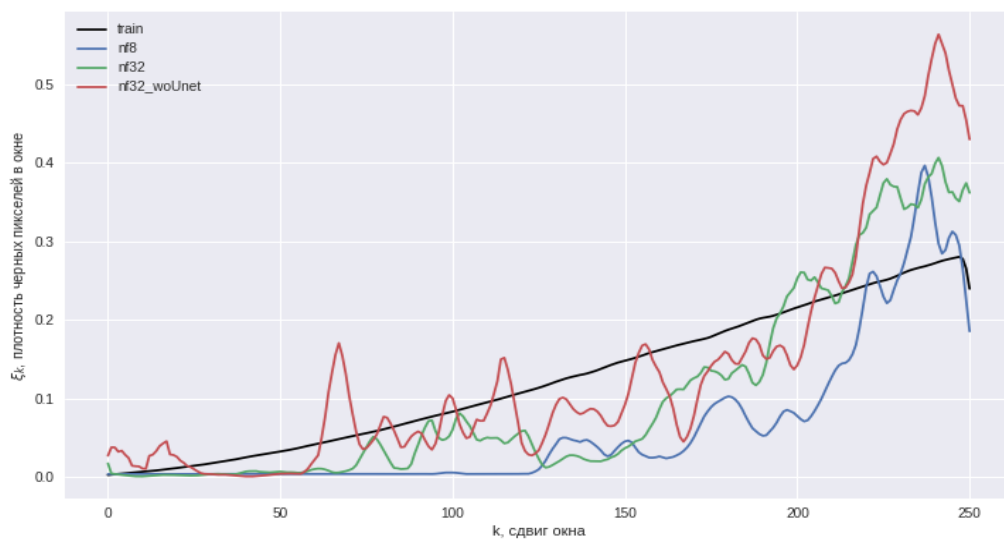


Рис. 6: Плотность черных пикселей в окне в зависимости от сдвига окна

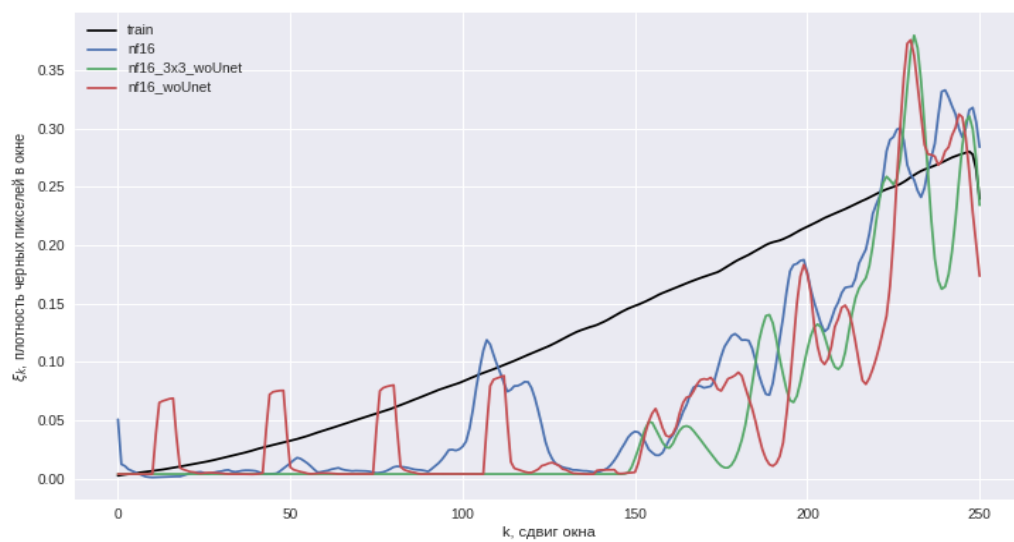


Рис. 7: Плотность черных пикселей в окне в зависимости от сдвига окна

Сеть	Метрика
nf16	0.00606
nf16-woUnet	0.00881
nf16-3x3-woUnet	0.01034

Таблица 2: Значения введенной метрики для разных сетей

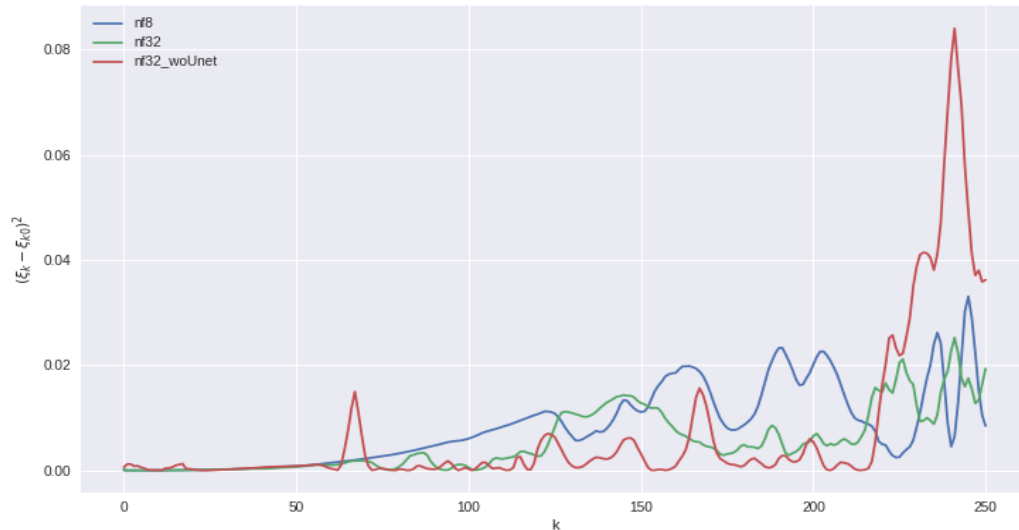


Рис. 8: Квадратичное отклонение от тренда для разных сетей

8 Выводы

В работе было:

- Исследовано применение архитектуры GAN для синтеза текстур с трендами
- Получены результаты синтеза при нескольких наборах гиперпараметров сети
- Проведено измерение качества генерации для каждого из наборов, используя специально введенную метрику

9 Заключение

Полученные результаты показывают, что в принципе нейросеть способна уловить тренд и воспроизвести его, однако на данный момент качество генерации относительно невысоко. Необходимо провести дальнейшее исследование оптимальных гиперпараметров, и, возможно, увеличить объем обучающей выборки. Также можно провести аналогичные эксперименты с другими архитектурами генераторов.

Список литературы

- [1] Воронцов К. В., "Математические методы обучения по прецедентам (теория обучения машин)".

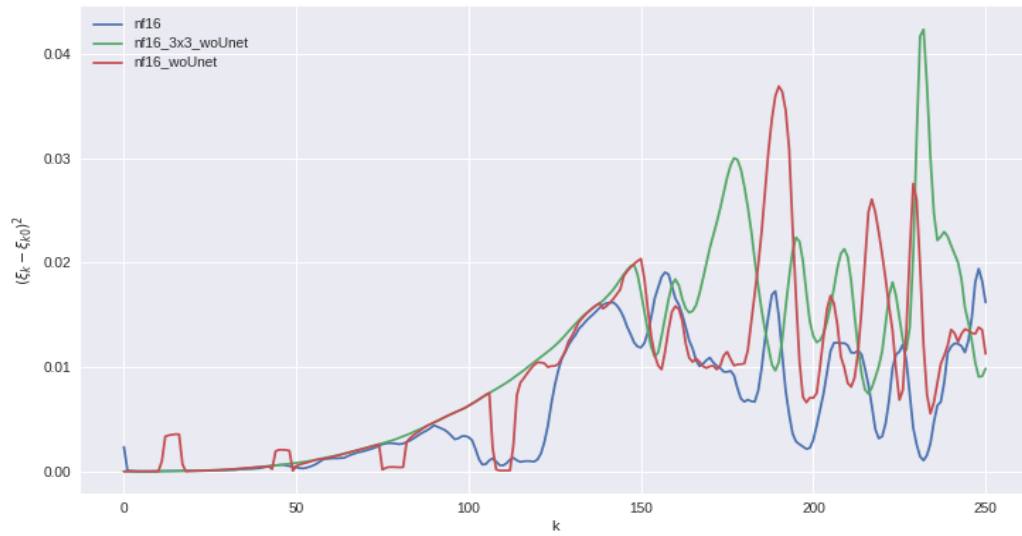


Рис. 9: Квадратичное отклонение от тренда для разных сетей

- [2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Nets". arXiv: 1406.2661 [stat.ML], 2014
- [3] Goodfellow, Ian, et al. "Generative adversarial nets". Advances in neural information processing systems. 2014
- [4] Pedro Costa, Adrian Galdran, Maria Inês Meyer, Michael David Abràmoff, Meindert Niemeijer, Ana Maria Mendonça, Aurélio Campilho, "Towards Adversarial Retinal Image Synthesis". arXiv: 1701.08974 [cs.CV], 2017