

Список используемых источников

1. “Документация базы данных Realm” [Электронный ресурс]. URL: <https://realm.netlify.app/docs/swift/latest/>
2. “Документация языка программирования Swift” [Электронный ресурс]. URL: <https://www.swift.org/documentation/>
3. “Realm tutorial: Getting started” [Электронный ресурс]. URL: <https://www.raywenderlich.com/1464-realm-tutorial-getting-started>
4. “Core Data tutorial: Getting started” [Электронный ресурс]. URL: <https://www.raywenderlich.com/7569-getting-started-with-core-data-tutorial>
5. “SQLite tutorial” [Электронный ресурс]. URL: <https://www.sqlitetutorial.net/>
6. “Серверная сторона Swift – объектно-реляционное отображение” [Электронный ресурс]. URL: <https://medium.com/@mzczachurski/server-side-swift-object-relational-mapping-orm-68879d9a1aa3>

УДК 77.06:004.032.26

Н.С. Павлова

Казанский национальный исследовательский технический университет
им. А.Н. Туполева – КАИ
Казань, Россия

УЛУЧШЕНИЕ КАЧЕСТВА ФОТО С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

Аннотация. В данной статье приведена информация о том, как работают нейронные сети, как можно использовать нейронные сети в работе с изображениями и как в этом может помочь Google Colaboratory.

N.S. Pavlova

Kazan National Research Technical University
named after A.N. Tupolev – KAI
Kazan, Russia

IMPROVING PHOTO QUALITY USING NEURAL NETWORKS

Abstract. This article provides information on how neural networks work, how neural networks can be used in working with images, and how Google Colaboratory can help with this.

Как работают нейронные сети? На данный момент существует разнообразие нейронных сетей и все они имеют общие черты. Они также как и мозг человека, состоят из большого числа связанных между собой однотипных элементов – нейронов, которые имитируют нейроны головного мозга. Нейрон же в сетях – базовая единица, а нейронная сеть — это последовательность нейронов, соединенных между собой синапсами. Нейроны объединяются собой слоями – входной, скрытые и выходной (рис.1). Во входной слой поступают данные, в скрытых производятся вычисления, а в выходном эти данные выводятся. Каждый слой соединяется с двумя соседними слоями – это и есть синапс, у него одно свойство – это вес, то есть как бы своя «важность», придаваемая значению сигнала.

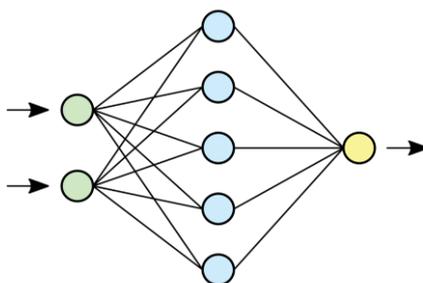


Рис. 1 - Схема простой нейросети

Технология распознавания в нейронных сетях. Рассмотрим простейший тип нейронной сети Перцептрон. Мы можем научить нейросеть распознавать породы собак, выберем для примера – бульдог и ретривер. Для этого мы даём нашей нейросети изображение породы собак – бульдог, она разбирает его на пиксели, обрабатывает через нейроны и синапсы и выдаёт нам результат, но так как наша нейронная сеть ещё не обучена, она не может выдать точный результат. Выходной слой будет выдавать пару чисел (от 0 до 1). Первое число будет (10) это насколько нейронная сеть уверена, что это порода собак – бульдог, а второе число (01) на сколько это ретривер. Поэтому мы сами даём ей значение 10, чтобы она пересчитала все веса своих связей, изменила их под правильный ответ и запомнила эти конкретные веса. Таким образом, нужно загружать больше количество данных и чем больше нейронов в нашей сети, тем точнее она будет выдавать ответ.

Синтез картинки с помощью нейронной сети. Мы рассмотрели, как нейросеть распознает, что находится на картинке, но теперь нам нужно изучить обратную задачу, то есть задачу синтеза, порождения изображений. Нейросеть должна просмотреть большое количество изображений, содержащих какие-то объекты, и, выучив, что это такое, она должна учиться создавать новые изображения.

На вход этой нейросети подается вектор, который кодирует класс, определенный тип объекта, и еще один вектор, который кодирует геометрические параметры камеры. На выходе эта нейросеть синтезирует изображение объекта, отделяющую объект от фона. Отличие такой развернутой нейросети от традиционной — в том, что она всё делает в обратном порядке. Изображение у нас теперь не на входе, а на выходе. И представления, которые возникают в этой нейросети, сначала являются просто векторами, а в некоторый момент превращаются в наборы изображений. Постепенно изображения комбинируются друг с другом с помощью обобщенных сверток, и на выходе получается изображение, то есть нейросеть выдает нам смесь объектов.

На основе этих методов в моей работе представлено улучшение качества изображения с помощью нейросети. Нейросеть будет и реставрировать фото и улучшать качество, каким образом это происходит? Для того, чтобы нейронная сеть нашла дефекты на фотографии, она должна сгенерировать маску, где будет отмечать повреждения фотографии, нужно её обучить путём загрузки большого количества фотографий и готовых масок, где уже обозначены дефекты и для того, чтобы научить нейронную сеть реставрировать эти фотографии, мы так же подаём изображение и маску, на которой единицами отмечаем чистые пространства, а нолями — те пиксели, которые хотим закрасить. Чтобы проще и быстрее обучать нейросеть, лучше искусственно добавлять дефекты, которые похожи по форме на те, что встречаются в реальной жизни и после этого обучать сеть восстанавливать недостающие части.

Для того, чтобы нейронная сеть работала быстро я использую Google Colab. Это облачная среда для работы с кодом от Google, она дает возможность работать с Python-библиотеками и её главная отличительная черта - бесплатные мощные графические процессоры GPU и TPU, благодаря которым можно заниматься исследованиями в области машинного обучения.

TPU — тензорный процессор, разработка Google. Он предназначен для тренировки нейросетей. У этого процессора в разы выше производительность при больших объемах вычислительных задач.

GPU - графический процессор, с помощью него обработка данных занимает минимальное время, так как процессор выполняет задачи параллельно, исключительно с графикой, поэтому на нем удобнее работать с изображением, что я и использую в данном случае.

Для того, чтобы работать с GPU, нужно изменить среду выполнения. Далее мы запускаем по очереди ячейки с кодом, подключаем все функции и библиотеки, программа показывает уже готовые изображения, тем самым демонстрируя функции, которые у неё есть. Это denoising(шумопонижение), tone readjustment(регулировка тона), face enhancement(улучшение лица) и physical damage restoration (восстановление физических повреждений).

Загрузка своей фотографии. Для того, чтобы нейронная сеть понимала, что ей нужно восстанавливать физические повреждения, у неё настроен код "--with_scratch". Запускаем ячейку "Try it on your own photos", она позволяет загружать своё изображение. В своей работе я буду использовать старую фотографию (рис.2), где видна структура бумаги, шум на фото и т.д.



Рис. 2 - Исходное изображение

Загружаем фотографию из своего компьютера. Нейронная сеть начинает обрабатывать изображение, создаёт маску повреждений, чтобы реставрировать фото, распознаёт лицо, выполняет улучшения лица.



Рис. 3 - Результат работы нейросети

В ходе моей работы были рассмотрены нейронные сети, показано как они могут работать и как их применять, они не только предназначены для распознавания изображений и текстов, но и для улучшения качества изображения. Благодаря применению бесплатной облачной среды Google Colab получилось за короткое время улучшить качество фотографии без ручной обработки фотографии, структура и интерфейс платформы довольно просты, поэтому данную работу может выполнять любой человек без глубокого изучения нейронных сетей.

Список использованных источников

1. Тарик Рашид «Создаем нейронную сеть», 2018.
2. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. - М.: Диалектика, 2019. - 1104 с.
3. Mohamad H.Hassoun. Fundamentals of Artificial Neural Networks. MIT Press, Cambridge, Massachusetts, 1995.
4. Google Colaboratory (Colab) - <https://blog.skillfactory.ru/chto-takoe-google-colaboratory-i-komu-on-nuzhen/>
5. Google Colaboratory (Colab) - <https://colab.research.google.com/notebooks/welcome.ipynb?hl=ru>