

Стилизация фотографий под картину

Антон Мордберг
Семен Поляков

Научный руководитель: Андроник Ордян

СПбАУ
21 декабря 2015

Задача

- Написать мобильное приложение для создания фотографий, стилизованных под картины известных художников
- Проблема:
 - Современные телефоны не могут производить нужные вычисления быстро
- Решение:
 - Разделить задачу (тонкий android-клиент и толстый сервер с машинным обучением)
 - Создать централизованное хранилище для результатов

Что сделал Антон

- Сервер
 - Database (PostgreSQL + SQLite)
 - Spring framework

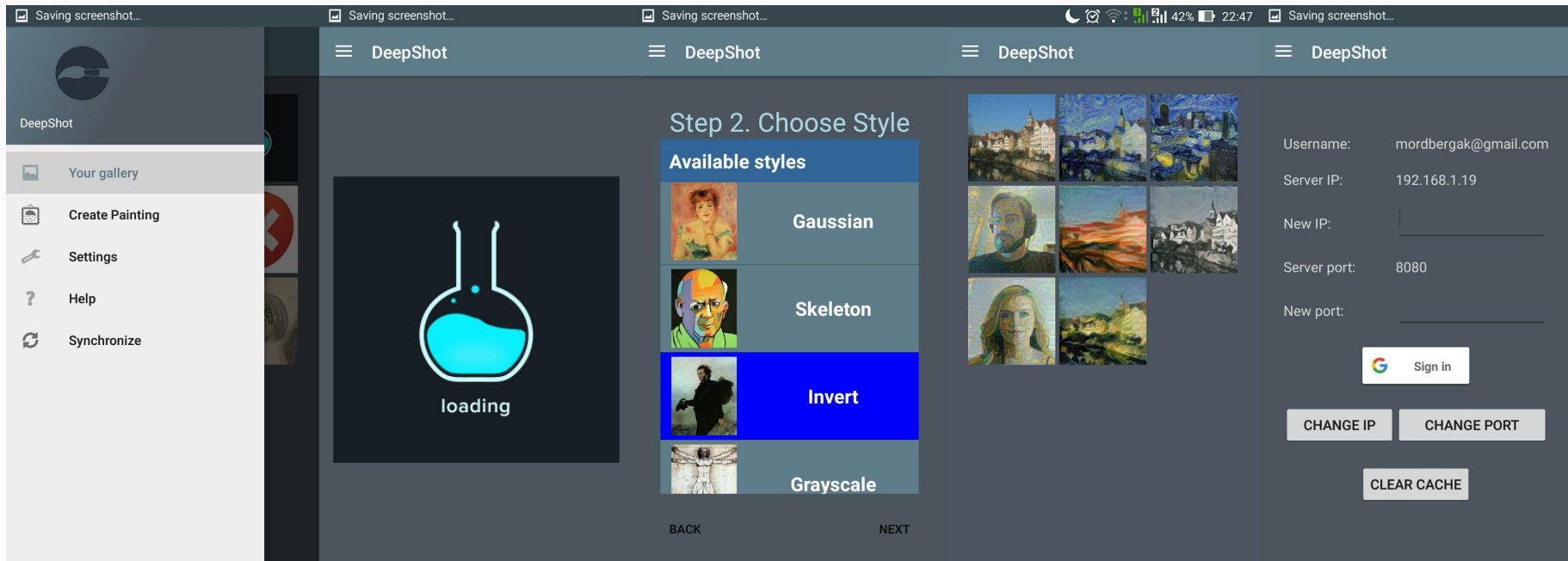
- Клиент
 - Android (поддержка начиная с Android 3.0)
 - Авторизация и оповещения (Google API)
 - Кеширование результатов
 - Дружелюбный интерфейс (Material design)

Сервер

- Rest API, которым пользуются клиенты
- Работа с базой данных
 - Запросы пользователей
 - Результаты (статус + изображение)
 - Список картин
- Распределение задач по потокам
 - `CachedThreadPool`
 - Управлением ресурсами
- Передача аргументов задачи нейронной сети
- Оповещение клиентов о результате (GCM)

Клиент

- **Галерея получившихся изображений**
 - Обновление при получении GCM сообщения
 - Кеширование всех изображений на телефоне
- **Интерфейс для создания новых изображений**
 - Выбор исходного изображения (файл / камера)
 - Выбор “стиля” (из тех, что есть на сервере)
- **Минимальный набор настроек (Имя пользователя / адрес сервера)**



1. Меню

2. Обработка

3. Выбор стиля

4. Галерея

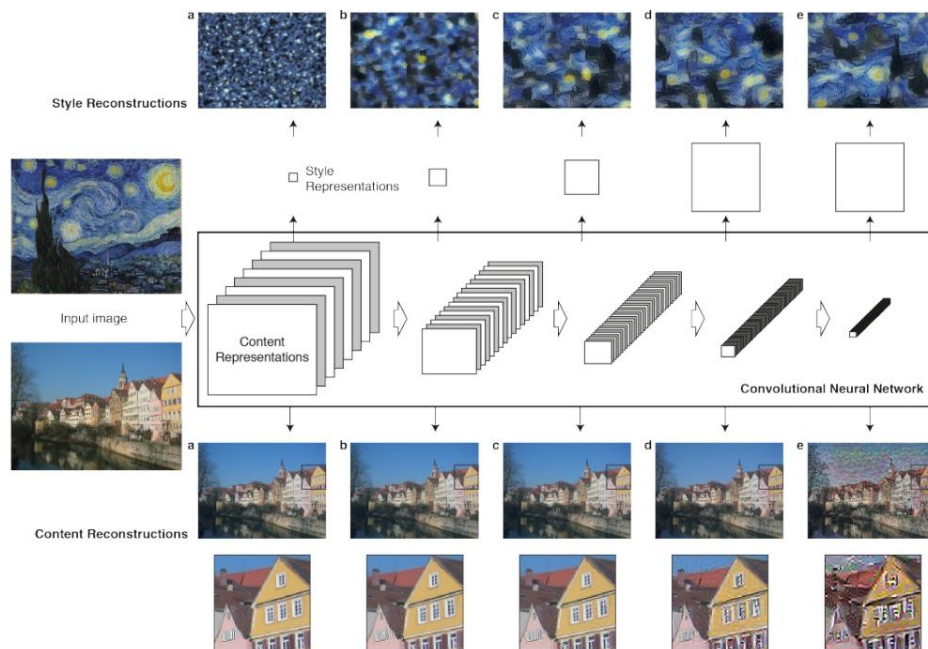
5. Настройки

Что сделал Семён

- Разобрался в устройстве сверточных нейронных сетей
- Разобрался в устройстве и работе нейронных сетей во фреймворке Caffe
- Реализовал алгоритм на основе статьи **A Neural Algorithm of Artistic Style***
- Добавил в python интерфейс Caffe методы для сброса кеша нужных блоков данных в нейронной сети

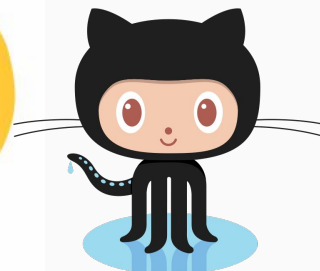
* - <http://arxiv.org/abs/1508.06576>, Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

Архитектура нейронной сети

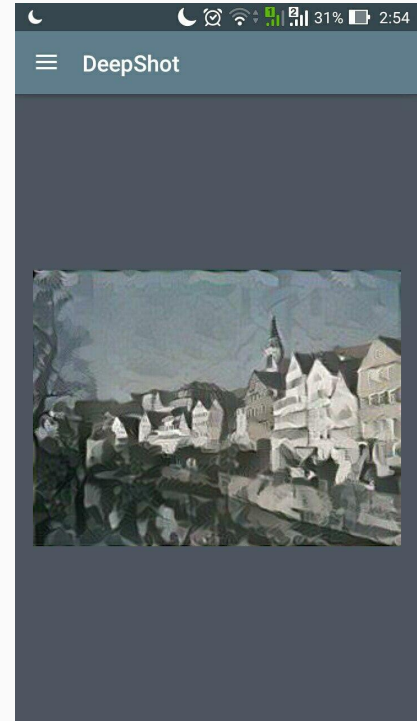
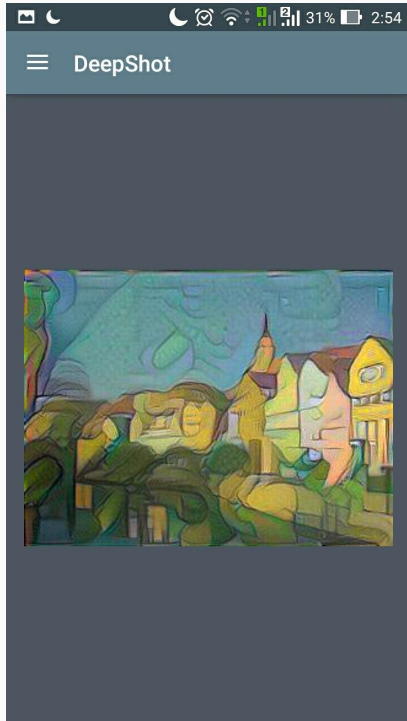


Используемые технологии

1. Python (+ numpy, scipy modules)
2. Caffe framework
3. Google Protobuf
4. CUDA v7.0
5. GitHub



Результаты



Спасибо за внимание!

Q&A?