

Детектор рептилоидов

Олег Свидченко, Татьяна Белова

Руководители: Богдан Бугаев, Антон Крыщенко

31 мая 2016 г.



Цель проекта

Создать мобильное приложение, которое делает фотографию человека и распознает на ней рептилоида с помощью компьютерного зрения

Задачи

1. Создать android-приложение

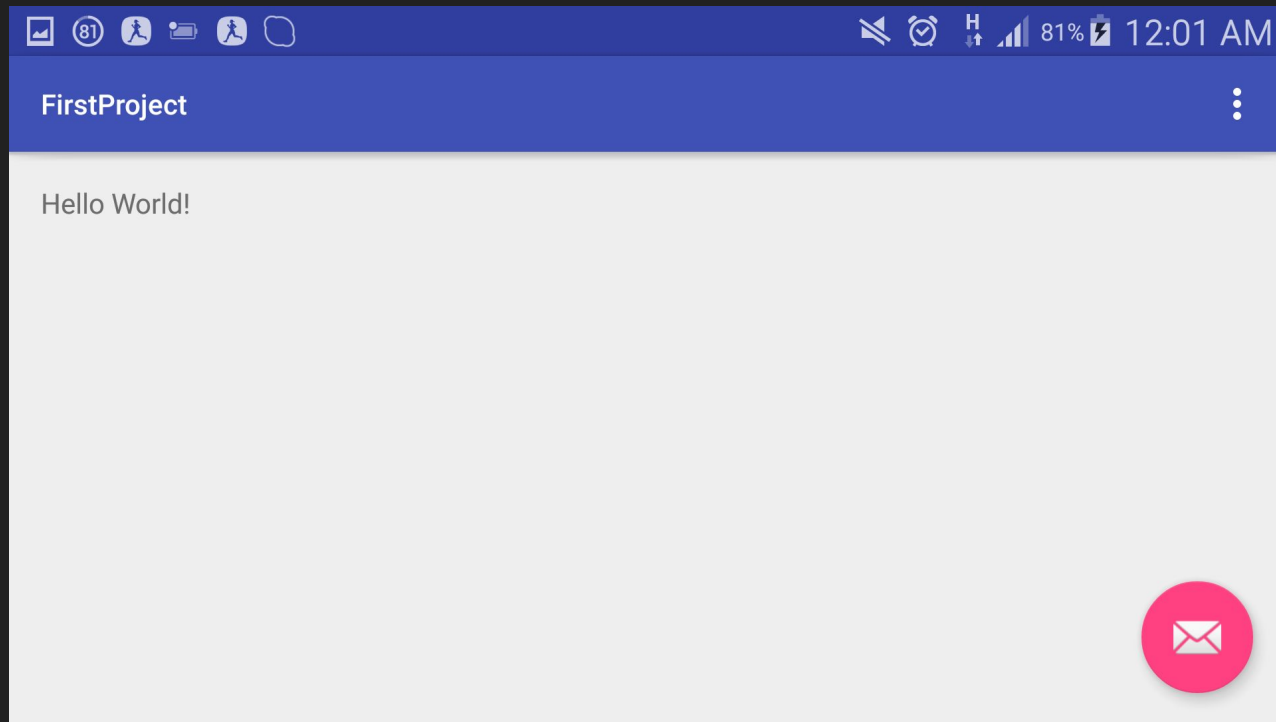
- a. научиться работать с камерой
- b. применить фильтр к полученному изображению
- c. показать результат

2. Написать фильтр для изображения

- a. Определить положение лица на фотографии, а также глаз, носа и рта.
- b. В полученной области лица выделить кожу
- c. Наложить маску рептилоида на выделенную область

Решение поставленных задач

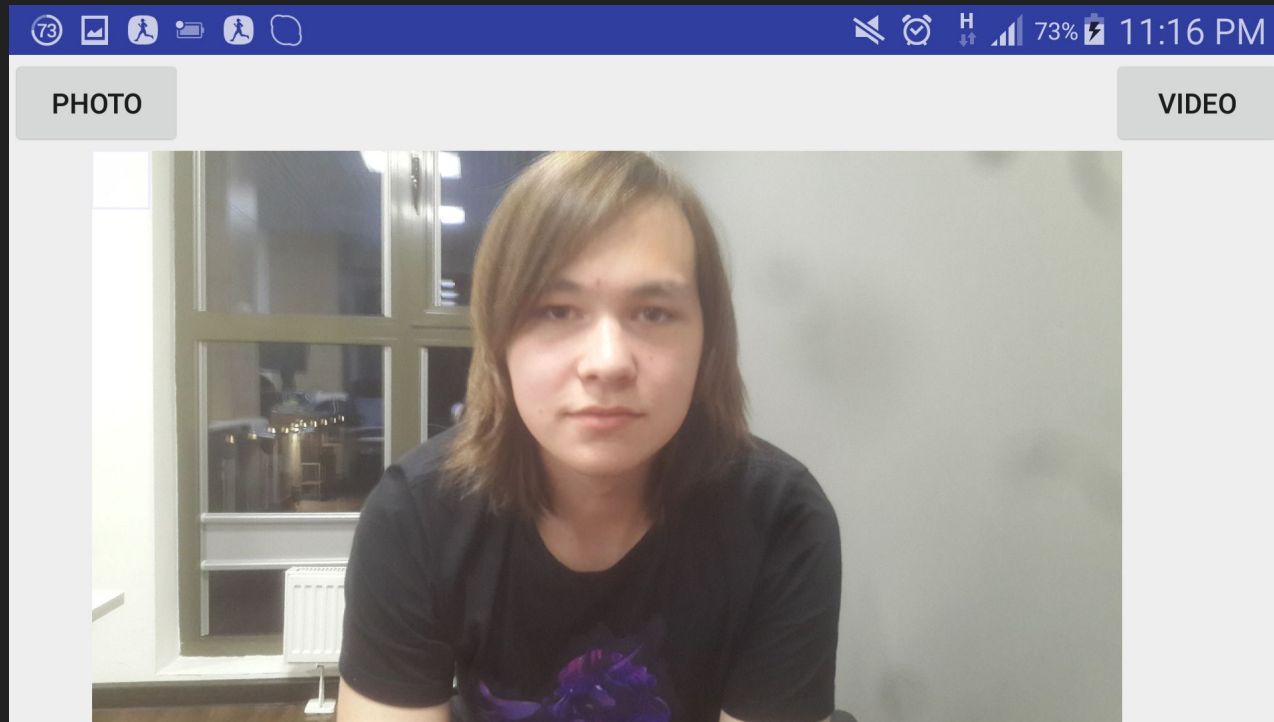
Hello world



«Hello World!»

- Настроили Android Studio
- Разобрались с Android API

Camera



- Запуск стандартной камеры
- Создание интерфейса
- Вывод изображения

Определение лица и его черт по фотографии

1. Используем метод Виолы — Джонса для нахождения лиц и их черт на фотографии

Качество распознавания из коробки оставляет желать лучшего.

Определение лица и его черт по фотографии

1. Используем метод Виолы — Джонса для нахождения лиц и их черт на фотографии
2. Подгоняем константы и закручиваем гайки
 - a. Должно быть распознано хотя бы три черты лица
 - b. Левый глаз может находиться только в левой верхней части лица
 - c. Правый глаз - в правой верхней
 - d. Рот находится в нижней части лица
 - e. Нос находится примерно в середине лица

Выделение кожи

1. Выделение только по цвету
2. Заливка от определенных точек
 - а. улучшение — использование ребер

Выделение кожи

1. Выделение только по цвету

Алгоритм: Мы считаем, что пиксели по центру изображения точно принадлежат коже. Найдем средний цвет этих пикселей и скажем, что все точки на изображении, цвет которых не сильно отличается от него, тоже принадлежат коже.

Выделение кожи

1. Выделение только по цвету

Проблема данного метода в том, что могут найтись точки, не принадлежащие коже. В то же время мы можем не найти точки на коже, цвет которых сильно отличается от тех, что мы брали за основу.

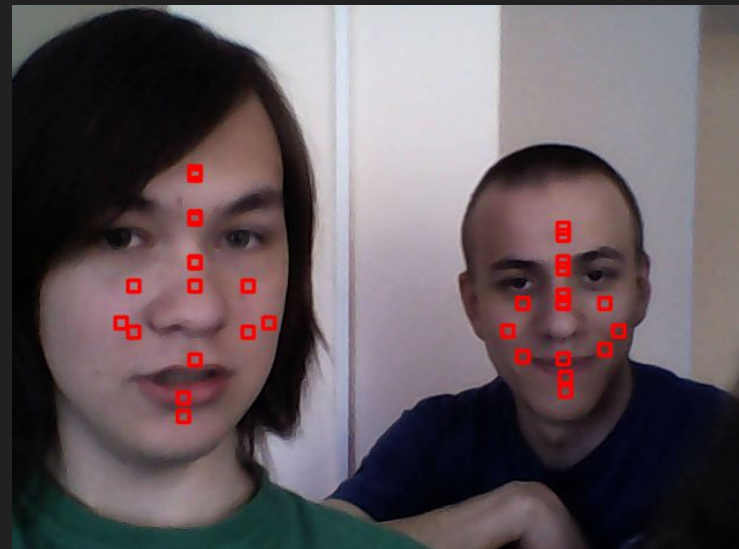
Выделение кожи

1. Выделение только по цвету
2. Заливка от определенных точек

Алгоритм: Начинаем обход из нескольких точек на лице

Проблемы: не работающие функции в JavaCV

Решение: написать свои



Выделение кожи

1. Выделение только по цвету
2. Заливка от определенных точек

Минусы этого способа в том, что мы все еще можем случайно найти области изображения, не принадлежащие коже.

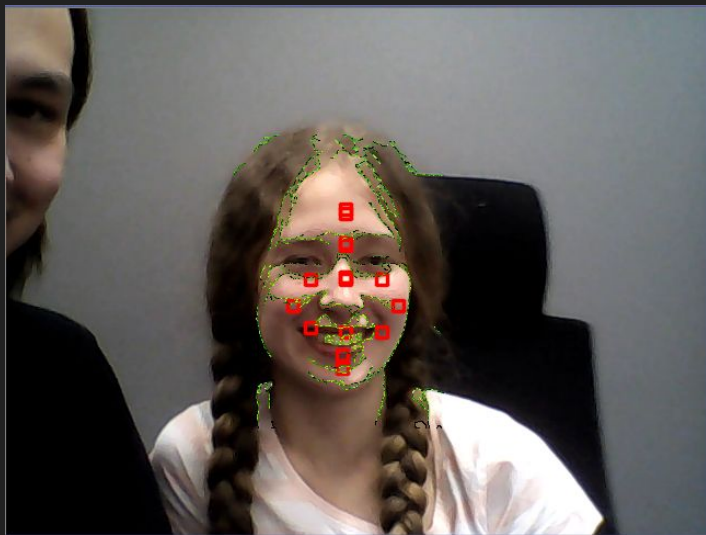
Выделение кожи

1. Выделение только по цвету
2. Заливка от определенных точек
 - а. Улучшение - Использование ребер

Алгоритм: Выделим кривые, в которых происходит резкое изменение яркости, и запретим алгоритму проходить через них

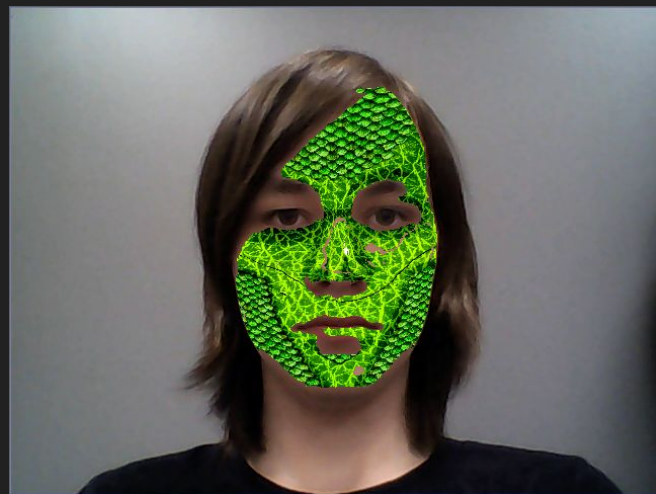


Выделение кожи



Точки и ребра на лице

Выделенная кожа



Фильтры: делаем рептилоида рептилоидом

Что хотим?

Хотим, прежде всего, сделать рептилоида рептилоидом. А у рептилоидов, как известно, кожа чешуйчатая. Поэтому просто покрасить кожу в зеленый цвет - не лучшее решение. Нужно накладывать текстуру на лицо.

Фильтры: делаем рептилоида рептилоидом

1. Аффинные преобразования

Просто взять и наложить чешую поверх лица - не лучшая идея.

Используем для “поворота” чешуи аффинные преобразования: точки лица у нас уже есть, точки текстуры мы выбираем сами.

Проблема: метод `Mat.put` есть только в документации.

Решение: использовать альтернативный конструктор матрицы, после чего поменять ее размер методом `Mat.reshape`

Фильтры: делаем рептилоида рептилоидом

1. Аффинные преобразования
2. Тени

Все хорошо, но текстура получилась плоской. Нужно добавить теней.

Возьмем серое изображение лица, инвертируем его и вычтем с коэффициентом из текстуры.

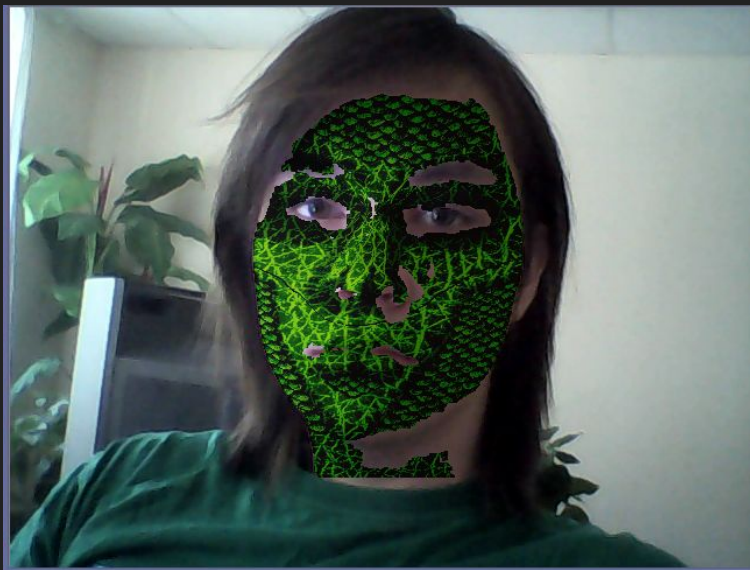
Фильтры: делаем рептилоида рептилоидом

1. Аффинные преобразования
2. Тени
3. Полупрозрачность

Относительно простой алгоритм выделения кожи, как оказалось, оставляет много “дырок”, из-за чего получившаяся композиция выглядит не лучшим образом.

Решение: добавить полупрозрачность текстуре.

Фильтры: делаем рептилоида рептилоидом



Прозрачности еще нет, зато есть тени.

Прозрачность и тени. Результат на лице.



Результаты

Фильтр для фотографии реализован полностью.

Андроид-приложение реализовано, фильтр импортирован, но осталась ошибка, из-за которой фильтр не отображается на телефоне.



Что мы узнали

1. Написали приложение под Андроид
2. Познакомились с языком Java
3. Поигрались с OpenCV
4. Узнали разные алгоритмы выделения кожи
5. Научились работать с Git и Gradle

Спасибо за внимание!

<https://github.com/ReptoidBusters/reptilian-detector>