МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»

Институт ИТКН

Кафедра инженерной кибернетики

Направление подготовки: «01.03.04 Прикладная математика»

Квалификация: бакалавр

Группа: БПМ-18-2

Отчёт по курсовой работе

по дисциплине
«Методы и средства обработки изображений»
на тему : «Распознавание «пальцев» на банкнотах»

Студент (ы) _____/Самсонова Екатерина Олеговна/
подпись (ф.и.о. полностью)

Руководитель _____/доцент, к.т.н. Полевой Д. В./
подпись должность, уч. степ. Фамилия И.О.

Оценка: _____

Дата: _____

Москва 2021

Оглавление

Актуальность ... 3
Содержательная постановка задачи ... 3
Алгоритм FLANN (Fast Library for Approximate Nearest Neighbors) ... 4
Алгоритм SURF (Speeded Up Robust Features) ... 4
Запуск программы ... 6
Работа программы ... 7
Выводы ... 9
Список литературы ... 10

Актуальность

Миллиарды людей ежедневно пользуются банкоматами для выдачи наличных денег или же для пополнения карты ими. Но наверняка только единицы задумываются на тему : «Почему банкомат не прижимает руки? Как он распознает, когда можно закрывать ячейку для приема купюр?» Я была именно из числа людей, которые не смогли оставить эти вопросы без ответов.

Главная цель данной работы – разобраться, с помощью каких алгоритмов возможно выявить наличие пальцев на купюрах, а также самостоятельно создать программу, способную воспроизвести данные алгоритмы. В процессе исследования я узнала, что похожие технологии используются и для создания искусственного зрения в роботах, беспилотных машинах и т.п.

Содержательная постановка задачи

В широком смысле с помощью SURF и FLANN мы можем решить задачу поиска фиксированного изображения.

Обнаружение объектов (препятствий) в зоне работы искусственного зрения банкомата является крайне актуальной и сложной задачей. В работе предложено применение для нахождения соответствия изображений в коллекции образов алгоритма SURF (Speeded Up Robust Features) и метода поиска ближайшего соседа в библиотеке «FLANN - Fast Library for Approximate Nearest Neighbors». Использование библиотеки FLANN совместно с алгоритмом SURF позволяет удовлетворить требованиям по обнаружению объектов в режиме реального времени. Результаты экспериментов показывают эффективность предложенного подхода при использовании в составе системы технического зрения мобильной робототехнической платформы.

Задача состоит в реализации алгоритмов FLANN и SURF и тестировании их эффективности для разных типов видео.

Алгоритм FLANN (Fast Library for Approximate Nearest Neighbors)

Мы можем решить проблему поиска ближайших соседей (NSS)следующим образом : учитывая набор точек P = $p_1, p_2, \dots, p_n$ в метрическом пространстве X, эти точки должны быть предварительно обработаны таким образом, чтобы при наличии нового запроса $q \epsilon X$ найти точку в P, ближайшую к q, можно было быстро. Проблема поиска ближайших соседей является одной из важнейших в различных приложениях, таких как распознавание изображений, сжатие данных, распознавание и классификация шаблонов, машинное обучение, системы поиска документов, статистика и анализ данных. Однако решение этой проблемы в пространствах с высокой размерностью представляется очень сложной задачей, и нет алгоритма, который выполнял бы ее значительно лучше, чем стандартный поиск грубой силы. Это привело к росту интереса к классу алгоритмов, которые выполняют приближенный поиск ближайших соседей, который оказался достаточно хорошим приближением в большинстве практических приложений и в большинстве случаев на порядки быстрее, чем алгоритмы, выполняющие точный поиск. FLANN (Быстрая библиотека для приблизительных ближайших соседей) - это библиотека для выполнения быстрого приблизительного поиска ближайших соседей. FLANN написан на языке программирования C++. FLANN можно легко использовать во многих контекстах через привязки C, MATLAB и Python, поставляемые с библиотекой.

Алгоритм SURF (Speeded Up Robust Features)

Метод SURF ищет ключевые точки и строит описание найденных ключевых точек через дескрипторы особенностей и является аналогом метода SIFT. Ключевой точкой является локальный экстремум детерминанта матрицы Гессе. Для двумерного случая детерминант матрицы Гессе определяется следующим образом:

$$\det(H) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2,$$

где $H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$ — матрица Гессе; $f(x,y)$ — функция изменения градиента яркости [5].

Гессиан инвариантен к сдвигу яркости изображения и повороту, но не инвариантен к масштабу. Данную проблему решают с помощью последовательного перебора различных масштабов и фильтров и поочередного применения их к одному пикселю. В качестве опорных точек выбирают локальные максимумы гессианов, соответствующие локальным максимумам изменения градиента яркости. После нахождения точек локальных максимумов определяют точка истинного максимума гессиана . Данная методика гарантирует, что в окрестности ключевой точки расположены участки с разными

градиентами. Дескриптор представляет собой массив чисел, определяющих опорную точку. Инвариантность дескриптора относительно поворота обеспечивается дисперсией (различием) дескрипторов для разных особых точек.

Для определения особых точек в методе SURF используется целочисленная аппроксимация детерминанта blob-детектора гессиана, который вычисляется с помощью трех операций с использованием предварительно вычисленного интегрального изображения. Blob-детектирование — выявление областей в цифровом изображении, которые отличаются по своим свойствам, таким как яркость или цвет, от прилегающих областей.

К недостаткам метода можно отнести то, что SURF используется для поиска объектов на изображении, но не работает непосредственно с самими объектами. Кроме того, область применения данного метода ограничена в связи с тем, что он в большинстве случаев не может быть использован для распознавания объектов простой формы и без ярко выраженной текстуры.

Сравнивая дескрипторы, полученные с разных изображений, можно определить совпадающие пары, и на основе этой информации сделать выводы об обнаружении искомого фрагмента на снимке.

Все обнаруженные ключевые точки можно подразделить на два типа:
а) точки, удовлетворяющие модели, «не-выбросы» (англ. inlier);
б) ложные точки, шумы — случайные включения в исходные данные, «выбросы» (англ. outlier).

Таким образом, алгоритм работы SURF предусматривает выполнение следующих этапов:
– масштабно-пространственное представление;
– расчет значений гессиана;
– поиск точек локальных максимумов;
– определение точки истинного максимума;
– определение ориентации опорной точки;
– формирование дескриптора опорной точки.

SURF-декрипторы используются для обнаружения и распознавания объектов, людей, жестов или лиц, для восстановления 3D-сцен, отслеживания объектов на видео и извлечения координат точек, представляющих интерес.

Запуск программы

1. Скачиваем проект из репозитория https://github.com/kate-coder/Kursovaya_finger.git
2. Запускаем проект в CLion
3. В верхней панели необходимо добавить Configurations (Рисунок 1)

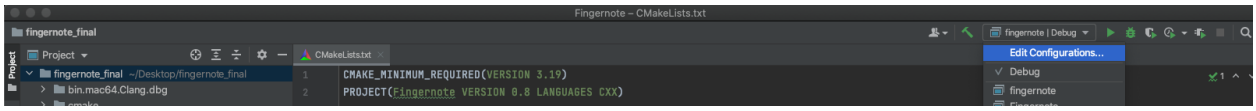

Рисунок 1

4. Далее прописываем Program arguments (Рисунок 2):
-v=./res/video/video_2.MOV -i=./res/banknotes/500_2.jpg

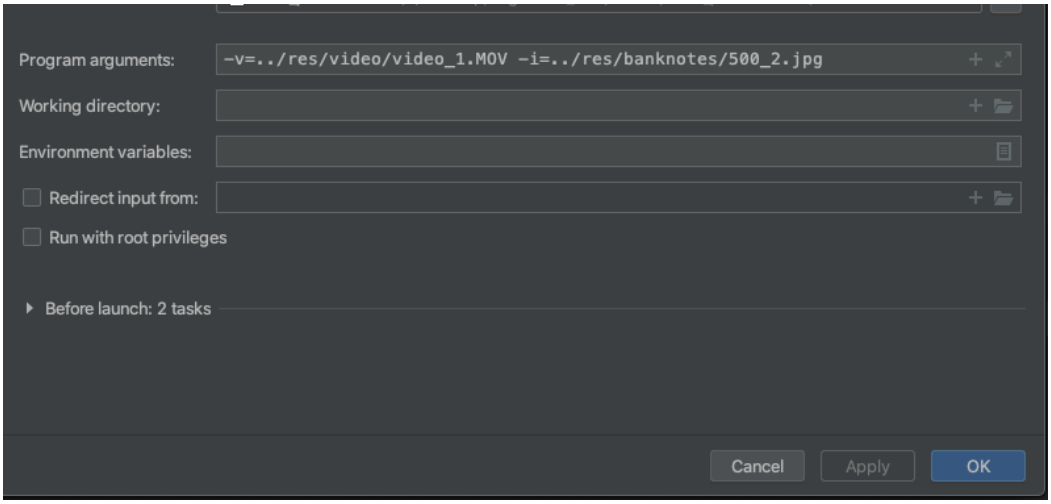

Рисунок 2

5. Аргументы можно изменять (Все доступные видео и банкноты можем увидеть в левой панели в папках Рисунок 3)

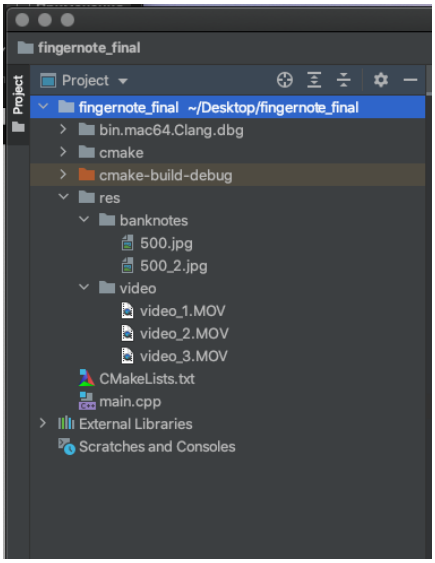

Рисунок 3

6. Запускаем программу, после чего на экране появится видео с нахождением пальцев (Рисунок 6)

Работа программы

1. Запрашиваем пути к двум изображениям: объекту (искомое изображение) и сцене (место, где ищут объект)
2. Находим детекторы и ключевые точки у примера банкноты с помощью SURF;
3. После шага 2 идет поиск пальцев на изображении (пиксели с характерным для кожи цветом)
4. Далее ищем дескрипторы и ключевые точки в изображении
5. Соотносим дескрипторы банкноты с изображением
6. После шага 5 отбрасываем «плохие» совпадения (Если «хороших» совпадений больше минимального необходимого кол-ва, то на изображении присутствует банкнота)
7. Находим границы купюры на изображении посредством преобразования перспективы (Рисунок 4)
8. Заполняем найденную область банкноты
9. Находим пересечение масок пальцев и банкноты на изображении (Если пальцы составляют больше определенного процента на маске банкноты, то это значит, что они лежат на ней и окрашиваются в красный оттенок)

Рисунок 4 (определение области работы программы, т.е. сцены)

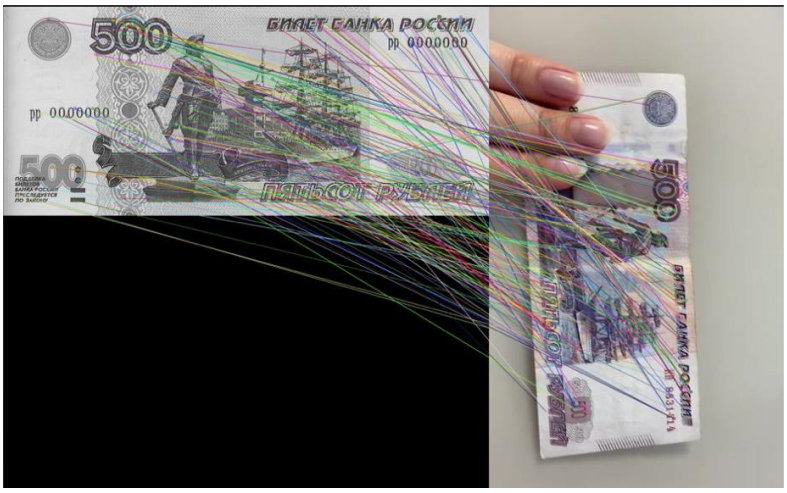

Рисунок 5 (наглядное представление работы программы во время поиска соответствий объекта и сцены)

Рисунок 6 (Вывод программы в случае отсутствия пальцем на видео)

Рисунок 7 (Найденные пальцы окрашены в красный цвет)

Для проверки точности работы программы запустим ее с другим видео, где пальцы передвинуты в другой край (Рисунок 5)

Рисунок 8 (Найденные пальцы окрашены в красный цвет)

Выводы

В ходе работы была разработана и протестирована программа для нахождения пальцев на банкнотах. Был проведен анализ работы алгоритмов SURF и FLANN для нахождения объектов на разных видео, а также были сделаны выводы по эффективности программы.

В процессе исследования моей работы, стоит отметить, что программа практически не способна понять, что ногти – это тоже часть пальцев. Скорее всего, это связано с тем, что на видео ногти блестят и создают блики, а также они отличаются от цвета кожи.

Данная программа и алгоритмы, используемые в ней, при доработке могут использоваться в самых важных и инновационных сферах жизни, таких как : распознавание лиц и создание беспилотного транспорта и многих других.

Список литературы

1. Репозиторий проекта [Электронный ресурс]. – Режим доступа: https://github.com/kate-coder/Kursovaya_finger.git
2. Национальная библиотека им. Н.Э.Баумана [Электронный ресурс]. – Режим доступа: https://ru.bmstu.wiki/index.php?title=Распознавание_изображения_SURF-методом&mobileaction=toggle_view_mobile. – Дата обращения: 01.05.2021.
3. Русские блоги [Электронный ресурс]. – Режим доступа: https://russianblogs.com/article/2466789331/. – Дата обращения : 01.05.2021.
4. OpenCV [Электронный ресурс]. – Режим доступа: https://docs.opencv.org/master/df/dd2/tutorial_py_surf_intro.html. – Дата обращения: 30.04.2021.
5. Пименова, М.Б. Распознавание характерных объектов на местности с использованием алгоритма SURF / М.Б. Пименова // Политехнический молодежный журнал. – 2019. – № 10. – С. 13