

Отчет по лабораторной работе №2

курс «Обработка и распознавание изображений»

кафедра ММП ВМК МГУ

Михеев Борис, 317 группа

8 мая 2022 г.

Постановка задачи

Цель работы - разработать и реализовать программу для классификации изображений ладоней. Требования к программе:

- Ввод и отображение на экране изображений.
- Классификация изображений ладоней.
- Определение «позы» ладоней по расположению сомкнутых пальцев. Пальцы нумеруются от 1 до 5, начиная с большого пальца против часовой стрелки. Поза ладони описывается кодом $1*2*3*4*5$, где символ «*» обозначает «-», если пальцы разомкнуты, или «+», если они прижаты друг к другу.
- Построение линии пальцев - ломанной линии, соединяющей точки на кончиках пальцев с точками в основаниях пальцев.
- Вход программы - набор изображений ладоней, выход - код позы ладони и координаты вершин линии пальцев: набор вершин, отвечающий кончикам пальцев, затем набор вершин, отвечающий основаниям пальцев, все в порядке, начиная с большого пальца. Вывод результатов осуществляется в .txt-файл согласно указанному в формулировке задания формату. Также необходимо визуализировать построенную линию пальцев для входного изображения.

Описание данных

Входные данные - цветные изображения левых ладоней людей, полученные с помощью сканера. Набор состоит из 67 изображений формата .tif размера 489×684 с разрешением 72 дп. Ладони на изображениях имеют разный размер, форму и ориентацию. Пальцы могут быть сомкнуты или разомкнуты. Также на изображениях могут присутствовать фрагменты одежды людей (рукава рубашек, браслеты и т. д.), и участки изображений могут иметь различную яркость. Предполагается, что разрабатываемая программа должна адекватно работать с такими особыенностями входных изображений. Стоит отметить, что почти на всех изображениях фон темный и не содержит посторонних объектов, что упрощает задачу распознавания ладоней.

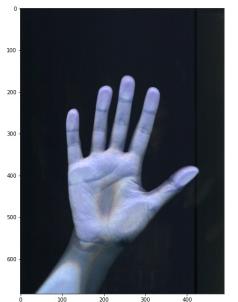


Рис. 1: Примеры входных изображений, предоставленных для отладки и тестирования. Ладони имеют разный размер, позу, форму и ориентацию, а также на снимках присутствуют фрагменты одежды.

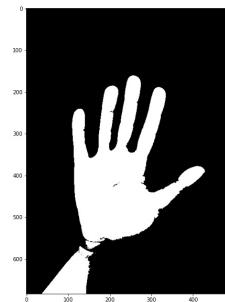
Описание метода решения

Решение задачи разделяется на следующие основные этапы. Сначала строится линия пальцев, затем по ней определяется поза ладони.

Рассмотрим процесс решения более детально. Исходное изображение переводится в черно-белое, затем бинаризуется. Программа позволяет указывать различные значения порога бинаризации для получения более качественного результата в зависимости от входного изображения.



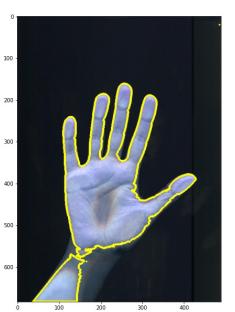
(a) Исходное изображение



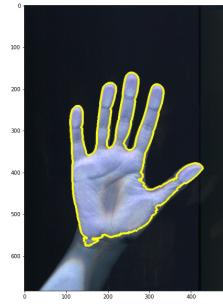
(b) Бинаризованное изображение

Рис. 2: Бинаризация исходного изображения.

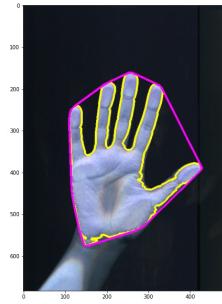
Затем строятся контуры областей на изображении. Хотя фон темный и не содержит сторонних объектов, на некоторых снимках присутствуют элементы одежды людей или части их запястий. Поэтому в результате могут быть получены несколько контуров, а не один связный. Среди них выбирается самый большой по длине, исходя из логичного и интуитивного предположения, что ладонь является ключевым объектом и самой большой связной областью на изображении. Можно выбирать и контур с самой большой охватываемой площадью, но экспериментально было установлено, что это приводит к аналогичным результатам. Далее для найденного контура ладони строится выпуклая оболочка. Ее точки упорядочиваются против часовой стрелки.



(а) Все контуры на изображении



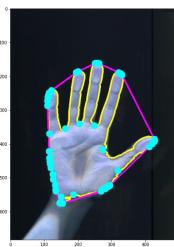
(б) Контур ладони, выбранный как самый длинный среди всех



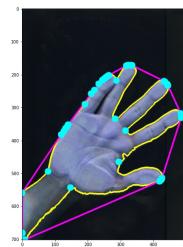
(с) Выпуклая оболочка контура ладони

Рис. 3: Выделение контура ладони и построение его выпуклой оболочки.

Для дальнейшего определения кончиков и оснований пальцев воспользуемся т. н. дефектами выпуклости, т. е. точками контура, наиболее отдаленными от двух последовательных точек его выпуклой оболочки. Каждый дефект выпуклости состоит из набора трех точек, образующих треугольник. Однако найденный контур ладони неидеален и может иметь сильно искривленную форму, особенно если самый длинный из контуров изображения захватывает ладонь вместе с запястьем или фрагментами одежды. Таким образом, дефектов выпуклости может быть больше, чем число промежутков между пальцами.



(а) Дефекты выпуклости контура ладони для случая, когда контур охватывает лишь ладонь



(б) Дефекты выпуклости контура ладони для случая, когда контур охватывает ладонь вместе с запястьем

Рис. 4: Дефекты выпуклости для контура ладони.

Возникает задача отбора троек точек, соответствующих именно промежуткам между пальцами. Воспользуемся предположением, что искомые точки будут соответствовать 4 самым глубоким среди дефектов выпуклости. Однако в некоторых случаях, как на рис. 4b, если контур охватил запястье, то довольно глубокие дефекты могут возникнуть на стыке ладони и запястья. Также, т. к. во входном наборе изображений присутствуют снимки ладоней с сомкнутыми пальцами, то для пар прижатых пальцев дефект выпуклости будет иметь малую глубину, как на рис. 5.



Рис. 5: Дефекты выпуклостей для ладоней с сомкнутыми пальцами.

Для различения таких ситуаций воспользуемся предположением, что углы между пальцами и углы между линиями, соединяющими «стык» и кончики для плотно сомкнутых пальцев, обычно не больше определенной величины. Для дефектов, соответствующих запястью и неровностям контура на ребре ладони и т. д., как на рис. 4a, углы все же как правило более близки к развернутому, в силу большого расстояния между точками выпуклой оболочки относительно расстояния от удаленной точки до самой выпуклой оболочки. Таким образом, среди всех найденных дефектов выпуклости отбираются сначала те, для которых косинус угла между линиями от кончиков до удаленной точки больше некоторого порога, а затем среди них отбираются 4 самых глубоких. Для получения более качественного результата в зависимости от входного изображения программа позволяет задавать различные значения порогового значения косинуса.

В итоге находим 4 тройки точек. Причем для указательного, среднего и безымянного пальцев найдется по 2 точки, и необязательно одинаковых. Для построения линии пальцев необходимо лишь 9 точек. Поступим следующим образом. Сначала необходимо отсортировать найденные наборы точек. Для нахождения дефектов выпуклости использовалась функция `convexityDefects` из библиотеки `opencv`, и найденные ей дефекты представляются как наборы точек вида «начальная точка на выпуклой оболочке - конечная точка на выпуклой оболочке - удаленная точка - расстояние до нее». Таким образом, нам известны начальные и конечные точки, соответствующие кончикам пальцев. Среди пар «начало - конец» будут соответствующие одному и тому же пальцу, и расстояние между ними будет мало. Найдем среди всех таких пар пару с наименьшим расстоянием, и с большой вероятностью она будет соответствовать указательному пальцу, т. к. при построении выпуклой оболочки точки упорядочивались против часовой стрелки, и в таком же порядке скорее всего находились дефекты выпуклости. Дальнейшие эксперименты показали состоятельность этого предположения. Далее относительно этих точек упорядочим остальные циклическим сдвигом. Затем для впадин, большого пальца и мизинца возьмем исходные точки из троек, для остальных же пальцев возьмем среднее из двух найденных точек. Такой прием позволил получить довольно точное положение кончиков в силу близости двух точек дефектов выпуклости для одного пальца. В итоге, выбрав 9 точек, можем построить линию пальцев. Для разомкнутых пальцев она получилась достаточно точной, однако для сомкнутых точки оснований попали на «стыки» между прижатыми пальцами. Точки кончиков при этом определяются достаточно точно. Таким образом, данный метод дает довольно точный результат для разомкнутых пальцев и

приближенный, неточный для сомкнутых. Однако в любом из случаев построенная линия позволяет определить позу ладони.

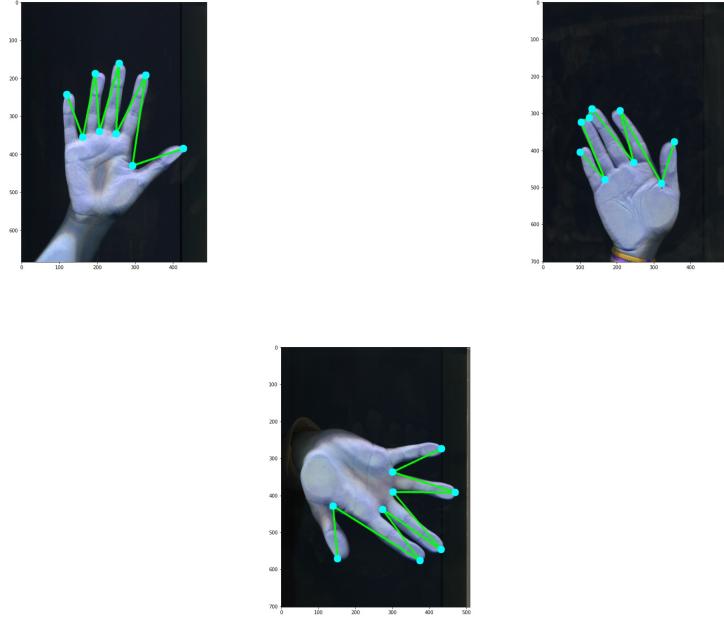


Рис. 6: Примеры получающихся линий пальцев, точных для разомкнутых пальцев и приближенных для сомкнутых.

Определение позы ладони производится следующим образом. Каждая тройка точек кончиков пальцев и основания между ними образует треугольник. Если пальцы сомкнуты, то либо основание было найдено правильно, и в таком случае угол между линиями от кончиков к основанию мал, либо пальцы прижаты сильно, и основание попало в «стык», и тогда три точки образуют достаточно маленький треугольник. Иначе же, если пальцы находятся на отдалении, то и соответствующий угол, и образуемый треугольник достаточно велики. Таким образом, для определения сомкнутости пальцев будем использовать площадь треугольника из основания и кончиков и угол между линиями от кончиков к основанию. Соответственно, если площадь меньше некоторого порога или угол меньше некоторого порога, то пальцы считаются сомкнутыми, иначе разомкнутыми. Определение близости по площади позволит выявить случаи «стыков», когда получаемый треугольник очень мал и когда точка основания не попала в истинное положение. Использование угла позволит определить близость, если точка основания оказалась достаточно близко к истинному положению, т. е. когда контур построился именно по пальцам, а не по их «стыку». Также это позволяет не учитывать разный размер ладоней, угол инвариантен к этому. Программа позволяет варьировать пороговые значения площади и угла для достижения корректного результата. Для удобства тестирования экспериментально полученные наборы входных параметров, дающих правильный результат на конкретных изображениях, приведены в приложении. В итоге поза ладони и координаты точек линии пальцев, сначала точек кончиков, затем точек оснований, записываются в текстовый файл согласно требуемому формату. Построенная линия пальцев визуализируется поверх исходного изображения.

Описание программной реализации

Программа написана на языке Python с использованием библиотек numpy, matplotlib, opencv. Для запуска программы необходимо открыть соответствующий .ipynb файл, например, в таких программах, как Jupyter Lab или Jupyter Notebook, или же через браузер в соответствующих облачных сервисах, например, Google Colab и т. д.. Входные данные для загрузки можно расположить в рабочей директории программы или в любом другом месте, но указав при этом в соответствующей ячейке кода нужный путь. Также можно загрузить .zip архив с изображениями в соответствующей ячейке кода, а затем распаковать его. При использовании Google Colab можно использовать изображения, загруженные в Google Drive. Результаты своей работы программа сохраняет в текущую рабочую директорию. Путь сохранения также можно изменить в соответствующей ячейке в коде. Также при желании можно сохранять промежуточные результаты (в соответствующих демонстрационных ячейках).

Программа разбита на функции бинаризации изображения, вычисления площади треугольника и косинуса угла, отбора дефектов выпуклости по углу, нахождения контура, выпуклой оболочки и всех дефектов выпуклости ладони, построения линии пальцев и определения позы ладони, визуализации результатов. Есть и возможность изменения различных параметров непосредственно в соответствующих ячейках.

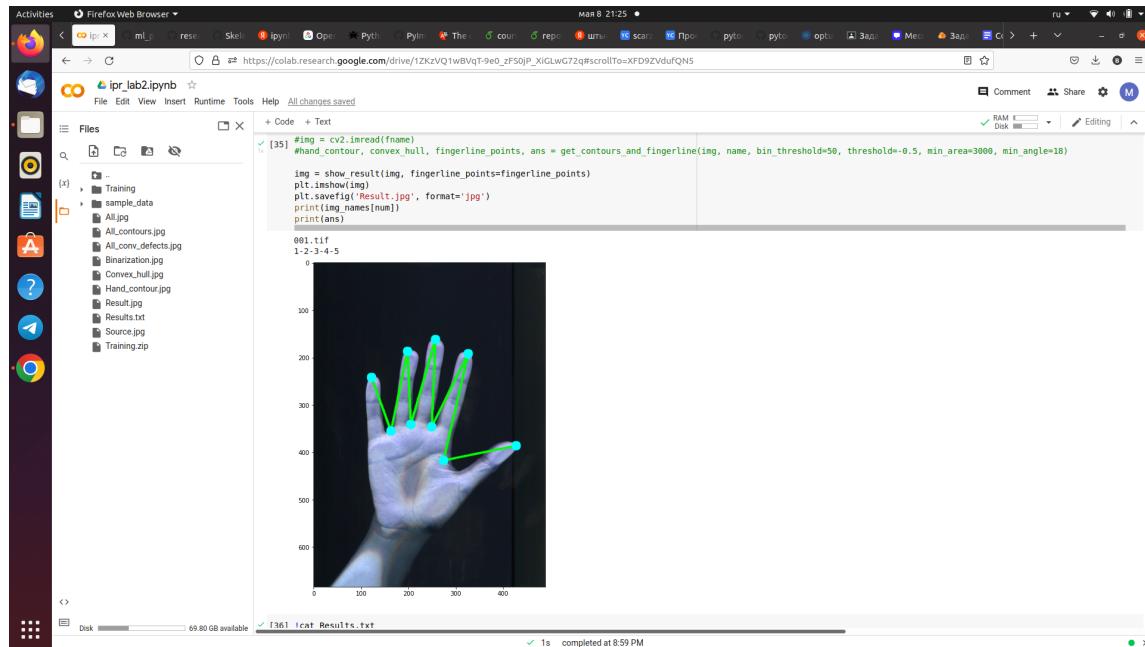


Рис. 7: Работа программы, запущенной в Google Colab

Эксперименты

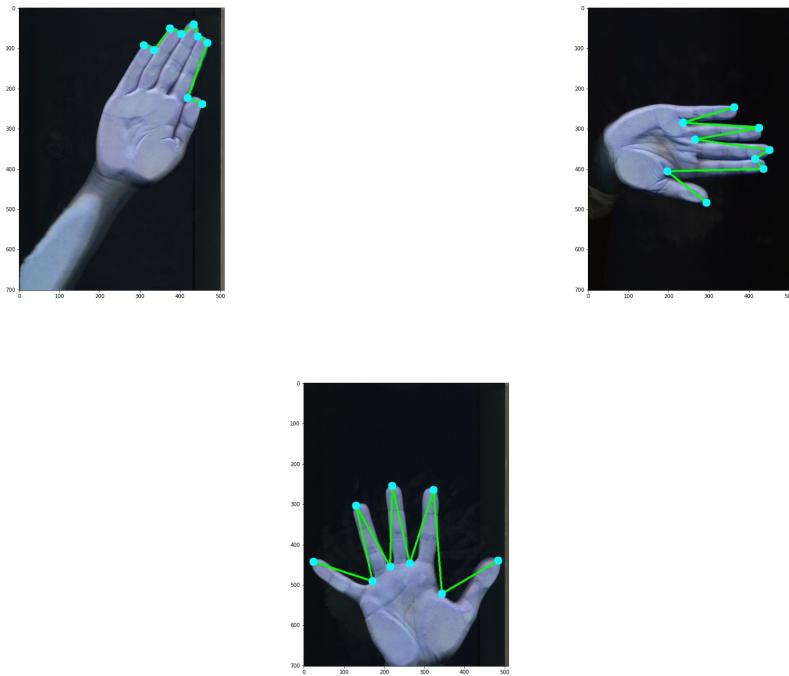
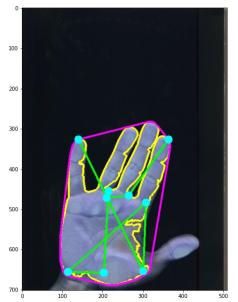


Рис. 8: Примеры результатов работы программы на разных входных изображениях.

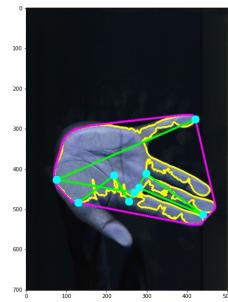
Рассмотрим результаты работы на нескольких входных изображениях. Программа строит линию пальцев не во всех случаях идеально, для разомкнутых пальцев она строится корректно, для сомкнутых - приближенно, точки оснований попадают в промежутки между сжатых пальцев, лежащие на контуре ладони. Однако это успешно позволяет определять позу ладони и при сомкнутых пальцах благодаря используемому методу решения. Программа позволяет указывать порог бинаризации, пороговое значение косинуса угла для отбора дефектов выпуклости, пороговое значение площади треугольника и угла между пальцами для определения близости. Это дает возможность строить более четкий и цельный контур ладони, следовательно, более точно по нему строить выпуклую оболочку и находить дефекты выпуклости. Варьирование порога площади и угла позволяет обрабатывать случаи сомкнутых пальцев и обеспечивает устойчивость у различному размеру и ориентации ладоней. Практически для всех изображений (о чём будет сказано подробнее позже) программа позволяет построить корректную линию для отделенных пальцев и достаточно качественную приближенную линию для сомкнутых. Получаемые линии в любом случае вполне адекватны и соответствуют структуре и форме ладони. По ним при определенных наборах задаваемых параметров для конкретного изображения программа безошибочно выдает корректный код позы. В ходе экспериментов не удалось установить универсальный набор параметров, обеспечивающий корректный ответ на всех изображениях. Однако были найдены соответствующие комбинации значений для 63 из 67 изображений, и в целом для большинства изображений они во многом схожи. В основном различия возникают в значении порога бинаризации и косинуса угла. Из-за несколько разной яркости входных изображений и наличия фрагментов одежды, запястий и т. п. контуры на промежуточных стадиях могут получаться разного качества и искривленности, и не всегда удается выделить саму ладонь без потерь частей пальцев и самой ладони. Однако варьирование входных параметров в целом позволяет успешно бороться с этим. Проводились также эксперименты с различными преобразованиями яркости исходных снимков, но улучшения ре-

зультата это не дало, порой выделялись отпечатки на сканере и т. д., появлялись шумовые объекты на фоне и т. д., что затрудняло дальнейшую работу программы.

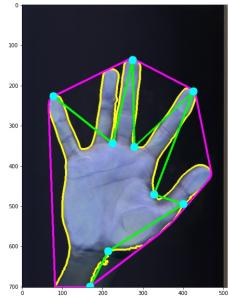
В итоге при должном подборе параметров удалось получить корректную позу ладони для 63 из 67 изображений, а также построить достаточно качественную линию пальцев для них, порой приближенную и не совсем корректную с точки зрения определения для сомкнутых пальцев. Однако для 4 изображений не удалось добиться построения сколь-нибудь корректной линии пальцев. Причиной некорректной работы могло стать низкое качество контура в силу свойств входного изображения, или же то, что безымянный палец на них имеет сравнительно небольшую длину, и вероятнее всего его кончик оказался чуть ниже построенной выпуклой оболочки. Однако при некоторых входных параметрах можно получить верный код позы, однако данный результат, очевидно, некорректен и является совпадением. Линия пальцев для указанных изображений в любом случае строится неверно.



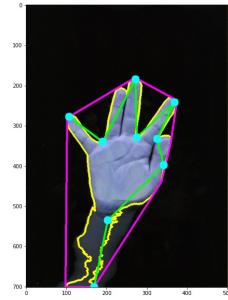
(a) 098.tif



(b) 165.tif



(c) 230.tif



(d) 284.tif

Рис. 9: Примеры некорректных результатов работы программы на некоторых входных изображениях.

Выводы

В результате данной лабораторной работы была разработана программа определения позы ладони и построения линии пальцев. Разработанный алгоритм в целом способен работать и с другими подобными снимками левых ладоней. Небольшим изменением кода отбора точек его можно адаптировать под снимки правых рук. Метод способен давать точный код позы ладони по построенной линии пальцев для подавляющего большинства входных изображений, 63 из 67. Задача построения линии пальцев не была решена идеально, для сомкнутых

пальцев она строится приближенно, для 4 изображений не удалось построить ее вовсе. Однако варьируя параметры метода, можно добиться верного результата. Таким образом, требуется тщательный подбор гиперпараметров, либо же использование или разработка более точных и мощных методов построения линии пальцев. Разработанная программа справляется с задачей для различных входных изображений. Также стоит отметить, что метод устойчив к различиям в размерах, форме и ориентации ладоней, и присутствует гибкость в выборе значений параметров. В итоге можно сказать, что метод требует усовершенствований и более тщательной отладки для получения абсолютно точного результата. Однако текущая реализация очень хорошо справляется с задачей, выдавая некорректный результат лишь на 4 из 67 изображений, что говорит о состоятельности алгоритма.

Приложение

Набор входных параметров программы, обеспечивающий верный результат для различных входных изображений

Имя файла	Порог бинаризации (bin_threshold)	Порог косинуса угла (threshold)	Порог площади (min_area)	Порог угла (min_angle)
001.tif	50	-0.5	3000	18
017.tif	50	-0.5	3000	18
043.tif	50	-0.5	3000	21
044.tif	50	-0.5	3000	21
055.tif	50	-0.2	3000	21
067.tif	50	-0.5	3000	21
070.tif	50	-0.7	3000	24
078.tif	50	-0.5	3000	22
090.tif	100	-0.5	3000	22
091.tif	100	-0.5	3000	22
098.tif	-	-	-	-
103.tif	50	-0.5	3000	22
107.tif	70	-0.5	3000	22
108.tif	50	-0.2	3000	22
125.tif	50	-0.5	3000	22
134.tif	50	-0.5	3000	22
144.tif	100	-0.5	3000	22
145.tif	50	-0.5	3000	22
145.tif	50	-0.5	3000	35
147.tif	50	-0.5	3000	22
150.tif	50	-0.5	3000	22
152.tif	100	-0.5	3000	22
156.tif	50	-0.5	3000	22
158.tif	50	-0.5	3000	22
162.tif	50	-0.5	3000	41
165.tif	-	-	-	-
168.tif	50	-0.5	3000	22
174.tif	100	-0.5	3000	22
175.tif	100	-0.5	3000	22

Имя файла	Порог бинаризации (bin_threshold)	Порог косинуса угла (threshold)	Порог площади (min_area)	Порог угла (min_angle)
177.tif	50	-0.5	3000	22
182.tif	50	-0.5	3000	22
184.tif	50	-0.5	3000	22
185.tif	50	-0.5	3000	22
187.tif	50	-0.5	4000	22
195.tif	50	-0.5	3000	22
196.tif	50	-0.5	3000	28
201.tif	50	-0.5	3000	22
209.tif	50	-0.5	3000	22
217.tif	50	-0.5	3000	22
225.tif	50	-0.5	3000	22
226.tif	80	-0.5	3000	22
230.tif	-	-	-	-
231.tif	100	-0.5	3000	23
235.tif	50	-0.5	3000	22
239.tif	80	-0.5	8500	22
240.tif	50	-0.4	3000	22
247.tif	50	-0.5	3000	22
248.tif	50	-0.5	3000	22
251.tif	50	-0.4	3000	22
252.tif	50	-0.5	3000	22
254.tif	50	-0.5	3000	22
265.tif	100	-0.5	3000	22
266.tif	50	-0.5	3000	22
267.tif	100	-0.5	3000	22
271.tif	100	-0.4	3000	22
272.tif	100	-0.5	3000	27
274.tif	90	-0.5	3000	24
277.tif	50	-0.5	3000	22
280.tif	50	-0.5	3000	28
281.tif	50	-0.5	3000	22
284.tif	-	-	-	-
286.tif	50	-0.5	3000	25
288.tif	50	-0.5	3000	22
289.tif	50	-0.5	3000	36
292.tif	50	-0.5	3000	22
295.tif	60	-0.7	3000	22
300.tif	50	-0.5	3000	24