Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана (Национальный исследовательский университет)»   
(МГТУ им. Н.Э. Баумана)

Факультет Информатики и систем управления (ИУ)

Кафедра Информационные системы и телекоммуникации (ИУ-3)

**Отчёт**

**по лабораторной работе №1.1**

**«Распознавание объекта по форме»**

**по курсу**

**«Цифровая обработка изображений»,**

**1 курс, 2-й семестр.**

Выполнил: студент группы ИУ3-21М

Смирнова К.Д.,

Проверил: Алфимцев А.Н.

Москва, 2020

**Задание**

Получить обучающую и тестовую выборки изображений двух фруктов у преподавателя в соответствии с вариантом.

Разработать программу по загрузке цифрового изображения.

Подключить функцию обращения к пикселем цифрового изображения.

Запрограммировать формулу перевода цифрового изображения в полутоновой формат.

Запрограммировать оператор выделения границ в соответствии с вариантом.

Запрограммировать работу программы распознавания фруктов в зависимости от геометрической формы (после работы оператора выделения границ).

Вариант №31

|  |  |  |
| --- | --- | --- |
| **Фрукт 1** | **Фрукт 2** | **Оператор** |
| Apple Red Delicious | Quince | Кэнни |

Пример изображений выборки Apple Red Delicious:



Рисунок 1 – Изображения выборки Apple Red Delicious

Пример изображений выборки Quince:

Рисунок 2 – Изображения выборки Quince

Сначала изображения обрабатываются оператором Кэнни для выделения границ. Целью Кэнни было разработать оптимальный алгоритм обнаружения границ, удовлетворяющий трём критериям:

* хорошее обнаружение (Кэнни трактовал это свойство как повышение отношения *сигнал/шум*);
* хорошая локализация (правильное определение положения границы);
* единственный отклик на одну границу.

Из этих критериев затем строилась целевая функция стоимости ошибок, минимизацией которой находится «оптимальный» линейный оператор для свёртки с изображением.

Алгоритм детектора границ не ограничивается вычислением градиента сглаженного изображения. В контуре границы оставляются только точки максимума градиента изображения, а не максимальные точки, лежащие рядом с границей, удаляются. Здесь также используется информация о направлении границы для того, чтобы удалять точки именно рядом с границей и не разрывать саму границу вблизи локальных максимумов градиента. Затем с помощью двух порогов удаляются слабые границы. Фрагмент границы при этом обрабатывается как целое. Если значение градиента где-нибудь на прослеживаемом фрагменте превысит верхний порог, то этот фрагмент остается также «допустимой» границей и в тех местах, где значение градиента падает ниже этого порога, до тех пор пока она не станет ниже нижнего порога. Если же на всем фрагменте нет ни одной точки со значением большим верхнего порога, то он удаляется. Такой гистерезис позволяет снизить число разрывов в выходных границах. Включение в алгоритм Кэнни шумоподавления с одной стороны повышает устойчивость результатов, а с другой — увеличивает вычислительные затраты и приводит к искажению и даже потере подробностей границ. Так, например, таким алгоритмом скругляются углы объектов и разрушаются границы в точках соединений.

Алгоритм Кэнни состоит из пяти отдельных шагов:

1. **Сглаживание**. Размытие изображения для удаления шума.
2. **Поиск градиентов**. Границы отмечаются там, где градиент изображения приобретает максимальное значение.
3. **Подавление не-максимумов**. Только локальные максимумы отмечаются как границы.
4. **Двойная пороговая фильтрация**. Потенциальные границы определяются порогами.
5. **Трассировка области неоднозначности**. Итоговые границы определяются путём подавления всех краёв, несвязанных с определенными (сильными) границами.

Перед применением детектора, преобразуем изображение в оттенки серого, чтобы уменьшить вычислительные затраты. Этот этап характерен для многих методов обработки изображений.

При пририменении детектора границ Кэнни в контуре границы оставляются только точки максимума градиента изображения, а не максимальные точки, лежащие рядом с границей, удаляются – поэтому нет нужды производить дополнительную фильтрацию оттенков серого.

Распознавание заключается в определении количества белых пикселей в выделении контура оператором Кэнни. Apple Red Delicious – тёмный и гладкий фрукт, и соответственно вмятины и различный рельеф не выделяются, а Quince светлый и рельефный фрукт, поэтому оператор выделяет массу деталей.

Опытным путём было определено, что у Apple Red Delicious за счёт не рельефной формы выделяются границы суммой не более чем 500 пикселей, а Quince больше 500.

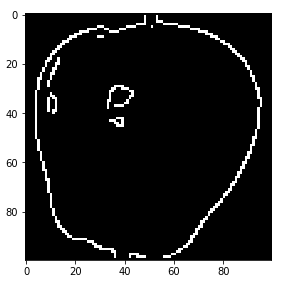
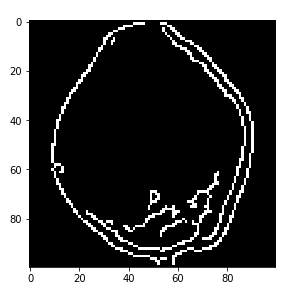
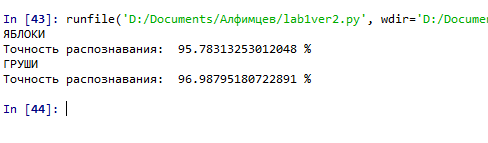
а)  б

Рисунок 3 – Изображения, обработанное оператором Кэнни:

1. Apple Red Delicious; б) Quince.

Результат работы программы:



Точность распознавания при подаче выборки Apple Red Delicious: 95,783 %

Точность распознавания при подаче выборки Quince: 96,988 %

Ошибки I и II рода:

|  |  |  |
| --- | --- | --- |
|  | Ожидается Apple Red Delicious | Ожидается Quince |
| Выход Apple Red Delicious | 159 | 5 |
| Выход Quince | 7 | 161 |

**Вывод**

При выполнении лабораторной работы было проведено распознавание одного из объекта на изображении по предварительно выделенному контуру. Был придуман и реализован метод обработки контура объекта для его классификации.

Оператор Кэнни позволил достичь необходимых результатов, но не оптимально. Результат фильтрации содержит нежелательные контуры внутри объекта (например отсветы на Apple Red Delicious).

По итогу выполнения была достигнута точность распознавания Apple Red Delicious в 95,783% и 96,988% точность распознавания Quince.

Исходный код алгоритма:

# -\*- coding: utf-8 -\*-

"""

Created on Fri May 15 22:41:06 2020

@author: karina

"""

from skimage import data, io, filters

from skimage.color import rgb2gray #Импорт функции перевода изображения в полутона

from math import sqrt

from skimage import feature #Импорт функции вычисления квадратного корня

import os

def Raspoznavatel(image):

A = Filter(image) #фильтруем, на выходе картинка с границами

B = Process(A)# подсчёт на выходе колво белых пикселей

if B <= 500: #белых пикселей меньше 500 - яблоко

Ans = 'Яблоко'

#print(Ans)

else:

Ans = 'Груша'

#print(Ans)

return Ans

def Filter(image):

image = io.imread(image) #Считывание изображения

image\_gray = rgb2gray(image) #Перевод в полутоновое изображения

image\_filter1 = feature.canny(image\_gray)#фильтр Кэнни

# io.imshow(image\_filter1)

# io.show()

return(image\_filter1)

def Process(image\_filter1):

k=0

for j in range(100):

for g in range(100):

if image\_filter1[j][g]==1: #белые пиксели

k+=1

#print(k)

return(k)

files = os.listdir('Testing\_Fruit') #список файлов в папке

print('ЯБЛОКИ')

true = 0

for j in range(len(files)):

Answer = Raspoznavatel('Testing\_Fruit/' + files[j]) #формируется название

if Answer == 'Яблоко':

true = true + 1

accuracy = (true / len(files)) \* 100

print('Точность распознавания: ', accuracy, '%')

files2 = os.listdir('Testing\_Fruit2')

print('ГРУШИ')

true = 0

for j in range(len(files2)):

Answer = Raspoznavatel('Testing\_Fruit2/' + files2[j])

if Answer == 'Груша':

true = true + 1

accuracy = (true / len(files2)) \* 100

print('Точность распознавания: ', accuracy, '%')