

**Лабораторные работы по курсу
Моделирование ч.3 «Компьютерное зрение»**

Лабораторная работа 4

Распознавание образов

Содержание

Введение.....	3
1. Краткая теоретическая справка	3
1.1. Задача распознавания образов	3
1.2. Общая схема системы распознавания образов	4
1.3. Искусственные нейронные сети	4
1.4. Применение перцептрона для распознавания образов.....	6
2. Подготовка к выполнению лабораторной работы	7
2.1. Установка пакетов.....	7
2.2. Дополнительные материалы	7
2.3. Описание проекта « <i>recognition</i> »	7
3. Индивидуальные задания	9
3.1. Задание №1	9
3.2. Задание №2	10
4. Список литературы	10

Введение

Целью лабораторной работы является освоение практических навыков в создании систем распознавания образов для анализа видеoinформации на языке *python* с применением библиотеки для компьютерного зрения *OpenCV* [1] и библиотеки для создания искусственных нейронных сетей *PyTorch* [2].

Лабораторная работа содержит краткую теоретическую справку и два задания для самостоятельного выполнения. Первое задание заключается в разработке системы распознавания образов с помощью аналитически описываемого решающего правила для бинарной классификации. Второе задание заключается в разработке системы бинарной классификации образов с помощью применения искусственных нейронных сетей.

Лабораторная работа рассчитана на 4 академических часа.

1. Краткая теоретическая справка

1.1. Задача распознавания образов

Распознавание образов – это научная дисциплина, целью которой является классификация объектов по нескольким категориям или классам. Объекты называются образами [3].

Можно выделить несколько базовых задач распознавания образов:

- бинарная классификация;
- классификация с учителем (распознавание);
- классификация без учителя (кластеризация).

Бинарная классификация представляет собой тип задач, в которых требуется отнести наблюдаемый объект или процесс к одному из двух противоположных классов. Частным случаем задачи бинарной классификации является задача обнаружения, когда требуется дать один из двух ответов: *объект присутствует* на изображении или *объект отсутствует* на изображении. Еще одним примером задачи бинарной классификации может служить отнесение наблюдаемого объекта к одному из двух заранее заданных классов: *яркий/темный, круг/квадрат* и т. д. Наличие третьего класса исключается.

Классификация с учителем, так же называемая задачей распознавания, предполагает отнесение наблюдаемого объекта к одному из N заранее известных классов. Например, по фотографии электронного компонента требуется определить его тип: резистор, конденсатор, диод, транзистор, операционный усилитель или иное.

Классификация без учителя, так же называемая задачей кластеризации, предполагает отсутствие заранее заданных классов. Все объекты, рассматриваемые в рамках данной задачи, должны быть разбиты на классы на основе сходства тех или иных признаков.

1.2. Общая схема системы распознавания образов

Общий подход к решению задачи распознавания образов представлен на Рисунке Рисунок 1.

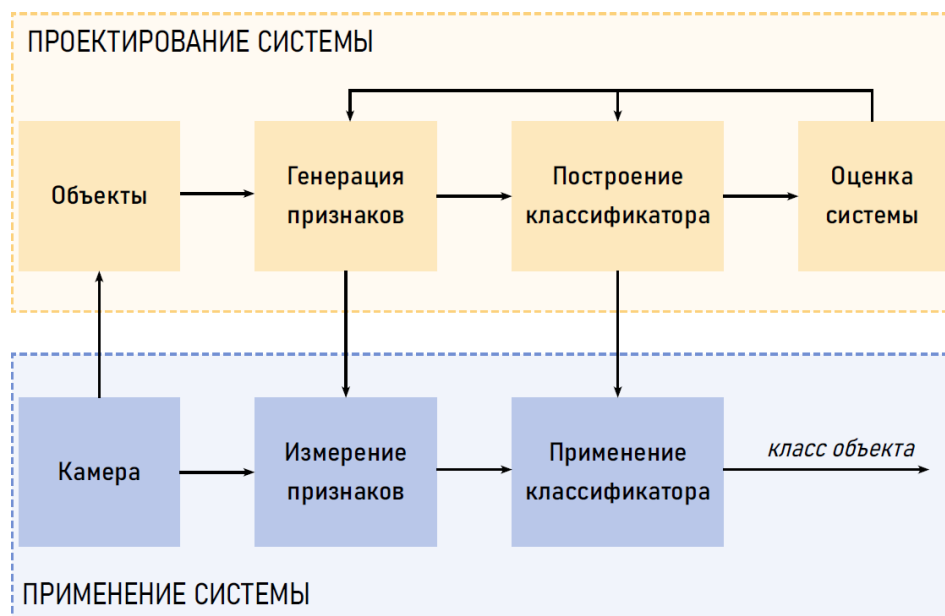


Рисунок 1. Поход к созданию системы распознавания образов

Создание системы распознавания образов разбивается на два параллельных и взаимосвязанных между собой маршрута: проектирование системы и применение системы. К маршруту проектирования системы относятся те этапы, которые выполняются разработчиком в процессе ее создания, и не выполняются в процессе ее эксплуатации. К маршруту применения системы относятся только те этапы, которые должны быть выполнены для решения задачи распознавания образа для вновь наблюдаемого объекта. Исключением является задача кластеризации, для решения которой должны выполняться оба маршрута.

Первичной информацией для проектирования системы являются изображения объектов. Желательно, чтобы они были получены с помощью той же камеры и при тех же условиях, при которых в дальнейшем планируется эксплуатации системы. В ходе совокупного анализа объектов формируется пространство признаков, внутри которого объекты могут быть разделены на классы. Для объединения объектов в один класс формируется аналитическое описание функции, на основе которой принимается решение о принадлежности объекта к тому или иному классу.

Зачастую ввиду сложности аналитического описания разделения объектов на классы для решения данной задачи применяются искусственные нейронные сети.

1.3. Искусственные нейронные сети

Искусственные нейронные сети представляют собой математическую модель обработки информации мозгом. Нейронные сети не программируются в классическом понимании этого слова, а обучаются [4]. Основным элементом искусственной

нейронной сети является математическая модель нейрона, которая схематично изображена на Рисунке Рисунок 2.

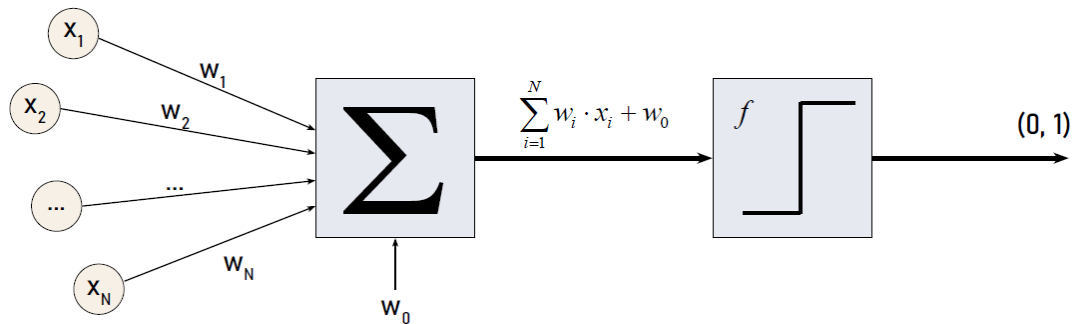


Рисунок 2. Математическая модель нейрона

Входами для нейрона являются признаки $x_1 \dots x_N$, которые сначала суммируются, а затем подаются на вход функции активации. Важность каждого входа для нейрона определяется весами $w_1 \dots w_N$:

$$r = \sum_{i=1}^N w_i \cdot x_i + w_0$$

С помощью параметра w_0 задается смещение с выхода сумматора. Результат суммы подается на вход «функции активации», которая в простейшем случае представляет собой функцию Хевисайда:

$$f = \begin{cases} 0, & r < 0 \\ 1, & r \geq 0 \end{cases}.$$

В результате нейрон представляет собой простейший бинарный классификатор, который формирует гиперплоскость для разделения пространства входных признаков $x_1 \dots x_N$ на два класса (Рисунок Рисунок 3).

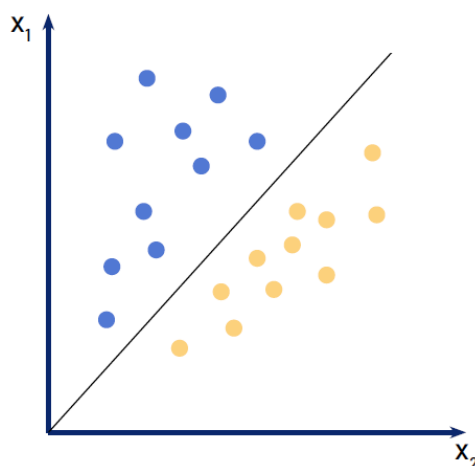


Рисунок 3. Разделение пространства признаков на два класса.

Положение гиперплоскости определяется значениями весов, а процедура обучения нейрона заключается в подстройке значений весов путем сравнения выходного значения нейрона с правильно заданным (например, с помощью разметки данных вручную).

На практике для получения численной оценки степени принадлежности набора признаков к тому или иному классу, а также для улучшения свойств обучаемости искусственной нейронной сети, в качестве функции активации применяют одну из дифференцируемых монотонно возрастающих функций [5]: линейная, сигмоида, гиперболический тангенс, *ReLU* и др. Так же на практике входные значения признаков нормируют для их изменения в диапазоне от 0 до 1.

1.4. Применение перцептрона для распознавания образов

Перцептрон является одной из первых архитектур нейронных сетей, созданных для моделирования работы нейронных связей в мозгу. Общее схематичное изображение «однослойного» перцептрона представлено на Рисунке Рисунок 4.

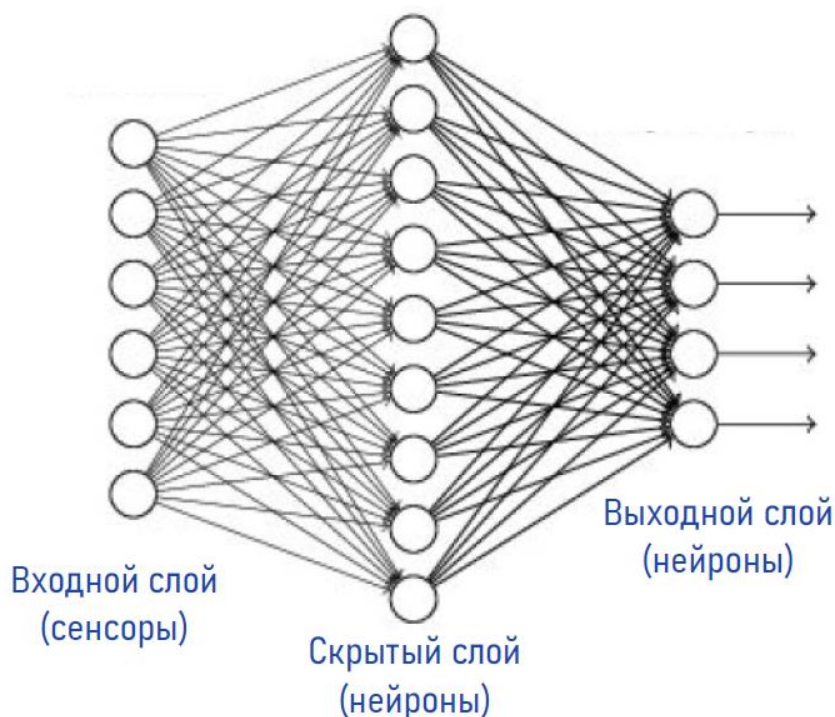


Рисунок 4. Архитектура однослойного перцептрона.

Он состоит из одного входного слоя, одного внутреннего (скрытого) слоя и одного выходного слоя. Входной слой представляет собой нейроны, которые не осуществляют обработку данных, а только передают входные признаки на дальнейшие слои. Количество входных нейронов определяет размерность пространства признаков. Скрытый слой состоит из набора однотипных нейронов. Каждый нейрон получает на вход все признаки и передает выходной результат на следующий слой. Количество нейронов выходного слоя определяет количество классов, на которые разбивается все

пространство признаков. Для решения задачи бинарной классификации требуется один нейрон в выходном слое.

При применении в нейронах перцептрона дифференцируемых функций активации, возможно обучение нейронной сети с помощью итерационного метода обратного распространения ошибки [6]. Для реализации процедуры обучения требуется исходный набор признаков и соответствующий ему набор правильных разделений признаков на классы.

2. Подготовка к выполнению лабораторной работы

2.1. Установка пакетов

2.1.1. Для выполнения работы необходимо установить следующие пакеты для языка *python*:

- *OpenCV* (*opencv-python*)
- *NumPy* (*numpy*)
- *Matplotlib* (*matplotlib*)
- *PyTorch* (*torch*)

2.2. Дополнительные материалы

2.2.1. Откройте каталог с дополнительными материалами к лабораторной работе.

2.2.2. В подкаталоге «*data*» расположены изображения четырех видов ягод: черной смородины (*black_currant*), ежевики (*blackberry*), черники (*blueberry*) и малины (*raspberry*), которые используются как входные изображения:

2.2.2.1. в подкаталоге «*data/dataset*» расположено по 5 изображений для каждого вида ягоды с обозначением ее названия;

2.2.2.2. в корне каталога «*data*» расположено по одному изображению каждой ягоды без указания ее названия.

2.2.3. В каталоге «*recognition*» располагается проект для *PyCharm IDE*, в котором реализована бинарная классификация образов ягод на основе пространства признаков с помощью двух методов: аналитического описания решающего правила и решающего правила на основе перцептрона.

2.3. Описание проекта «*recognition*»

2.3.1. Откройте проект «*recognition*» в *PyCharm IDE*.

2.3.2. Проект состоит из следующих файлов исходных кодов:

2.3.2.1. *berry.py*

В данном файле реализован класс «*Berry*» для загрузки изображения ягоды из файла и класс «*Features*» для выделения следующих признаков из изображения:

- размер – вычисляется как отношение количества пикселей ягоды к общему количеству пикселей изображения;

- яркость – вычисляется как средняя яркость трех цветовых компонент пикселей, принадлежащих изображению ягоды, приведенная к градициям серого и нормированная к диапазону от 0.0 до 1.0.

2.3.2.2. rule.py

В данном файле реализован класс «*Rule*», позволяющий по входящему перечню точек пространства признаков сформировать линейные отрезки, которые будут ограничивающей фигурой для выделяемого в пространстве признаков класса. Метод «*check*» позволяет проверить положения точки относительно заданных линий.

2.3.2.3. perceptron.py

В данном файле реализован класс «*Perceptron*», представляющий собой искусственную нейронную сеть с одним входным слоем для приема двух признаков, одним скрытым слоем и одним выходным слоем с одним нейроном. Конструктор класса в качестве входных аргументов использует:

- *data* – двумерный массив точек обучающей выборке данных;
- *answer* – массив правильной классификации для точек из *data*;
- *numofneurons* – количество нейронов скрытого слоя (по умолчанию 128);
- *epochs* – количество циклов обучения (по умолчанию 10000).

Метод класса «*check*» позволяет подать массив входных данных на вход нейронной сети и получить ответ о принадлежности каждой точки к одному или другому классу.

2.3.2.4. visualization.py

Вспомогательный файл, в котором для визуализации процедуры обработки данных реализованы следующие функции:

- «*img_concat*» – объединение всех входных изображений в один массив;
- «*all_image_plot*» – отображение всех изображений ягод с комментариями;
- «*berry_image_view*» – отображение одного изображения ягоды при тестировании;
- «*features_separate*» – перенос признаков из общего массива объектов в отдельный массив для каждого класса;
- «*plot_feautres*» – отображение точек в пространстве признаков;
- «*show_and_wait_any_key*» – отображение фигуры и ожидание нажатия клавиши на клавиатуре.

2.3.2.5. main.py

В данном файле реализована процедура бинарной классификации ягод. В качестве первого класса используется «*blueberry*», в качестве второго класса – все остальные ягоды.

В первой части исходного кода формируется решающее правило на основе аналитической функции, заданной набором отрезков, для решения задачи

классификации, после чего проверяется каждое изображение ягод из «data» для тестирования созданного решающего правила.

Во второй части исходного кода решающее правило формируется с помощью нейронной сети, после чего проверяется каждое изображение ягод из «data» для тестирования.

2.3.3. Изучите исходные кода файлов проекта.

2.3.4. Запустите проект на выполнение.

2.3.5. После запуска этапы выполнения *main.py* будут визуализироваться на фигуре (см. Рисунок 5)

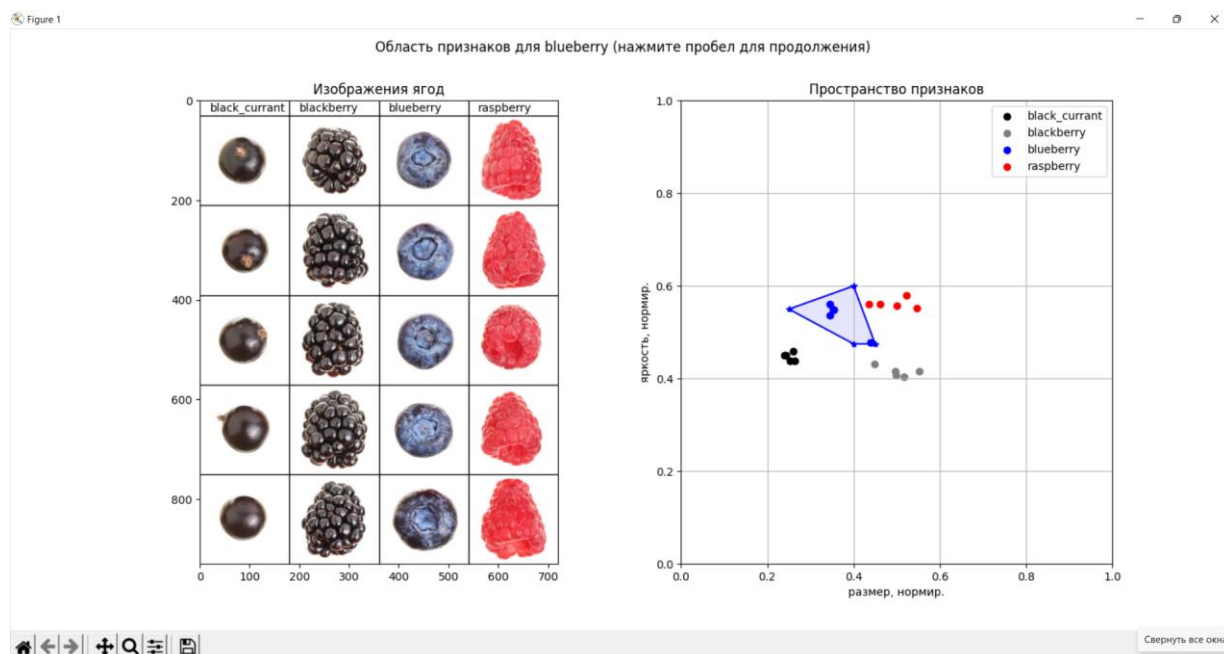


Рисунок 5. Визуализация выполнения проекта

2.3.6. В левой части фигуры будут показаны изображения ягод, в правой части – пространство признаков.

2.3.7. Для перехода к следующему этапу обработки нажимайте клавишу «пробел».

3. Индивидуальные задания

3.1. Задание №1

3.1.1. Доработать исходный код проекта *recognition* для бинарной классификации с помощью аналитически заданного решающего правила с разбиением на классы в зависимости от

$$i = (N \% 2),$$

где N – ваш номер в общем списке группы:

3.1.1.1. класс 1: *raspberry* и *blackberry*, класс 2: *blueberry* и *black_currant*

3.1.1.2. класс 1: *blueberry* и *blackberry*, класс 2: *raspberry* и *black_currant*

3.2. Задание №2

3.2.1. Доработать исходный код проекта *recognition* для бинарной классификации с помощью перцептрона с разбиением на классы в зависимости от

$$i = (N \% 2),$$

где N – ваш номер в общем списке группы:

3.2.1.1. класс 1: *blueberry* и *blackberry*, класс 2: *raspberry* и *black_currant*

3.2.1.2. класс 1: *raspberry* и *blackberry*, класс 2: *blueberry* и *black_currant*

4. Список литературы

1. OpenCV [Электронный ресурс] // OpenCV: [сайт]. URL: <https://opencv.org/>
2. PyTorch [Электронный ресурс] // PyTorch: [сайт]. URL: <https://pytorch.org/>
3. М. М.Л. Математические методы распознавания образов. Москва: МГУ, ВМиК, 2002-2004.
4. Лекун Я. Как учится машина. Революция в области нейронных сетей и глубокого обучения. М.: Альпина нон-фикшн, 2001.
5. Wikipedia | Функция активации [Электронный ресурс] // Wikipedia: [сайт]. [2021]. URL: https://ru.wikipedia.org/wiki/Функция_активации
6. Обратное распространение ошибки [Электронный ресурс] // Университет ИТМО: [сайт]. [2021]. URL: https://neerc.ifmo.ru/wiki/index.php?title=Обратное_распространение_ошибки