

Библиотека OpenCV для задачи распознавания показаний артериального давления по дисплеям тонометров

А.З. Туйгунов

Факультет информатики и робототехники
Уфимский государственный авиационный
технический университет
Уфа, Россия
e-mail: nequeyjd2009@yandex.ru

Аннотация¹

Данное исследование посвящено инструменту по распознаванию изображений, а именно библиотеке OpenCV. Решается задача по автоматизированному мониторингу артериального давления [2]. Проводится сравнительный анализ существующих инструментов по распознаванию образов, выявляются их достоинства и недостатки, а также описываются модели и методы распознавания изображений. В качестве исходных данных используются изображения дисплеев автоматических тонометров.

Ключевые слова: OpenCV; распознавание образов; артериальное давление; мониторинг; дисплей.

1. Введение

Уровень артериального давления является одним из самых важных показателей состояния здоровья [2]. Для того чтобы быть в курсе состояния здоровья своей сердечно-сосудистой системы, необходим постоянный мониторинг артериального давления и фиксация измерений. В связи с тем, что у большинства населения накопилось большое количество автоматических тонометров, не содержащих конструктивно каналов связи с персональным компьютером, то появляется классическая проблема унаследованных систем (legacy system). Поскольку заменить весь этот парк устройств за короткое время невозможно, то предлагается вариант решения – система считывания показаний с дисплея тонометра с последующим их распознаванием [1] и хранением в электронном виде.

Труды Седьмой всероссийской научной конференции "Информационные технологии интеллектуальной поддержки принятия решений", 28-30 мая, Уфа-Ставрополь, Ханты-Мансийск, Россия, 2019

Для проектируемой системы по распознаванию изображений с показаниями тонометра, конечно же, необходим тонометр с дисплеем, то есть полуавтоматический или автоматический, так как само распознавание будет выполняться по цифрам, характеризующим систолическое и диастолическое артериальное давление.

Для задачи распознавания изображений с показаниями тонометра предлагается инструмент, а именно библиотека компьютерного зрения – OpenCV [5].

2. Описание библиотеки OpenCV

OpenCV – это библиотека компьютерного зрения с открытым исходным кодом. Изначально она была написана на языке *C++/C*, но сейчас она уже существует на многих языках, таких как *java*, *python* и др. Эта библиотека включает в себя различные алгоритмы компьютерного зрения, распознавания изображений и многое другое. *OpenCV* доступна для общего использования, как в образовательных целях, так и коммерческих проектах. Она содержит следующие алгоритмы:

- распознавание объектов в видеопотоке;
- распознавание печатного и рукописного текста;
- устранение искажений картинки;
- выявление сходства и формы объектов;
- слежение за перемещением объекта;
- распознавание движений, жестов и многое другое.

С момента своего релиза в январе 1999 года, *OpenCV* была использована во многих приложениях и научно-исследовательских работах, в том числе: наложение обычных карт и фотографий со спутника, выравнивание документов при сканировании, удаление шума из медицинских изображений, анализ объектов, системы безопасности, автоматическое наблюдение, системы контроля качества на производстве, калибровка камер, а также

VII Всероссийская научная конференция "Информационные технологии интеллектуальной поддержки принятия решений", Уфа-Ставрополь-Ханты-Мансийск, Россия, 2019

беспилотные летательные, наземные и подводные аппараты. Она даже была использована для распознавания звука и музыки, где методы распознавания образов были применены к изображениям спектрограмм звука. Библиотека стала ключевой частью системы зрения робота «Stanley» из Стэнфорда, выигравшего 2.000.000\$ на Больших Пустынных Гонках Роботов *DARPA*.

Открытая лицензия для *OpenCV* была составлена таким образом, чтобы было возможно создавать коммерческие приложения, используя любые возможности *OpenCV*. Эту библиотеку используют такие крупные компании как *IBM*, *Microsoft*, *Intel*, *Sony*, *Siemens*, *Google*, а также научно-исследовательские центры, такие как Стэнфорд, Массачусетский технологический институт, *CMU*, Кембридж, и *INRIA*. *OpenCV* популярна во всём мире, причём большие сообщества пользователей можно найти в Китае, Японии, России, Европе и Израиле [9].

Не мало важной частью библиотеки *OpenCV* является модуль обработки изображений. Он включает в себя:

- Базовые операции над изображениями (фильтрация, геометрические преобразования, преобразование цветковых пространств и т.д.).
- Анализ изображений (выбор отличительных признаков, морфология, поиск контуров, гистограммы).
- Структурный анализ (описание форм, плоские разбиения и т.д.).
- Анализ движения, слежение за объектами.
- Обнаружение объектов, в частности лиц.
- Калибровка камер, элементы восстановления пространственной структуры.

3. Модели распознавания изображений

3.1 Бинаризация изображений

Бинаризация изображения это перевод полноцветного изображения в монохромное, то есть где присутствует только два типа пикселей: темные и светлые. Существуют различные методы к бинаризации изображения, которые имеют две группы:

- Пороговые.
- Адаптивные.

Пороговые работают со всем изображением и, найдя какую-либо характеристику (порог), переводят изображение в черно-белое. Адаптивные методы используются при неравномерном освещении объектов и работают с такими участками изображения.

Выбираемый подход бинаризации часто зависит от изображений, которые необходимо распознавать. Например, при распознавании неравномерно освещенного текста можно воспользоваться

адаптивными функциями с размерами блоков, превышающими максимальный размер ячейки символа. При этом адаптивные методы бинаризации работают медленнее пороговых, что также влияет на выбор метода.

Примеры работы адаптивного преобразования:

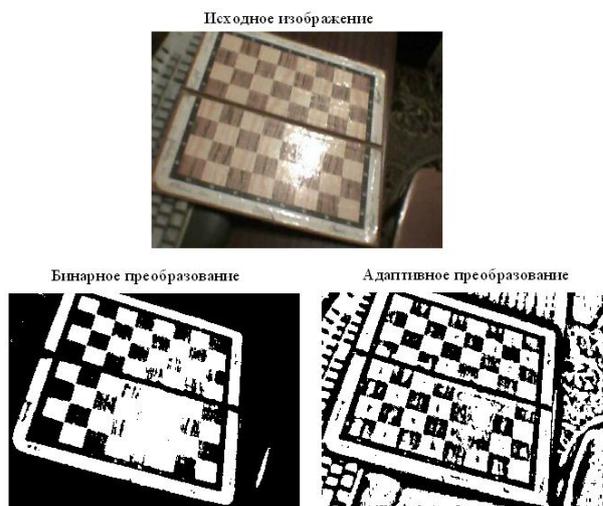


Рис. 1. Бинарное и адаптивное преобразование

Адаптивное преобразование является более эффективным методом и преобразует изображение в оттенках серого к монохромному изображению по формуле (1,2):

$$dst(x, y) = \begin{cases} max_value & \text{if } src(x, y) > T(x, y) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$dst(x, y) = \begin{cases} 0 & \text{if } src(x, y) > T(x, y) \\ max_value & \text{otherwise} \end{cases} \quad (2)$$

где $T(x, y)$ – пороговое значение пикселя,

$src(x, y)$ – *RGB*-значения пикселя исходного изображения,

$dst(x, y)$ – *RGB*-значения пикселя конечного изображения.

3.2. Сглаживание

Сглаживание, или размытие – операция, которая часто применяется для подавления шума изображений и для других целей. Выполняется она путем применения линейных фильтров к изображению. Это означает, что новое значение пикселя (x_i, y_j) вычисляется как взвешенная сумма значений исходных пикселей в той же позиции и ее окрестности. Веса пикселей обычно хранятся в матрице, называемой ядром. Таким образом, фильтр можно представлять себе как скользящее окно коэффициентов.

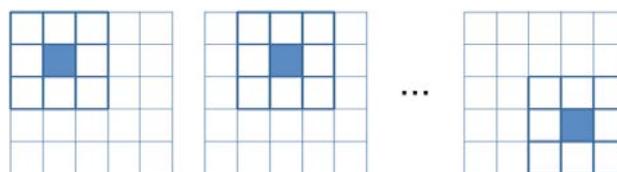


Рис. 2. Представление окрестности пикселя

Обозначим K – ядро, а I и O – входное и выходное изображение. Тогда новое значение пикселя в позиции (i, j) вычисляется по формуле (3):

$$O(i, j) = \sum_{m, n} I(i + m, j + n) * K(m, n) \quad (3)$$

В OpenCV для сглаживания чаще всего применяется медианный фильтр, фильтр Гаусса и двусторонний фильтр. Медианный фильтр очень хорош для устранения зернистости изображения («соль с перцем»), а фильтр Гаусса – в качестве предварительного шага для обнаружения границ. Что касается двусторонней фильтрации, то она применяется, в основном, для сглаживания изображений с сохранением резких границ.

3.3. Резкость изображения

Фильтры, повышающие резкость, используются, чтобы выделить границы и другие детали. Они основаны на вычислении первой и второй производной. Первая производная описывает градиент интенсивности изображения, а вторая определяется как дивергенция градиента. Поскольку при цифровой обработке изображений мы имеем дело с дискретными величинами (значениями пикселей), то для повышения резкости вычисляются дискретные производные. Оператор Собеля вычисляет первую производную изображения I по формуле (4):

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \quad (4)$$

Модуль градиента тогда можно вычислить по формуле (5):

$$G = \sqrt{G_x^2 + G_y^2} \quad (5)$$

Первые производные дают более жирные границы и широко применяются для выделения границ. Вторые же производные используются для улучшения изображения, потому что лучше реагируют на мелкие детали. Для вычисления производных чаще всего используются операторы Собеля и Лапласа.

3.4. Обесцвечивание

Обесцвечивание – процесс преобразования цветного изображения в полутоновое. Такое преобразование относится к числу базовых, оно есть и в OpenCV и в любой другой библиотеке обработки изображений. Пусть есть два цвета, A и B , и мы хотим преобразовать их в оттенки серого. Предположим, что B получается из A разнонаправленным изменением каналов R и G (формула 6):

$$A = (R, G, B) \Rightarrow G = \frac{R+G+B}{3}, \quad (6)$$

Стандартное преобразование создано на линейном комбинировании каналов R , G и B . Возникает проблема в том, что при таком преобразовании может теряться контрастность из-за того, что два разных цвета (воспринимаемые в исходном изображении как

контрастные) отображаются на одно и то же значение яркости.

Приведем пример обесцвечивания:

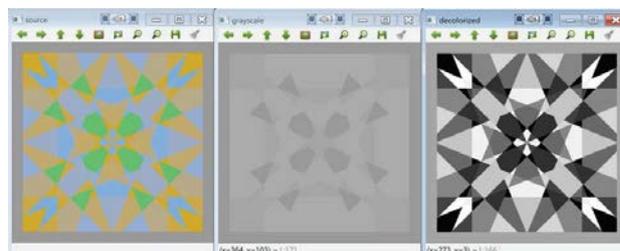


Рис. 3. Пример обесцвечивания изображения

3.5. Геометрические преобразования

Геометрическое преобразование не изменяет содержание изображения, а деформирует его посредством изменения системы координат. То есть для каждого входного пикселя мы берем его координаты, применяем к ним преобразование и копируем пиксель, оказавшийся в точке с новыми координатами, в выходное изображение.

$$O(x, y) = I(f_x(x, y), f_y(x, y))$$

При этом может возникнуть две проблемы:

Экстраполяция: значения $f_x(x, y)$ и $f_y(x, y)$ могут оказаться вне границ изображения. К геометрическим преобразованиям применяются те же режимы экстраполяции, что в случае фильтрации, плюс один дополнительный: *BORDER_TRANSPARENT*.

Интерполяция: $f_x(x, y)$ и $f_y(x, y)$ обычно принимают вещественные значения. В OpenCV есть выбор между интерполяцией по ближайшему соседу и полиномиальной интерполяцией. В первом случае значение координаты с плавающей точкой округляется до ближайшего целого. Поддерживаются следующие методы интерполяции:

- *INTER_NEAREST*: это описанная выше интерполяция по ближайшему соседу;
- *INTER_LINEAR*: билинейная интерполяция, подразумевается по умолчанию;
- *INTER_AREA*: передискретизация по области пикселей;
- *INTER_CUBIC*: бикубическая интерполяция по окрестности 4×4 ;
- *INTER_LANCZOS4*: метод Ланцоша по окрестности 8×8 .

OpenCV поддерживает аффинные преобразования (масштабирование, параллельный перенос, поворот и т. п.) и перспективные преобразования.

3.6. Морфологические операции

Морфологические операции используют для изображения «структурный элемент», в результате чего формируется новое изображение, в котором

пиксель в позиции (x_i, y_j) выходит путем сравнения пикселя исходного изображения в той же позиции с его соседями. В зависимости от выбранного структурного элемента морфологическая операция может быть более чувствительной к одним формам и менее чувствительной к другим.

Существуют две основные морфологические операции – наращивание и эрозия. В примере наращивания к границам объектов в изображении добавляются пиксели из фона, а в случае эрозии эти пиксели удаляются. Структурный элемент определяет, какие пиксели добавлять или удалять. При наращивании значение результирующего пикселя вычисляется как максимум, а при эрозии – как минимум всех пикселей в его окрестности. Приведем пример:



Рис. 4. Пример наращивания и эрозии

Путем комбинирования наращивания и эрозии можно определять и другие операции обработки изображений, например, размыкание, замыкание и морфологический градиент. Операция размыкания (открытия) определяется как эрозия, за которой следует наращивание, а операция замыкания (закрытия) наоборот – как наращивание с последующей эрозией. Таким образом, размыкание удаляет мелкие объекты, оставляя крупные, а замыкание удаляет небольшие дырки, оставляя более крупные. Морфологический градиент определяется как разность между наращиванием и эрозией изображения [3].

3.7. Алгоритм ближайшего соседа

KNN (*k Nearest Neighbor*) – алгоритм ближайшего соседа, который является одним из простых алгоритмов классификации, часто используется в машинном обучении. Это задача отнесения объекта к заранее обозначенным классам по формально выделенным признакам. Каждый из объектов представляется в виде вектора в N -мерном пространстве, где измерение подразумевает определенное описание какого-либо признака объекта. Например, нужно классифицировать принтеры: измерениями будут выступать стоимость, наличие *wi-fi* модуля, скорость печати и др.

Для классификации объектов тестовой выборки нужно выполнить следующие действия:

- Вычислить расстояние до каждого из объектов обучающей выборки.
- Отобрать k объектов обучающей выборки, расстояние до которых минимально.

Наиболее часто встречающийся класс среди k ближайших соседей и будет классом классифицируемого объекта.

Для нахождения расстояния используется функция расчета, она должна отвечать следующим требованиям:

$$d(x, y) \geq 0, d(x, y) = 0,$$

$$d(x, y) = d(y, x),$$

$$d(x, z) \leq d(x, y) + d(y, z),$$

где x, y, z – векторы признаков сравниваемых объектов.

4. Сравнение существующих решений

Для решения поставленной задачи существует не мало вариантов. Возьмем популярные инструменты распознавания образов – OpenCV, Tesseract и Tensorflow.

Выбирать библиотеку будем по таким критериям, как размер библиотеки, скорость ее распознавания, точность распознавания. На выбор – известные библиотеки по распознаванию: *OpenCV*, *TensorFlow*, *TesseractOCR*, *Caffe*.

Для обоснования принятия решения по выбору библиотеки приведем анализ метода иерархий [4].

Метод анализа иерархий (МАИ) является систематической процедурой для иерархического представления элементов, определяющих суть любой проблемы. Метод состоит в декомпозиции проблемы на все более простые составляющие части и дальнейшей обработке последовательности суждений на основе парных сравнений. В результате может быть выражена относительная степень (интенсивность) взаимодействия элементов в иерархии. Эти суждения затем выражаются численно. МАИ включает процедуры синтеза множественных суждений, получения приоритетности характеристик и нахождения вариантов решений. Полезно отметить, что полученные таким образом значения являются оценками в шкале отношений и соответствуют так называемым жестким оценкам.

Составим иерархию задачи по выбору библиотеки для распознавания изображения:



Рис. 5. Иерархия задачи

На основании матриц парных сравнений для трех уровней, была составлена результирующая таблица:

Таблица 1. Результирующая таблица

	Размер	Скорость	Точность	
	0,55	0,27	0,18	
OpenCV	0,48	0,09	0,06	0,29
TensorFlow	0,09	0,51	0,51	0,28
TesseractOCR	0,14	0,27	0,22	0,19
Caffe	0,30	0,14	0,22	0,24

На основании значения компонентов вектора обобщенных приоритетов, выбран вариант, имеющий наибольшую величину – OpenCV.

5. Заключение

В ходе исследования для задачи считывания показаний артериального давления тонометров по дисплею выбран инструмент для распознавания образов – библиотека OpenCV.

После применения методов для распознавания изображений, таких как перевод изображения в черно-белый оттенок (оттенки серого), размытия изображения, применение обычного порогового преобразования, чтобы достигнуть определенного порогового значения между пикселями, а также использование алгоритма ближайшего соседа, получаем результат – распознанные показания артериального давления по дисплеям тонометров.

Приведем пример работы программы, выполненной на языке программирования Python [6].



Рис. 6. Входное изображение

После загрузки изображения, происходит экспозиция изображения (регулировка яркости):



Рис.7. Окно с измененной яркостью изображения

Применяем обычное пороговое преобразование пикселей изображения:



Рис.8. Обычное пороговое преобразование

Чтобы легче и четче было найти контуры цифр, применяем размытие (уменьшаем расстояние разрыва между символами):



Рис.9. Применение наращивания

В результате получим распознанные показания артериального давления в текстовом формате (в консоли):

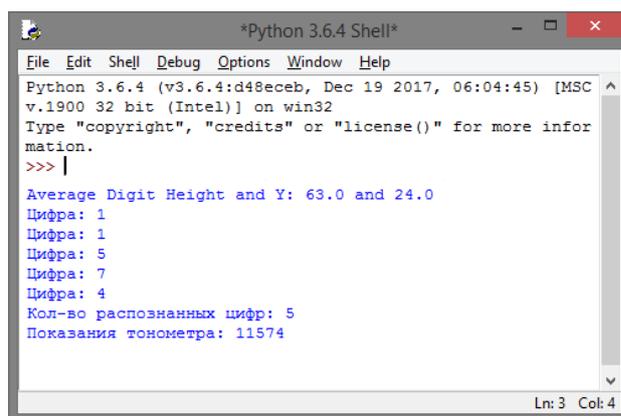


Рис.10. Результат работы программы

Таким образом, рассмотрен инструмент распознавания образов – библиотека OpenCV, сравнение ее аналогов, а также приведено краткое описание его моделей распознавания изображений и пример работы программы.

Список используемых источников

1. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python / Мюллер А. Гвидо С. – Самиздат, 2017. – 393 с.
2. Кобалава Ж.Д., Котовская Ю.В. Мониторирование артериального давления: методические аспекты и клиническое значение / Моисеев В.С.- Москва, 1999.
3. Глория Б.Г., Оскар Д.С., Хосе Л.А., Хесус С.Т., Исмаэль С.Г., Ноэлия В.Э. Обработка изображений с помощью OpenCV / пер. с англ. Слинкин А.А. – М.: ДМК Пресс, 2016. – 210 с.
4. Насыров Р.В. Принятие решений на основе метода анализа иерархий к лабораторной работе N5 по курсу "Теория принятия решений / Методические указания ФГБОУ ВПО Уфимск. гос. техн. унив. – Уфа, 2011. – 23 с.
5. Документация библиотеки OpenCV [Электронный ресурс] – Режим доступа: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html, свободный (Дата обращения 20.03.2019).
6. Официальный сайт Python [Электронный ресурс] – Режим доступа: <https://www.python.org/doc/>, свободный (Дата обращения 15.03.2019).