УДК 004.932

ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ ДИАГНОСТИКИ ПРОЧНОСТНЫХ СВОЙСТВ ТОНКИХ ПЛЕНОК

С. И. РОГОВСКИЙ, канд. физ.-мат. наук, доц. С. А. ВАБИЩЕВИЧ, Н. В. ВАБИЩЕВИЧ
(Полоцкий государственный университет имени Евфросинии Полоцкой, Беларусь)

Аннотация. *Рассмотрены вопросы применения цифровой обработки изображений отпечатков, полученных в ходе испытаний материала на микротвердость. Построен алгоритм и реализована программа обработки изображений. Результаты моделирования использованы для оценки прочностных характеристик полимерных пленок после испытаний на микротвердость.*

Ключевые слова: *анализ и обработка изображения, программирование, поверхность, микроиндентирование, микротвердость.*

Введение. В настоящее время сфера цифровой обработки изображений становится неотъемлемой составляющей в различных областях науки и технологий. Одним из перспективных направлений применения цифровых методов обработки изображений является их использование в качестве инструмента для определения физических и химических свойств материалов. Новаторские подходы, основанные на алгоритмах обработки цветных изображений, позволяют с применением технологии компьютерного зрения эффективно анализировать и характеризовать разнообразные материалы, внося значительный вклад в развитие современных технологий материаловедения.

Цель работы состояла в развитии оригинального программного обеспечения, позволяющего проводить полноценную обработку изображений отпечатков индентора, полученных в ходе испытаний на микротвердость полимерных пленок (рисунок 1). Указанная обработка позволит определить контуры и площади зон разрушения вокруг отпечатков, контуры и длины диагоналей отпечатков и длины образовавшихся в углах отпечатков трещин [1, 2].

Входное изображение, как правило, состоит из нескольких компонентов, которые определяют его структуру, содержание и характеристики. Вот основные компоненты, из которых может состоять входное изображение [3]:

– Пиксели (Pixels): Пиксели являются основными элементами изображения. Каждый пиксель представляет собой точку на изображении, имеющую определенные значения интенсивности или цвета. Количество пикселей определяет разрешение изображения.

– Разрешение (Resolution): Разрешение изображения определяет количество пикселей, содержащихся в изображении по горизонтали и вертикали. Оно обычно измеряется в пикселях по горизонтали (ширина) и пикселях по вертикали (высота).

– Цветовая модель (Color Model): Цветовая модель определяет способ представления цвета на изображении. Некоторые из наиболее распространенных цветовых моделей включают RGB (красный, зеленый, синий), CMYK (циан, маджента, желтый, черный), HSV (оттенок, насыщенность, значение) и оттенки серого.

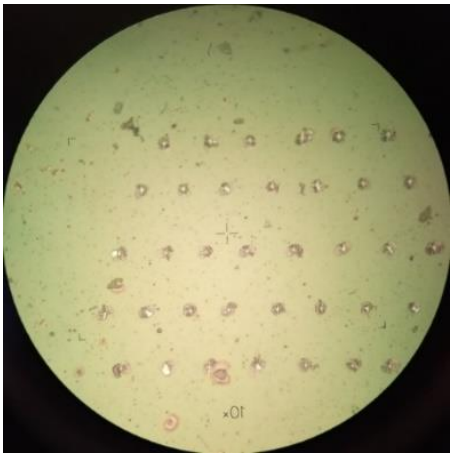
– Каналы (Channels): Каждая цветовая модель может быть представлена в виде набора каналов, каждый из которых представляет один из цветовых компонентов. Например, в RGB-изображении у каждого пикселя есть три канала: красный, зеленый и синий.

– Глубина цвета (Color Depth): Глубина цвета определяет количество битов, используемых для представления цвета каждого пикселя на изображении. Чем выше глубина цвета, тем больше цветов может быть представлено на изображении, и тем более детализированным будет изображение.

– Метаданные (Metadata): Метаданные представляют собой дополнительную информацию об изображении, такую как дата создания, разрешение, камера или устройство, с которого было сделано изображение, и другие сведения.

Учитывая эти компоненты, анализ и обработка изображений могут быть адаптированы под конкретные характеристики входного изображения, что способствует повышению эффективности применения технологий компьютерного зрения и достоверности полученных результатов.

Рисунок 1.– Входное изображение, включающее отпечатки индентора на поверхности пленки полимера



Для успешного анализа и определения свойств материалов при помощи цифровой обработки изображений программисты используют различные этапы обработки, включающие в себя следующие ключевые функции библиотеки OpenCV: cv2.medianBlur(), cv2.cvtColor, cv2.findContours, cv2.drawContours, cv2.contourArea(contours[i]).

Функция cv2.medianBlur() представляет собой метод медианного сглаживания изображений. Этот метод применяется для уменьшения шумов на изображении путем замены каждого пикселя на медианное значение яркости в окрестности этого пикселя. Медианный фильтр является одним из наиболее эффективных методов сглаживания изображений, поскольку он сохраняет резкие края объектов на изображении, в отличие от других методов, таких как среднее значение или гауссовское размытие. Эта функция принимает следующие значения:

– src: Исходное изображение, которое требуется обработать. Изображение должно быть в формате grayscale или color.

– ksize: Размер окна медианного фильтра, должен быть нечетным положительным числом.

Функция возвращает отфильтрованное изображение того же размера и типа данных, что и исходное.

Функция cv2.cvtColor() используется для изменения цветового пространства изображения. Она позволяет перевести изображение из одного цветового пространства в другое. Это может быть полезно для адаптации изображения под требования конкретной задачи обработки. Функция принимает следующие значения:

– src: Исходное изображение, которое требуется преобразовать. Изображение должно быть в формате grayscale, RGB или другом цветовом формате, поддерживаемом OpenCV.

– code: Код цветового пространства, в которое требуется перевести изображение. Например, для перевода изображения в оттенки серого используется код cv2.COLOR_BGR2GRAY, а для перевода из BGR в HSV – cv2.COLOR_BGR2HSV.

Функция возвращает преобразованное изображение в новом цветовом пространстве.

Функция cv2.findContours() используется для выделения контуров объектов на бинарном изображении. Контур представляет собой кривую, состоящую из набора точек, представляющих границы объекта. Функция принимает следующие значения:

– image: Бинарное изображение (чаще всего монохромное), на котором производится поиск контуров. Обычно это результат применения пороговой обработки или других операций обнаружения краев.

– mode: Определяет режим поиска контуров. Может быть задан одним из значений, таких как cv2.RETR_EXTERNAL, cv2.RETR_LIST, cv2.RETR_TREE и другие. Эти режимы определяют, как контуры связаны друг с другом и как они организованы в иерархии.

– method: Метод аппроксимации контуров. Это аппроксимация контуров, используемая для сжатия контура. Часто используется cv2.CHAIN_APPROX_SIMPLE.

Функция возвращает список контуров (список массивов координат точек) и, в некоторых случаях, иерархию контуров, особенно если был использован метод cv2.RETR_TREE.

Функция cv2.drawContours() используется для отображения контуров на изображении. Эта функция позволяет визуализировать найденные контуры, что упрощает визуальное понимание результатов обработки изображений. Функция принимает следующие значения.

- image: Изображение, на котором будут отрисованы контуры.
- contours: Список контуров, полученных из функции cv2.findContours().
- contourIdx: Индекс контура, который должен быть отрисован. Указание -1 приводит к отрисовке всех контуров из списка.
- color: Цвет контура.
- thickness: Толщина линии контура.

Функция не возвращает значения, но изменяет переданное изображение, добавляя контуры на него.

Функция cv2.contourArea() используется для расчета площади контура, заданного вектором точек. Площадь контура является важным параметром при анализе размеров и формы объектов на изображении.

Функция принимает следующие значения

- contours[i]: Контур, площадь которого требуется вычислить. Каждый контур представлен в виде списка точек.

Функция возвращает площадь контура в единицах пикселей.

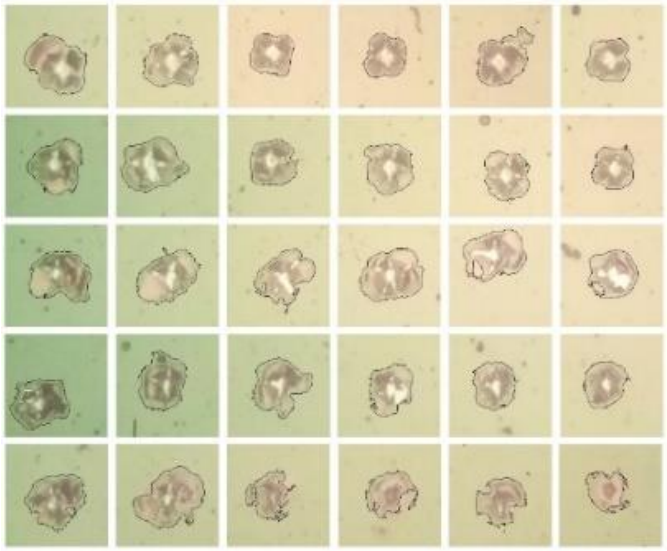
Все рассмотренные методы цифровой обработки изображений в библиотеке OpenCV представляют собой инструменты, позволяющие эффективно анализировать, визуализировать и извлекать информацию из изображений. Их применение открывает новые возможности для разработки высокоточных систем распознавания, анализа и классификации объектов на изображениях.

Пример кода, включающий в себя все вышеописанные функции обработки изображений:

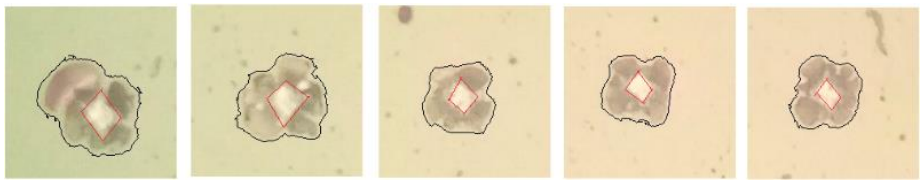
```
import cv2
# Загрузка изображения
image = cv2.imread('input_image.jpg')
# Применение медианного сглаживания
blurred_image = cv2.medianBlur(image, 5)
# Преобразование изображения в оттенки серого
gray_image = cv2.cvtColor(blurred_image, cv2.COLOR_BGR2GRAY
# Поиск контуров
contours, hierarchy = cv2.findContours(gray_image, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE
# Отрисовка контуров на изображении
image_with_contours = cv2.drawContours(image.copy(), contours, -1, (0, 255, 0), 2)
# Расчет площади контуров
for contour in contours:
area = cv2.contourArea(contour)
print(f"Площадь контура: {area}")
```

```
# Отображение результата
cv2.imshow('Image with Contours', image_with_contours)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Этот код загружает изображение, применяет к нему медианное сглаживание, преобразует его в оттенки серого, находит контуры, рисует их на исходном изображении и выводит площади контуров. Для улучшения процесса подбора параметров можно использовать методы оптимизации, такие как методы оптимизации гиперпараметров или автоматический подбор значений при помощи алгоритмов оптимизации. Эти методы могут помочь автоматизировать процесс подбора параметров, ускорить его и повысить точность обработки изображений. Имея понимание о том, как каждый параметр влияет на конечный результат обработки изображений, и использование эффективных методов оптимизации, можно достичь более точных и надежных результатов. Несмотря на сложности в подборе параметров, современные методы обработки изображений и инструменты оптимизации позволяют преодолевать эти трудности и достигать высококачественных результатов, как показано на рисунке 2.



а



б

Рисунок 2. – Результат выделения контуров вокруг зоны разрушения (*а*) (внешний контур) и отпечатков индентора (*б*) (внутренний контур) на поверхности фоторезиста, нанесенного на кремниевую подложку

Заключение. В данной работе успешно разработан и реализован алгоритм обработки цветных изображений, полученных в ходе фотографирования поверхности материала после проведения испытаний на микротвердость. Основная цель данного алгоритма заключается в точном определении геометрических параметров разрушений и отпечатков на поверхности материала, что имеет критическое значение для объективной оценки прочностных характеристик материалов и автоматизации процесса измерения в области материаловедения. Работа выполнена в рамках задания 2.16 Государственной программы научных исследований «Материаловедение, новые материалы и технологии», подпрограмма «Наноструктурные материалы, нанотехнологии, нанотехника («Наноструктура»)».

ЛИТЕРАТУРА

1. Литвинов, Ю.М. Методология определения механических свойств полупроводниковых материалов с помощью метода непрерывного вдавливания индентора/ Ю.М. Литвинов, М.Ю. Литвинов // Известия вузов. Материалы электронной техники. – 2004. – № 4. – С. 11–16.
2. Колесников, Ю.В. Механика контактного разрушения / Ю.В. Колесников, Е.М. Морозов. – М.: Наука, 1989. – 220 с.
3. Bradski, G. Learning OpenCV. Computer vision with the OpenCV library/ G.Bradski, A.Kaehler // O'Reilly Media, Inc. – 2008.