

Министерство образования и науки Российской Федерации
ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»
Кафедра информационных технологий

Оценка работы _____

Руководитель от УрФУ Спиричева Н.Р.

Программная реализация методов сегментации изображения с
использованием библиотеки компьютерного зрения OpenCV.

ОТЧЕТ

по производственной практике

Руководитель от предприятия

Чирышев Ю.В.

Студент

Смирнов Д.А.

Специальность (направление подготовки)

Информатика и вычислительная техника

Группа

РИ-250003

Введение

В промышленности очень часто возникает задача контроля и улучшения качества изделий. Для ее решения необходимо обнаруживать и локализовывать дефекты в выпускаемой продукции или в сырье. В данной работе рассматривается предметная область, связанная с производством стеклопластика – это композитный материал, который представляет собой стеклоткань, пропитанную различными лаками и смолами. Готовую продукцию, как правило, получают пропитыванием стеклоткани эпоксидной, полиэфирной или фенолформальдегидной смолой с последующим отверждением и распрессовкой. Обрабатываемая таким образом стеклоткань может иметь различные дефекты, образовавшиеся на поверхности волокон. Основные дефекты стеклопластика связаны с неравномерным нанесением – пропиткой стеклянного волокна полимерными связующими (рис. 1). Для достижения оптимального уровня качества выпускаемой продукции необходимо контролировать толщину нанесения полимеров. Происходит это за счет фотографирования стеклоткани на этапе нанесения смол и отвердителя с последующей обработкой изображений специализированным программным обеспечением. Задача состоит в обнаружении и сегментации дефектов на полученных фотоизображениях.

В современных приложениях, работающих с изображениями, все чаще используются сложные математические алгоритмы обработки изображений. Многие разработчики таких приложений заинтересованы в использовании программных модулей, которые помогали бы им решать типовые задачи компьютерного зрения. Одним из таких средств является OpenCV – библиотека алгоритмов компьютерного зрения и обработки изображений с открытым исходным кодом. Она реализована на языке C/C++ и призвана упростить разработку приложений в данной области. Множество функций библиотеки позволяют выполнять базовые операции обработки числовых массивов и изображений. В частности, выполнять фильтрацию изображений, нахождение отличительных признаков изображений, анализ движения, сравнение изображений, обнаружение объектов, восстановление изображений, морфологический анализ и многое другое. Исходя из этого, целью данной работы будет являться разработка алгоритма обнаружения и сегментации дефектов на изображениях стеклянной ткани и его программная реализация с использованием библиотеки OpenCV. Для этого я ставлю перед собой ряд задач: установка необходимых библиотек, изучение литературы по теме, разработка и реализация алгоритма.

Интерфейс

Разработанный интерфейс приложения прост и управляется двумя элементами: кнопкой Open, нужной для открытия файла изображения и ползунком, нужным для установки порогового значения. При перемещении ползунка прогресс обработки отображается в прогрессбаре. Результат, сведения о том, сколько процентов составляет дефекты, выводится в строке состояния.

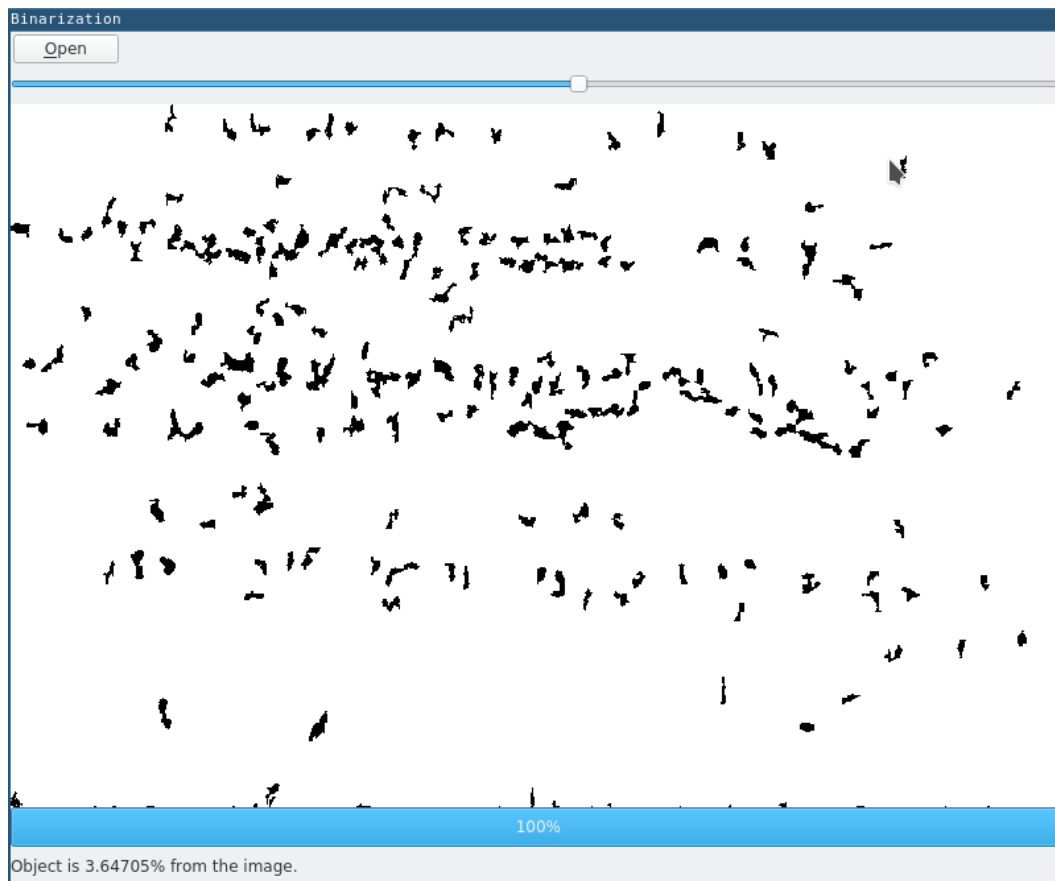


Рис. 1. Скриншот окна программы

При запуске программа просит выбрать файл с изображением. После открытия файла программа установит минимальное и максимальное значения ползунка в соответствии с минимальным и максимальным значением яркости пикселей изображения. Далее производится ручная корректировка порогового значения и чтение результата. Если требуется проанализировать другой файл, то следует нажать кнопку Open.

Алгоритм обработки изображения

Обработка изображения производится в несколько этапов.

Для начала исходное изображение загружается из файла. Исходное изображение изображено на рисунке 2.

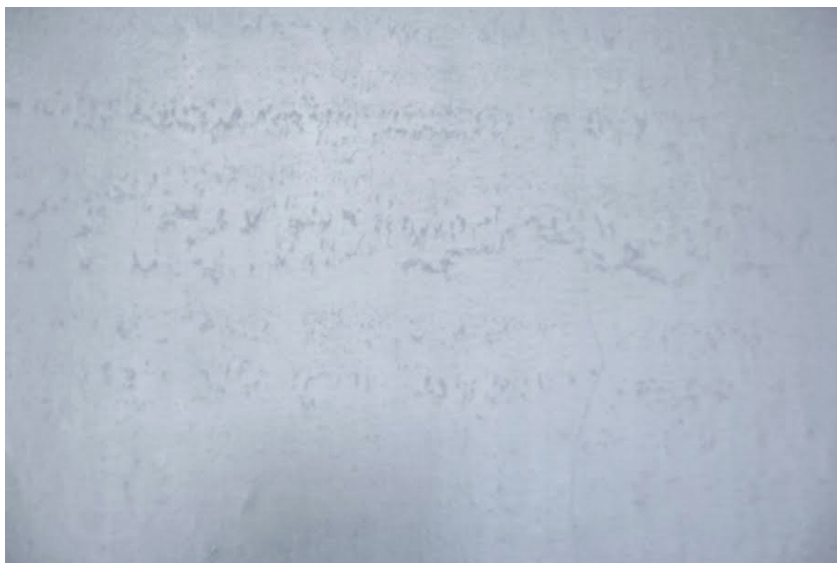


Рис. 2. Исходное изображение

Для более удобной обработки изображение переводится в серые тона. Результат изображен на рисунке 3.

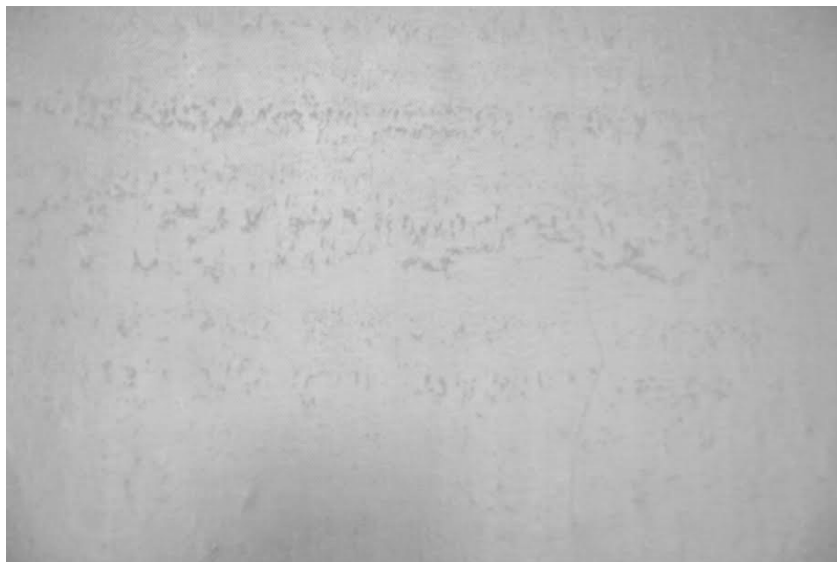


Рис. 3. Изображение в серых тонах

Можно заметить, что исходное изображение имеет очень не равномерный уровень освещенности для его различных областей. Для коррекции применяется следующий алгоритм:

1. Для каждого пикселя вычисляется средний уровень освещенности его окрестности определенного радиуса (для примера был экспериментально установлен оптимальный

радиус равный 10 пикселям). Так как изображение было переведено в серые тона, то значение каждого пикселя представляется одним числом – его уровнем освещенности.

2. Вычисляется средняя освещенность всего изображения.
3. Значение каждого пикселя результирующего изображения устанавливается по следующей формуле: $v_{рез} = v_{исх} + l_{изображения} - l_{пикселя}$, где $v_{рез}$ – результирующее значение, $v_{исх}$ – исходное значение, $l_{изображения}$ – средняя освещенность изображения, $l_{пикселя}$ – средняя освещенность окрестности пикселя.

Результат работы алгоритма изображен на рисунке 4.



Рис. 4. Изображение с выровненной яркостью

Далее производится бинаризация изображения. Результат показан на рисунке 5.

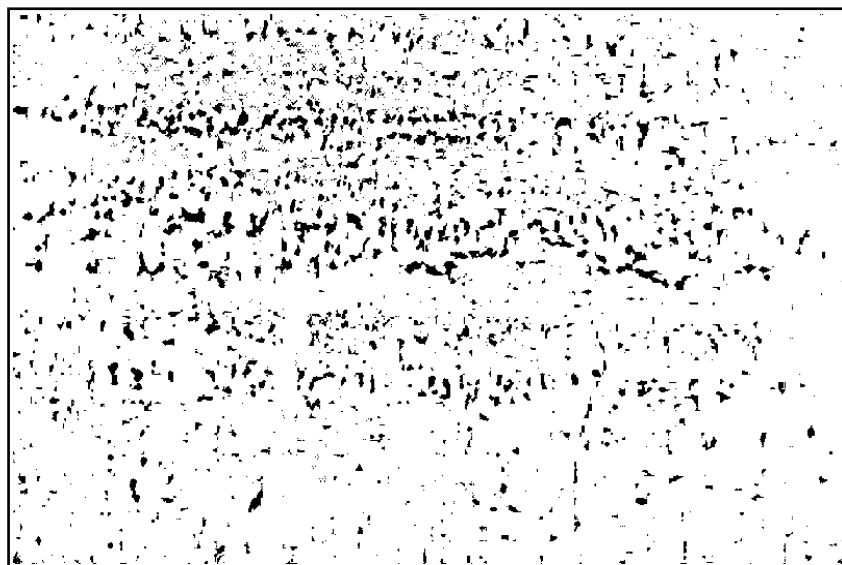


Рис. 5. Бинаризованное изображение

На бинаризованном изображении можно заметить много мусора состоящего из мелких пятен. Эти пятна являются не дефектами нанесения смолы, а возникают из-за съемки на камеру.

Для их ликвидации производится удаление всех пятен площади меньшей определенного критического значения. Это значение было подобрано экспериментально. Результат изображен на рисунке 6.

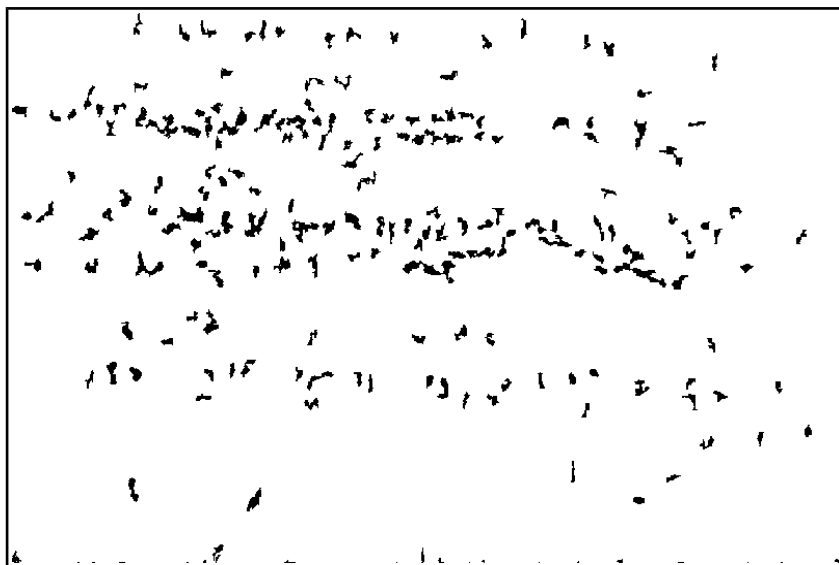


Рис. 6. Очищенное от мусора изображение

Дальше просто производится подсчет процента площади изображения, занимаемой черными пикселями. Для этого подсчитывается количество черных пикселей и подставляется в

следующую формулу:
$$\text{percent} = \frac{\text{black}}{\text{width} \cdot \text{height}} \cdot 100\%$$
 В приведенном примере полученное

значение равно 3,6%.

Изученные классы и функции библиотеки OpenCV

В ходе работы я изучил и применил на практике следующие классы и функции библиотеки OpenCV:

- `Mat` – класс матрицы
- `Mat imread(const string& filename, int flags = 1)` – функция загрузки файла изображения
- `void cvtColor(InputArray src, OutputArray dst, int code, int dstCn = 0)` – функция преобразования цветового пространства. Использовалась для преобразования изображения из цветного в оттенки серого, для этого параметр `code` был установлен в `CV_BGR2GRAY`
- `double threshold(InputArray src, OutputArray dst, double thresh, double maxval, int type)` – функция бинаризации. Использовалась для выделения дефектов.
- `void adaptiveThreshold(InputArray src, OutputArray dst, double maxValue, int adaptiveMethod, int thresholdType, int blockSize, double C)` – функция адаптивной бинаризации, позволяющая для каждого пикселя вычислять собственное пороговое значение. Была изучена, но на практике давала результаты хуже, чем простая бинаризация.
- `void resize(InputArray src, OutputArray dst, Size dsize, double fx = 0, double fy = 0, int interpolation = INTER_LINEAR)` – функция масштабирования матрицы изображения, может применяться в двух вариантах: масштабирование к фиксированному размеру и масштабирование с коэффициентом. Применялась для уменьшения изображения до приемлемого размера.

Вывод

В ходе работы я установил необходимые библиотеки, изучил литературу по теме и разработал алгоритм обнаружения и сегментации дефектов на изображениях стеклянной ткани и его реализацию с использованием библиотеки OpenCV. Таким образом, цель практической работы достигнута.