

1 Исходные данные

В качестве исходных данных было представлено изображение плохого качества.

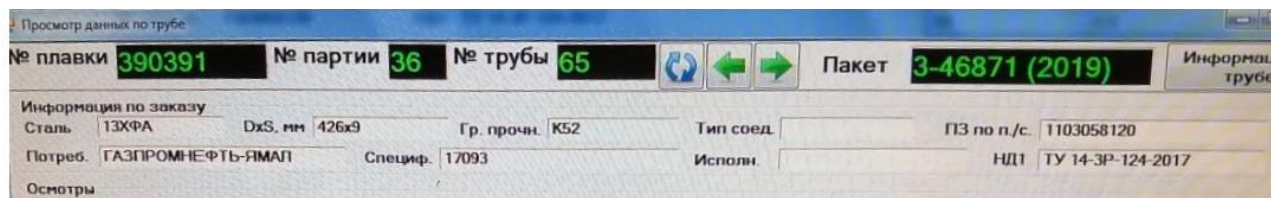


Рисунок 1 – Исходное изображение

Целью данной работы было обработать исходное изображение и получить как можно больше полезной информации.

2 Извлечение информации

Местоположение и размеры полей на изображении будут постоянными. Этого можно добиться простым printscreen-ом или, для особых случаев, захватом видеопотока с рабочего стола.

На первом этапе происходит выделение полей изображения с полезной информацией и получается двоичная маска:



Рисунок 2 – Двоичное изображение

Далее происходит разделение двоичной маски на составные части (13 частей) и сохраняются в отдельные файлы. Для отсутствия потери качества необходимо использовать расширение png для всех файлов изображений.

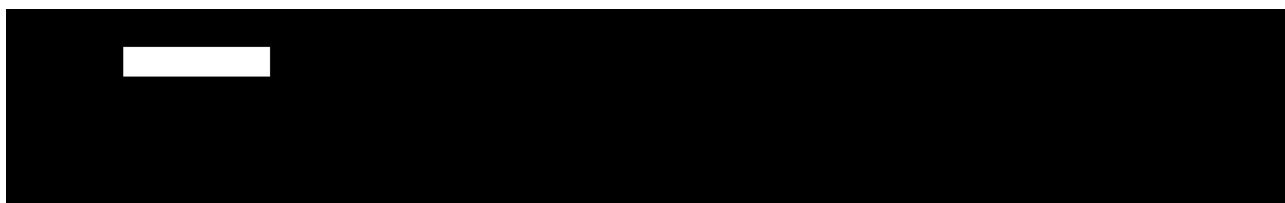


Рисунок 3 - Пример одной маски с именем mask1.png

Далее заполняется файл description.txt

Таблица 1 – Заполнение файла

Название маски	Параметр
mask1.png	№ плавки
mask2.png	№ партии
mask3.png	№ трубы
mask4.png	Пакет
mask5.png	Сталь
mask6.png	DxS, мм
mask7.png	Гр. прочн.
mask8.png	Тип соедин.
mask9.png	ПЗ по п./с.
mask10.png	Потреб.
mask11.png	Специф.
mask12.png	Исполн.
mask13.png	НД1

В таблице 1 создана ассоциация между именем файла маски и именем соответствующего поля. По большей части эта ассоциация влияет только на то, как программа будет называть распознанные данные, поэтому эти имена полей можно менять.

3 Процесс обработки изображения

Предварительно вычисляются параметры масок и вырезаются нужное изображение из исходного:

Производится увеличение изображения и бикубическая интерполяция, что позволяет сгладить границы увеличенного изображения и получить более высокое качество распознавания текста при данных низкого разрешения:

Убираются шумы серого и белых цветов, вызванные матрицей монитора. Выделяются значимые изменения цвета по динамическому порогу Оцу, чтобы выделить текст. Инвертируется светлый текст на черном фоне:

Настраивается модуль Tesseract с последующим его вызовом.

4 Общее описание команд

«--l rus» установка русского языка;

«--oem 2» установка «ocrenginemode»

2 Запустить оба (Tesseract и Cube) и объединить результаты - лучшая точность

«--psm 7» Установка «pagesegmode»

7 Обработать изображение как одну текстовую строку.

```
config = ("--l rus --oem 2 --psm 7")
```

```
text = pytesseract.image_to_string(gray, config=config)
```

5 Оценка точности

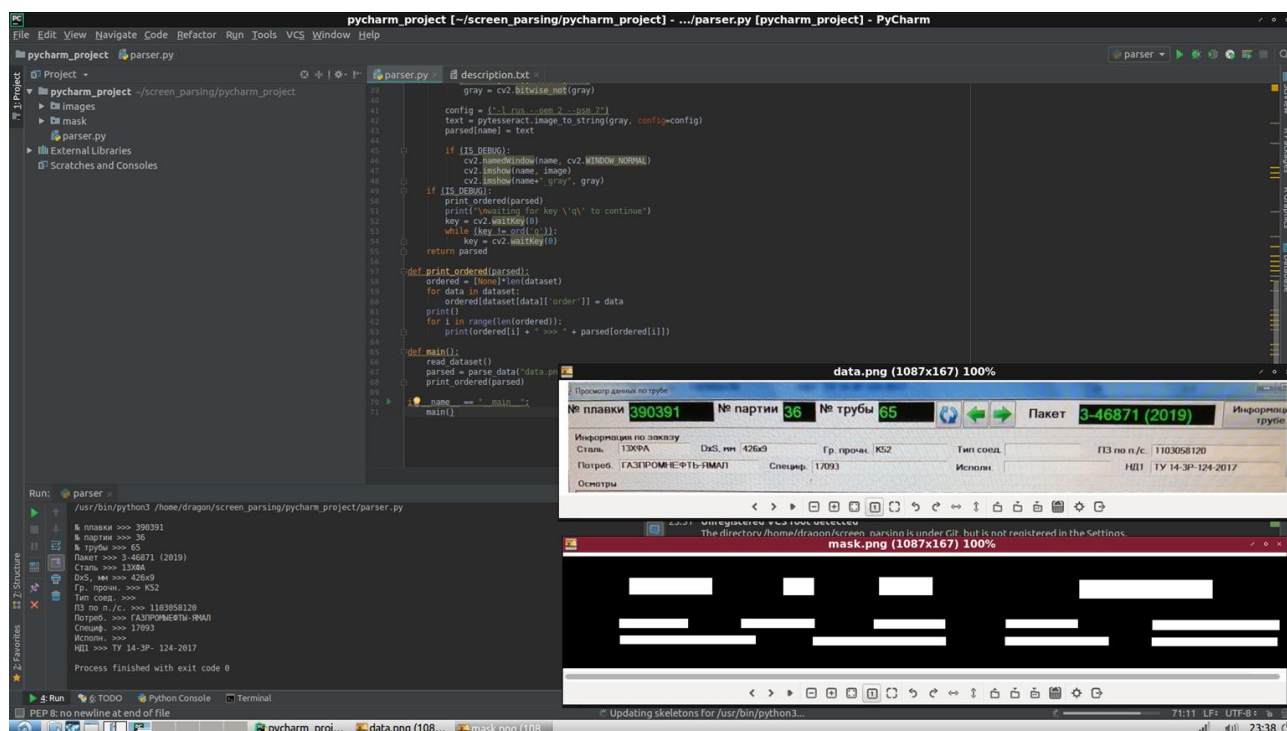


Рисунок 4 – Пример работы программы

Таблица 2 – Результат работы

Название маски	Параметр
№ плавки	390391
№ партии	36
№ трубы	65
Пакет	3-46871 (2019)
Сталь	13ХФА
DxS, мм	426x9
Гр. прочн.	K52
Тип соед.	NaN
ПЗ по п./с.	1103058120
Потреб.	ГАЗПРОМЫЕФТЫ-ЯМАЛ
Специф.	17093
Исполн.	NaN
НД1	ТУ 14-3Р- 124-2017

6 Разбор недочетов

Несерьёзная ошибка - появление лишних пробелов.

Пример «ТУ 14-3Р-124-2017» вместо «ТУ 14-3Р-124-2017»

Серьёзная ошибка — ошибка распознавание текста.

Пример «ГАЗПРОМЫЕФТЫ-ЯМАЛ» вместо «ГАЗПРОМНЕФТЬ-ЯМАЛ»

Но так как данные были достаточно плохого качества:



ГАЗПРОМНЕФТЬ-ЯМАЛ

При таком исходном разрешении изображения не удивительно, что символы распознаются неверно.

А вот это уже результат работы фильтров. Ситуация улучшилась, но разобрать и отделить некоторые символы по-прежнему сложно — не хватает качества изображения.

ГАЗПРОМНЕФТЬ-ЯМАЛ

Выводы

Из исходных данных удалось получить достаточно информации для успешного распознавания чисел и простого текста. Однако для успешного распознавания любых текстов необходимо значительно более высокое качество изображения.