

ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Т.Г. ШЕВЧЕНКО

Рыбницкий филиал

Кафедра информатики и программной инженерии

НАУЧНЫЙ ПРОЕКТ

на тему:

«РАСПОЗНАВАНИЕ НОМЕРОВ И ПОКАЗАНИЙ СЧЁТЧИКОВ»

Команда проекта:
студенты 1-го курса,

направления «Программная инженерия» профиля
«Разработка программно- информационных систем»

Стефанцов Дмитрий,

Глинка Илья,

Строжевский Дмитрий,

Полищук Евгений,

Лазарев Павел

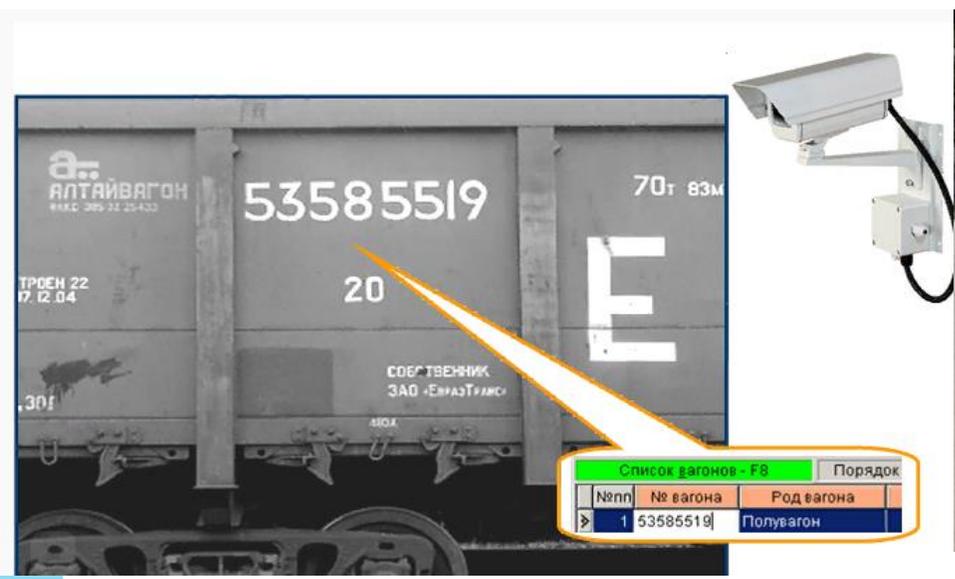
Научные руководители:

доцент Козак Л.Я.,

ст. преподаватели Глазов А.Б., Гарбузняк Е.С.

АКТУАЛЬНОСТЬ

Необходимость разработки решения данной проблемы обусловлена упрощением и ускорением работы предприятий, в основе которых лежит многократная проверка номеров автомобилей, вагонов поездов и счетчиков.



НЕ! - Цифры в красной рамочке (после запятой)
нужны! - Нули вначале

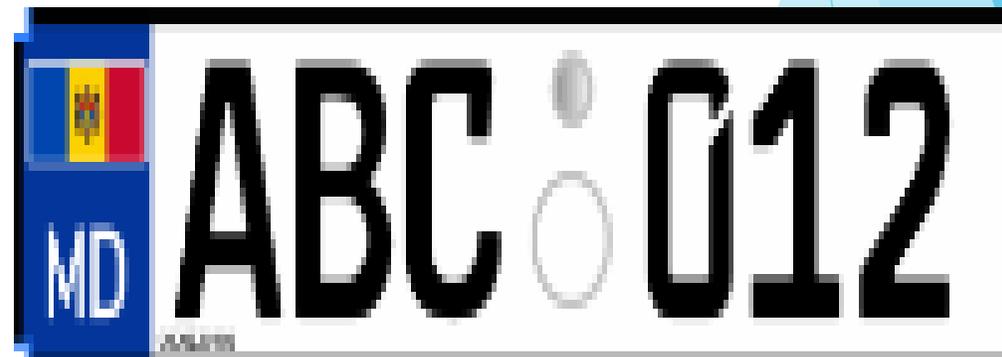
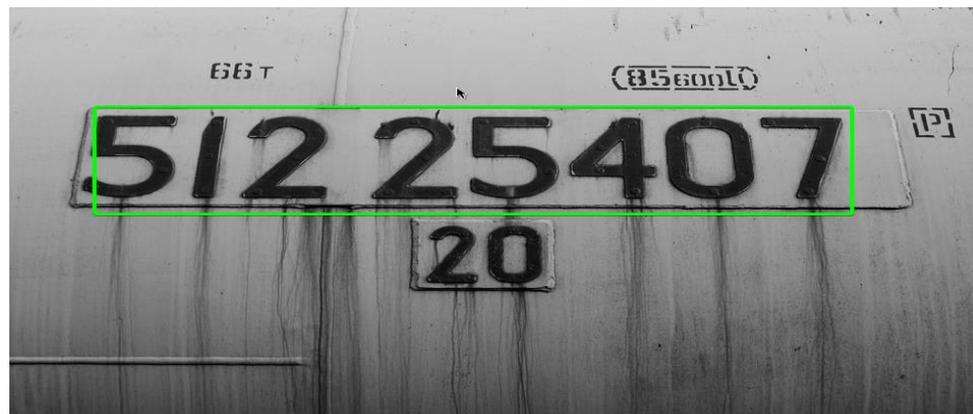
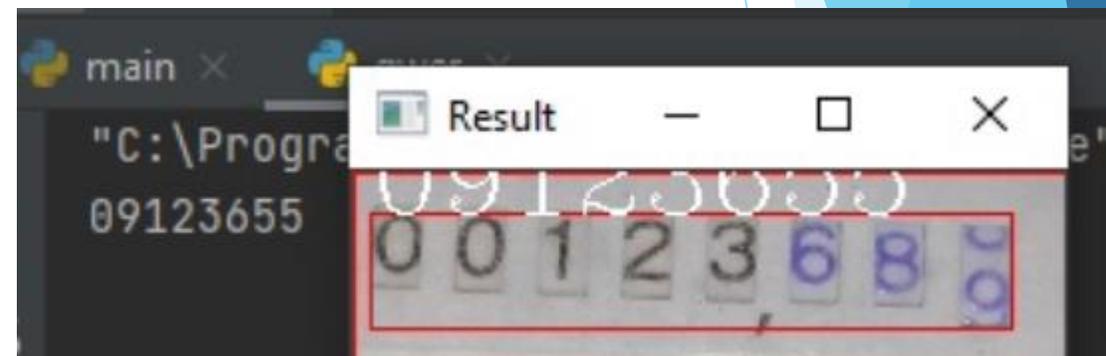
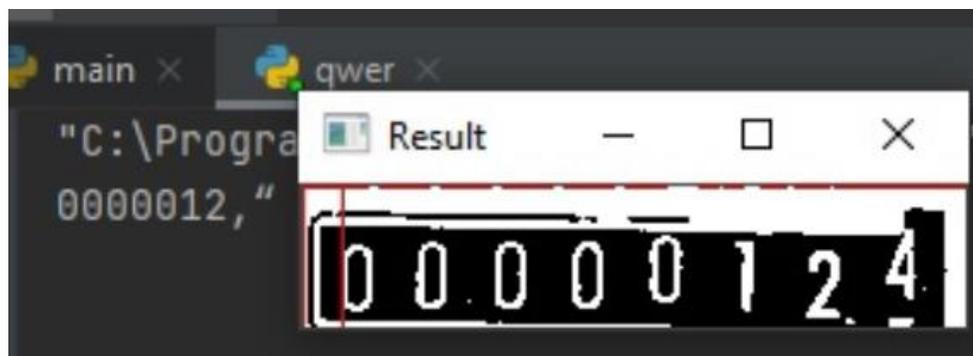
НОРМАИС
00431.629 м³
СВК-15Г
К=0,8888 м³/имп.
Qn 1,5 м³/ч
В-Н/А-У
1 МПа
0:49
90°

2

Показания, которые нужно передать: 431

ОБЪЕКТ И ПРЕДМЕТ

- ▶ **Объектом** научного проекта является распознавание изображений.
- ▶ **Предметом** научного проекта являются методы нахождения цифр из изображений с помощью множественной фильтрации и нейронной сети.

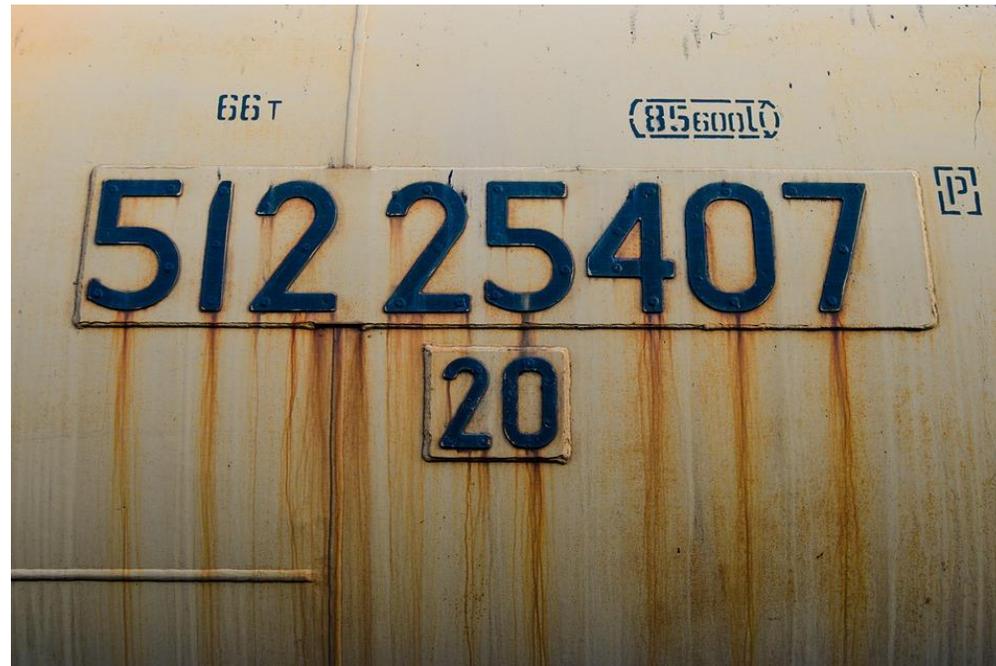


ЦЕЛЬ И ЗАДАЧИ

Цель научного проекта – разработка программы для нахождения номеров.

Задачи:

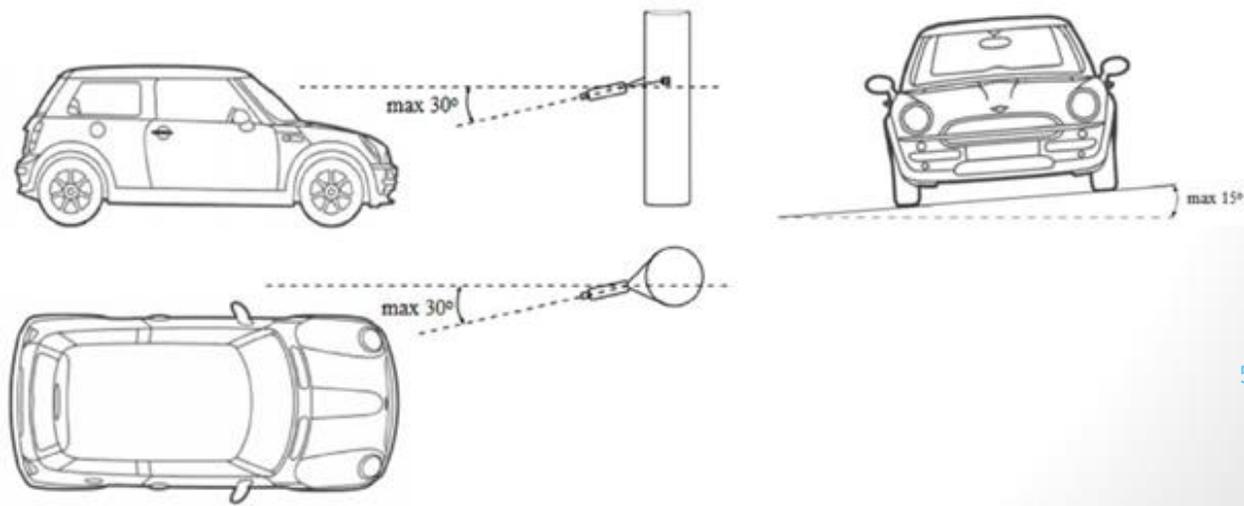
1. Нахождение и определение номеров вагонов и машин с помощью множественной фильтрации.
2. Нахождение и распознавание показаний счетчиков с помощью бинаризации и оптического движка (Tesseract).
3. Переход от фильтрации к нейронной сети.



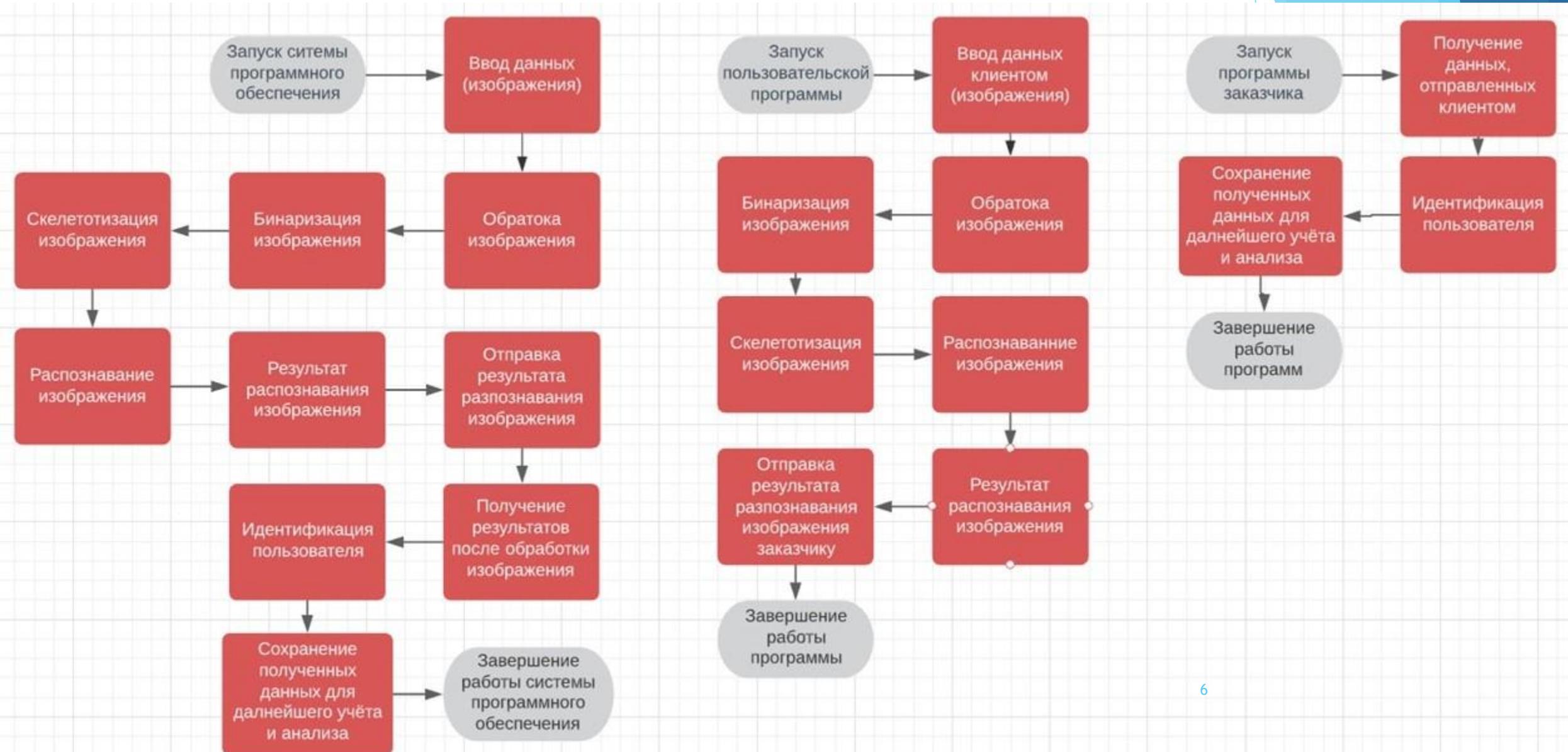
АЛГОРИТМ РАСПОЗНАВАНИЯ ТРАНСПОРТНЫХ НОМЕРОВ

Требования к получаемому изображению автомобильного и вагонного номеров:

- ▶ отсутствие видимого глазом скоростного «смазывания» на изображении номера движущегося автотранспортного средства;
- ▶ размер однострочного номера на изображении должен быть не менее (ширина x высота): 120 x 20 пикселей;
- ▶ перспективные искажения изображения номера, возникающие в результате наклона и поворота плоскости номерной пластины относительно оси камеры на угол, не должны превышать 30° ;
- ▶ отклонение изображения номерной пластины по горизонтали не должно превышать 15° .



АЛГОРИТМ РАСПОЗНАВАНИЯ ПОКАЗАНИЙ СЧЁТЧИКОВ



МЕТОДЫ НАХОЖДЕНИЯ ЦИФР

1. С помощью фильтров PyTesseract.

- Множественная фильтрация и вывод правильного варианта из всех найденных чисел

2. С помощью искусственного интеллекта.

- Обучение нейронной сети.

Фильтрация PyTesseract

```
print("1 стадия")
for i in range (0, 255, 10):...

print ("2 стадия")
for i in range (0, 255, 10):...

print("3 стадия")
for i in range(0, 255, 10):...

print("4 стадия")
for i in range(0, 255, 10):...

print("5 стадия")
for i in range(0, 255, 10):...

print("6 стадия")
for i in range(0, 255, 10):|...

print("7 стадия")
for i in range(0, 255, 10):...

print("8 стадия")
for i in range(0, 255, 10):...
```

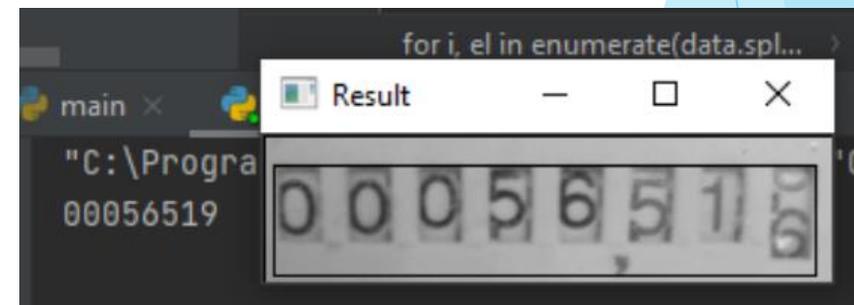
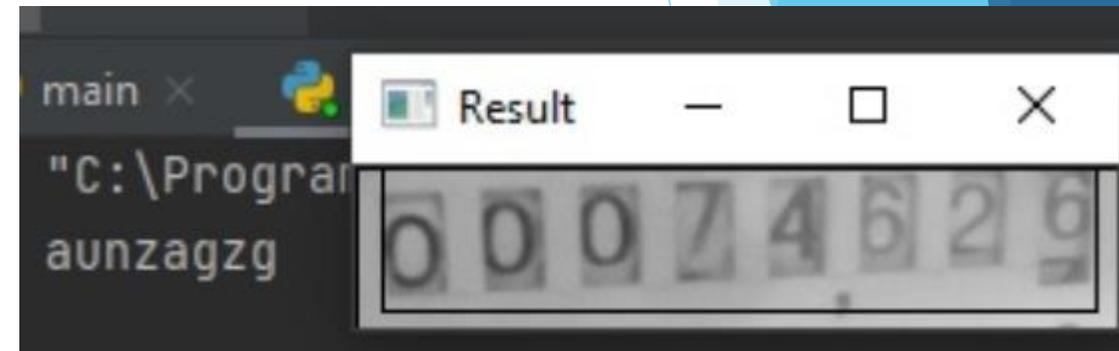
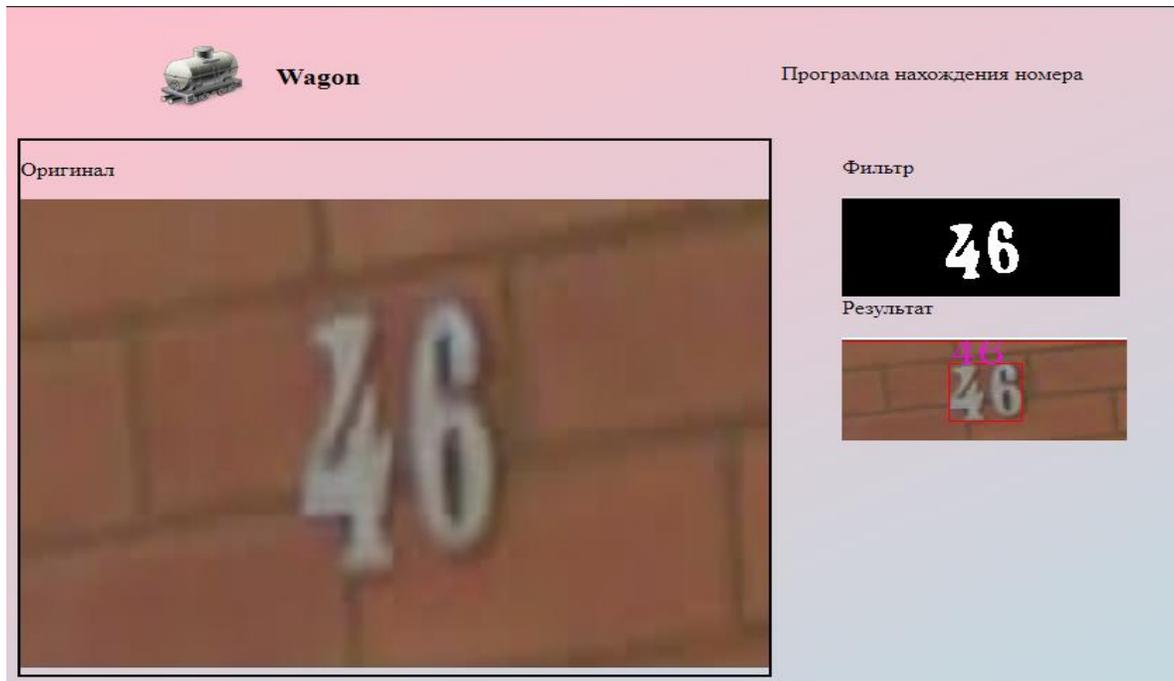
РАЗБОР ФИЛЬТРОВ

```
print("1 стадия")
for i in range(0, 255, 10):
    blur = cv2.GaussianBlur(gray, (5, 5), 0)
    thresh = cv2.threshold(blur, i, 255, cv2.THRESH_BINARY)[1]
    data = pytesseract.image_to_data(thresh, config=tesdata_dir_config)
    for k, el in enumerate(data.splitlines()):
        if k == 0:
            continue
        try:
            el = el.split()
            x, y, w, h = int(el[6]), int(el[7]), int(el[8]), int(el[9])
            cv2.rectangle(img, (x, y), (w + x, h + y), (0, 0, 255), 1)
            cv2.putText(img, el[11], (x, y), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 255), 1)
            if isinstance(int(el[11]), int):
                print(int(el[11]))
                print(i)
                cv2.imshow("Результат", thresh)
                cv2.waitKey(0)
                cv2.imshow("Результат", img)
                cv2.waitKey(0)
        except:
            print("...")
```

```
thresh = cv2.threshold(blur, i, 255, cv2.THRESH_BINARY)[1]
thresh = cv2.threshold(blur, i, 255, cv2.THRESH_BINARY_INV)[1]
thresh = cv2.threshold(blur, i, 255, cv2.THRESH_TRUNC)[1]
thresh = cv2.threshold(blur, i, 255, cv2.THRESH_TOZERO)[1]
thresh = cv2.threshold(blur, i, 255, cv2.THRESH_TOZERO_INV)[1]
thresh = cv2.threshold(blur, i, 255, cv2.THRESH_OTSU)[1]
thresh = cv2.threshold(blur, i, 255, cv2.THRESH_MASK)[1]
thresh = cv2.threshold(blur, i, 255, cv2.THRESH_TRIANGLE)[1]
```

Работа программы:

1. Импорт библиотеки и фото
2. Серая фильтрация фотографий
3. Множественная обработка фотографий в количестве 8-ми этапов
4. Вывод результатов



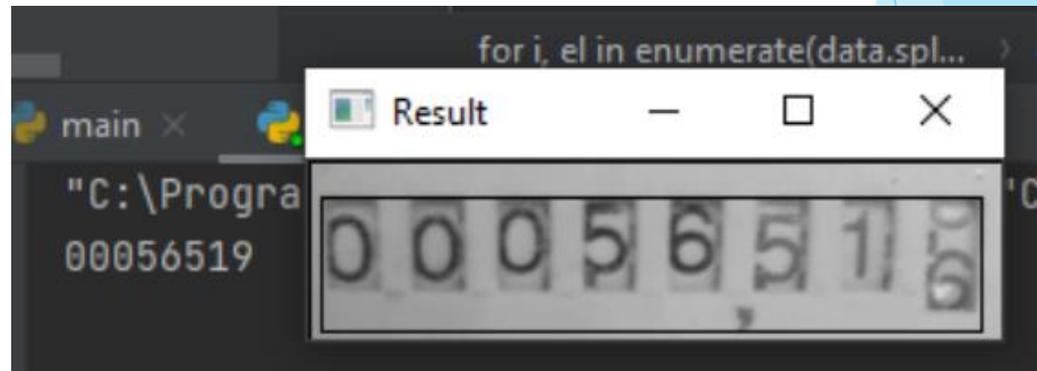
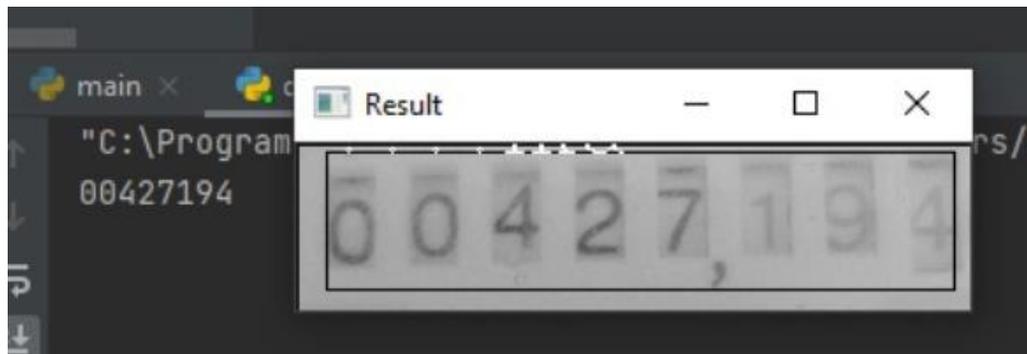
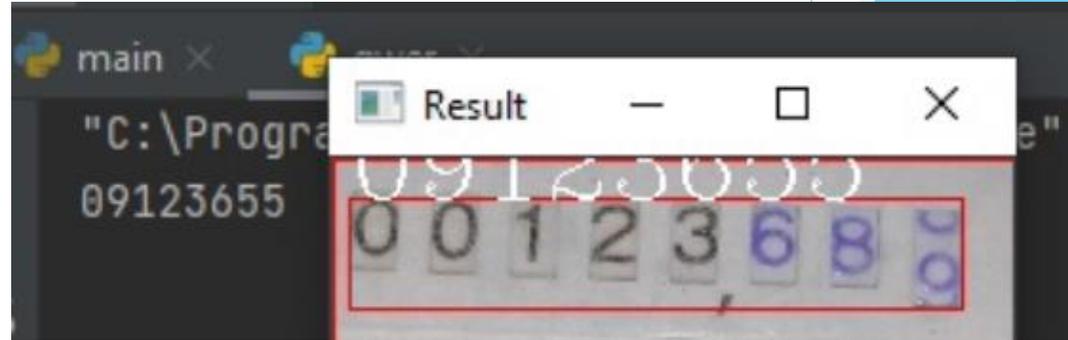
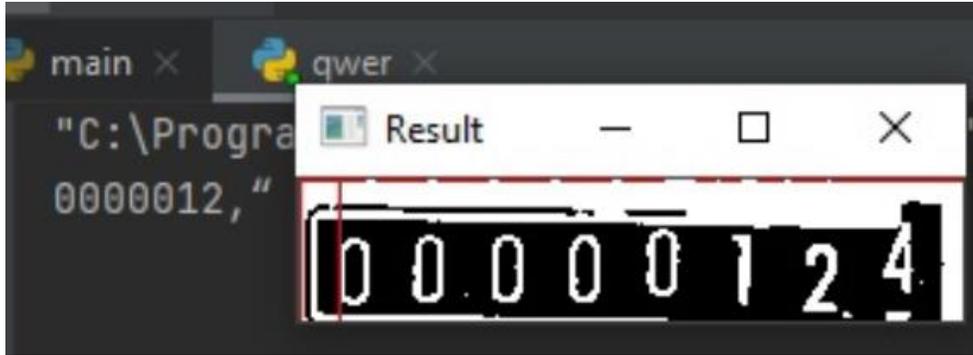
БИНАРИЗАЦИЯ ИЗОБРАЖЕНИЯ С ПОКАЗАНИЯМИ СЧЁТЧИКОВ

```
# Бинаризация PIL
def binar(self):
    ld = self.image.load()
    width, height = self.image.size
    for y in range(height):
        for x in range(width):
            r, g, b = ld[x, y]
            h, s, v = colorsys.rgb_to_hsv(r / 255., g / 255., b / 255.)

            if s > 0.5:
                ld[x, y] = (0, 0, 0)
            else:
                ld[x, y] = (255, 255, 255)

    self.photo = ImageTk.PhotoImage(self.image)
    self.display.itemconfigure(self.display_img, image=self.photo, anchor="nw")
```

ОТПРАВКА ИЗОБРАЖЕНИЯ В TESSERACT



ПРИЧИНЫ ПЕРЕХОДА НА НЕЙРОСЕТИ

- ▶ Точность распознавания показаний счётчиков мала, около 80% (любая ошибка критична в распознавании).
- ▶ Скорость распознавания низкая(относительно).
- ▶ Нейросети в данном случае могут дать в точности до 95% при достаточно большом наборе данных.

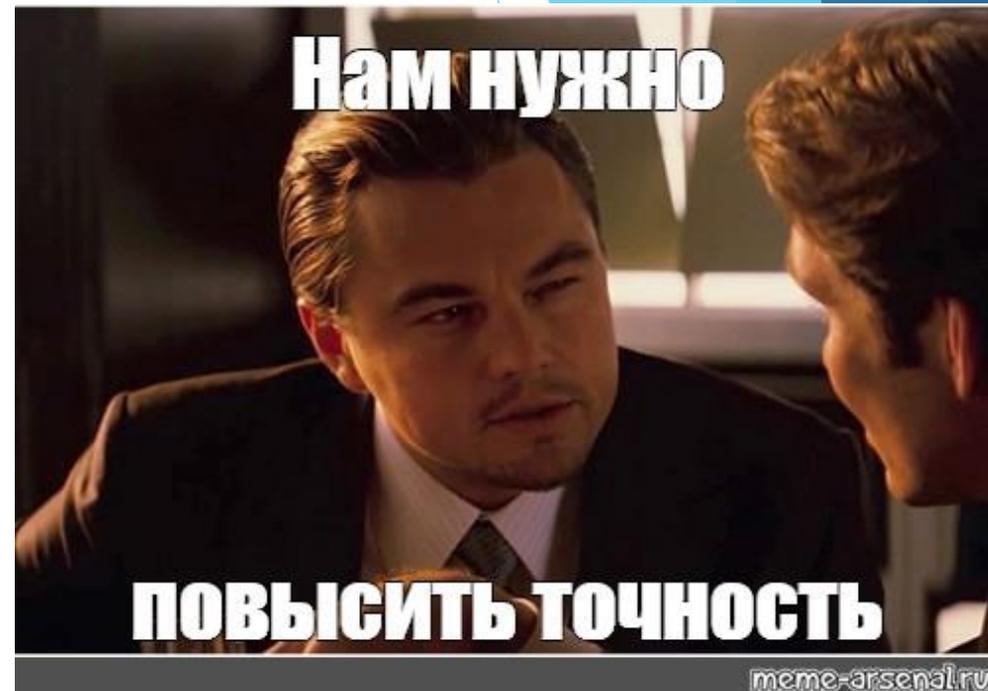
Результаты по распознаванию изображений:

Плюсы:

1. Скорость составляет: 1мин.30сек. на 5 фото.
2. Точность 75%.
3. Распознавание на любой поверхности.

Минусы:

1. Программа способна вычислить цифры (двузначные и трехзначные)
2. 1мин.30сек. – слишком долго!



Результаты по распознаванию нейронной сети:

Плюсы:

1. Скорость составляет: 15 сек
2. Распознавание с электронных файлов.

Минусы:

1. У программы есть погрешность (относительно счетчиков) по распознаванию цифр.
2. Точность 80% (любая ошибка критична в распознавании).



В ПЕРСПЕКТИВЕ ПЛАНИРУЕТСЯ

1. Повысить программную точность распознавания показаний счетчиков.
2. Разработать нейросеть, определяющую символы на счётчике (в нашем случае цифры и точку).
3. Определить нейроны, составные элементы нейронных сетей.
4. Разработать алгоритмическую базу для представленной модели, включающей в себя алгоритм выделения признаков и алгоритм распознавания изображений.
5. Реализовать алгоритмический комплекс в виде программы для ПК, включающая интерфейс для клиентов и сотрудников предприятия.
6. Оценить производительность разработанного способа и критериев достижения поставленной цели.

