

## Лабораторная работа №2

### Основы работы с микрокомпьютерами серии PI

#### Цель работы:

Изучить принципы анализа изображений и выявления различий с использованием библиотеки алгоритмов компьютерного зрения *OpenCV*.

#### Порядок работы:

1. Модифицируем имеющуюся программу так, чтобы она использовала *opencv*, а не *pygame*. Код программы приведен на рисунке 1.

```
Ввод [ ]: import cv2
import telepot
import time

def handle(msg):
    chat_id=msg['chat']['id']
    command=msg['text']
    print('Got command: %s' %command)
    print('From: %s' %chat_id)
    if (chat_id==58127159) and (command=='/photo'):
        cam=cv2.VideoCapture(1)
        ret,frame=cam.read()
        cam.release()
        cv2.imwrite('photo.jpg', frame)
        bot.sendPhoto(chat_id, photo=open('photo.jpg','rb'))
bot=telepot.Bot('6556171571:AAGVGRGH0tG8G9d_cYd_gP00i9Pi_9FNgtg')
bot.message_loop(handle)
print('I am listening...')
while 1:
    time.sleep(10)
```

Рисунок 1 – Модифицированная программа

2. Модифицируем программу таким образом, чтобы она отправляла авторизованному пользователю фотографию в случае смены содержания кадра. Код программы приведен на рисунке 2.

```

from skimage.metrics import structural_similarity
import time
import cv2
import numpy as np
import telepot

Id = '58127159'
Token = '6556171571:AAGVGRGH0tG8G9d_cYd_gP00i9Pi_9FNgtg'
bot = telepot.Bot(Token)

before = cv2.imread('left.jpg')
after = cv2.imread('right.jpg')

before_gray = cv2.cvtColor(before, cv2.COLOR_BGR2GRAY)
after_gray = cv2.cvtColor(after, cv2.COLOR_BGR2GRAY)

(score, diff) = structural_similarity(before_gray, after_gray, full=True)
print("Сходство изображений: {:.4f}%".format(score * 100))

diff = (diff * 255).astype("uint8")
diff_box = cv2.merge([diff, diff, diff])

if score < 0.85:
    print("Зафиксированно изменение")
    bot.sendPhoto(Id, photo=open('right.jpg', 'rb'))
else:
    print("Изменений нет")
time.sleep(1)

```

Рисунок 2 – Модифицированная программа

Используемые в ходе работы изображения приведены на рисунках 3-4.

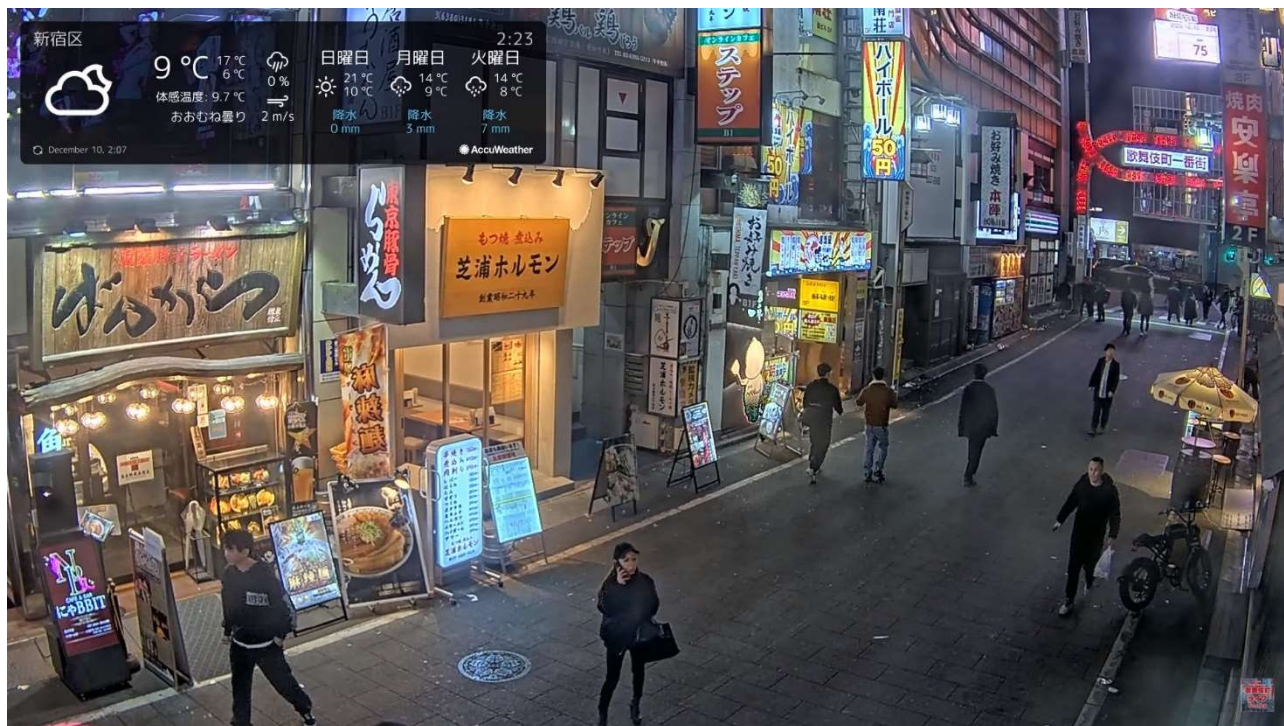


Рисунок 3 – Исходное изображение



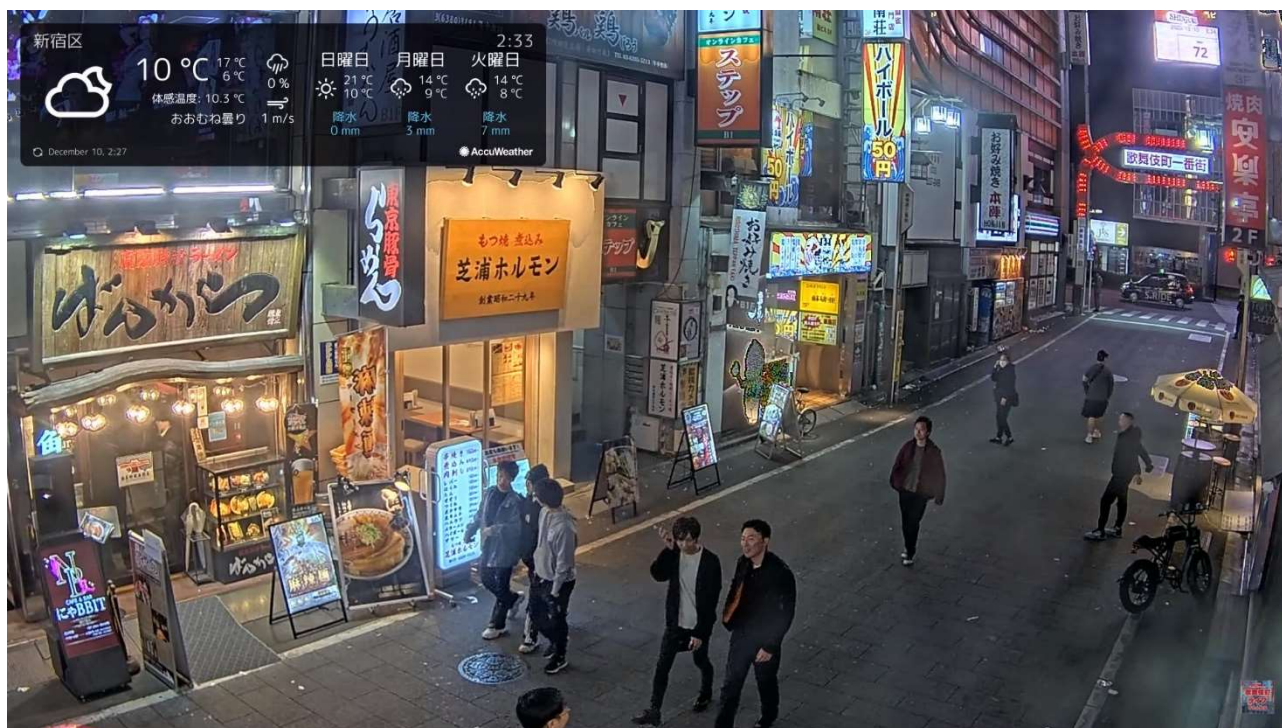


Рисунок 4 – Измененное изображение

Результат работы приведен на рисунках 5-6.

Сходство изображений: 81.5549%

Зафиксированно изменение

Рисунок 5 – Результат работы программы



Рисунок 6 – Изображение, присланное ботом

**Вывод:** получили опыт работы с библиотекой алгоритмов компьютерного зрения *OpenCV*, с библиотекой *Telepot* для работы с *Telegram Bot API*, создали программу для определения различий между похожими изображениями с функцией оповещения посредством отправки снимка с камеры в *Telegram*-бот.