Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторным работам №1-2

Выполнил:
студент группы ИУ5-35Б
Серопегин Я.Р.
Подпись и дата:

Проверил:
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2024 г

Описание:
Разработаны программы для подготовки, форматирования и умножения датасетов картинок, чтобы в дальнейшем можно было обучаться нейросети типа pix2pix. Данная лабораторная работа была оценена как выполнение нескольких лабораторных работ.

Принцип работы:
Имеется два варианта картинок: начальный и конечный. Нейросети предстоит научиться из начальных генерировать конечные изображения. Но размер ограничен, а картинки могут иметь самый разный размер.
Программа состоит из нескольких программ, каждая запускается поочередно. Первая программа нарезает картинку начальную и конечную на много более маленьких (под размер параметров нейросети).

Программа (функция получения файла картинки):
```
from os import listdir
from os.path import isfile, join

def imgs(path='.'):
    imgs = []
    onlyfiles = [f for f in listdir(path) if
isfile(join(path, f))]
    for file in onlyfiles:
        if file.split('.')[-1] == 'png' or file.split('.')[-
1] == 'jpg':
            imgs.append(file)
    return imgs
```

Программа (для нарезки картинки на картинки с необходимым размером):

```
from PIL import Image
import datetime
import random as rd
time = str( datetime.datetime.today().strftime("%H-%M-%S") )
# Opens a image in RGB mode
im = Image.open("input.png").convert('RGB')
ot = Image.open("output.png").convert('RGB')

width, height = im.size
```

```
width = int(width//256+1)*256
height = int(height//256+1)*256
print(width, height)
im0 = Image.new("RGB", (width, height), (0, 0, 0))
im0.paste(im, (0,0))
im = im0
for i in range(0, width-256+1, 4*rd.randint(2, 8)):
    for j in range(0, height-256+1, 4*rd.randint(2, 8)):
        top = j
        bottom = 256+j
        right = 256+i
        left = i
        im1 = im.crop((left, top, right, bottom))
        ot1 = ot.crop((left, top, right, bottom))
        #im1.save('img'+str(i)+'-'+str(j)+'-'+time+'.png',
quality=100)

        bg = Image.new(im.mode, (512, 256), (0, 0, 0)) #
upper size
        bg.paste(im1, (0, 0))
        bg.paste(ot1, (256, 0))

bg.save('sizers/sizer_'+time+'_'+str(i)+'_'+str(j)+'_.png')
```

Программа для умножения полученных картинок на множество подобных картинок, но с другими цветами, поворотом и прочими параметрами:

```
from PIL import Image, ImageFilter, ImageDraw, ImageOps,
ImageChops
import numpy as np
import random as rd
import _parse_
i = _parse_.imgs('my/')
o = _parse_.imgs('blurs/')

def color(img, dtR=0, dtG=0, dtB=0):
    img = img.convert('HSV')
    draw = ImageDraw.Draw(img)
    pix = img.load()
```

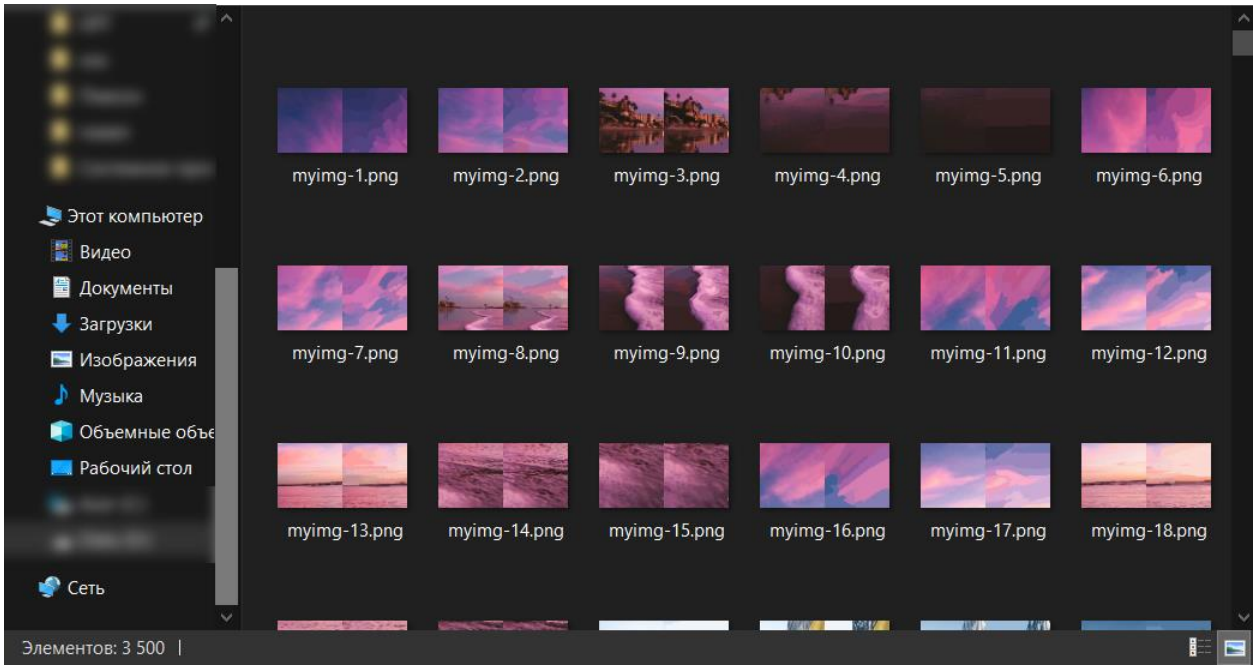
```
    mnh = 0
    mns = 0
    mnv = 0
    mxh = 0
    mxs = 0
    mxv = 0
    for x in range(img.size[0]):
        for y in range(img.size[1]):
            h = pix[x, y][0]
            s = pix[x, y][1]
            v = pix[x, y][2]
            mnh = min(mnh, h)
            mns = min(mns, s)
            mnv = min(mnv, v)
            mxh = max(mxh, h)
            mxs = max(mxs, s)
            mxv = max(mxv, v)
    dtH = rd.randint(0, 255)
    dtS = rd.randint(-50, 100)
    dtV = [rd.randint(-100, -30), rd.randint(30,
150)][rd.randint(0, 1)]
    #print(dtH, dtS, dtV)
    if dtS < 0 and dtV > 70:
        dtV = dtV - 50
    for x in range(img.size[0]):
        for y in range(img.size[1]):
            h = pix[x, y][0]
            s = pix[x, y][1]
            v = pix[x, y][2]
            h2 = int(dtH)
            s2 = int(s + dtS)
            v2 = int(v + dtV)
            draw.point((x, y), (h2, s2, v2))
    img = img.convert('RGB')
    return img


for ind in range(len(i)):
    if ind%10==0:
        print(ind*100//len(i), '%')
```

```
    q = Image.open('my/'+i[ind]).convert('RGB')
    q.save('delete/'+i[ind])

    q = color(q)
    q1 = q.crop((0, 0, 256, 256))
    q2 = q.crop((256, 0, 512, 256))

    q1 = ImageOps.flip((q1))
#(q1.rotate(270))#.filter(ImageFilter.BoxBlur(rd.randint(1,
3)))   # ImageOps.flip()   ImageChops.invert()  .rotate(90)
    q2 = ImageOps.flip((q2)) # (q2.rotate(270))
    bg = Image.new(q.mode, (512, 256), (0, 0, 0)) # upper
size
    bg.paste(q1, (0, 0))
    bg.paste(q2, (256, 0))
    bg.save('delete/myimg_1_'+i[ind])

    q1 = ImageOps.mirror((q1))
#(q1.rotate(270))#.filter(ImageFilter.BoxBlur(rd.randint(1,
3)))   # ImageOps.flip()   ImageChops.invert()  .rotate(90)
    q2 = ImageOps.mirror((q2)) # (q2.rotate(270))
    bg = Image.new(q.mode, (512, 256), (0, 0, 0)) # upper
size
    bg.paste(q1, (0, 0))
    bg.paste(q2, (256, 0))
    bg.save('delete/myimg_2_'+i[ind])

    q1 = (q1.filter(ImageFilter.BoxBlur(1)))
#(q1.rotate(270))#   # ImageOps.flip()   ImageChops.invert()
.rotate(90)
    q2 = (q2) # (q2.rotate(270))
    bg = Image.new(q.mode, (512, 256), (0, 0, 0)) # upper
size
    bg.paste(q1, (0, 0))
    bg.paste(q2, (256, 0))
    bg.save('delete/myimg_3_'+i[ind])

    q1 = (q1.rotate(90)) #(q1.rotate(270))#   #
ImageOps.flip()   ImageChops.invert()  .rotate(90)
    q2 = (q2.rotate(90)) # (q2.rotate(270))
```

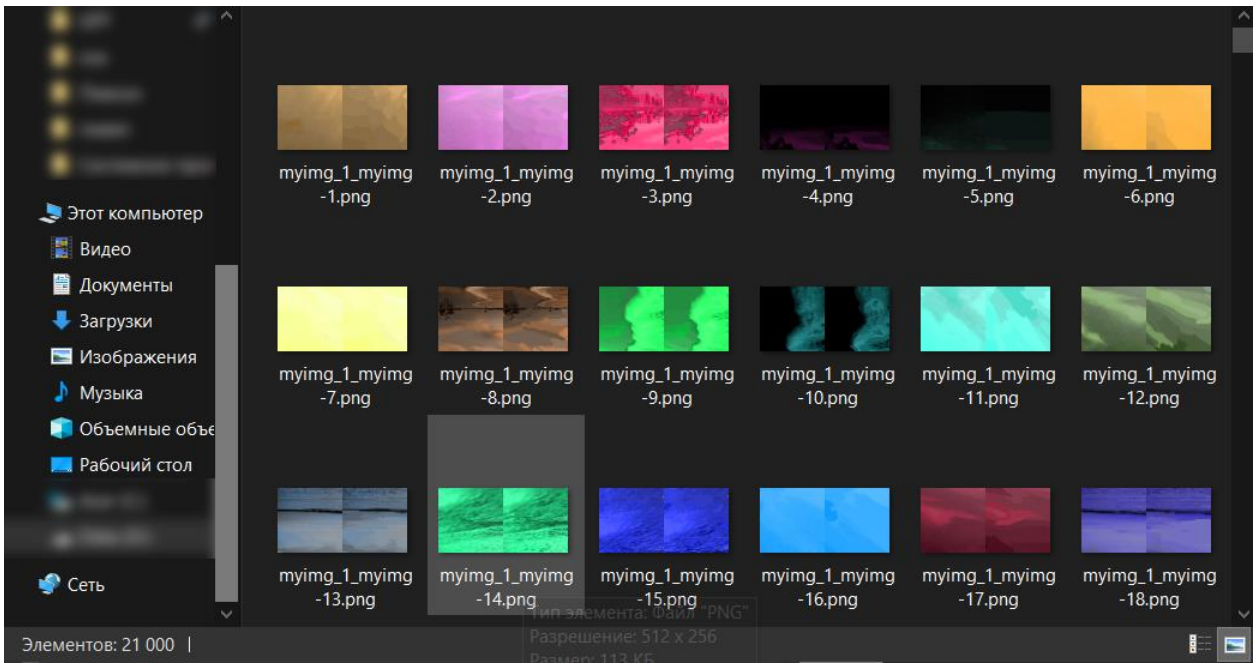
```
    bg = Image.new(q.mode, (512, 256), (0, 0, 0)) # upper
size
    bg.paste(q1, (0, 0))
    bg.paste(q2, (256, 0))
    bg.save('delete/myimg_5_'+i[ind])

    q1 = ImageChops.invert(q1.rotate(270))
#(q1.rotate(270))#   # ImageOps.flip()   ImageChops.invert()
.rotate(90)
    q2 = ImageChops.invert(q2.rotate(270)) #
(q2.rotate(270))
    bg = Image.new(q.mode, (512, 256), (0, 0, 0)) # upper
size
    bg.paste(q1, (0, 0))
    bg.paste(q2, (256, 0))
    bg.save('delete/myimg_6_'+i[ind])
```

Результаты:

Создано 3500 изображений первой программой.



И создано 21000 изображений второй программой:


Этот компьютер
Видео
Документы
Загрузки
Изображения
Музыка
Объемные объе
Рабочий стол
Сеть
myimg_1_myimg-1.png
myimg_1_myimg-2.png
myimg_1_myimg-3.png
myimg_1_myimg-4.png
myimg_1_myimg-5.png
myimg_1_myimg-6.png
myimg_1_myimg-7.png
myimg_1_myimg-8.png
myimg_1_myimg-9.png
myimg_1_myimg-10.png
myimg_1_myimg-11.png
myimg_1_myimg-12.png
myimg_1_myimg-13.png
myimg_1_myimg-14.png
myimg_1_myimg-15.png
myimg_1_myimg-16.png
myimg_1_myimg-17.png
myimg_1_myimg-18.png
Элементов: 21 000