УДК 004

Разработка системы перевода изображения в мозаику на языке программирования Python

Кизянов Антон Олегович
Приамурский государственный университет имени Шолом-Алейхема
студент

Аннотация
В данной статье продемонстрирован процесс создания системы перевода изображения в мозаику из других изображений.
Ключевыеслова: Python, Pillow.

Development of the image transfer system in a mosaic in the Python programming language

Kizyanov Anton Olegovich
Sholom-Aleichem Priamursky State University
student

Abstract
This article demonstrated the process of creating the translation system image into a mosaic of other images.
Keywords: Python, Pillow.

Научный руководитель:
Лучанинов Дмитрий Васильевич
Приамурский государственный университет имени Шолом-Алейхема
старший преподаватель кафедры информационных систем, математики и методик преподавания

Современная фотография не стоит на месте и людипридумывают все разные способы их обработки. Одна из них мозаика, состоящая из множества мелких снимков образуя вместе картину.

Целью данной статьи является описание алгоритма и системы в целом, по созданию мозаик из других картинок.

Для ознакомления с языком программирования Pythonпрочтите следующие статьи. В.А. Машков, В.И. Литвиненко рассказали о применение языка программирования python для решения задач самодиагностики на системном уровне [1]. Г.Д.Бухарова, и П.С.Комельских рассказали о важность и необходимости внедрения языка программирования Python в процесс обучения студентов [2]. Г.С.Сейдаметов продемонстрировал

особенности использования языка программирования python в подготовке будущих инженеров-программистов [3]. Э.А.Усеинов продемонстрировал использование объектно-ориентированного программирования в рамках дисциплины «язык программирования python» [4].

Для работы потребуется библиотека Pillow.

Библиотека Pillow это библиотека языка Python, предназначенная для работы с растровой графикой[5]. Устанавливается следующей командой **pipinstallPillow** в консоли.

Скрипт начинается с импорта нужных библиотека.

```
from PIL import Image
importos
```

Дальше идут функции.

```
def distance(color1, color2):
distance = 0
fori in range(3):
temp = color1[i] - color2[i]
distance = distance + (temp * temp)
   return distance
```

Функция `distance` рассчитывает расстояние между цветами (RGB).

```
Def putBlock(block, destination, location):
destination.paste(block, location)
return destination

def scale(origImage, widthFactor, heightFactor):
w, h = origImage.size
    w = int(w * widthFactor)
    h = int(h * heightFactor)
im = origImage.resize((w, h))
returnim

def scaleTarget(image, newSize):
w, h = image.size
nw, nh = newSize
sw = nw * 1.0 / w
sh = nh * 1.0 / h
im = scale(image, sw, sh)
   returnim
```

Функция`scaleTarget`строит новое изображение на основе старого.

```
def average(image):
r = 0
```

```
    g = 0
    b = 0
w, h = image.size
fori in range(w):
for j in range(h):
color = image.getpixel((i, j))
            r += color[0]
            g += color[1]
            b += color[2]
average_r = r / (w * h)
average_g = g / (w * h)
average_b = b / (w * h)
return (average_r, average_g, average_b)
```

Функция `average` считает среднее значение цвета каждого пикселя.

```
def buildTile(image, tileSize):
w, h = image.size
tw, th = tileSize
sw = tw * 1.0 / w
sh = th * 1.0 / h
im = scale(image, sw, sh)
average_color = average(im)
return (im, average_color)
```

Функция `buildTile` строит маленькие картинки для будущего изображения.

```
def findBestThumbnail(color, tileList):
nesrest = None
for tile in tileList:
dis = distance(color, tile[1])
if nesrest == None or nesrest> dis:
nesrest = dis
best_tile = tile
   return best_tile[0]
```

Функция `findBestThumbnail` ищетсамуюпохожуюплитку.

```
def makeTileList(imageFileList, tileSize):
tiles = []
for filename in imageFileList:
im = Image.open(filename)
tile = buildTile(im, tileSize)
tiles.append(tile)
```

```
   return tiles
```

Функция `makeTileList` собирает плитки для картинки.

```
def stitchMosaic(image, tileList):
w, h = image.size
tw, th = tileList[0][0].size
im = Image.new('RGB', (w * tw, h * th))
fori in range(w):
for j in range(h):
color = image.getpixel((i, j))
thumbnail = findBestThumbnail(color, tileList)
im = putBlock(thumbnail, im, (i * tw, j * th))
   return im
```

Функция `stitchMosaic` собирает картинку.

```
def   buildMosaic(targetImageFile,   imageDirectory,   tileSize,
   resolution):
imgFileList = os.listdir(imageDirectory)
for item in range(len(imgFileList)):
imgFileList[item] = imageDirectory + os.sep + imgFileList[item]

tileList = makeTileList(imgFileList, tileSize)
im = Image.open(targetImageFile)
im = scaleTarget(im, resolution)
im = stitchMosaic(im, tileList)
im.save("mosaic.jpg")
   print("Finish! the output image is mosaic.jpg")
```

Функция `buildMosaic` вызывает другие функции и управляет ими.

```
def testBuildMosaic():
sourceFileName = "Dodge Challenger.jpg"
directory = "./tile"
tileSize = (5, 5)
mosaicSize = (400, 250)
   buildMosaic(sourceFileName, directory, tileSize, mosaicSize)
```

Функция `testBuildMosaic` настройки.

```
if __name__ == "__main__":
   testBuildMosaic()
```

Вызываем функцию настроек.
Для тестирования выбран Рисунок 1.

Рисунок 1

После запуска скрипта в папке появляется Рисунок 2.

Рисунок 2

В функции настроек можно самому настраивать, каких размеров делать будущую картинку, откуда брать картинки в роли мозаики и размеры плитки.

Вывод: Разработана система перевода изображения в мозаику.

Библиографический список

1. МашковВ.А., Литвиненко В.И. Использование языка программирования python 3 и системы компьютерной алгебры sympy на факультативных занятиях по теории чисел // В сборнике: Электротехнические и компьютерные системы Издательство: Одесский национальный политехнический университет (Одесса)С. 48-54
2. Бухарова Г. Д., Комельских П. С. важность и необходимость внедрения языка программирования python в процесс обучения студентов // В сборнике: новые информационные технологии в образовании Материалы VII международной научно-практической конференции. Российский государственный профессионально-педагогический университет. 2014 Издательство: Российский государственный профессионально-педагогический университет (Екатеринбург) С. 40-42.
3. СейдаметовГ. С. Особенности использования языка программирования python в подготовке будущих инженеров-программистов // В сборнике: INTERNATIONAL SCIENTIFIC REVIEW Издательство: Олимп (Иваново) С. 50-51
4. Усеинов Э.А. объектно-ориентированное программирование в рамках дисциплины «язык программирования python» // В сборнике: ученые записки крымского инженерно-педагогического университета Издательство: Государственное бюджетное образовательное учреждение высшего образования Республики Крым "Крымский инженерно-педагогический университет" (Симферополь) С. 157-160.
5. Pillow. [Электронный ресурс]. URL: https://pypi.python.org/pypi/Pillow/3.3.1 (дата обращения: 5.09.2016)