Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем Дисциплина «Объектно-ориентированное программирование»

*К защите допустить:*

Магистр технических наук,

старший преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А. П.Горбач

\_\_\_.\_\_\_. 20\_\_\_

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

на тему

**ПРОГРАММА ЭКВАЛИЗАЦИИ ПОТОКА ВИДЕОДАННЫХ**

БГУИР КР 1-40 05 01-10 069 ПЗ

Бабинский М.А  
Курсовая работа представлена

на проверку \_\_\_.\_\_\_. 20\_\_\_

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(подпись студента)  (подпись студента)

Минск 2022

**РЕФЕРАТ**

БГУИР КР 1-40 05 01 10 069 ПЗ

Бабинский М.А Программа эквализации видеопотока

/М.А. Бабинский – Минск: БГУИР, 2021. –с.

Пояснительная записка 35 с., рис. 6, источников 5, приложения 3

ЭКВАЛИЗАЦИЯ ВИДЕОПОТОКА

Предмет: изменение гистограмм плотности распределения видеопотока.

Объект: эвализирование видеопотока.

Цель: разработка алгоритма эквализации видеопотока, создание приложения на его основе.

Методология проведения работы: в процессе решения поставленных задач использованы принципы системного подхода, знания психологии и мышления в экстренных ситуациях. Для реализации программного кода были использованы основы алгоритмизации на языке Python.

Результаты работы: выполнен анализ источников по теме; реализованы компоненты и алгоритмы для новостей и пособия, результаты представлены в виде таблицы результатов тренировок.

Программа предназначена для понимания способов улучшения своих игровых навыков и получения некоторых новостей.

Область применения результатов: бизнес, киберспорт.

**Содержание**

Введение…………………………………………………………………... 6

1. Описание проекта………………………………………………………. 7

1.1 Описание серверной части……………………………………... 7

1.2 Описание клиентской части……………………………………... 7

2 Обоснование выбора технологий……………………………………... 17

2.1Технологии программирования……………………….……...... 17

2.2Реализация объктно-ориентированных технологий программирования в современных средах разработки.……………...... 17

3 Инструментарий. …................................................................................. 20

3.1 Обоснование используемых инструментов…………………... 19

3.2 Ссылка на репозиторий с проектом и использование технологий GIT………………………………………………………………….......... 20

4 Архитектурный шаблон проектирования MVC……………………... 22

4.1 Разработка схемы алгоритма……............................................ 22

5 Шаблон проектирования практических задач…………………….. 24

5.1 Использование шаблонов проектирования практических решений для решения практических задач …………………………… 24

Заключение …………………………………………………………….. 31

Список использованных источников ……………………………….......... 32

Приложение А(обязательное) Листинг программного кода......................... 33

Приложение Б(обязательное) Графический материал.................................. 38

Приложение В(обязательное) Ведомость документов…………...…… 39

**ВВЕДЕНИЕ**

**Киберспорт** — спортивные соревнования по компьютерным видеоиграм. История киберспорта началась с игры Doom, которая имела режим сетевой игры через локальную вычислительную сеть. Благодаря популярности игры Quake, в 1997 году в США появилась первая лига киберспортсменов — Cyberathlete Professional League (CPL). Соревнования по киберспорту проводятся по всему миру, в том числе и международные: World Cyber Games, The CPL, ESWC, ESL Pro Series, ECG, KODE5. Наиболее значительным из них является турнир World Cyber Games, который организован подобно Олимпийским играм. Впервые WCG были проведены в Южной Корее, родоначальнице киберспорта и неизменным лидером в мировых состязаниях по Старкрафту, и с тех пор соревнования устраиваются каждый год в разных странах.

Цель работы – создание удобного приложения для получения информации о новостях киберспорта и изучению возможностей для улучшения своих навыков.

Задачи:

1 Изучить понятие киберспорта, его основные аспекты, наиболее популярные соревнования и аспекты. А также анализ киберспортивных матчей и форумов, для понимания основ игры.

2 Разработать свой небольшой реферат, который мог бы помочь новичкам понять принципы игры, все её элементы. Этот реферат должен дать основу для улучшения своей игры, игровых навыков.

3 Разработка удобного интерфейса, подключение новостей, создание базы данных с новостями, рефератом.

4 Подключение функций, классов и базы данных к интерфейсу.

5 Тестирование приложения, тестирование работоспособности советов по улучшению своих игровых навыков.

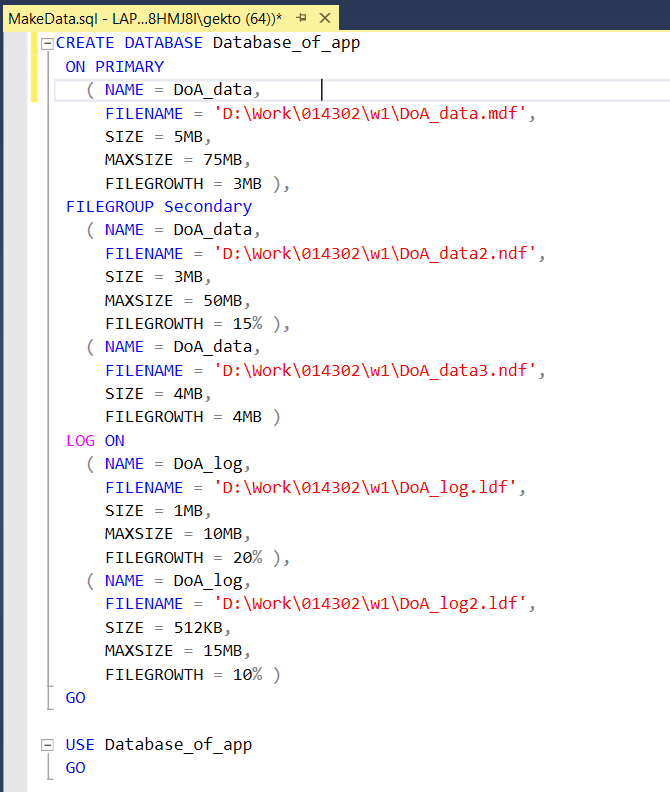
**1 Описание проекта**

* 1. **Описание серверной части**

Серверная часть проекта содержит большое количество информации, которое затем интегрируется в проект.

База данных была создана на основе программы SQL Server Management Studio. Для начала, был создан скрипт и внесены туда нужные данные и часть информации.





Затем после срабатывания скрипта, создаётся нужная нам база данных, которая может управляться с помощью языка программирования Python.

Сама база данных содержит 3 таблицы: Новости по CS:GO, Новости по Dota 2, Обучение по CS:GO.

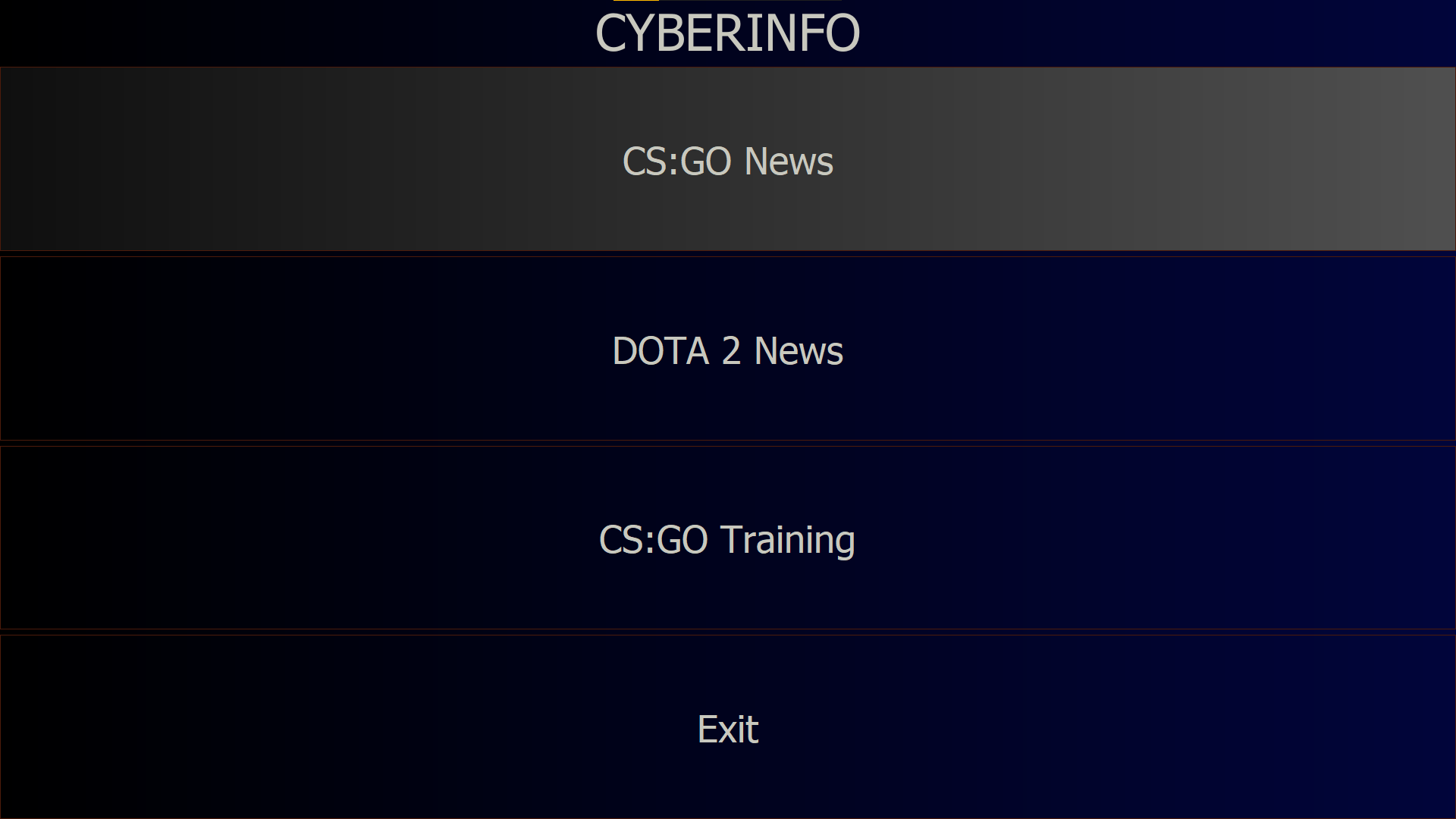
В каждой новостной таблице содержатся ID, Дата новости, Название новости, Название картинки для новости, сама новость.

В таблице Обучение содержится ID, Название темы, Название(имя) картинки, информация темы.

За интегрирование и управление серверной части отвечает библиотека pyodbc, которая позволяет управлять таблицами, правами и интегрировать в питон SQL базу данных.

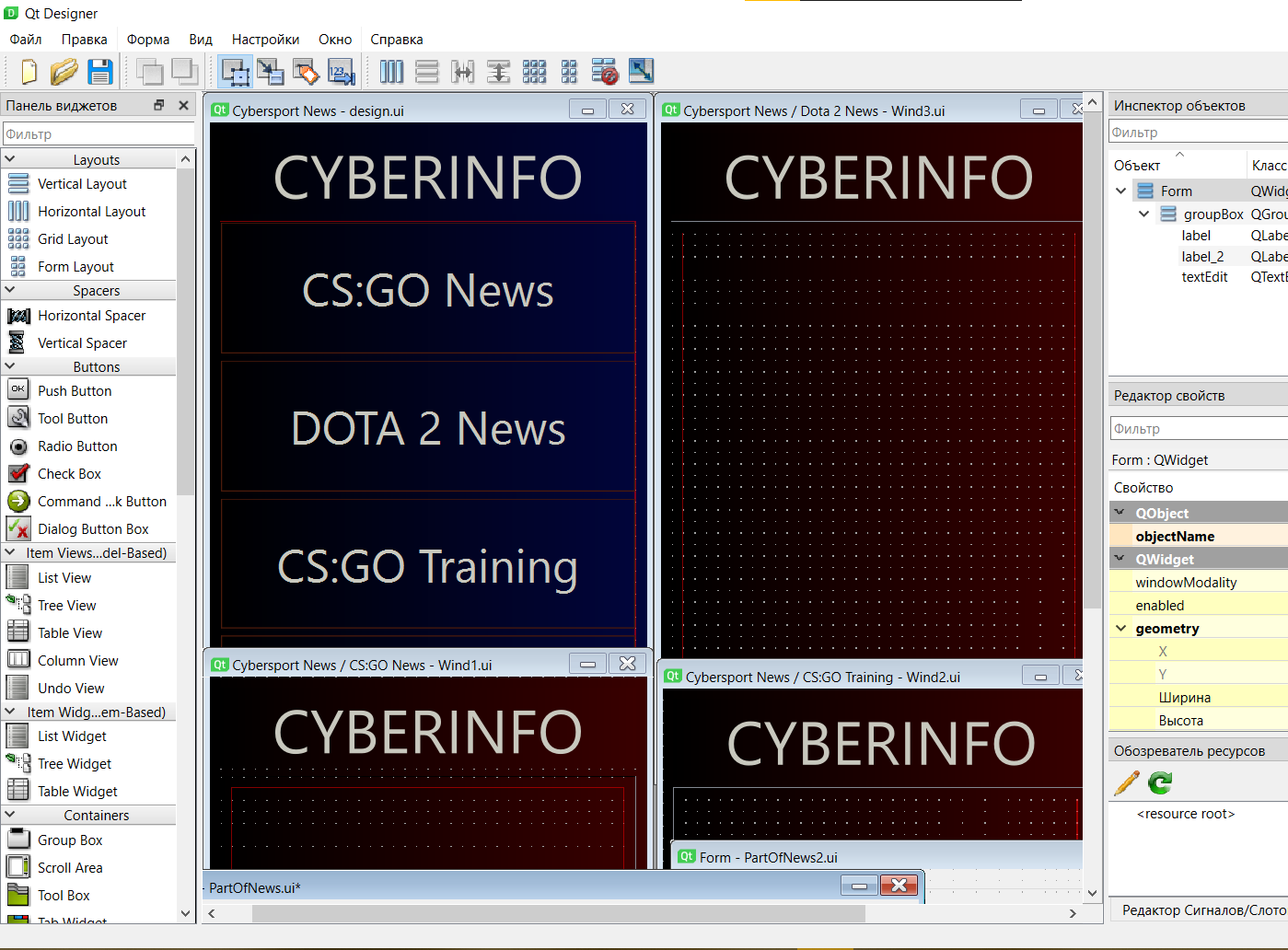
* 1. **Описание клиентской части**

Клиентская часть – эта та часть, которую видит сам клиент(пользователь), который скачал приложение.



В клиентской части находятся основные классы, которые отвечают за соединение функций кнопок с самими кнопками, а также за отображение самого приложения и его внешний вид.

Создание внешнего вида происходило с помощью библиотеки PyQt5, который позволяет открыть редактор PyDesigner, в котором создаётся внешний вид, кнопки, можно подключить небольшой скриптинг.



Управление кнопками происходит с помощью массива, точнее с помощью изменения индекса используемой переменной в массиве.

**2 Обоснование выбора технологий**

**2.1 Технологии программирования**

**2.2 Реализация объектно-ориентированных технологий программирования в современных средах разработки**

**3 Инструментарий**

**3.1 Обоснование используемых инструментов**

**3.2 Ссылка на репозиторий с проектом и использование технологий GIT**

**4 Архитектурный шаблон проектирования MVC**

**4.1 Разработка схемы алгоритма**

**5 Шаблон проектирования практических задач**

**5.1 Использование шаблонов проектирования практических решений для решения практических задач**

**ЗАКЛЮЧЕНИЕ**

В процессе работы на курсовой работой проведено проектирование графического пользовательского интерфейса для фильтрации видеофайла посредством эквализации гистограмм интегральной плотности уровней яркостной матрицы каждого кадра, с возможностью задания полного пути к видеофайлу, выбора функции и ввода параметров эквализации яркостного компонента кадров видеофайла, сохранения результата эквализации в полутоновом виде в отдельном видеофайле. В ходе работы рассмотрены: математическая модель эквализации изображений, форматы видеофайлов, объектно-ориентированные технологии программирования, проектирование графического пользовательского интерфейса средствами COM-объектов, алгоритм функционирования программного обеспечения, программная реализация алгоритма эквализации. В работе приведена перечень графического материала : структура графического пользовательского интерфейса, схема алгоритма, диаграмма последовательности, а также диаграмма состояний. В ходе работы были рассмотрены: понятие изображения, понятие видеопотока, понятия и типы эквализации . Были рассмотрены гистограммы плотности и интегральной плотности уровней. Проведено математическое описание процесса эквализации. Был проведен анализ литературных источников и описание форматов видеофайлов, а также анализ функций современных языков программирования для работы с видеофайлами. Была выполнена классификация цветовых моделей. В ходе работы рассмотрены функции вычисления гистограмм плотности и интегральной плотности уровней яркостного компонента кадра видеофайла как изображения. Приведена подпрограмма эквализации яркостного компонента кадра видеофайла с детальным описанием её участков. Были изучены основы теорий вероятностей : плотность распределения и функция вероятности. Проведен анализ результатов решения поставленной задачи.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Histogram equalization [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/Histogram_equalization>

## [2] Преобразование гистограмм, эквализация [Электронный ресурс]. – Режим доступа: https://scask.ru/a\_book\_kir.php?id=10

# [3] Работа с изображениями и видео [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/otus/blog/558426/

# [4] Обработка видео с использованием Python и OpenCV [Электронный ресурс]. – Режим доступа: https://overcoder.net/q/1520693/%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0-%D0%B2%D0%B8%D0%B4%D0%B5%D0%BE-%D1%81-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D0%BC-python-%D0%B8-opencv

# [5] ПРЕОБРАЗОВАНИЕ ГИСТОГРАММ ЭКВАЛИЗАЦИЯ, ГИПЕРБОЛИЗАЦИЯ [Электронный ресурс]. – Режим доступа: https://studref.com/524525/informatika/preobrazovanie\_gistogramm\_ekvalizatsiya\_giperbolizatsiya

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Листинг кода**

from tkinter import \*  
import cv2  
import os  
import glob  
import matplotlib.pyplot as plt  
  
images = []  
fname = ""  
Defname = "D:\\Data\\"  
  
  
def clicked1():  
 global fname  
 res = txt.get()  
 lbl1.configure(text="Выбран файл: " + res)  
 fname = chose\_way(res)  
  
  
def clicked2():  
 lbl2.configure(text="Проводится разбиение видео на кадры...")  
 video\_fragmentation()  
 lbl2.configure(text="Разбиение видео на кадры... Успешно")  
  
  
def clicked3():  
 lbl2.configure(text="Проводится выравнивание...")  
 making\_video(images)  
 lbl2.configure(text="Выравнивание... Успешно")  
  
  
def clicked4():  
 lbl2.configure(text="Делаем анимацию...")  
 img\_changes()  
 lbl2.configure(text="Анимацию... Успешно")  
  
  
def clicked5():  
 n\_txt = ""  
 for filename in glob.glob('D:/Data/\*.mp4'):  
 n\_txt += (filename + "\n")  
 lbl3.configure(text=n\_txt)  
  
  
window = Tk()  
  
window.title("Добро пожаловать в приложение PythonRu")  
window.geometry('600x400')  
  
lbl = Label(window, text="Выравнивание яркости видео:", font=("Arial Bold", 20))  
lbl.grid(column=0, row=0)  
  
  
btn = Button(window, text="Ввести путь файла", bg="gainsboro", command=clicked1, width=85, height=2)  
btn.grid(column=0, row=1)  
  
txt = Entry(window, width=40, bg="gainsboro", font=("Arial Bold", 20))  
txt.grid(column=0, row=2)  
  
lbl1 = Label(window, text="Выбран файл:", font=("Arial Bold", 20))  
lbl1.grid(column=0, row=3)  
  
btn = Button(window, text="Сделать фрагментирование видео", bg="gainsboro", command=clicked2, width=85, height=2)  
btn.grid(column=0, row=10)  
  
btn = Button(window, text="Собрать видео с выравненной яркостью", bg="gainsboro", command=clicked3, width=85, height=2)  
btn.grid(column=0, row=12)  
  
btn = Button(window, text="Сделать анимацию изменеия гистограмм изображений", bg="gainsboro", command=clicked4, width=85, height=2)  
btn.grid(column=0, row=13)  
  
btn = Button(window, text="Поиск видео в папке", bg="gainsboro", command=clicked5, width=85, height=2)  
btn.grid(column=0, row=14)  
  
lbl3 = Label(window, text="", font=("Arial Bold", 10))  
lbl3.grid(column=0, row=15)  
  
  
lbl2 = Label(window, text="", font=("Arial Bold", 10))  
lbl2.grid(column=0, row=16)  
  
  
  
  
def chose\_way(pname):  
 print("checking way")  
 return Defname + pname  
  
  
def img\_changes():  
 print("in prossec")  
 frameSize = (600, 650)  
 out = cv2.VideoWriter('output\_changes.avi', cv2.VideoWriter\_fourcc(\*'DIVX'), 25, frameSize)  
 for i in range(0, len(glob.glob('D:/Data/Gists/\*.jpg'))):  
 for filename in glob.glob('D:/Data/Gists/\*.jpg'):  
 if filename == "D:/Data/Gists\\s\_fig" + str(i) + ".jpg":  
 print('Creating...' + str(i))  
 img = cv2.imread(filename)  
 img = cv2.resize(img, (600, 650))  
 out.write(img)  
 os.remove(filename)  
  
 out.release()  
  
  
def histogram\_equalization(img, i):  
 img\_yuv = cv2.cvtColor(img, cv2.COLOR\_BGR2YUV)  
 # equalize the histogram of the Y channel  
 img\_yuv[:, :, 0] = cv2.equalizeHist(img\_yuv[:, :, 0])  
 # convert the YUV image back to RGB format  
 img\_output = cv2.cvtColor(img\_yuv, cv2.COLOR\_YUV2BGR)  
 plt.figure(figsize=(6, 4))  
 plt.hist(img.ravel(), 256, [0, 256])  
 plt.hist(img\_output.ravel(), 256, [0, 256])  
 plt.savefig('D:\\Data\\Gists\\s\_fig' + str(i) + '.jpg')  
 return img\_output  
  
  
def video\_fragmentation():  
 global images, fname  
 video\_capture = cv2.VideoCapture(fname)  
 currentFrame = 0  
 flag = True  
 while flag:  
 # Capture frame-by-frame  
  
 ret, frame = video\_capture.read()  
  
 if ret:  
 name = 'D:\\Data\\vid1\\' + str(currentFrame) + '.jpg'  
 print('Creating...' + name)  
 cv2.imwrite(name, frame)  
 currentFrame += 1  
 images.append(frame)  
 else:  
 flag = False  
 video\_capture.release()  
 cv2.destroyAllWindows()  
  
  
def making\_video(images):  
 frameSize = (640, 360)  
 out = cv2.VideoWriter('output\_video.avi', cv2.VideoWriter\_fourcc(\*'DIVX'), 25, frameSize)  
 for i in range(0, len(glob.glob('D:/Data/vid1/\*.jpg'))):  
 for filename in glob.glob('D:/Data/vid1/\*.jpg'):  
 if filename == "D:/Data/vid1\\" + str(i) + ".jpg":  
 print('Creating...' + str(i))  
 img = cv2.imread(filename)  
 img2 = histogram\_equalization(img, i)  
 os.remove(filename)  
 out.write(img2)  
 out.release()

def equalizeHist(img\_yuv[:, :, 0]):

totalPixels = img\_yuv[:, :, 0]

for y in 0..img.Rows:

for x in 0..img.Cols:

s = int(<byte>(y, x))

fx = (float(cdx.[s]) - float(cdxMin))/(float(totalPixels - 1))\*255

equalizeImage.Circle(x, y, 1, Scalar(double(fx)))

returnequalizeImage  
  
window.mainloop()

**ПРИЛОЖЕНИЕ Б**

**(обязательное)**

**Графический материал**

**ПРИЛОЖЕНИЕ В**

**(обязательное)**

**Ведомость документов**