Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «Северо-Восточный федеральный университет

имени М.К Аммосова»

Институт математики и информатики

Кафедра «Информационные технологии»

Документация по созданию интерфейса.

на тему: Распознавание эмоций по фотографии

Выполнили: студенты 1 курса СВФУ ИМИ

гр. М-ИВТ-21 Заморщикова Д.A.

Проверил: доцент научно-исследовательской кафедры ВТ

Григорьев Александр Виссарионович

Якутск, 2022

**Цель работы**

Разработать модель интерфейса приложения для распознавания эмоций по фотографиям.

**Интерфейс**

Графи́ческий интерфейс пользователя (англ. graphical user interface, GUI) — система средств для взаимодействия пользователя с электронными устройствами, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана (окон, значков, меню, кнопок, списков и т. п.).

Чаще всего элементы интерфейса в GUI реализованы на основе метафор и отображают их назначение и свойства, что облегчает понимание и использование электронных устройств неподготовленными пользователями.

Графический интерфейс пользователя является частью пользовательского интерфейса и определяет взаимодействие с пользователем на уровне визуализированной информации.

**Язык программирования и библиотеки**

В качестве языка для программирования графического интерфейса пользователя был выбран язык Python. Это обусловлено тем, что сама модель нейронной сети для распознавания эмоций разрабатывается на языке программирования Python, таким образом, в дальнейшей интеграции не возникнет конфликтов.

Выделяют пять лучших библиотек графического интерфейса Python, это:

* PyQt5
* Python Tkinter
* PySide 2
* Киви
* wxPython

В результате сравнения библиотек, была выбрана библиотека PyQt5.

PyQt5-это графический пользовательский интерфейс (GUI) для Python. Он очень популярен среди разработчиков, и графический интерфейс может быть создан с помощью кодирования или QT-дизайнера. Фреймворк разработки QT-это визуальный фреймворк, который позволяет перетаскивать виджеты для создания пользовательских интерфейсов.

Это бесплатное программное обеспечение с открытым исходным кодом для привязки и реализовано для кросс-платформенной платформы разработки приложений. Он используется в Windows, Mac, Android, Linux и Raspberry PI.

PyQt-это зрелый набор привязок Python к Qt для кросс-платформенной разработки настольных приложений. Он предлагает богатый выбор встроенных виджетов и инструментов для создания пользовательских виджетов для формирования сложных графических интерфейсов, а также надежную поддержку баз данных SQL для подключения к базам данных и взаимодействия с ними.

**Предварительная визуализация на Figma**

Чтобы предварительно согласовать функционал приложения с членами команды и с целью моделирования всех необходимых окон в приложении, были отрисованы все экраны с учетом полной модели поведения пользователя и функциональности нашей нейронной сети. Результаты представлены на рис.1.

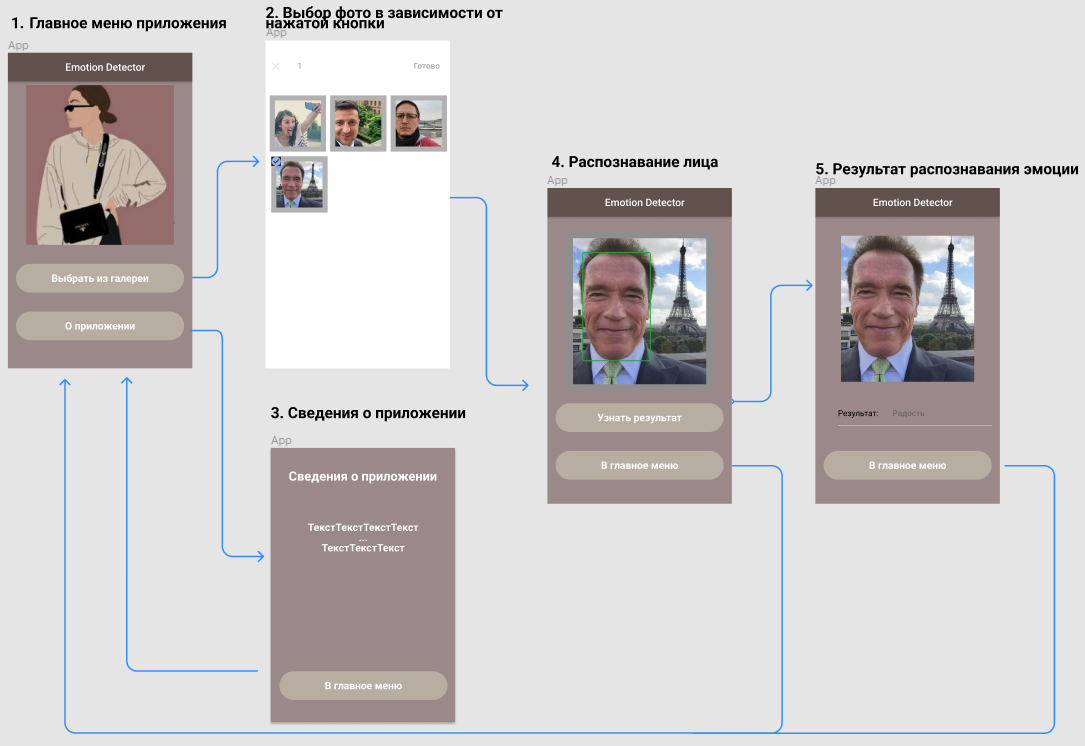


Рис. 1. Интерфейс приложения и основные user story.

1. Главное меню приложения.

В главном меню приложения будет приветственная картинка, а также две кнопки:

* Выбрать из галереи: при нажатии которой должен будет открыться проводник, и дать выбрать изображение для анализа.
* О приложении: при нажатии будет открываться окно с информацией о разработчиках и самом приложении

1. Выбор фото в зависимости от нажатой кнопки.

Переход на данное окно происходит при нажатии на кнопку “выбрать из галереи” в главном меню приложения. Данное окно представляет из себя проводник ПК пользователя, в котором он может выбрать изображение для распознавания эмоции.

После выбора изображения произойдет переход на окно приложения “распознавание лица” в котором будет отражено выбранное изображение.

1. Сведения о приложении.

Переход на данное окно происходит при нажатии на кнопку “ О приложении” в главном меню приложения. В ней отображаются сведения о приложении и его разработчиках.

Для выхода из этого окна существует кнопка “В главное меню”, при нажатии которой происходит переход в главное меню приложения.

1. Распознавание лица.

Переход на данное окно происходит если в окне номер 2, было выбрано изображение из проводника ПК. В этом окне отображается выбранное пользователем изображение и кнопки:

* Узнать результат: при нажатии данной кнопки запускается нейронная сеть, которая подтягивает к себе выбранное в предыдущем пункте изображение, обрабатывает полученную информацию и выдает результат.
* В главное меню: при нажатии происходит переход в главное меню приложения.

1. Результат распознавания эмоции.

Переход в данное окно происходит при удачном выборе и анализе изображения, в этом окне отображается результат работы нейронной сети и кнопка “В главное меню”, при нажатии которой происходит переход в главное меню приложения, и можно повторить процесс повторно.

Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «Северо-Восточный федеральный университет

имени М.К Аммосова»

Институт математики и информатики

Кафедра «Информационные технологии»

Документация по созданию интерфейса.

на тему: Распознавание эмоций по фотографии

Выполнили: студенты 1 курса СВФУ ИМИ

гр. М-ИВТ-21 Заморщикова Д.A.

Проверил: доцент научно-исследовательской кафедры ВТ

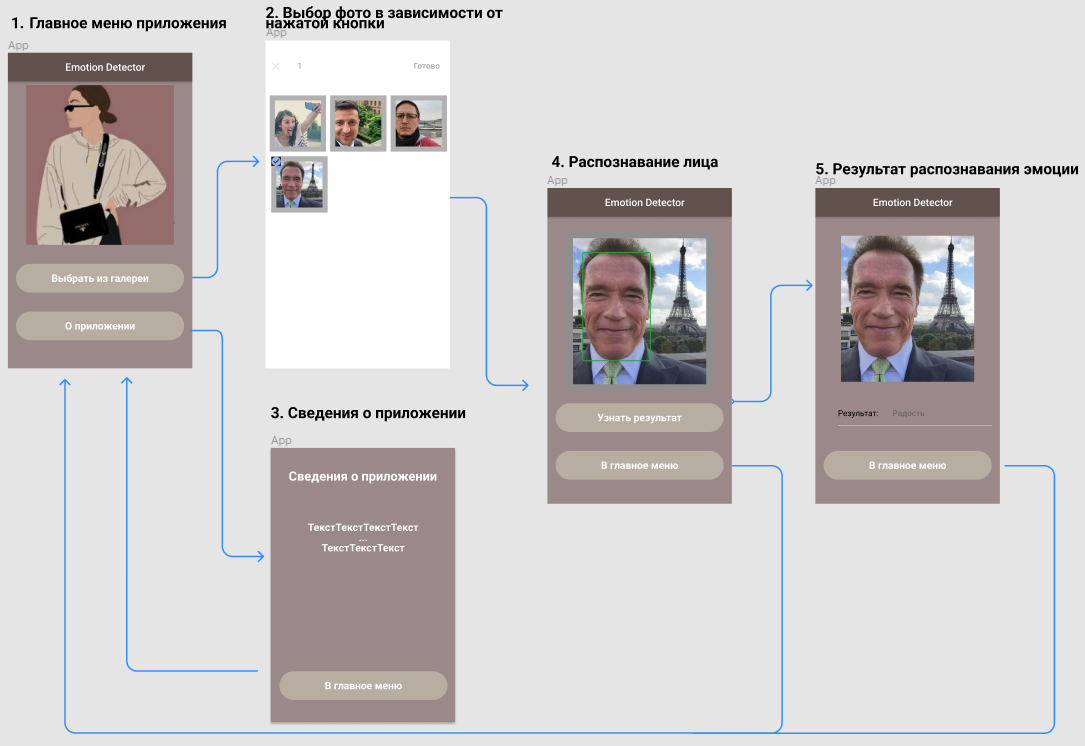
Григорьев Александр Виссарионович

Якутск, 2022

**Цель работы**

Разработать интерфейс приложения для распознавания эмоций по фотографиям.

В предыдущей работе был разработан предварительный интерфейс приложения с использованием Figma:



По результатам первой работы была начата работа по реализации приложения. Использование Figma оказалось очень удобным и полезным. Оттуда были взяты все коды цветов и расцветок различных окон. Далее будет описана работа готового приложения со скриншотами работы программы.

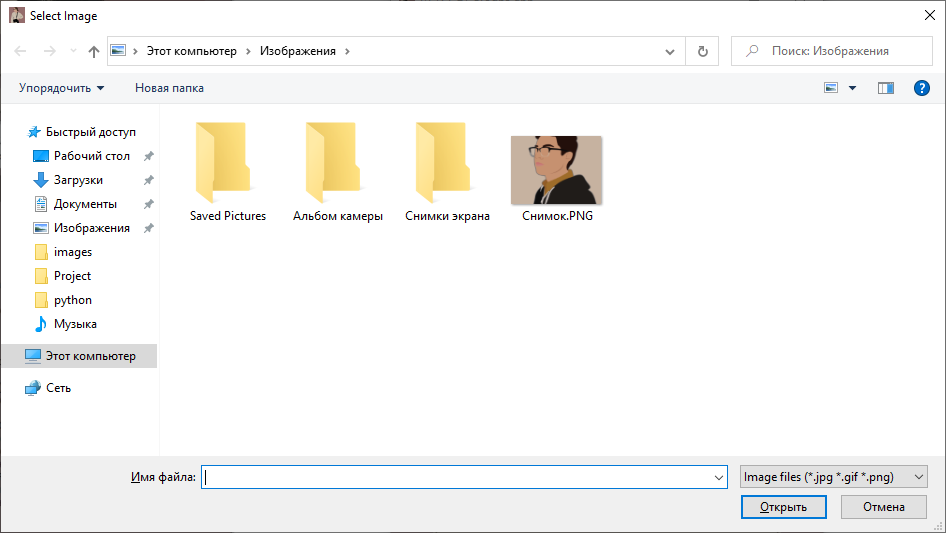
1. Главное меню приложения.



В главном меню приложения будет приветственная картинка, а также две кнопки:

* Выбрать из галереи: при нажатии которой должен будет открыться проводник, и дать выбрать изображение для анализа.
* О приложении: при нажатии будет открываться окно с информацией о разработчиках и самом приложении

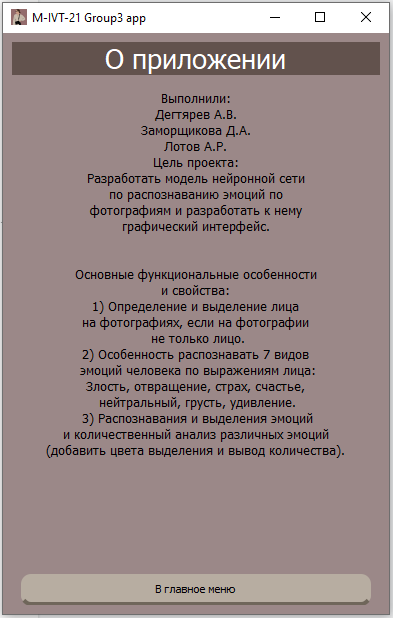
1. Выбор фото в зависимости от нажатой кнопки.



Переход на данное окно происходит при нажатии на кнопку “выбрать из галереи” в главном меню приложения. Данное окно представляет из себя проводник ПК пользователя, в котором он может выбрать изображение для распознавания эмоции.

После выбора изображения произойдет переход на окно приложения “распознавание лица” в котором будет отражено выбранное изображение.

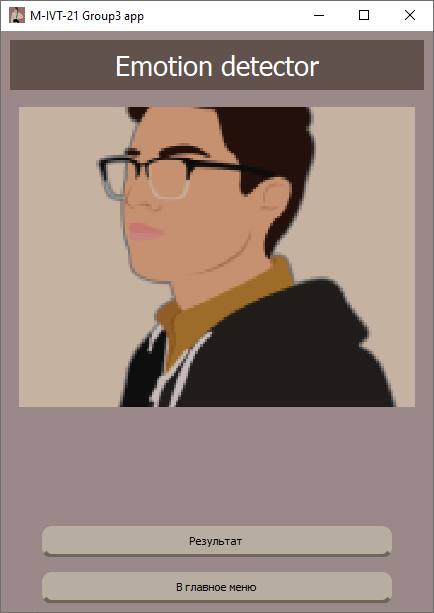
1. Сведения о приложении.



Переход на данное окно происходит при нажатии на кнопку “ О приложении” в главном меню приложения. В ней отображаются сведения о приложении и его разработчиках.

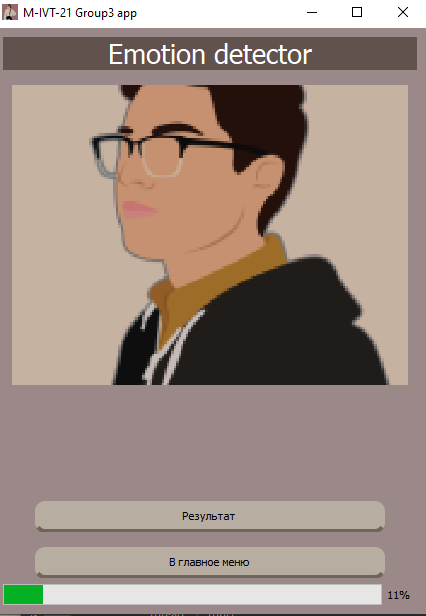
Для выхода из этого окна существует кнопка “В главное меню”, при нажатии которой происходит переход в главное меню приложения.

1. Распознавание лица.



Переход на данное окно происходит если в окне номер 2, было выбрано изображение из проводника ПК. В этом окне отображается выбранное пользователем изображение и кнопки:

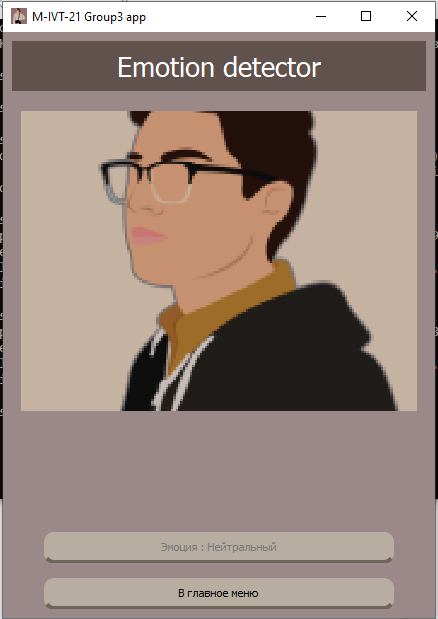
* Узнать результат: при нажатии данной кнопки запускается нейронная сеть, которая подтягивает к себе выбранное в предыдущем пункте изображение, обрабатывает полученную информацию в процессе которого отображается Progress Bar, который исчезает при появлении результата.



* В главное меню: при нажатии происходит переход в главное меню приложения.

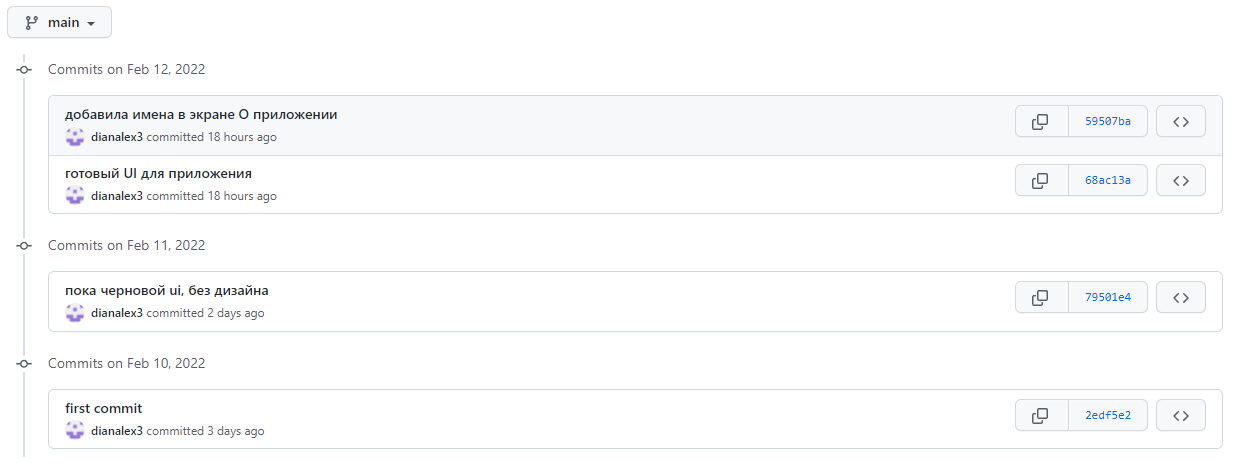
1. Результат распознавания эмоции.

Переход в данное окно происходит при удачном выборе и анализе изображения, в этом окне отображается результат работы нейронной сети и кнопка “В главное меню”, при нажатии которой происходит переход в главное меню приложения, и можно повторить процесс повторно.



Созданный UI был подгружен в общий репозиторий на GitHub:





ПРИЛОЖЕНИЕ

Листинг программы

import sys

from PyQt5.QtWidgets import \*

import time

from PyQt5 import QtGui

from PyQt5 import QtCore

from PyQt5.QtCore import QThread, pyqtSignal

class App(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.title = 'M-IVT-21 Group3 app'

self.left = 200

self.top = 200

self.width = 350

self.height = 500

self.queryimg=None

self.dir\_path=None #"C:/Users/vovaz/Documents/example/Image-Classifier/images/001\_0007.jpg"

self.initUI()

self.setWindowIcon(QtGui.QIcon('icon.png'))

self.threadpool = QtCore.QThreadPool()

def initUI(self):

self.central=QWidget()

self.setCentralWidget(self.central)

self.setWindowTitle(self.title)

self.setGeometry(self.left, self.top, self.width, self.height)

self.mainbox=QVBoxLayout()

self.central.setLayout(self.mainbox)

self.createUI()

self.show()

def createUI(self):

for i in reversed(range(self.mainbox.count())):

self.mainbox.itemAt(i).widget().setParent(None)

self.labelmain = QLabel()

self.labelmain.setAlignment(QtCore.Qt.AlignCenter)

self.labelmain.setMaximumHeight(50)

self.labelmain.setText("Emotion detector")

self.labelmain.setStyleSheet("background-color : #62524D;"

"color : white")

self.central.setStyleSheet("background-color : #9B8888;")

font = QtGui.QFont('SansSerif', 20)

self.labelmain.setFont(font)

self.mainbox.addWidget(self.labelmain)

self.chooseBtn = QPushButton()

self.chooseBtn.setText('Выбрать изображение')

self.chooseBtn.setFixedSize(350,32)

self.chooseBtn.setStyleSheet("background-color : #B7ADA1;"

"border-radius : 10px;"

"border-bottom: 4px solid #6F665A")

self.aboutBtn = QPushButton(self)

self.aboutBtn.setText('О приложении')

self.aboutBtn.setFixedSize(350, 32)

self.aboutBtn.setStyleSheet("background-color : #B7ADA1;"

"border-radius : 10px;"

"border-bottom: 4px solid #6F665A")

self.groupbox = QGroupBox()

self.box1 = QHBoxLayout()

self.groupbox.setFixedHeight(40)

self.groupbox.setLayout(self.box1)

self.box1.addWidget(self.aboutBtn)

self.groupbox.setStyleSheet("border : 0px")

self.groupbox1 = QGroupBox()

self.box2 = QHBoxLayout()

self.box2.addWidget(self.chooseBtn)

self.groupbox1.setLayout(self.box2)

self.groupbox1.setFixedHeight(40)

self.groupbox1.setStyleSheet("border : 0px")

self.groupbox2 = QGroupBox()

self.queryLayout = QHBoxLayout()

self.groupbox2.setLayout(self.queryLayout)

self.groupbox2.setStyleSheet("background-image : url(icon.png);"

"background-repeat : no-repeat;"

"background-position : center;"

"border : 0px")

self.mainbox.addWidget(self.groupbox2)

self.mainbox.addWidget(self.groupbox1)

self.mainbox.addWidget(self.groupbox)

self.chooseBtn.clicked.connect(self.openFileNameDialog)

self.aboutBtn.clicked.connect(self.start)

def start(self):

self.groupbox2.setStyleSheet("")

self.labelmain.setText("О приложении")

self.labelabout = QLabel()

self.labelabout.setAlignment(QtCore.Qt.AlignHCenter)

self.labelabout.setText("Выполнили:\n"

"Дегтярев А.В.\n"

"Заморщикова Д.А.\n"

"Лотов А.Р.\n"

"Цель проекта:\n"

"Разработать модель нейронной сети \n"

"по распознаванию эмоций по \n"

"фотографиям и разработать к нему\n"

"графический интерфейс.\n"

"\n"

"\n"

"Основные функциональные особенности\n"

"и свойства:\n"

"1) Определение и выделение лица\n"

"на фотографиях, если на фотографии\n"

" не только лицо.\n"

"2) Особенность распознавать 7 видов\n"

" эмоций человека по выражениям лица:\n"

"Злость, отвращение, страх, счастье,\n"

" нейтральный, грусть, удивление.\n "

"3) Распознавания и выделения эмоций\n"

"и количественный анализ различных эмоций\n"

"(добавить цвета выделения и вывод количества).")

font = QtGui.QFont('SansSerif', 10)

self.labelabout.setFont(font)

self.queryLayout.addWidget(self.labelabout)

self.groupbox2.setStyleSheet("border : 0px")

self.chooseBtn.setVisible(False)

self.aboutBtn.setVisible(False)

self.menuBtn = QPushButton()

self.menuBtn.setText('В главное меню')

self.menuBtn.setFixedSize(350, 32)

self.menuBtn.setStyleSheet("background-color : #B7ADA1;"

"border-radius : 10px;"

"border-bottom: 4px solid #6F665A")

self.groupbox3 = QGroupBox()

self.box3 = QHBoxLayout()

self.groupbox3.setFixedHeight(40)

self.groupbox3.setLayout(self.box3)

self.box3.addWidget(self.menuBtn)

self.groupbox3.setStyleSheet("border : 0px")

self.menuBtn.clicked.connect(self.createUI)

self.mainbox.addWidget(self.groupbox3)

def openFileNameDialog(self): ## USE THIS TO GET PATH OF QUERY IMAGE...

options = QFileDialog.Options()

options |= QFileDialog.DontUseNativeDialog

fileName, \_ = QFileDialog.getOpenFileName(self,"Select Image", "","Image files (\*.jpg \*.gif \*.png)")

self.groupbox2.setStyleSheet("")

if fileName:

self.queryimg = fileName

self.im1 = QtGui.QPixmap(fileName).scaled(400, 300, QtCore.Qt.KeepAspectRatio)

self.label1 = QLabel()

self.label1.setAlignment(QtCore.Qt.AlignCenter)

self.label1.setPixmap(self.im1)

if self.queryLayout.count()==0:

self.queryLayout.addWidget(self.label1)

else:

self.queryLayout.itemAt(0).widget().setParent(None)

self.queryLayout.addWidget(self.label1)

self.groupbox2.setStyleSheet("border : 0px")

self.chooseBtn.setVisible(False)

self.aboutBtn.setVisible(False)

self.resultBtn = QPushButton()

self.resultBtn.setText('Результат')

self.resultBtn.setFixedSize(350, 32)

self.resultBtn.setStyleSheet("background-color : #B7ADA1;"

"border-radius : 10px;"

"border-bottom: 4px solid #6F665A")

self.groupbox4 = QGroupBox()

self.box4 = QHBoxLayout()

self.groupbox4.setFixedHeight(40)

self.groupbox4.setLayout(self.box4)

self.box4.addWidget(self.resultBtn)

self.groupbox4.setStyleSheet("border : 0px")

self.resultBtn.clicked.connect(self.result)

self.mainbox.addWidget(self.groupbox4)

self.menuBtn = QPushButton()

self.menuBtn.setText('В главное меню')

self.menuBtn.setFixedSize(350, 32)

self.menuBtn.setStyleSheet("background-color : #B7ADA1;"

"border-radius : 10px;"

"border-bottom: 4px solid #6F665A")

self.groupbox5 = QGroupBox()

self.box5 = QHBoxLayout()

self.groupbox5.setFixedHeight(40)

self.groupbox5.setLayout(self.box5)

self.box5.addWidget(self.menuBtn)

self.groupbox5.setStyleSheet("border : 0px")

self.mainbox.addWidget(self.groupbox5)

self.menuBtn.clicked.connect(self.createUI)

def result(self):

self.pbar = QProgressBar(self)

self.pbar.setValue(0)

self.mainbox.addWidget(self.pbar)

self.thread = Thread()

self.thread.\_signal.connect(self.signal\_accept)

self.thread.start()

#код вставить тут

#путь к изображению хранится в path\_to\_image = self.queryimg

#результат сохранить в переменной result в def signal\_accept

path\_to\_image = self.queryimg

def signal\_accept(self, msg):

self.pbar.setValue(int(msg))

if self.pbar.value() == 99:

self.pbar.setValue(100)

result = "Эмоция : Нейтральный" # В виде строки

self.resultBtn.setText(result)

self.resultBtn.setDisabled(True)

self.pbar.setVisible(False)

return

class Thread(QThread):

\_signal = pyqtSignal(int)

def \_\_init\_\_(self):

super(Thread, self).\_\_init\_\_()

def \_\_del\_\_(self):

self.wait()

def run(self):

for i in range(100):

time.sleep(0.11)

self.\_signal.emit(i)

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

ex = App()

sys.exit(app.exec\_())