

Документація до програми «Background remover»

Спочатку я знайшов модель для виконання основної функції програми. Дослідивши складніші моделі розпізнавання об'єктів з застосуванням методів глибокого навчання, я обрав для подальшої роботи архітектуру U2-Net, що базується на моделі Silent Object Detection. Для свого додатку я використав модель U2-Net, що була розроблена студентами Альбертського університету в Канаді. Дана модель розміщена у відкритому доступі на GitHub. U2-Net порівняно нова архітектура, яку можна класифікувати як модель Silent Object Detection (SOD).

Розробники прагнули вирішити 2 питання: по-перше, чи можна навчити SOD модель з нуля, без попереднього тренування на існуючих класифікаторах зображень (таких як AlexNet, ImageNet та ін.), і отримати аналогічні або кращі результати, ніж у моделей, які попередньо

натреновуються на згаданих класифікаторах; і, по друге, чи можна зробити мережу глибше, ніж у попередників, при цьому залишивши таку ж високу якість карт ознак та низькі потреби в пам'яті та обчислювальній здатності машин, на якій буде використовуватися модель.

У результаті дослідження автори змогли досягти цих результатів, у чому можемо впевнитися, поглянувши на рисунок 1, на якому зображена порівняльна характеристика 76 різних метрик SOD – моделей, на трьох різних наборах даних. Червоним позначені найкращі результати, зеленим і синім – другі і треті найкращі результати відповідно.

Method	Backbone	Size(MB)	DUT-OMRON (5168)						DUTS-TE (5019)						HKU-IS (4447)					
			$maxF_{\beta}$	MAE	F_{β}^w	S_m	$relaxF_{\beta}^s$	$maxF_{\beta}$	MAE	F_{β}^w	S_m	$relaxF_{\beta}^s$	$maxF_{\beta}$	MAE	F_{β}^w	S_m	$relaxF_{\beta}^s$	$maxF_{\beta}$	MAE	F_{β}^w
MDF _{TIP16}	AlexNet	112.1	0.694	0.142	0.565	0.721	0.406	0.729	0.099	0.543	0.723	0.447	0.860	0.129	0.564	0.810	0.594			
UCF _{ICCV17}	VGG-16	117.9	0.730	0.120	0.573	0.760	0.480	0.773	0.112	0.596	0.777	0.518	0.888	0.062	0.779	0.875	0.679			
Amulet _{ICCV17}	VGG-16	132.6	0.743	0.098	0.626	0.781	0.528	0.778	0.084	0.658	0.796	0.568	0.897	0.051	0.817	0.886	0.716			
NLDF _{CVPR17}	VGG-16	428.0	0.753	0.080	0.634	0.770	0.514	0.813	0.065	0.710	0.805	0.591	0.902	0.048	0.838	0.879	0.694			
DSS _{CVPR17}	VGG-16	237.0	0.781	0.063	0.697	0.790	0.559	0.825	0.056	0.755	0.812	0.606	0.916	0.040	0.867	0.878	0.706			
RAS _{ICCV18}	VGG-16	81.0	0.786	0.062	0.695	0.814	0.615	0.831	0.059	0.740	0.828	0.656	0.913	0.045	0.843	0.887	0.748			
PAGRN _{CVPR18}	VGG-19	-	0.771	0.071	0.622	0.775	0.582	0.854	0.055	0.724	0.825	0.692	0.918	0.048	0.820	0.887	0.762			
BMPM _{CVPR18}	VGG-16	-	0.774	0.064	0.681	0.809	0.612	0.852	0.048	0.761	0.851	0.699	0.921	0.039	0.859	0.907	0.773			
PICANet _{CVPR18}	VGG-16	153.3	0.794	0.068	0.691	0.826	0.643	0.851	0.054	0.747	0.851	0.704	0.921	0.042	0.847	0.906	0.784			
MLMS _{CVPR19}	VGG-16	263.0	0.774	0.064	0.681	0.809	0.612	0.852	0.048	0.761	0.851	0.699	0.921	0.039	0.859	0.907	0.773			
AFNet _{CVPR19}	VGG-16	143.0	0.797	0.057	0.717	0.826	0.635	0.862	0.046	0.785	0.855	0.714	0.923	0.036	0.869	0.905	0.772			
MSWS _{CVPR19}	Dense-169	48.6	0.718	0.109	0.527	0.756	0.362	0.767	0.098	0.586	0.749	0.376	0.856	0.084	0.685	0.818	0.438			
R ³ Net _{ICCV18}	ResNet-101	215.0	0.795	0.063	0.728	0.817	0.599	0.828	0.058	0.763	0.817	0.601	0.915	0.036	0.877	0.895	0.740			
CapSal _{CVPR19}	ResNet-101	-	0.699	0.101	0.482	0.674	0.396	0.823	0.072	0.691	0.808	0.605	0.882	0.062	0.782	0.850	0.654			
SRM _{ICCV17}	ResNet-50	189.0	0.769	0.069	0.658	0.798	0.523	0.826	0.058	0.722	0.824	0.592	0.906	0.046	0.835	0.887	0.680			
DGRI _{CVPR18}	ResNet-50	646.1	0.779	0.063	0.697	0.810	0.584	0.834	0.051	0.760	0.836	0.656	0.913	0.037	0.865	0.897	0.744			
PICANetR _{CVPR18}	ResNet-50	197.2	0.803	0.065	0.695	0.832	0.632	0.860	0.050	0.755	0.859	0.696	0.918	0.043	0.840	0.904	0.765			
CPD _{CVPR19}	ResNet-50	183.0	0.797	0.056	0.719	0.825	0.655	0.865	0.043	0.795	0.858	0.741	0.925	0.034	0.875	0.905	0.795			
PoolNet _{CVPR19}	ResNet-50	273.3	0.808	0.056	0.729	0.836	0.675	0.880	0.040	0.807	0.871	0.765	0.932	0.033	0.881	0.917	0.811			
BASNet _{CVPR19}	ResNet-34	348.5	0.805	0.056	0.751	0.836	0.694	0.860	0.047	0.803	0.853	0.758	0.928	0.032	0.889	0.909	0.807			
U ² -Net (Ours)	RSU	176.3	0.823	0.054	0.757	0.847	0.702	0.873	0.044	0.804	0.861	0.765	0.935	0.031	0.890	0.916	0.812			
U ² -Net [†] (Ours)	RSU	4.7	0.813	0.060	0.731	0.837	0.676	0.852	0.054	0.763	0.847	0.723	0.928	0.037	0.867	0.908	0.794			

Рисунок 1 – Порівняння сучасних SOD моделей

Враховуючи те, що ця модель давала дуже високі результати, а також є відкритою і активно оновлюваною – я вирішив використати саме її у своєму додатку.

Базово модель має такий функціонал: на вхід подається зображення (рис.2), а на виході отримується чорно-біла маска у форматі png, що розділяє передній і задній плани.

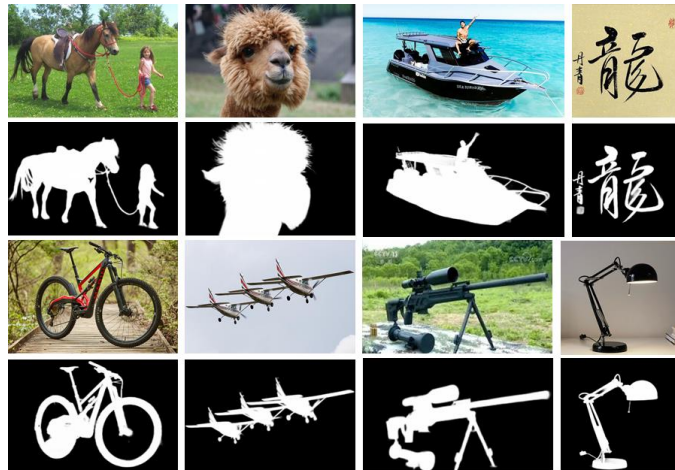


Рис 2 – Початкові зображення та результат виконання u2-net

На основі цієї моделі я написав тестовий код, щоб перевірити як працює ця модель. Для початку встановлюю потрібну бібліотеку через рір інсталер (рис. 3).

```
(Курсовий проект 2 курс Політех) PS D:\Робочі файли\Курсовий проект 2 курс Політех> pip install rembg
Collecting rembg
  Downloading rembg-2.0.29-py3-none-any.whl (13 kB)
Collecting tqdm==4.04.1
  Downloading tqdm-4.64.1-py2.py3-none-any.whl (78 kB)
    | 78 kB 1.5 MB/s
Collecting python-multipart==0.0.5
  Downloading python-multipart-0.0.5.tar.gz (32 kB)
  Preparing metadata (setup.py) ... done
Collecting numpy==1.23.5
  Downloading numpy-1.23.5-cp310-cp310-win_amd64.whl (14.6 MB)
    | 14.6 MB 6.8 MB/s
Collecting pillow==9.3.0
  Downloading Pillow-9.3.0-cp310-cp310-win_amd64.whl (2.5 MB)
    | 2.5 MB 6.4 MB/s
Collecting opencv-python-headless==4.6.0.66
  Downloading opencv_python_headless-4.6.0.66-cp36-abi3-win_amd64.whl (35.5 MB)
    | 35.5 MB 6.8 MB/s
Collecting watchdog==2.1.9
  Downloading watchdog-2.1.9-py3-none-win_amd64.whl (78 kB)
    | 78 kB 5.1 MB/s
Collecting filetype==1.2.0
  Downloading filetype-1.2.0-py2.py3-none-any.whl (19 kB)
Collecting pooch==1.6.0
```

Рис 3 – встановлення бібліотеки rembg

Пишу код:

```
from rembg import remove
from PIL import Image
from pathlib import Path

def remove_bg():                                     ## основна
функція                                             функція
    list_of_extensions = ['*.png', '*.jpg']         ## список з
розширеннями які нам потрібні                     розширеннями які нам потрібні
    all_files = []

    for ext in list_of_extensions:                  ## цикл з
перебором усіх файлів у вказаній папці             перебором усіх файлів у вказаній папці
        all_files.extend(Path('D:\Робочі файли\Курсовий проект 2 курс
Політех\input_imgs').glob(ext))                    ## шлях до папки, з якої братимуться файли

    for index, item in enumerate(all_files):
        input_path = Path(item)
        file_name = input_path.stem
```

```

        output_path = f'D:\Робочі файли\Курсовий проект 2 курс
Політех\output_imgs/{file_name}_output.png'  ## шлях до папки, куди
зберігатимуться готові файли

        input_img = Image.open(input_path)
        output_img = remove(input_img)
        output_img.save(output_path)

        print(f'Completed: {index + 1}/{len(all_files)}')  ##
вивід у консолі прогресу

def main():  ## функція main,
у якій викликається основна функція
    remove_bg()

if __name__ == '__main__':  ## умова, за якої
викликається функція main
    main()

```

Дана тестова функція бере зображення із папки input_imgs, обробляє їх, та зберігає результат обробки у папку output_imgs. Ось вхідні зображення (рис.4), а ось результати роботи у консолі (рис.5) та у вихідній папці (рис.6).

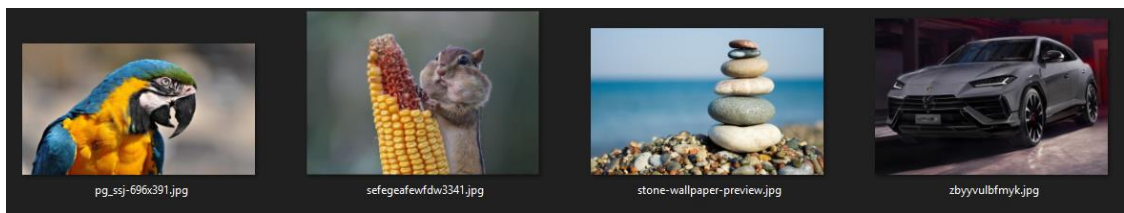


Рис.4 – вхідні зображення у папці input_imgs

```

"D:\Робочі файли\Політехніка 2 курс\Прикладне програмува
20002020-01020-00070 01090:00060:02040.0505040403000 0
Completed: 1/4
Completed: 2/4
Completed: 3/4
Completed: 4/4

Process finished with exit code 0
|

```

Рис. 5 – результат роботи програми у консолі

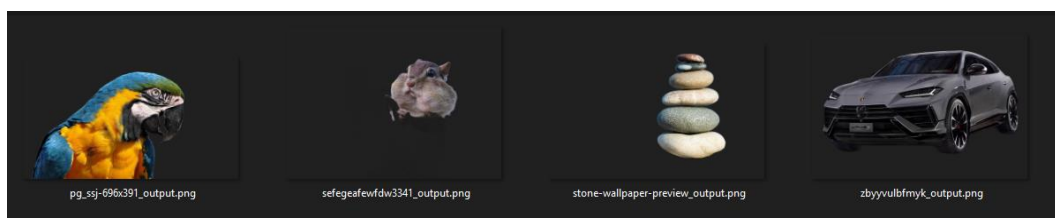


Рис. 6 – вихідні зображення у папці ourput_imgs

Як бачимо по другому зразку, модель працює не без недостатків, але результат для мене більш ніж підходить.

Далі я вирішив розробити цій програмі інтерфейс. Для цього я обрав PyQt5.

Щоб створити графічний інтерфейс для моїх вікон і діалогових вікон у PyQt, я можу піти двома основними шляхами: використовувати Qt Designer або закодувати графічний інтерфейс користувача вручну в простому коді Python.

Я обрав зробити інтерфейс у програмі QtDesigner та потім переформатувати його у файл .ру.

Далі я створив інтерфейс для трьох послідовних вікон (рис.7,8,9). Задній фон поправлю потім у python-файлі.

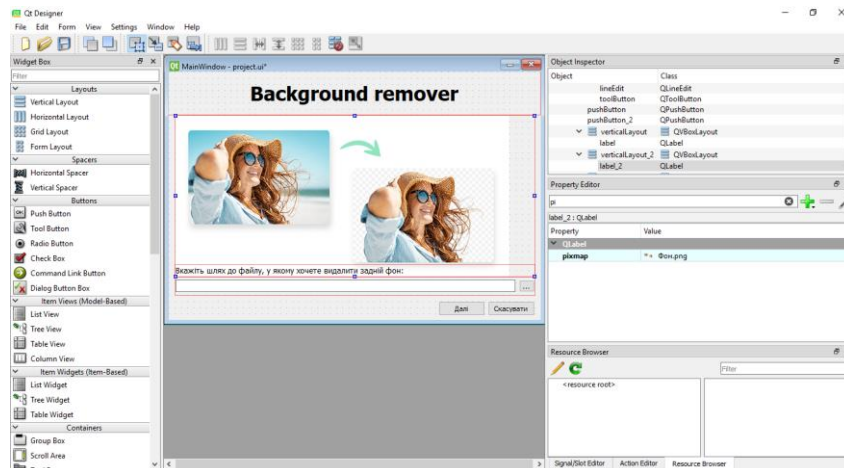


Рис. 7 – дизайн першого вікна у програмному середовищі QtDesigner

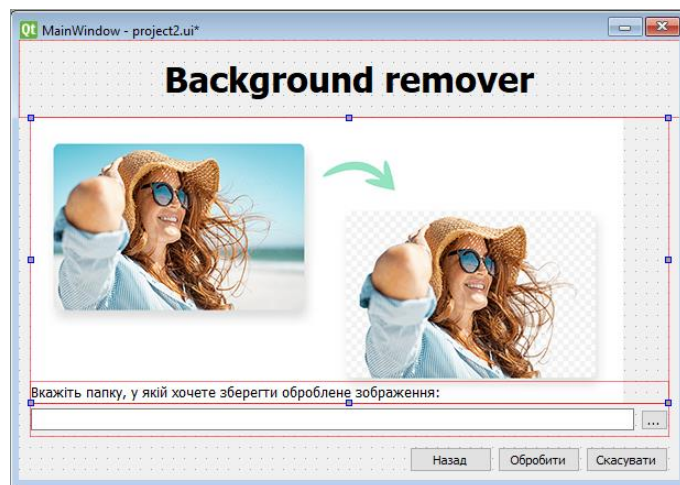


Рис. 8 – дизайн другого вікна

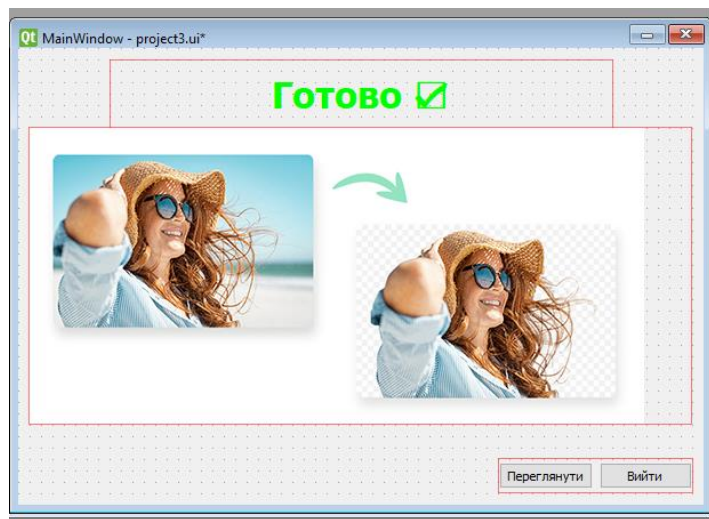


Рис. 9 – дизайн третього вікна

Для того, щоб перетворити ці файли у формат python-файлів, перейшов у консоль Windows, там перемістився у папку із проектом та ввів наступну команду (рис.10):

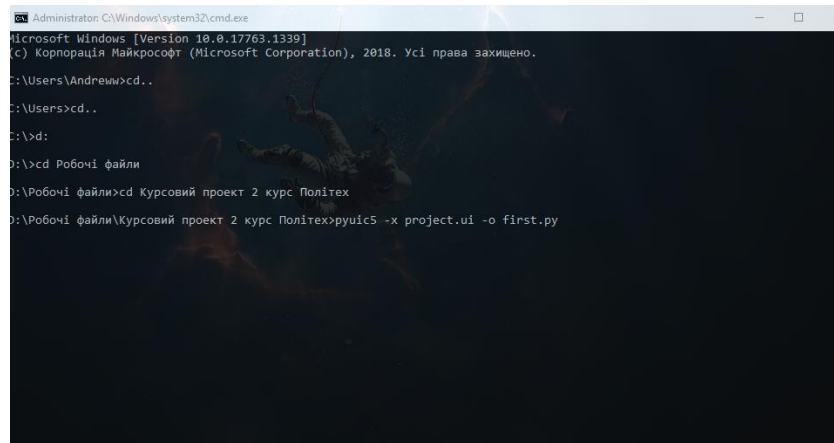


Рис 10 – перетворення .ui файлів у .py

Ми получили файл із ось таким змістом:

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'project.ui'
#
# Created by: PyQt5 UI code generator 5.15.7
#
# WARNING: Any manual changes made to this file will be lost when pyuic5 is
# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(600, 400)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setStyleSheet("background-image: url(D:Робочі файлиПолітехніка 2 курсСайтДизайн сайту/wall.jpg);")
        self.centralwidget.setObjectName("centralwidget")
        self.verticalLayoutWidget = QtWidgets.QWidget(self.centralwidget)
```



```

        self.verticalLayoutWidget.setGeometry(QtCore.QRect(-1, -1, 601, 71))
        self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
        self.verticalLayout =
QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout.setObjectName("verticalLayout")
        self.label = QtWidgets.QLabel(self.verticalLayoutWidget)
        self.label.setObjectName("label")
        self.verticalLayout.addWidget(self.label)
        self.verticalLayoutWidget_2 = QtWidgets.QWidget(self.centralwidget)
        self.verticalLayoutWidget_2.setGeometry(QtCore.QRect(10, 70, 581,
261))
        self.verticalLayoutWidget_2.setObjectName("verticalLayoutWidget_2")
        self.verticalLayout_2 =
QtWidgets.QVBoxLayout(self.verticalLayoutWidget_2)
        self.verticalLayout_2.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout_2.setObjectName("verticalLayout_2")
        self.label_2 = QtWidgets.QLabel(self.verticalLayoutWidget_2)
        self.label_2.setText("")
        self.label_2.setPixmap(QtGui.QPixmap("Фонн.jpg"))
        self.label_2.setObjectName("label_2")
        self.verticalLayout_2.addWidget(self.label_2)
        self.horizontalLayoutWidget = QtWidgets.QWidget(self.centralwidget)
        self.horizontalLayoutWidget.setGeometry(QtCore.QRect(10, 330, 581,
31))
        self.horizontalLayoutWidget.setObjectName("horizontalLayoutWidget")
        self.horizontalLayout =
QtWidgets.QHBoxLayout(self.horizontalLayoutWidget)
        self.horizontalLayout.setContentsMargins(0, 0, 0, 0)
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.lineEdit = QtWidgets.QLineEdit(self.horizontalLayoutWidget)
        self.lineEdit.setObjectName("lineEdit")
        self.horizontalLayout.addWidget(self.lineEdit)
        self.toolButton = QtWidgets.QToolButton(self.horizontalLayoutWidget)
        self.toolButton.setObjectName("toolButton")
        self.horizontalLayout.addWidget(self.toolButton)
        self.verticalLayoutWidget_3 = QtWidgets.QWidget(self.centralwidget)
        self.verticalLayoutWidget_3.setGeometry(QtCore.QRect(10, 310, 581,
21))
        self.verticalLayoutWidget_3.setObjectName("verticalLayoutWidget_3")
        self.verticalLayout_3 =
QtWidgets.QVBoxLayout(self.verticalLayoutWidget_3)
        self.verticalLayout_3.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout_3.setObjectName("verticalLayout_3")
        self.label_3 = QtWidgets.QLabel(self.verticalLayoutWidget_3)
        self.label_3.setObjectName("label_3")
        self.verticalLayout_3.addWidget(self.label_3)
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(515, 370, 75, 23))
        self.pushButton.setObjectName("pushButton")
        self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_2.setGeometry(QtCore.QRect(435, 370, 75, 23))
        self.pushButton_2.setObjectName("pushButton_2")
        MainWindow.setCentralWidget(self.centralwidget)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.label.setText(_translate("MainWindow", "<html><head/><body><p
align=\"center\"><span style=\" font-size:24pt; font-weight:600;\">Background
remover</span></p></body></html>"))

```

```

        self.toolButton.setText(_translate("MainWindow", "..."))
        self.label_3.setText(_translate("MainWindow",
"<html><head></head><body><p><span style='font-size:10pt;'>Вкажіть шлях до
файлу, у якому хочете видалити задній фон:</span></p></body></html>"))
        self.pushButton.setText(_translate("MainWindow", "Скасувати"))
        self.pushButton_2.setText(_translate("MainWindow", "Далі"))

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```

Так само зробив із наступними двома файлами.

Щоб зробити перехід між цими трьома вікнами я створив головний файл програми Ptoject.py.

Туди імпортував усі бібліотеки та три python-файли вікон.

Задав фіксований розмір вікна:

```
MainWindow.setFixedSize(600, 400)
```

Також поміняв назву вікна та встановив іконку

```

MainWindow.setWindowTitle(_translate("Background Remover", "Background
Remover"))
MainWindow.setWindowIcon(QtGui.QIcon('icon.png'))

```

```

import sys
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QFileDialog, QApplication
from first import Ui_MainWindow
from second import Ui_SecondWindow
from third import Ui_ThirdWindow
from PyQt5.QtWidgets import QMessageBox

from rembg import remove
from PIL import Image
from pathlib import Path

```

Налаштування які відповідають за запуск першого вікна:

```

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()
ui = Ui_MainWindow()
ui.setupUi(MainWindow)
MainWindow.show()

```

Додав функцію, яка робить кнопку неактивною «Далі неактивною»

```

def test():
    if ui.lineEdit1.text() == '':
        ui.next1.setEnabled(False)
    elif ui.lineEdit1.text() != '':
        ui.next1.setEnabled(True)

test()

```

Створив функції при натисненні на різні кнопки першого вікна:

При натисненні кнопки «...» відкриється провідник Windows, де можна обрати зображення, яке потрібно обробити. Після вибору шлях запишеться в lineEdit. Додав умову, яка перевіряє чи обрано файл, і відповідно активує, або не активує кнопку «Далі»

```
def search1():
    global res
    res = QFileDialog.getOpenFileName(None, 'toolButton', 'D:/', 'JPG File (*.jpg);; PNG File (*.png)')
    ui.lineEdit1.setText(str(res[0]))
    if ui.lineEdit1.text() == '':
        ui.next1.setEnabled(False)
    elif ui.lineEdit1.text() != '':
        ui.next1.setEnabled(True)
```

При натисненні кнопки «Скасувати» програма закриється

```
def cancell(self):
    QApplication.quit()
```

При натисненні кнопки «Далі» відкриється наступне вікно.

```
def openSecondWindow():
    inp = res[0]
    global SecondWindow, ui
    SecondWindow = QtWidgets.QMainWindow()
    ui = Ui_SecondWindow()
    ui.setupUi(SecondWindow)
    FirstWindow.hide()
    SecondWindow.show()
```

Вигляд першого вікна програми, та вікна провідника(рис. 11, 12, 13).

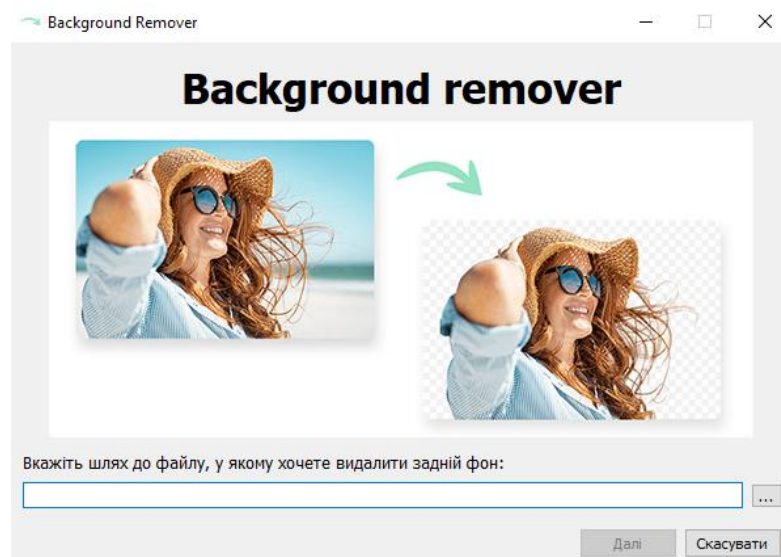


Рис. 11 – перше вікно програми Background remover

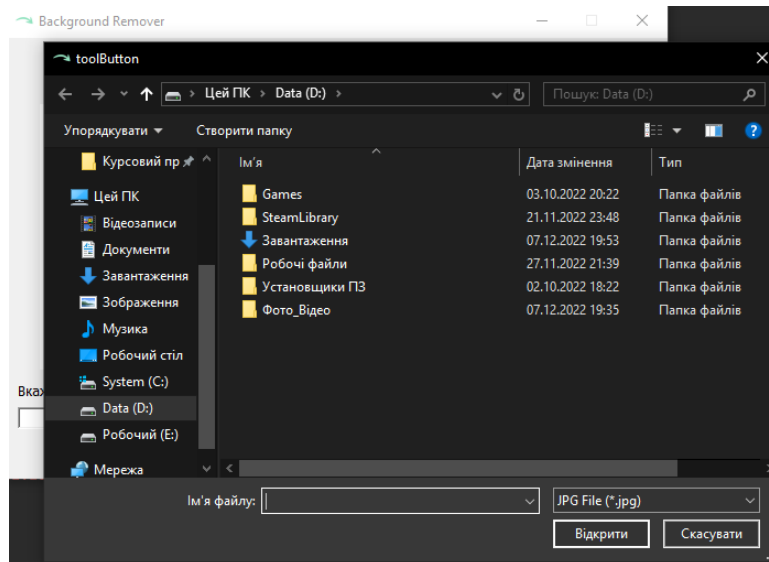


Рис. 12 – перше вікно програми Background remover
(вибір файлу)

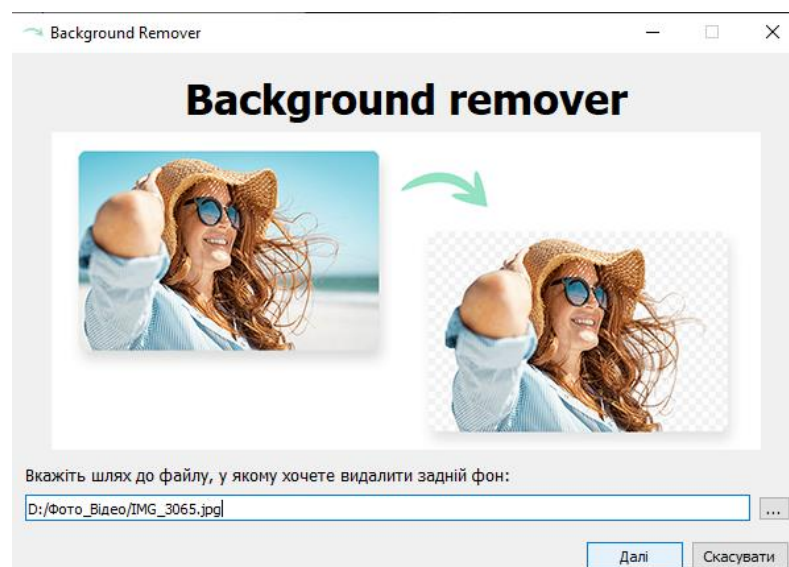


Рис. 13 – перше вікно програми Background remover
(коли файл обрано)

Функції на другому вікні:

При натисненні на кнопку «Назад» відбувається перехід на попереднє вікно

```
def Back2First():
    SecondWindow.hide()
    FirstWindow.show()
```

При натисненні на кнопку «...» відкривається провідник для вибору папки. Також додав функцію, яка перевіряє чи обрано папку і відповідно активує кнопку «>>»

```
def search2():
    global dirname
    dirname = QFileDialog.getExistingDirectory(None, 'Вибрати папку' 'D:/')
    ui.lineEdit2.setText(str(dirname))
    if ui.lineEdit2.text() == '':
```

```

        ui.go.setEnabled(False)
    elif ui.lineEdit2.text() != '':
        ui.go.setEnabled(True)

```

При натисненні на кнопку «Обробити» запускається головна функція програми. Вона приймає значення які обрав користувач та обробляє зображення. Після закінчення обробки відкривається третє вікно.

```

def go():
    input_path = Path(inp)
    file_name = input_path.stem
    output_path = f'{dirname}/{file_name}_output.png'
    input_img = Image.open(input_path)
    output_img = remove(input_img)
    output_img.save(output_path)

    global ThirdWindow
    ThirdWindow = QMainWindow()
    ui = Ui_ThirdWindow()
    ui.setupUi(ThirdWindow)
    SecondWindow.close()
    ThirdWindow.show()

```

Вигляд другого вікна програми, вікна провідника, та вікна із попередженням (рис. 14, 15, 16).

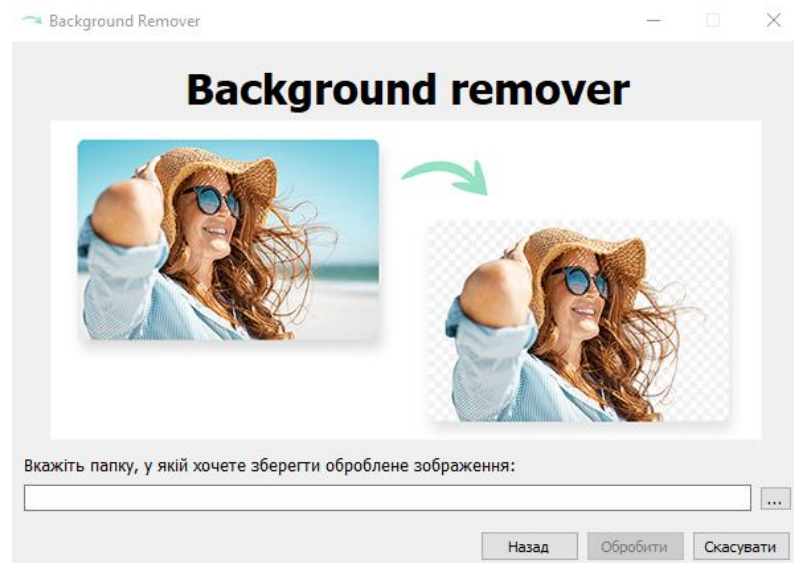


Рис. 14 – друге вікно програми Background remover

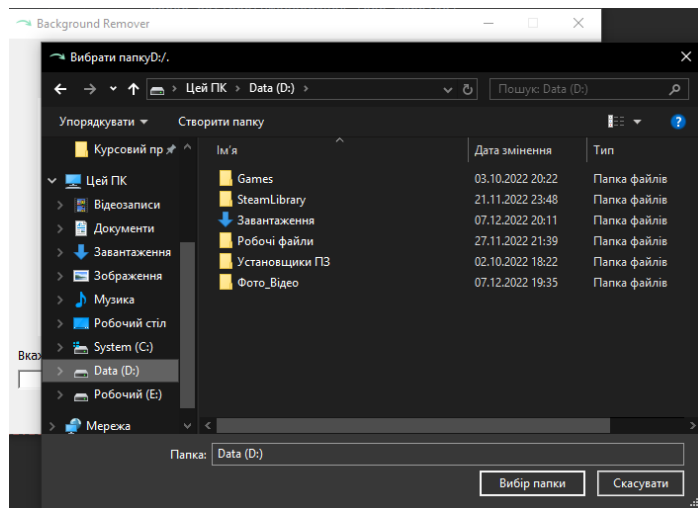


Рис. 15 – друге вікно програми Background remover
(вибір папки)

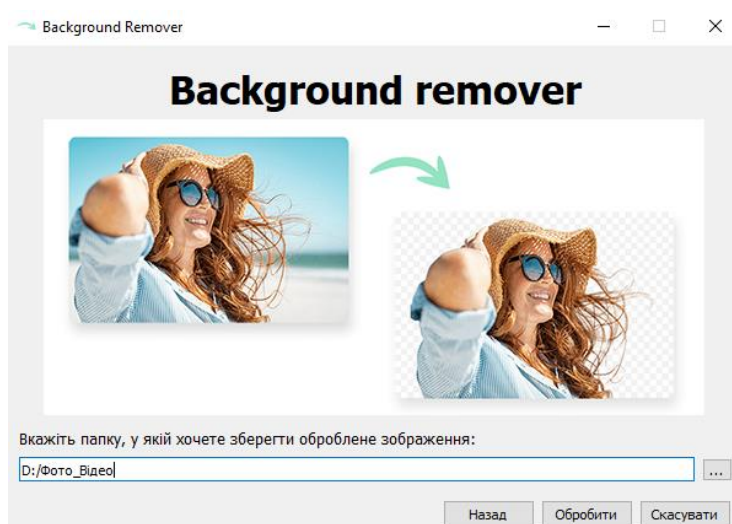


Рис. 16 – друге вікно програми Background remover
(коли папку обрано)

Функції на третьому вікні:

При натисненні кнопки «Вихід» програма завершить роботу.

```
def exit():
    QApplication.quit()
```

При натисненні кнопки «Переглянути» відкриється оброблене зображення програмою, заданою за замовчуванням у Windows.

```
def view():
    img = Image.open(output_path)
    img.show()
```

Вигляд третього вікна програми, та зображення результату. (рис. 17, 18).

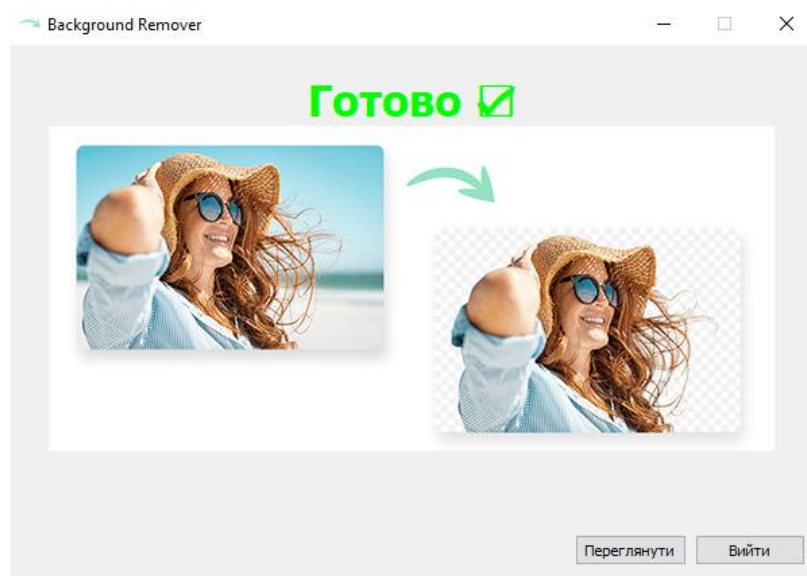


Рис. 17 – третє вікно програми Background remover

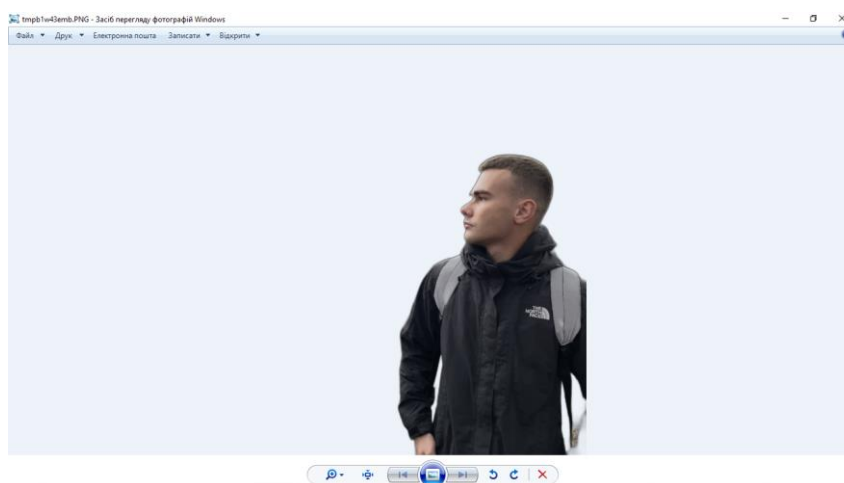


Рис. 18 – перегляд результату при натисненні кнопки «Переглянути»

Налаштування для підключення кнопок до їхніх функцій:

```
ui.view.clicked.connect(view)
ui.exit.clicked.connect(exit)

ui.go.clicked.connect(go)
ui.search2.clicked.connect(search2)
ui.back1.clicked.connect(Back2First)
ui.cancell2.clicked.connect(cancell)

ui.cancell1.clicked.connect(cancell)
ui.search1.clicked.connect(search1)
ui.next1.clicked.connect(openSecondWindow)

sys.exit(app.exec_())
```