###### A

Mini Project On

#### ACCURATE AND ROBUST SALIENCY DETECTION

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

###### COMPUTER SCIENCE AND ENGINEERING

By

###### P.SATVIKA(197R1A05N7)

###### PARANKUSHAM VAISHNAVI(197R1A05N8)

###### MAMINDLA MOUNIKA(197R1A05M7)

Under the Guidance of

**DR.K.SRUJAN RAJU**

(Professor & Hod)



##### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**CMR TECHNICAL CAMPUS UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2019-2023**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



#### CERTIFICATE

This is to certify that the project entitled **“ACCURATE AND ROBUST SALIENCY DETECTION”** being submitted by **P.SATVIKA(197R1A05N7),PARANKUSHAM VAISHNAVI(197R1A05N8) & MAMINDLA MOUNIKA(197R1A05M7)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year

2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**Dr.K.Srujan raju Dr. A. Raji Reddy**

(Professor & Hod) DIRECTOR

INTERNAL GUIDE

**Dr. K. Srujan Raju EXTERNAL EXAMINER**

HOD

**Submitted for viva voice Examination held on**

##### ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Dr.K.Srujan raju,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr. M. Subha Mastan Rao & J. Narasimharao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

P.SATVIKA (197R1A05N7)

**PARANKUSHAM VAISHNAVI (197R1A05N8)**

**MAMINDLA MOUNIKA (197R1A05M7)**

##### ABSTRACT

In this project, we design a simple yet powerful deep network architecture. U2 -Net, for saliency detection. The architecture of our U2 -Net is a two-level nested U-structure. The design has the following advantages: (1) it is able to capture more contextual information from different scales thanks to the mixture of receptive fields of different sizes in our proposed Residual U-blocks (RSU).(2) it increases the depth of the whole architecture without significantly increasing the computational cost because of the pooling operations used in these RSU blocks. This architecture enables us to train a deep network from scratch without using backbones from image classification tasks. We instantiate two models of the proposed architecture, U2 - Net (176.3 MB, 30 FPS on GTX 1080Ti GPU) and U2 - Net(4.7 MB, 40 FPS), to facilitate the usage in different environments. Both models achieve competitive performance on six saliency detection datasets.U2-Net architecture allows the network to go deeper, attain high resolution, without significantly increasing the memory and computation cost. This is achieved by a nested U-structure: on the bottom level, with a novel Residual U-block (RSU) module, which is able to extract intra-stage multi-scale features without degrading the feature map resolution; on the top level, there is a U-Net like structure, in which each stage

is filled by a RSU block.

##### FIGURE NO FIGURE NAME PAGE NO

###### Figure 3.1 Project Architecture for 7

###### Accurate and robust saliency detection

###### Figure 3.2 Use Case Diagram for 8

###### Accurate and robust saliency detection

###### Figure 3.3 Class Diagram for 9

###### Accurate and robust saliency detection

###### Figure 3.4 Sequence diagram for 10

###### Accurate and robust saliency detection

###### Figure 3.5 Activity diagram for 11

Accurate and robust saliency

detection

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NO.** | **SCREENSHOT NAME** | **PAGE NO**. |
| Screenshot 5.1 | User page | 16 |
| Screenshot 5.2 | Selecting a picture | 17 |
| Screenshot 5.3 | Uploading the picture | 18 |
| Screenshot 5.4 | Salient image with transparent background | 19 |
| Screenshot 5.5  Screenshot 5.6 | Salient image with black background  Salient image with white background | 20      21 |

ABSTRACT i

LIST OF FIGURES ii

LIST OF SCREENSHOTS iii

1. [INTRODUCTION 1](#_TOC_250034)
   1. [PROJECT SCOPE 1](#_TOC_250033)
   2. [PROJECT PURPOSE 1](#_TOC_250032)
   3. [PROJECT FEATURES 1](#_TOC_250031)
2. [SYSTEM ANALYSIS 2](#_TOC_250030)
   1. [PROBLEM DEFINITION 2](#_TOC_250029)
   2. [EXISTING SYSTEM 2](#_TOC_250028)
      1. LIMITATIONS OF THE EXISTING SYSTEM 3
   3. [PROPOSED SYSTEM 3](#_TOC_250027)
      1. ADVANTAGES OF PROPOSED SYSTEM 3
   4. [FEASIBILITY STUDY 4](#_TOC_250026)
      1. [ECONOMIC FEASIBILITY 4](#_TOC_250025)
      2. [TECHNICAL FEASIBILITY 5](#_TOC_250024)
      3. SOCIAL FEASIBILITY 5
   5. [HARDWARE & SOFTWARE REQUIREMENTS 5](#_TOC_250023)
      1. [HARDWARE REQUIREMENTS 5](#_TOC_250022)
      2. [SOFTWARE REQUIREMENTS 6](#_TOC_250021)
3. [ARCHITECTURE 8](#_TOC_250020)
   1. [PROJECT ARCHITECTURE 7](#_TOC_250019)
   2. [DESCRIPTION 7](#_TOC_250018)
   3. [USE CASE DIAGRAM 9](#_TOC_250017)
   4. [CLASS DIAGRAM 1](#_TOC_250016)0
   5. [SEQUENCE DIAGRAM 1](#_TOC_250015)1
   6. [ACTIVITY DIAGRAM 1](#_TOC_250014)2
4. [IMPLEMENTATION 1](#_TOC_250013)3
   1. [SAMPLE CODE 1](#_TOC_250012)3
5. RESULTS 19
6. [TESTING 2](#_TOC_250011)5
   1. [INTRODUCTION TO TESTING 2](#_TOC_250010)5
   2. [TYPES OF TESTING 2](#_TOC_250009)5
      1. [UNIT TESTING 2](#_TOC_250008)5
      2. [INTEGRATION TESTING 2](#_TOC_250007)6
      3. [FUNCTIONAL TESTING 2](#_TOC_250006)6
   3. [TEST CASES 2](#_TOC_250005)7
      1. [CLASSIFICATION 2](#_TOC_250004)7
7. CONCLUSION & FUTURE SCOPE 28
   1. [PROJECT CONCLUSION 2](#_TOC_250003)8
   2. [FUTURE SCOPE 2](#_TOC_250002)8
8. REFERENCES 29
   1. [REFERENCES 2](#_TOC_250001)9
   2. [GITHUB LINK 2](#_TOC_250000)9

# INTRODUCTION

#### INTRODUCTION

##### PROJECT SCOPE

This project is titled “Accurate and Robust Saliency Detection” will consequently increase the demand for the development of high precision and real-time computer vision techniques.The salient regions in images can facilitate subsequent high-level vision tasks for improved efficiency and optimal resource usage.

##### PROJECT PURPOSE

This project has been developed for predicting where people look in a scenario, which is relevant in many applications. For example, in advertising, it may be important for the producer to know if the key concept catches the viewer’s eye. Saliency Detection is a preprocessing step in computer vision which aims at finding salient objects in an image.

##### PROJECT FEATURES

The main features of this project are that this model allows the network to go deeper, attain high resolution, without significantly increasing the memory and computation cost. Saliency refers to unique features (pixels, resolution etc.) of the image in the context of visual processing. These unique features depict the visually alluring locations in an image. Saliency map is a topographical representation of them.

## SYSTEM ANALYSIS

##### SYSTEM ANALYSIS

**SYSTEM ANALYSIS**

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

##### PROBLEM DEFINITION

A general statement of saliency detection is aiming to segment the most attractive objects in an image which can be used in robots and autonomous vehicles to detect the important objects in a scenario to avoid errors.

.

##### 2.2EXISTING SYSTEM

Saliency Detection aims at segmenting the most visually attractive objects in an image.It is widely used in many fields, such as visual tracking and image segmentation. Recently, with the development of deep convolutional neural networks (CNNs), especially the rise of Fully Convolutional Networks (FCN) in image segmentation, the salient object detection has been improved significantly. There is a common pattern in the design of most saliency detection networks , that is, they focus on making good use of deep features extracted by existing backbones, such as

Alexnet, VGG, ResNet, ResNeXt, DenseNet, etc. However, these backbones are all originally designed for image classification. They extract features that are representative of semantic meaning rather than local details and global contrast information, which are essential to saliency detection. And they need to be pretrained on ImageNet data which is data-inefficient especially if the target data follows a different distribution than ImageNet.

**2.2.1 DISADVANTAGES OF EXISTING SYSTEM**

Following are the disadvantages of existing system:

* It uses backbones like Alexnet, ResNet,ResNext,DenseNet.
* Open source code is hard to read.

##### PROPOSED SYSTEM

Our main contribution is to propose a simple network architecture, called U 2 -Net, that addresses the two questions above. First, U2 -Net is a two-level nested U-structure that is designed for saliency detection without using any pre-trained backbones from image classification. It can be trained from scratch to achieve competitive performance. Second, the novel architecture allows the network to go deeper, attain high resolution, without significantly increasing the memory and computation cost. This is achieved by a nested U-structure: on the bottom level, we design a novel Residual U-block (RSU), which is able to extract intra-stage multi-scale features without degrading the feature map resolution; on the top level, there is a

U-Net like structure, in which each stage is filled by a RSU block. The two-level

configuration results in a nested U-structure. Our U2 -Net (176.3 MB) achieves competitive performance against the state-of-the-art (SOTA) methods on six public datasets, and runs at real time (30 FPS, with input size of 320×320×3). To facilitate the usage of our design in computation and memory constrained environments, we provide a small version of our U2 -Net, called U 2 -Net† (4.7 MB).

* + 1. ADVANTAGES OF THE PROPOSED SYSTEM
* It is able to capture more contextual information from different scales thanks to the mixture of receptive fields of different sizes in our proposed Residual U-blocks (RSU).
* It increases the depth of the whole architecture without significantly increasing the computational cost because of the pooling operations used in these RSU blocks.

##### 

##### 2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

* EconomicFeasibility
* TechnicalFeasibility
* SocialFeasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

* + 1. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

* + 1. BEHAVIORAL FEASIBILITY

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible

##### HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

* Processor : Minimum of Intel Core i3
* Hard disk : 4GB(minimum)
* RAM : 8GB and above
* Input devices : Keyboard, mouse.
* Monitor:14’ color Monitor.

##### SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

* Operating system : Windows 7 and above.
* Languages : Python.
* Design:XML.
* Tools : Pycharm.

## ARCHITECTURE

##### 3.ARCHITECTURE

##### PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

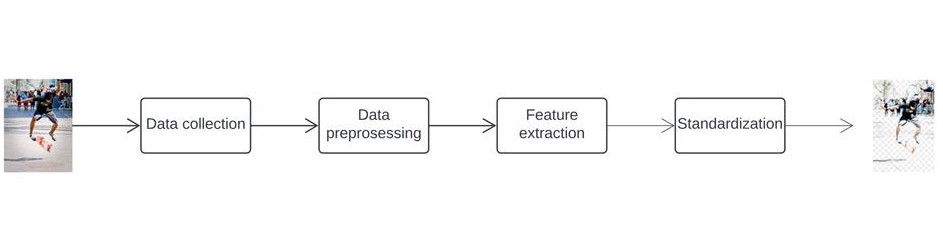


Figure 3.1: Project Architecture of Accurate and Robust Saliency Detection.

* 1. DESCRIPTION

This project is totally based on identifying the important or particularly noticeable features in an image. The model is built for saliency detection which is an internal functionality of robotics. the model is built using u2net architecture which is based on u-net machine learning model. the data from the image is collected and preprocessed. Then features are extracted from the image based on pixels and standardization is performed.

* 1. USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

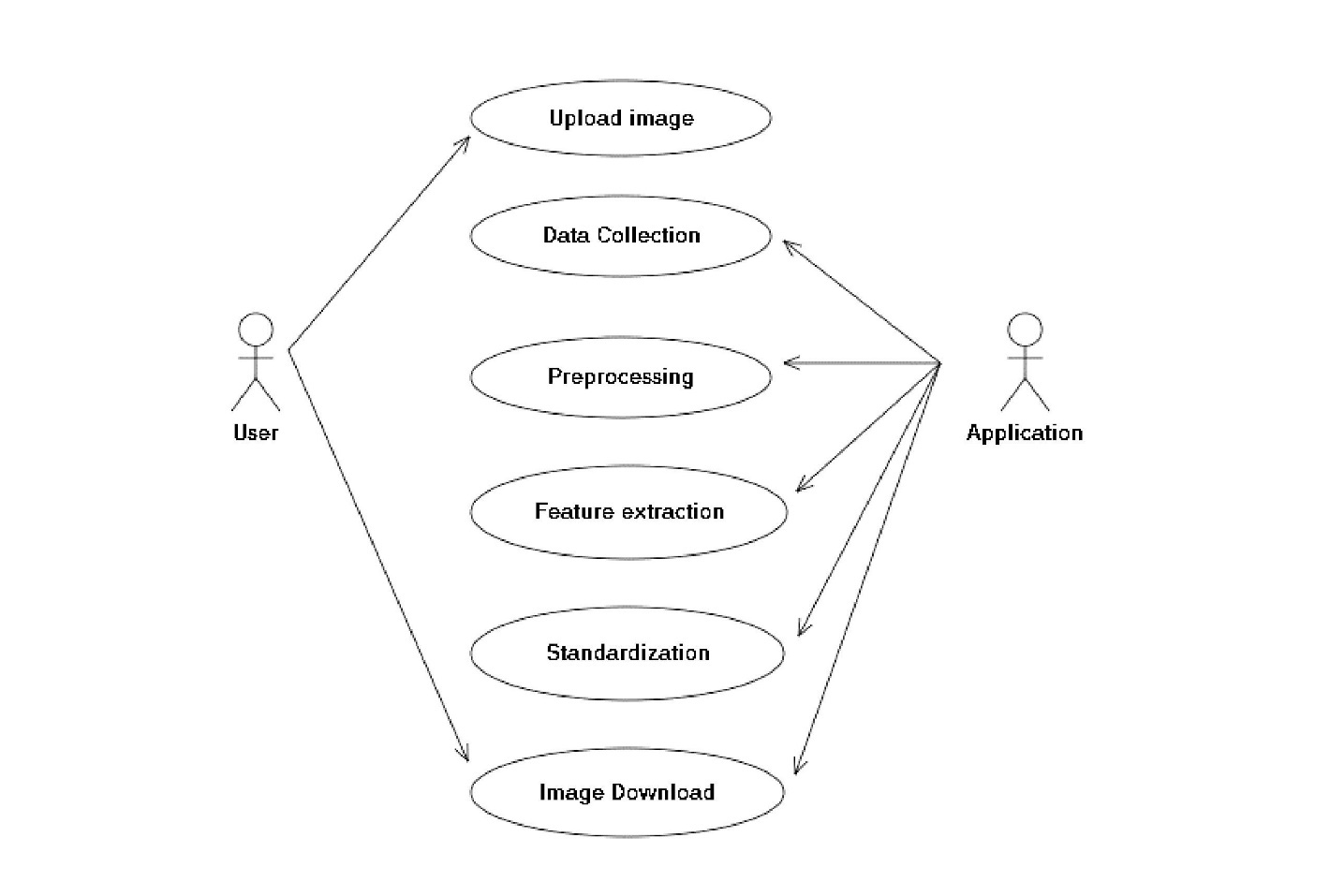


Figure 3.2: Use Case Diagram for Accurate and Robust Saliency Detection.

##### CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations(or methods), and the relationships among objects.

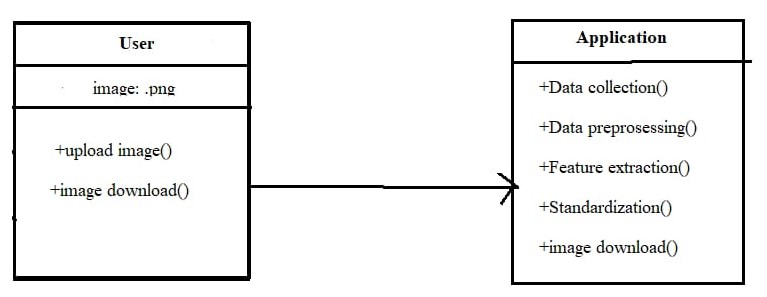


Figure 3.3: Class Diagram for Accurate and Robust Saliency Detection.

##### 3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

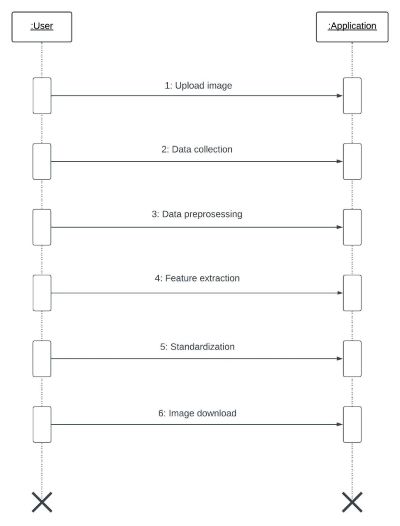


Figure 3.4: Sequence Diagram for Accurate and Robust Saliency Detection.

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

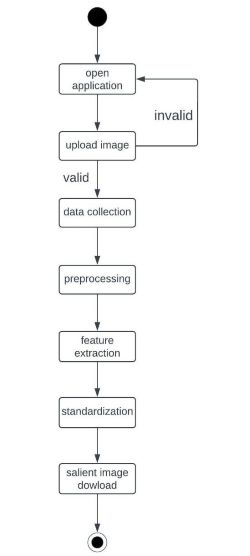


Figure 3.5: Activity Diagram for Accurate and Robust Saliency Detection.

## 4.IMPLEMENTATION

##### 4.1 SAMPLE CODE

import os

import base64

import math

#from app import app

from flask import flash, request, redirect, url\_for, render\_template,Flask

from werkzeug.utils import secure\_filename

import io as io\_norm

from skimage import io

import torch

from torch.autograd import Variable

from torch.utils.data import Dataset, DataLoader

from torchvision import transforms

import numpy as np

from PIL import Image

import glob

from data\_loader import RescaleT

from data\_loader import ToTensor

from data\_loader import ToTensorLab

from data\_loader import SalObjDataset

from model import U2NET # full size version 173.6 MB

from model import U2NETP # small version u2net 4.7 MB

UPLOAD\_FOLDER = "static/uploads"

FINAL\_FOLDER = "static/final"

app = Flask(\_\_name\_\_)

app.secret\_key = "secret key"

app.config["UPLOAD\_FOLDER"] = UPLOAD\_FOLDER

app.config["FINAL\_FOLDER"] = FINAL\_FOLDER

app.config["MAX\_CONTENT\_LENGTH"] = 16\*1024\*1024

def normPRED(d):

ma = torch.max(d)

mi = torch.min(d)

dn = (d-mi)/(ma-mi)

return dn

def convert\_img(image\_name,pred,d\_dir, org\_img\_path):

predict = pred

predict = predict.squeeze()

predict\_np = predict.cpu().data.numpy()

org\_img = Image.open(org\_img\_path)

im = Image.fromarray(predict\_np\*255).convert('RGB')

img\_name = image\_name.split(os.sep)[-1]

image = io.imread(image\_name)

imo = im.resize((image.shape[1],image.shape[0]),resample=Image.BILINEAR)

# print(type(imo))

pb\_np = np.array(imo)

aaa = img\_name.split(".")

bbb = aaa[0:-1]

imidx = bbb[0]

for i in range(1,len(bbb)):

imidx = imidx + "." + bbb[i]

cubw, cubh = 20,20

wcubic = Image.new("RGB", (cubw, cubh), (255,255,255))

gcubic = Image.new("RGB", (cubw, cubh), (238,238,238))

whiteg = Image.new("RGB", imo.size, (255,255,255))

blackg = Image.new("RGB", imo.size, (0,0,0))

backg = Image.new("RGB", imo.size)

width, height = imo.size

for i in range(math.ceil(width/cubw)):

for j in range(math.ceil(height/cubh)):

if (i+j)%2:

backg.paste(wcubic,(i\*cubw, j\*cubh))

else:

backg.paste(gcubic, (i\*cubw, j\*cubh))

im1 = Image.Image.split(imo)

alpha = im1[0]

# checkerboard

backg.paste(org\_img, (0,0), alpha)

# white

whiteg.paste(org\_img, (0,0), alpha)

#black

blackg.paste(org\_img, (0,0), alpha)

# transparent

rgba = org\_img.convert("RGBA")

rgba.putalpha(alpha)

# returns [original image, choose mode img]

return [org\_img, rgba, backg, whiteg, blackg]

def run\_model():

# --------- 1. get image path and name ---------

model\_name='u2net'

image\_dir = os.path.join(os.getcwd(), 'static', 'uploads')

prediction\_dir = os.path.join(os.getcwd(), 'static', "final" + os.sep)

model\_dir = os.path.join(os.getcwd(), 'saved\_models', model\_name, model\_name + '.pth')

img\_name\_list = glob.glob(image\_dir + os.sep + '\*')

#print(img\_name\_list)

# --------- 2. dataloader ---------

#1. dataloader

test\_salobj\_dataset = SalObjDataset(img\_name\_list = img\_name\_list,

lbl\_name\_list = [],

transform=transforms.Compose([RescaleT(320),

ToTensorLab(flag=0)])

)

test\_salobj\_dataloader = DataLoader(test\_salobj\_dataset,

batch\_size=1,

shuffle=False,

num\_workers=1)

# --------- 3. model define ---------

if(model\_name=='u2net'):

print("...load U2NET---173.6 MB")

net = U2NET(3,1)

elif(model\_name=='u2netp'):

print("...load U2NEP---4.7 MB")

net = U2NETP(3,1)

if torch.cuda.is\_available():

net.load\_state\_dict(torch.load(model\_dir))

net.cuda()

else:

net.load\_state\_dict(torch.load(model\_dir, map\_location='cpu'))

net.eval()

# --------- 4. inference for each image ---------

final\_dict = {}

for i\_test, data\_test in enumerate(test\_salobj\_dataloader):

print("inferencing:",img\_name\_list[i\_test].split(os.sep)[-1])

inputs\_test = data\_test['image']

inputs\_test = inputs\_test.type(torch.FloatTensor)

if torch.cuda.is\_available():

inputs\_test = Variable(inputs\_test.cuda())

else:

inputs\_test = Variable(inputs\_test)

d1,d2,d3,d4,d5,d6,d7= net(inputs\_test)

# normalization

pred = d1[:,0,:,:]

pred = normPRED(pred)

final\_dict[img\_name\_list[i\_test].split(os.sep)[-1]] = convert\_img(img\_name\_list[i\_test],pred,prediction\_dir, img\_name\_list[i\_test])

del d1,d2,d3,d4,d5,d6,d7

# returns dict mapping file name to [original image, nobg image, checker, whiteg, blackg]

return final\_dict

ALLOWED\_EXTENSIONS = set(['png', 'jpg', 'jpeg', 'gif'])

def allowed\_file(filename):

return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

@app.route('/')

def upload\_form():

return render\_template('upload.html')

@app.route('/', methods=['POST'])

def upload\_image():

print(request.files)

if 'file' not in request.files:

flash('No file part')

return redirect(request.url)

file = request.files["file"]

file2 = request.files.getlist("file")

print(file2)

for file in file2:

if file.filename == '':

flash('No image selected for uploading')

return redirect(request.url)

if file and allowed\_file(file.filename):

filename = secure\_filename(file.filename)

file.save(os.path.join(app.config['UPLOAD\_FOLDER'], filename))

else:

flash('Allowed image types are -> png, jpg, jpeg, gif')

return redirect(request.url)

# dict mapping file name to [original image, nobg image, checker, whiteg, blackg]

final\_img\_dict = run\_model()

image\_dir = os.path.join(os.getcwd(), 'static', 'uploads')

img\_name\_list = glob.glob(image\_dir + os.sep + '\*')

for img in img\_name\_list:

try:

os.remove(img)

except OSError as e:

print("Error: %s:%s" % (img, e.strerror))

base64\_img\_crop\_list = {}

base64\_img\_list = {}

# [final, checker, white, black]

for i in range(1,5):

for final\_img in final\_img\_dict:

bytesio = io\_norm.BytesIO()

print(type(final\_img\_dict[final\_img]))

final\_img\_dict[final\_img][i].save(bytesio, format = "PNG")

bytesio.seek(0)

base64\_img = base64.b64encode(bytesio.getvalue())

base64\_img = base64\_img.decode("ascii")

base64\_img\_list[final\_img]=[base64\_img]

# [original, final, checker, white, black]

for i in range(5):

for final\_img in final\_img\_dict:

bytesio = io\_norm.BytesIO()

resized\_image = final\_img\_dict[final\_img][i]

resized\_image.thumbnail((400,400))

resized\_image.save(bytesio, format = "PNG")

bytesio.seek(0)

base64\_img = base64.b64encode(bytesio.getvalue())

base64\_img = base64\_img.decode("ascii")

if final\_img in base64\_img\_crop\_list:

base64\_img\_crop\_list[final\_img].append(base64\_img)

else:

base64\_img\_crop\_list[final\_img]=[base64\_img]

#print(base64\_img)

# print('upload\_image filename: ' + filename)

message = 'Image successfully uploaded and displayed below'

# base64\_img\_list maps filename to [final, crop\_orig, crop\_final, crop\_checker, crop\_white, crop\_black]

# print(base64\_img\_list.values())

print([x[-1][-1] for x in base64\_img\_list.values()])

return render\_template('upload.html', base64\_img = base64\_img\_list, message = message, base64\_img\_crop\_list=base64\_img\_crop\_list)

@app.route('/display/<filename>')

def display\_image(filename):

print('display\_image filename: ' + filename)

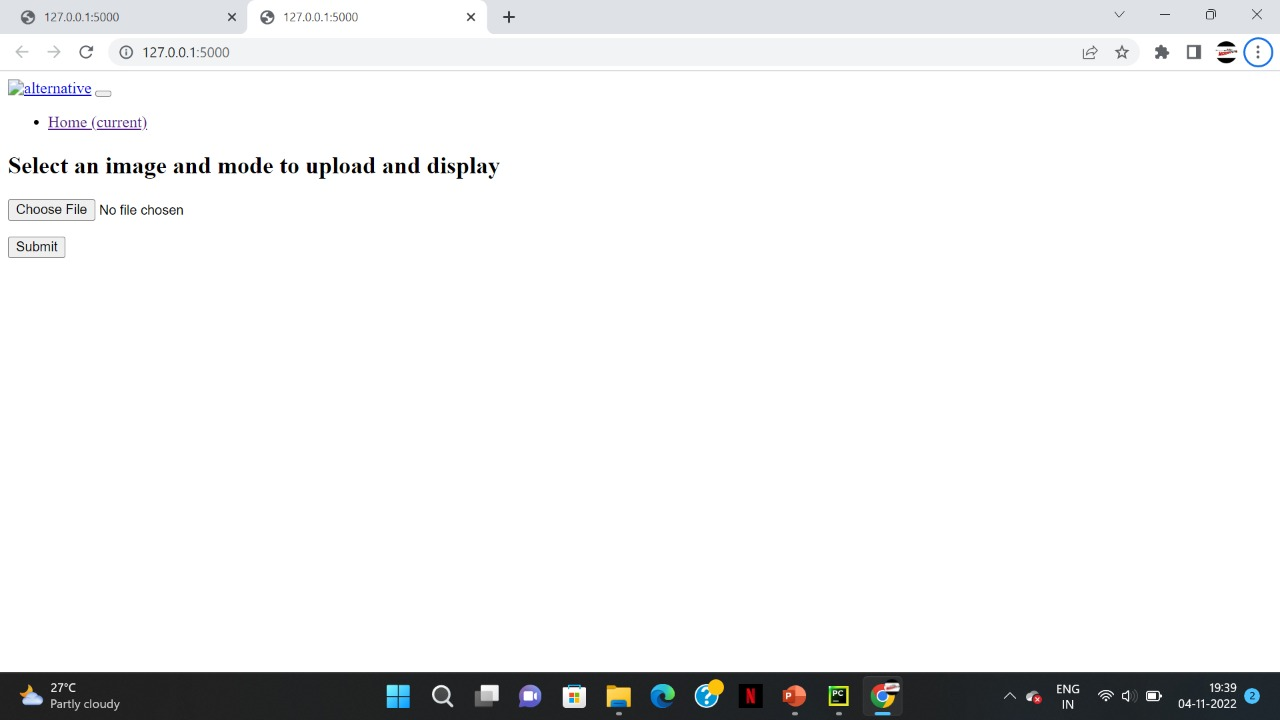
return redirect(url\_for('static', filename='final/' + filename.rsplit('.',1)[0]+'.png'), code=301)

if \_\_name\_\_ == "\_\_main\_\_":

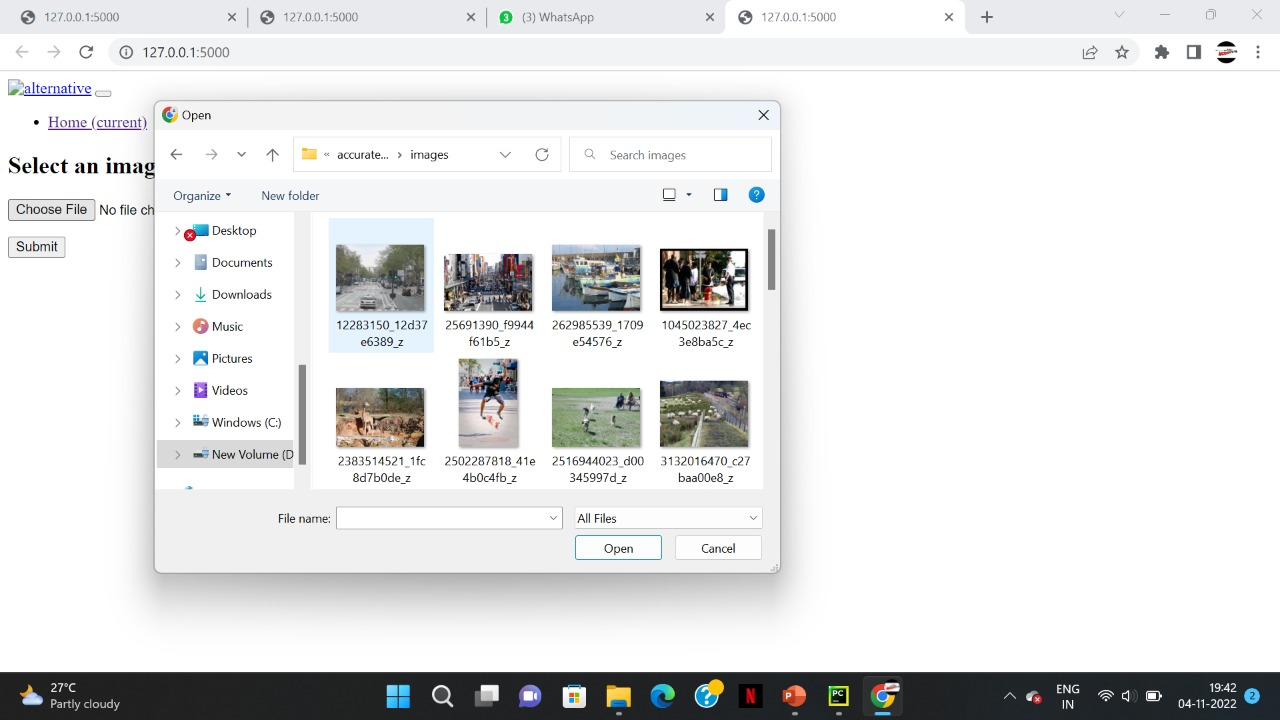
app.run(debug=True)

## 5.RESULTS

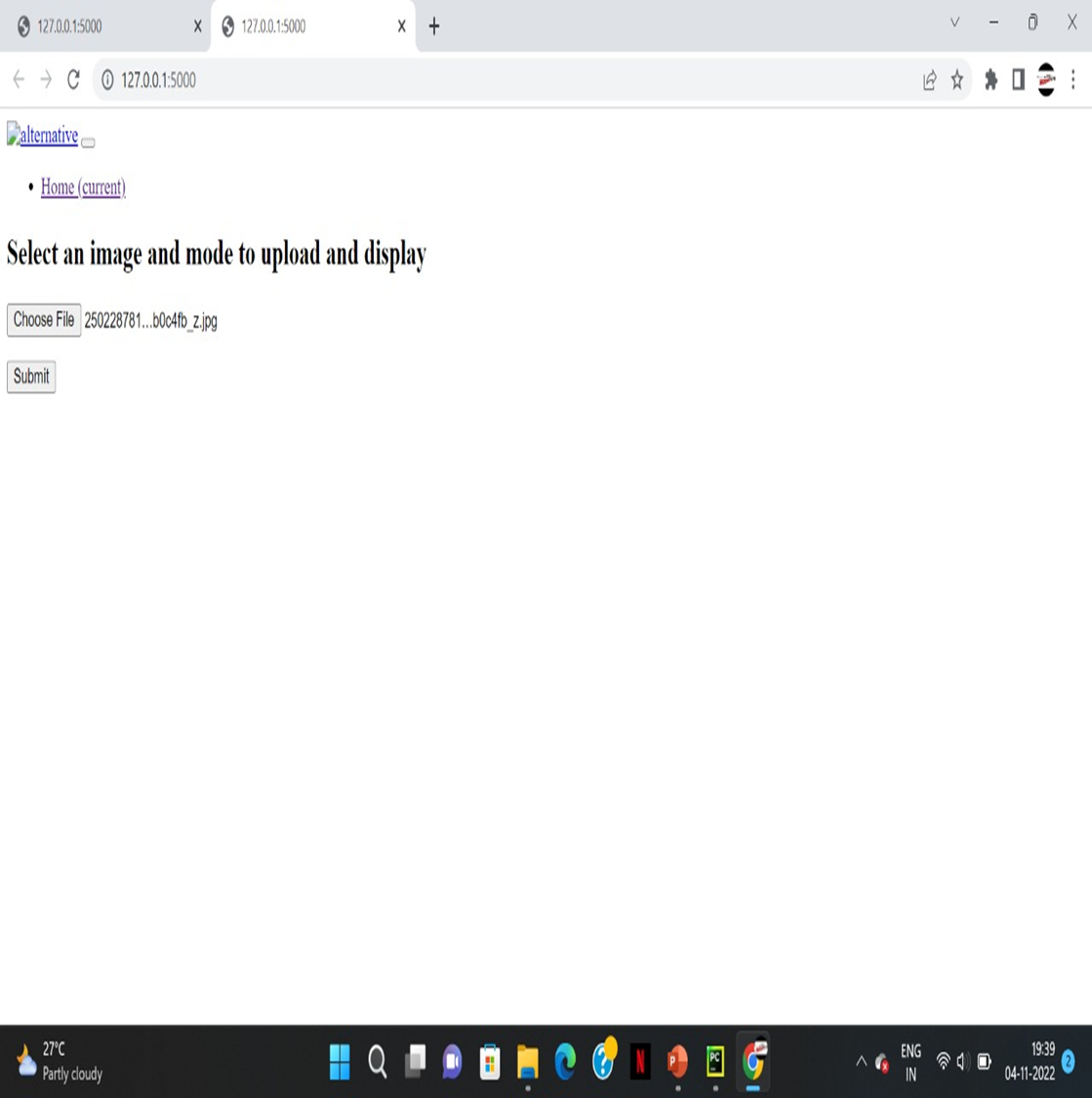
**5.RESULTS**

****

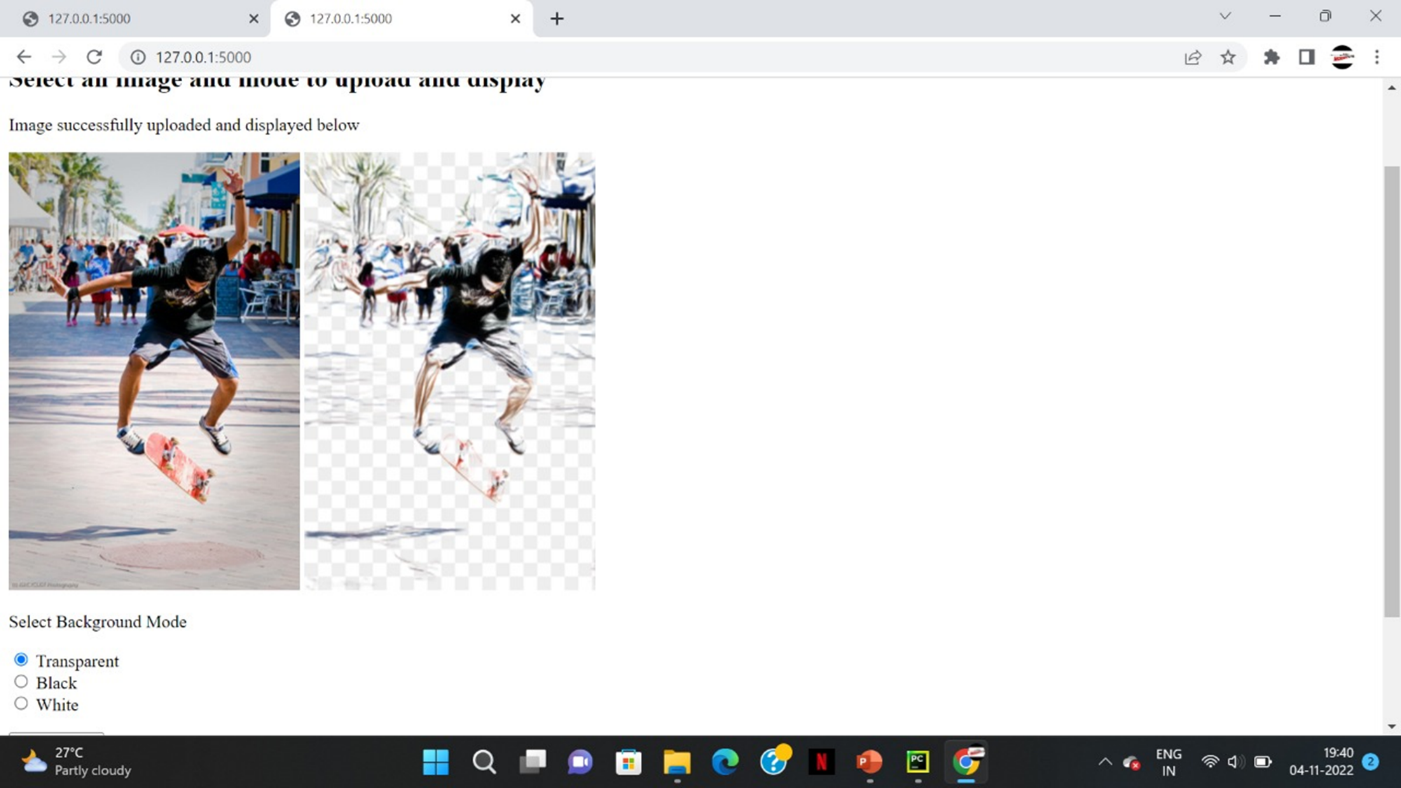
Screenshot 5.1: User page.



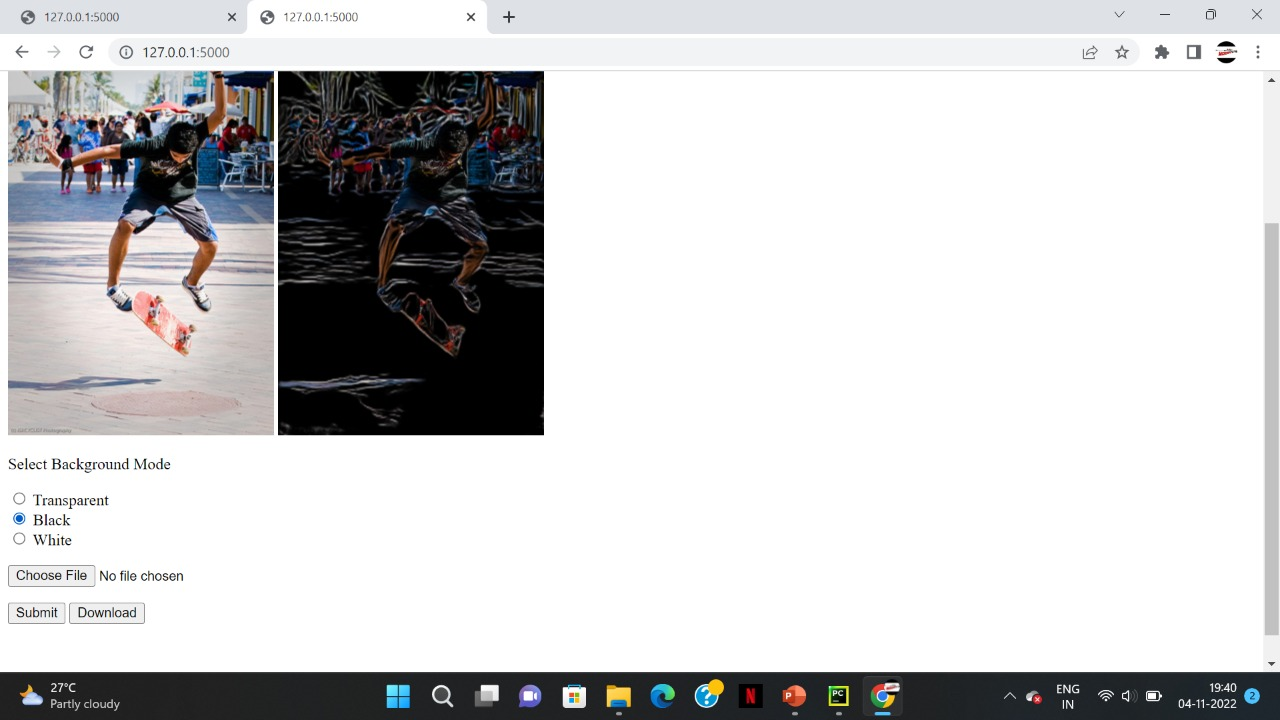
Screenshot 5.2:Selecting a picture.



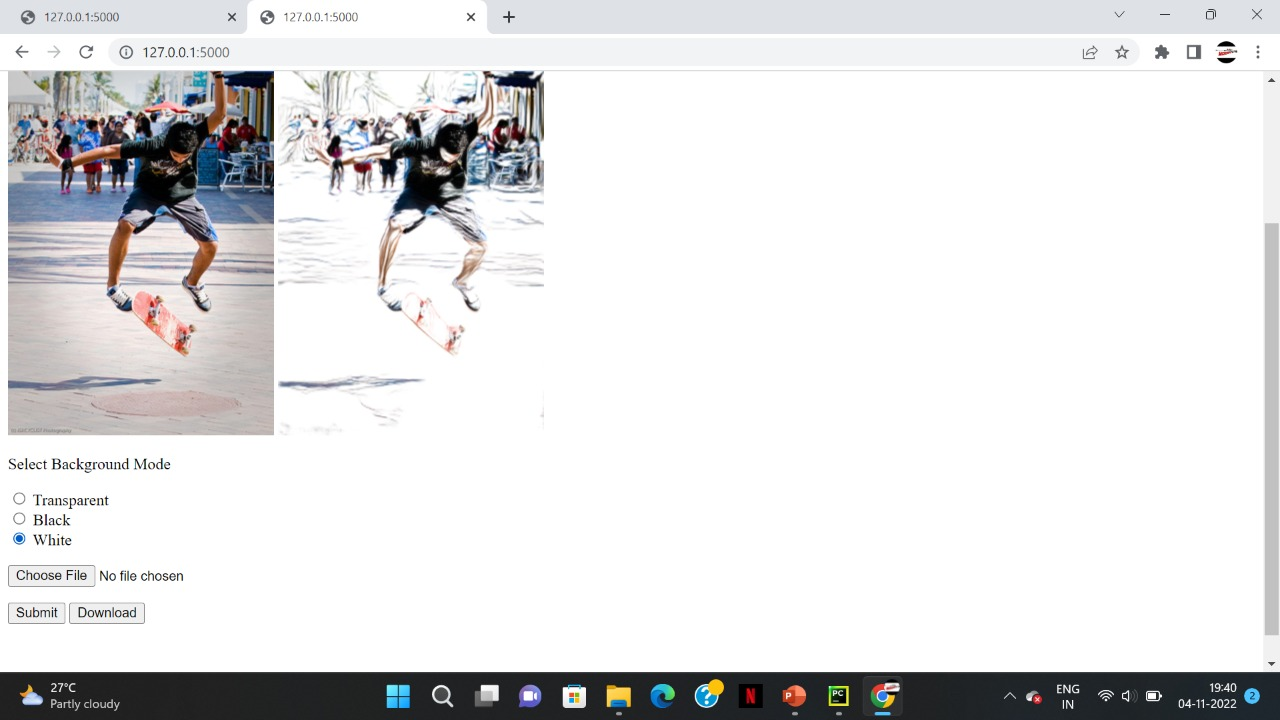
Screenshot 5.3:Uploading the picture.



Screenshot 5.4:Salient image with transparent background.



Screenshot 5.5: Salient image with black background.



Screenshot 5.6: Salient image with white background.

## 6.TESTING

#### 6.TESTING

##### INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

##### TYPES OF TESTING

* + 1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

* + 1. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

* + 1. FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input

: identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

##### TEST CASES

* + 1. CLASSIFICATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Test case name | Purpose | Input | Output |
| 1 | Saliency Detection | To detect Saliency | The user gives the input in the form of an image | An output is  Saliency  image. |

**7.CONCLUSION**

##### 7.CONCLUSION & FUTURE SCOPE

##### PROJECT CONCLUSION

The main is to propose a novel deep network: U2 - Net, for saliency detection. The main architecture of our U2 -Net is a two-level nested U-structure. The nested U structure with our newly designed RSU blocks enables the network to capture richer local and global information from both shallow and deep layers regardless of the resolutions. Compared with those saliency detection models built upon the existing backbones, our U2 -Net is purely built on the proposed RSU blocks which makes it possible to be trained from scratch and configured to have different model size according to the target environment constraints. We provide a full size U2 - Net (176.3 MB, 30 FPS) and a smaller size version U2 –Net (4.7 MB, 40 FPS) in this paper. Experimental results on six public salient object detection datasets demonstrate that both models achieve very competitive performance against other 20 state-of-the-art methods in terms of both qualitative and quantitative measures. Although our models achieve competitive results against other state-of-the-art methods, faster and smaller models are needed for computation and memory limited devices, such as mobile phones, robots, etc. In the near future, we will explore different techniques and architectures to further improve the speed and decrease the model size. In addition, larger diversified salient object datasets are needed to train more accurate and robust models.

##### FUTURE SCOPE

Although our models achieve competitive results against other state-of-the-art methods, faster and smaller models are needed for computation and memory limited devices, such as mobile phones, robots, etc. In the near future, we will explore different techniques and architectures to further improve the speed and decrease the model size. In addition, larger diversified salient object datasets are needed to train more accurate and robust models.

### 8.BIBLIOGRAPHY

##### 8. BIBLIOGRAPHY

##### REFERENCES

1. S. Li, M. Xu, Y. Ren, and Z. Wang, “Closed-form optimization on saliency-guided image compression for hevc-msp,” *IEEE Transactions on Multimedia*, 2017.
2. M. Ding and R.-F. Tong, “Content-aware copying and pasting in images,” *The Visual Computer*, vol. 26, no. 6-8, pp. 721–729, 2010.
3. Y. Fang, Z. Chen, W. Lin, and C. Lin, “Saliency detection in the compressed domain for adaptive image retargeting,” *IEEE transactions on image processing*, vol. 21, no. 9, pp. 3888–3901, 2012.
4. H. Wu, G. Li, and X. Luo, “Weighted attentional blocks for probabilistic object tracking,” *The Visual Computer*, vol. 30, no. 2, pp. 229–243, 2014.

##### GITHUB LINK

<https://github.com/sravya666/Face-recognition-audio-output/tree/master>