

**SRI VENKATESWARA INTERNSHIP PROGRAM
FOR RESEARCH IN ACADEMICS
(SRI-VIPRA)**

Project Report of 2022: SVP-2222

“App Designing using MATLAB & Python”



IQAC

Sri Venkateswara College

University of Delhi



Dhaulta Kuan

New Delhi -110021



SRIVIPRA PROJECT 2022

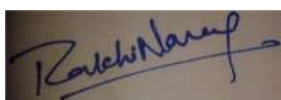
Title: App Designing using MATLAB & Python.

List of Mentors:

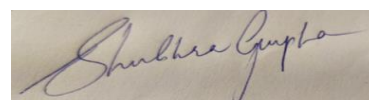
<p>Name of Mentor: Dr. Rakhi Narang Name of Department: Electronics Designation: Assistant Professor</p>	
<p>Name of Mentor: Ms. Shubhra Gupta Name of Department: Electronics Designation: Assistant Professor</p>	

List of Students:

S.No	Name of the student	Course	Photo
1	Sachin	B.Sc. (Hons.) Electronics Semester IV	
2	Kanishak	B.Sc. (Hons.) Electronics Semester IV	
3.	Praveen Regmi	B.Sc.(H) Electronics Semester IV	
4.	Sonakshi Singh	B.Sc.(H) Electronics Semester IV	
5.	Shobhraj Sharma	B.Sc.(H) Electronics Semester IV	
6.	Deepak Kumar	B.Sc. (Hons.) Physics Semester IV	



Signature of Mentor



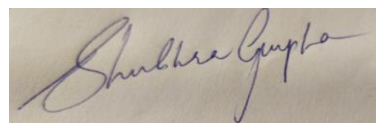
Signature of Mentor

Certificate

This is to certify that the aforementioned students from Sri Venkateswara College have participated in the summer project SVP-2222 titled “**App designing using MATLAB & Python**”. The participants have carried out the research project work under my guidance and supervision from 04/06/2022 to 14/11/2022. The work carried out is original and carried out in an offline mode.



Signature of Mentor



Signature of Mentor

Acknowledgment

On this great occasion of accomplishment of our project on **App Designing using MATLAB & Python**, we would like to sincerely express our gratitude to **Dr Rakhi Narang** and **Ms. Shubhra Gupta**, who has been supported through the completion of this project. As without their valuable knowledge and motivation this project would never be completed.

We would also be thankful to our principal **Prof. C. Sheela Reddy** of **Sri Venkateswara College**, for providing all the required facilities in completion of this project.

List of Contents:

Chapter	Name of Topic	Page No.
1.	MATLAB & App Designer.	7
2.	Simple & Scientific Calculator.	11
3.	Image Processing using MATLAB.	34
4.	Image Processing using Python.	50
5.	Conclusion	69
6.	References	70

ABSTRACT

The report will be containing the App Designing using MATLAB Software and Image Processing in MATLAB as well as Python. The first chapter of this report will be based on familiarisation with MATLAB. As MATLAB provides us the perfect APP Designing environment in there GUI interface it also have some basic functions that makes the app designing easier for beginners. After this the second chapter of this report will be dealing with more practical usage of that GUI by making a Basic Calculator APP and Scientific Calculator APP. The third chapter of this report is dealing with the Image Processing part using MATLAB Software as well as designing an APP for the Image Processing that takes images as input and provides the processed image at the output depending upon which function is chosen by the application user. The fourth chapter will be dealing with how image processing techniques can be used for detecting different 2-D shapes using python.

Chapter 1

MATLAB



MATLAB is a multi-paradigm programming language and numeric computing environment developed by **MathWorks**. **MATLAB** allows matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfacing with programs written in other languages. It is a programming platform specially designed for engineers and scientists to analyze data, develop algorithms and create models.

MATLAB was invented by mathematician and programmer **Cleve Moler**. In the beginning MATLAB was not a programming language, it was a simple interactive matrix calculator. There were no programs, no toolboxes, no graphics and no ODEs or FFTs.

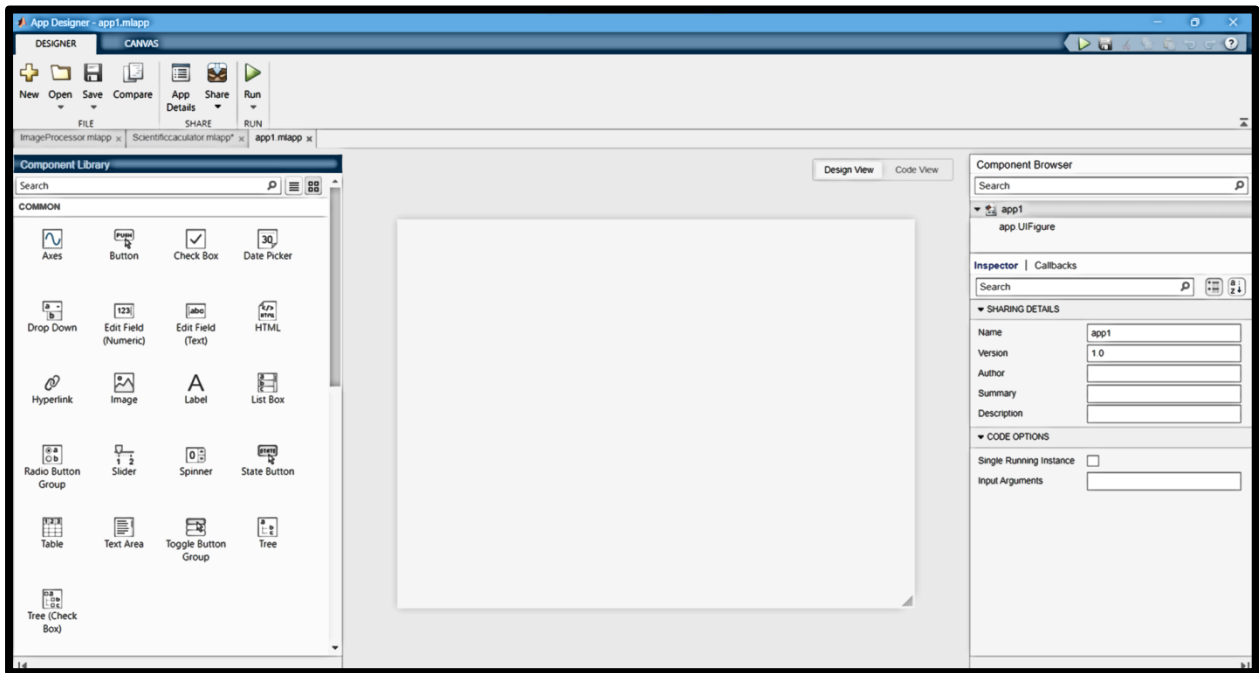
The first early version of MATLAB was completed in the late 1970s. The software was published for the first time in February 1979 at the Naval Postgraduate School in California. The early versions of MATLAB were simple Matrix calculators with 71 pre-built functions.

In 1980, **Cleve Moler** and **John N. Little** decided to reprogram MATLAB in C and market it for the IBM desktops that were replacing the mainframe computers at the time. John Little and programmer **Steve Bangert** re-programmed MATLAB in C, created the MATLAB programming language and developed features for toolboxes. In 2000, MathWorks added a Fortran based library for linear algebra in MATLAB 6, replacing the subroutines that were in C.

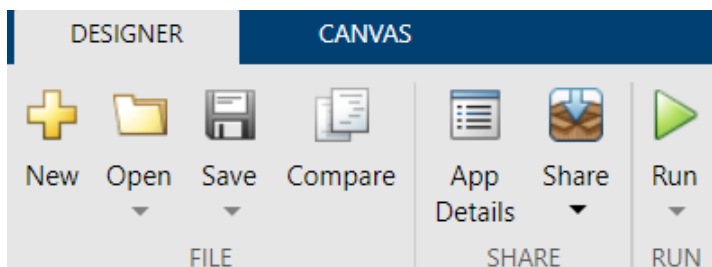
APP DESIGNER in MATLAB -

App Designer is an interactive development environment for designing an app layout and programming its behaviour. It provides a fully integrated version of MATLAB Editor and a large set of interactive UI components. It also offers a grid layout manager to organize user interface and automatic reflow options to make the app detect and respond to Change in the screen size. It is a next generation tool for creating and designing apps in MATLAB.

Interface of App Designer-



Basic Functions –



New: For creating new MLab file.

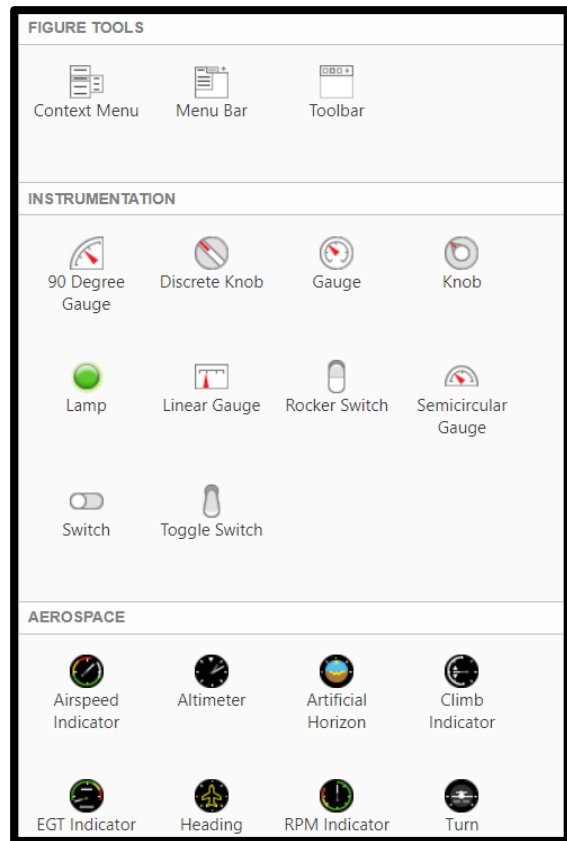
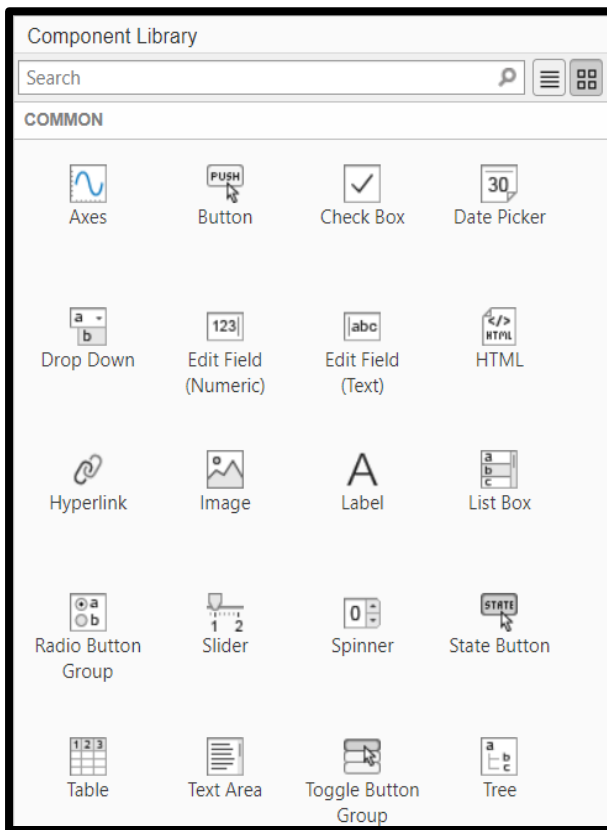
Open: For opening the apps which has been created as so far.

Save: For saving the file.

Run: To run the app designed.

App Details: Details of the app created.

Share: To share the file.



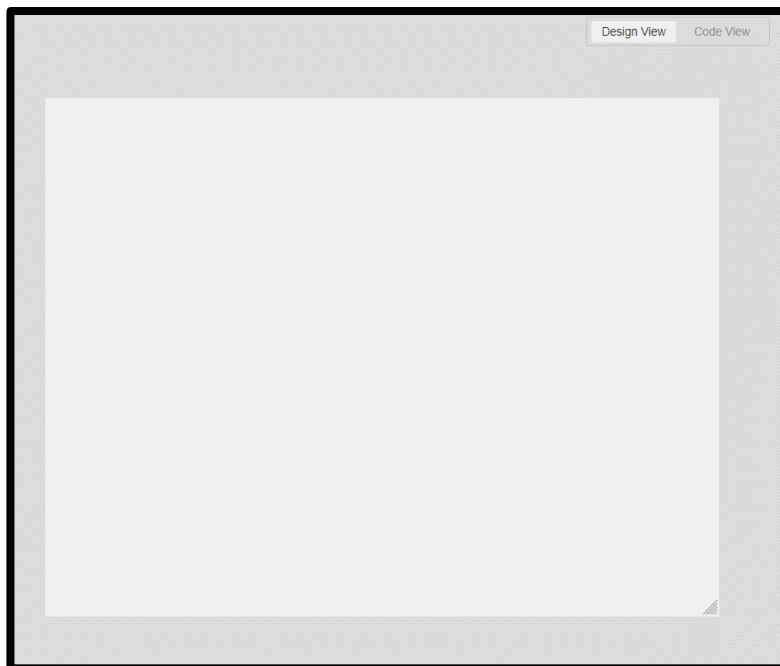
Component Library: Components used in designing the app can search here.

Common: Common components which are mostly used in designing the app.

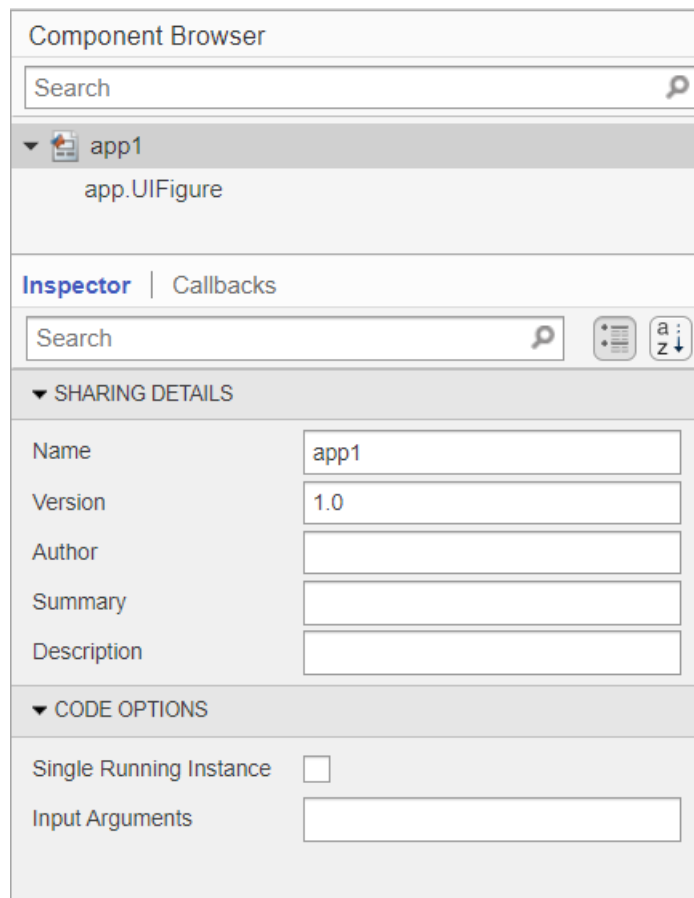
Figures: To display the figures mainly.

Instruments: Instruments which are inbuilt can be used for various purposes.

Aerospace: Inbuilt sensors that can be used in designing the apps like block detection on road from a moving car etc.



Here is the interface for designing the app. All the functions and components will appear here. Also, the code behind using the components will appear here.



Component browser: We can search the functions that we used in designing the interface of the app.

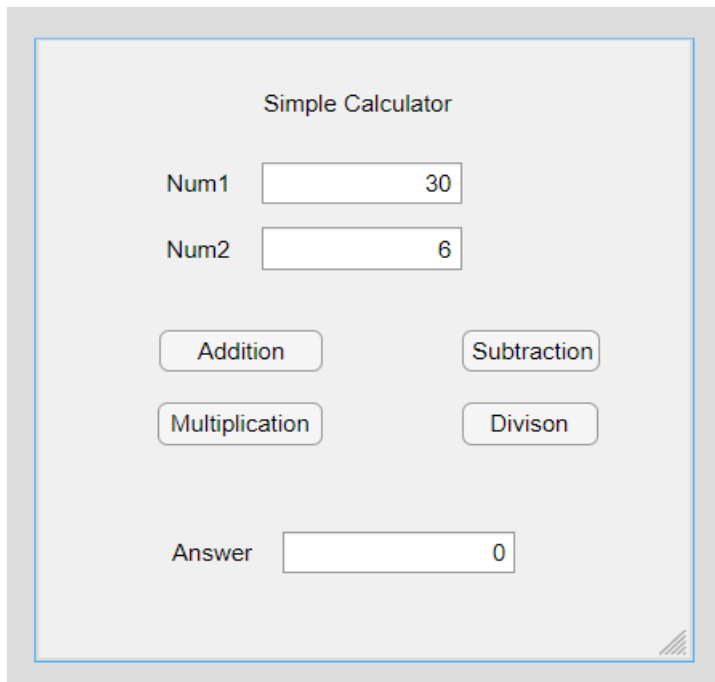
Inspector and Callbacks: Here the functions that can be used to design the interface of the app according to our choice like fonts color, size, look etc.

Chapter – 2

Simple Calculator:

We have designed a simple calculator to perform some simple tasks like Addition, Subtraction, Multiplication and Division.

Interface:



Here, we used three Edit Field(numeric) buttons for Num1, Num2 and Answer for displaying the input and output respectively.

One Add Label button and four pushed button for Addition, Subtraction, Multiplication and Division respectively.

Code –

```
classdef app2 < matlab.apps.AppBase
```

```
% Properties that correspond to app components
```

```
properties (Access = public)
```

```
    UIFigure          matlab.ui.Figure
```

```
    DivisonButton     matlab.ui.control.Button
```

```
    MultiplicationButton matlab.ui.control.Button
```

```
    SubtractionButton matlab.ui.control.Button
```

```
    SimpleCalculatorLabel matlab.ui.control.Label
```

```
    AnswerEditField   matlab.ui.control.NumericEditField
```

```
    AnswerEditFieldLabel matlab.ui.control.Label
```

```
    AdditionButton    matlab.ui.control.Button
```

```
    Num2EditField     matlab.ui.control.NumericEditField
```

```
    Num2EditFieldLabel matlab.ui.control.Label
```

```
    Num1EditField     matlab.ui.control.NumericEditField
```

```

Num1EditFieldLabel matlab.ui.control.Label
end

% Callbacks that handle component events
methods (Access = private)

% Button pushed function: AdditionButton
function AdditionButtonPushed(app, event)
    a=app.Num1EditField.Value;
    b=app.Num2EditField.Value;
    c=a+b;
    app.AnswerEditField.Value=c;
end

% Button pushed function: SubtractionButton
function SubtractionButtonPushed(app, event)
    a=app.Num1EditField.Value;
    b=app.Num2EditField.Value;
    c=a-b;
    app.AnswerEditField.Value=c;
end

% Button pushed function: MultiplicationButton
function MultiplicationButtonPushed(app, event)
    a=app.Num1EditField.Value;
    b=app.Num2EditField.Value;
    c=a*b;
    app.AnswerEditField.Value=c;
end

% Button pushed function: DivisonButton
function DivisonButtonPushed(app, event)
    a=app.Num1EditField.Value;
    b=app.Num2EditField.Value;
    c=a/b;
    app.AnswerEditField.Value=c;
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 325 321];
app.UIFigure.Name = 'MATLAB App';

```

```
% Create Num1EditFieldLabel
```

```
app.Num1EditFieldLabel = uilabel(app.UIFigure);  
app.Num1EditFieldLabel.HorizontalAlignment = 'right';  
app.Num1EditFieldLabel.Position = [59 237 38 22];  
app.Num1EditFieldLabel.Text = 'Num1';
```

```
% Create Num1EditField
```

```
app.Num1EditField = uieditfield(app.UIFigure, 'numeric');  
app.Num1EditField.Position = [112 237 100 22];  
app.Num1EditField.Value = 30;
```

```
% Create Num2EditFieldLabel
```

```
app.Num2EditFieldLabel = uilabel(app.UIFigure);  
app.Num2EditFieldLabel.HorizontalAlignment = 'right';  
app.Num2EditFieldLabel.Position = [59 203 38 22];  
app.Num2EditFieldLabel.Text = 'Num2';
```

```
% Create Num2EditField
```

```
app.Num2EditField = uieditfield(app.UIFigure, 'numeric');  
app.Num2EditField.Position = [112 203 100 22];  
app.Num2EditField.Value = 6;
```

```
% Create AdditionButton
```

```
app.AdditionButton = uibutton(app.UIFigure, 'push');  
app.AdditionButton.ButtonPushedFcn = createCallbackFcn(app, @AdditionButtonPushed, true);  
app.AdditionButton.Position = [61 150 82 22];  
app.AdditionButton.Text = 'Addition';
```

```
% Create AnswerEditFieldLabel
```

```
app.AnswerEditFieldLabel = uilabel(app.UIFigure);  
app.AnswerEditFieldLabel.HorizontalAlignment = 'right';  
app.AnswerEditFieldLabel.Position = [62 45 46 22];  
app.AnswerEditFieldLabel.Text = 'Answer';
```

```
% Create AnswerEditField
```

```
app.AnswerEditField = uieditfield(app.UIFigure, 'numeric');  
app.AnswerEditField.Position = [123 45 116 22];
```

```
% Create SimpleCalculatorLabel
```

```
app.SimpleCalculatorLabel = uilabel(app.UIFigure);  
app.SimpleCalculatorLabel.Position = [113 279 99 22];  
app.SimpleCalculatorLabel.Text = 'Simple Calculator';
```

```
% Create SubtractionButton
```

```
app.SubtractionButton = uibutton(app.UIFigure, 'push');
```

```
app.SubtractionButton.ButtonPushedFcn = createCallbackFcn(app, @SubtractionButtonPushed, true);
app.SubtractionButton.Position = [212 150 69 22];
app.SubtractionButton.Text = 'Subtraction';
```

```
% Create MultiplicationButton
```

```
app.MultiplicationButton = uibutton(app.UIFigure, 'push');
app.MultiplicationButton.ButtonPushedFcn = createCallbackFcn(app, @MultiplicationButtonPushed, true);
app.MultiplicationButton.Position = [60 112 83 22];
app.MultiplicationButton.Text = 'Multiplication';
```

```
% Create DivisonButton
```

```
app.DivisonButton = uibutton(app.UIFigure, 'push');
app.DivisonButton.ButtonPushedFcn = createCallbackFcn(app, @DivisonButtonPushed, true);
app.DivisonButton.Position = [212 112 68 22];
app.DivisonButton.Text = 'Divison';
```

```
% Show the figure after all components are created
```

```
app.UIFigure.Visible = 'on';
end
end
```

```
% App creation and deletion
```

```
methods (Access = public)
```

```
% Construct app
```

```
function app = app2
```

```
% Create UIFigure and components
```

```
createComponents(app)
```

```
% Register the app with App Designer
```

```
registerApp(app, app.UIFigure)
```

```
if nargin == 0
```

```
clear app
```

```
end
```

```
end
```

```
% Code that executes before app deletion
```

```
function delete(app)
```

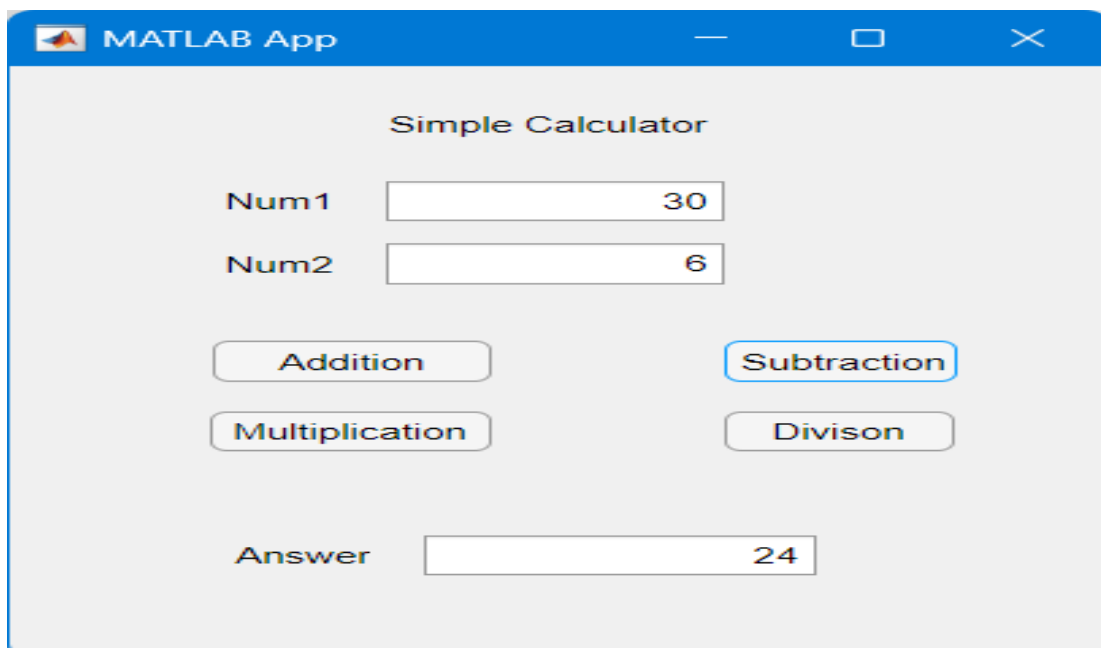
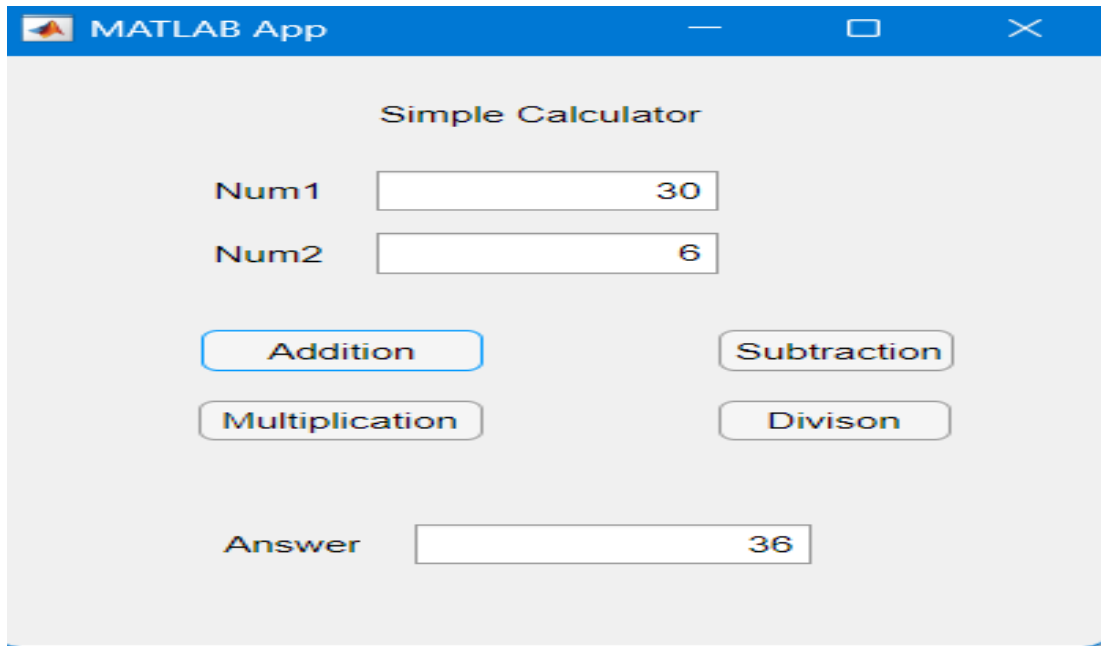
```
% Delete UIFigure when app is deleted
```

```
delete(app.UIFigure)
```

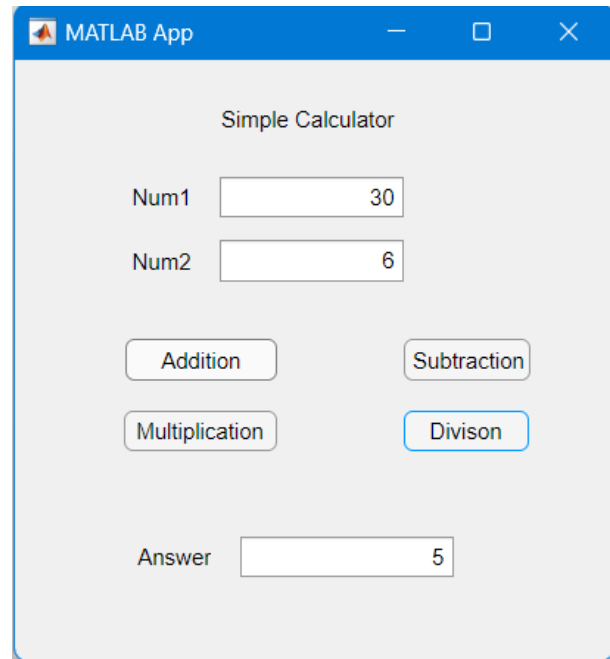
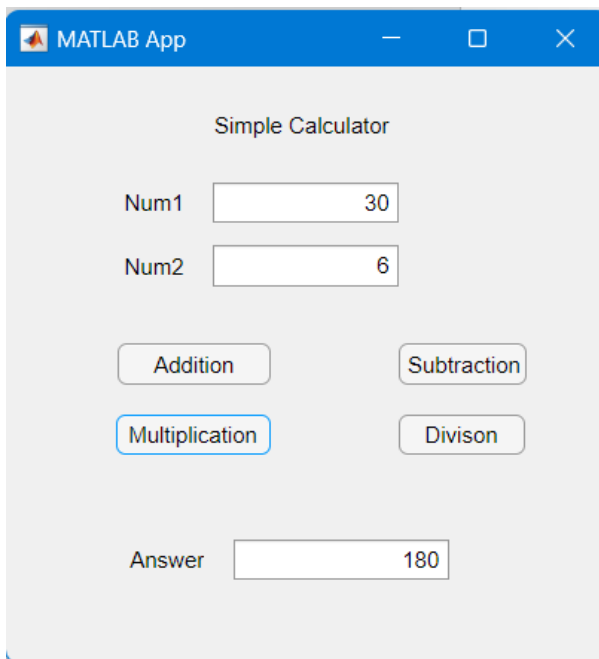
end
end
end

Tasks-

Addition & Subtraction –



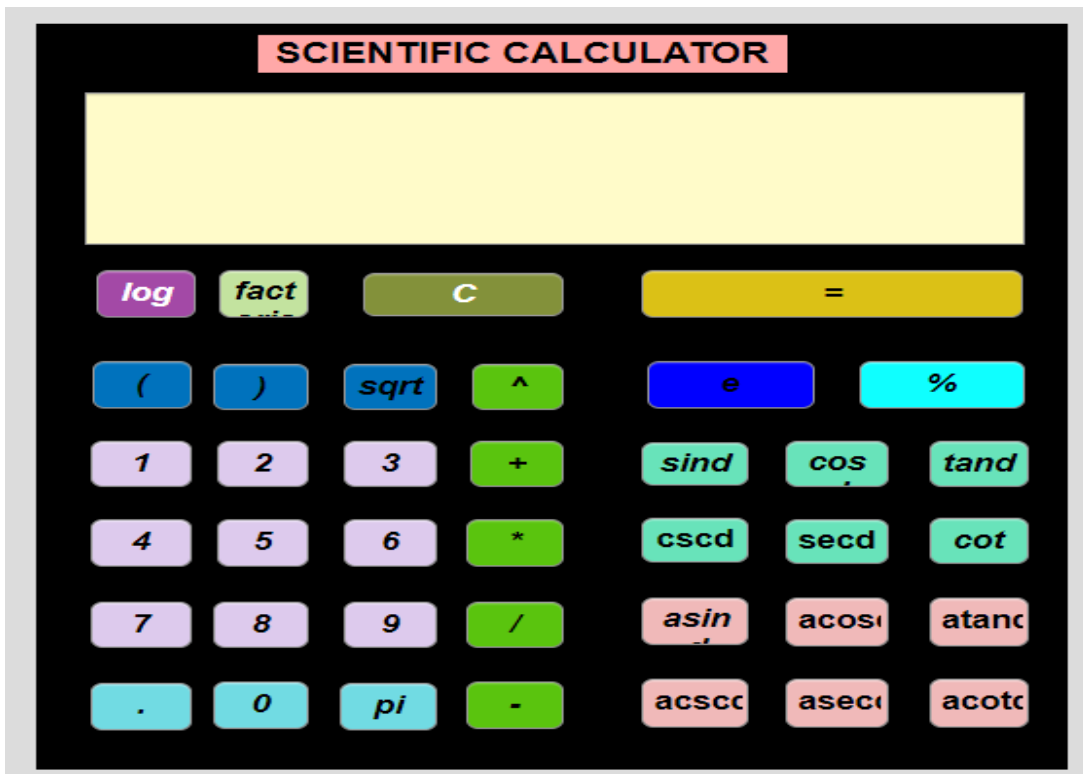
Multiplication & Division



Scientific calculator –

It is an advance calculator used to perform mathematical tasks like logarithm, integration, Differentiation, trigonometric, exponential modulus.

Interface:



Display Elements –

Scientific Calculator – Add Label button is used to give the title to the app we are designing.

Add Display – Axis button is used to display the calculations or tasks we are performing.

Others Buttons – So there are some other main buttons we have used that are – log(for taking logarithm), fact(for factorial), sqrt(for square root), sind, cosd (trigonometric functions in degree), 0 – 9(numeric digits, = (equal to), pi, dot, e(exponent) etc.

Code:

```
Classdef Scientificcalculator < matlab.apps.AppBase
```

```
    % Properties that correspond to app components  
    Properties (Access = private)
```

```
        UIFigure                matlab.ui.Figure  
        eButton                 matlab.ui.control.Button  
        acotdButton            matlab.ui.control.Button  
        asecdButton            matlab.ui.control.Button  
        acscdButton            matlab.ui.control.Button  
        atandButton            matlab.ui.control.Button  
        acosdButton            matlab.ui.control.Button  
        secdButton              matlab.ui.control.Button  
        cscdButton              matlab.ui.control.Button  
        Button_24               matlab.ui.control.Button  
        asindButton             matlab.ui.control.Button  
        piButton                matlab.ui.control.Button  
        factorialButton         matlab.ui.control.Button  
        Button_21               matlab.ui.control.Button  
        logButton               matlab.ui.control.Button  
        Button_20               matlab.ui.control.Button  
        sqrtButton              matlab.ui.control.Button  
        cotdButton              matlab.ui.control.Button  
        tandButton              matlab.ui.control.Button  
        cosdButton              matlab.ui.control.Button  
        Button_18               matlab.ui.control.Button  
        TextArea                matlab.ui.control.Button  
        Button_16               matlab.ui.control.Button  
        sindButton              matlab.ui.control.Button  
        Button_15               matlab.ui.control.Button  
        Button_14               matlab.ui.control.Button  
        Button_13               matlab.ui.control.Button  
        Button_23               matlab.ui.control.Button  
        CButton                 matlab.ui.control.Button  
        Button_10               matlab.ui.control.Button  
        Button_9                matlab.ui.control.Button  
        Button_8                matlab.ui.control.Button  
        Button_7                matlab.ui.control.Button  
        Button_6                matlab.ui.control.Button  
        Button_5                matlab.ui.control.Button
```

```

        Button_4           matlab.ui.control.Button
        Button_3           matlab.ui.control.Button
        Button_2           matlab.ui.control.Button
        Button             matlab.ui.control.Button
        SCIENTIFICCALCULATORLabel matlab.ui.control.Label
    end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: Button, Button_10, Button_2,
    % Button_24, Button_3, Button_4, Button_5, Button_6,
    % Button_7, Button_8, Button_9
    function ButtonPushed(app, event)
        app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
    end
    % Button pushed function: CButton
    function Buttondel(app, event)
        app.TextArea.Value = '';
    end

    % Button pushed function: Button_16
    function ButtonEqual(app, event)
        try
            app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text
            .....
            num2str(eval(app.TextArea.Value{1}))];
        catch
            app.TextArea.Value = 'Error';
        end
    end

    % Button pushed function: Button_13, Button_14, Button_15,
    % Button_23
    function ButtonMulti(app, event)
        app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
    end

    % Button pushed function: cosdButton, cotButton,
    % cscdButton, secdButton, sindButton, tandButton
    function ButtonSin(app, event)
        app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
    end

    % Button pushed function: Button_17, Button_18
    function Button123(app, event)
        app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
    end

% Callback function

```

```

function ButtonFun(app, event)
global shift
shift = 1;
end

% Callback function
function cosdButtonPushed(app, event)
app.TextArea.Value = 'cosd';
end

% Callback function
function tandButtonPushed(app, event)
app.TextArea.Value = 'tand';
end

% Callback function
function cotButtonPushed(app, event)
app.TextArea.Value = 'cotd';
end

% Button pushed function: sqrtButton
function ButtonSqrt(app, event)
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
end

% Callback function
function ButtonSquare(app, event)
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
end

% Button pushed function: Button_20
function ButtonDecimal(app, event)
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
end

% Callback function
function ButtonCuberoot(app, event)
app.TextArea.Value = 'cbrt';
end

% Button pushed function: logButton
function ButtonLog(app, event)
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
end

% Callback function
function Buttonantilog(app, event)
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];

```

```
end
```

```
% Button pushed function: Button_21, eButton, piButton  
function Buttonvalue(app, event)  
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];  
end
```

```
% Button pushed function: factorialButton  
function Buttonfactorial(app, event)  
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];  
end
```

```
% Button pushed function: acosdButton, acotdButton,  
% acscdButton, asecdButton, asindButton, atandButton  
function Buttoncosec(app, event)  
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];  
end
```

```
% Callback function  
function Buttonton(app, event)  
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];  
end
```

```
% Callback function  
function Buttonsec(app, event)  
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];  
end
```

```
% Callback function  
function Butttonx(app, event)  
app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];  
end  
end
```

```
% Component initialization  
methods (Access = private)
```

```
% Create UIFigure and components  
function createComponents(app)
```

```
% Create UIFigure and hide until all components are created  
app.UIFigure = uifigure('Visible', 'off');  
app.UIFigure.Color = [0 0 0];  
app.UIFigure.Colormap = [0.2431 0.149 0.6588;0.2431 0.1529 0.6745;0.2471  
0.1569 0.6863;0.2471 0.1608 0.698;0.251 0.1647 0.7059;0.251 0.1686  
0.7176;0.2549 0.1725 0.7294;0.2549 0.1765 0.7412;0.2588 0.1804 0.749;0.2588  
0.1843 0.7608;0.2627 0.1882 0.7725;0.2588 0.1882 0.7804;0.2627 0.1961  
0.7922;0.2667 0.2 0.8039;0.2667 0.2039 0.8157;0.2706 0.2078 0.8235;0.2706
```

0.2157 0.8353;0.2706 0.2196 0.8431;0.2745 0.2235 0.851;0.2745 0.2275
0.8627;0.2745 0.2314 0.8706;0.2745 0.2392 0.8784;0.2784 0.2431 0.8824;0.2784
0.2471 0.8902;0.2784 0.2549 0.898;0.2784 0.2588 0.902;0.2784 0.2667
0.9098;0.2784 0.2706 0.9137;0.2784 0.2745 0.9216;0.2824 0.2824 0.9255;0.2824
0.2863 0.9294;0.2824 0.2941 0.9333;0.2824 0.298 0.9412;0.2824 0.3059
0.9451;0.2824 0.3098 0.949;0.2824 0.3137 0.9529;0.2824 0.3216 0.9569;0.2824
0.3255 0.9608;0.2824 0.3294 0.9647;0.2784 0.3373 0.9686;0.2784 0.3412
0.9686;0.2784 0.349 0.9725;0.2784 0.3529 0.9765;0.2784 0.3569 0.9804;0.2784
0.3647 0.9804;0.2745 0.3686 0.9843;0.2745 0.3765 0.9843;0.2745 0.3804
0.9882;0.2706 0.3843 0.9882;0.2706 0.3922 0.9922;0.2667 0.3961 0.9922;0.2627
0.4039 0.9922;0.2627 0.4078 0.9961;0.2588 0.4157 0.9961;0.2549 0.4196
0.9961;0.251 0.4275 0.9961;0.2471 0.4314 1;0.2431 0.4392 1;0.2353 0.4431
1;0.2314 0.451 1;0.2235 0.4549 1;0.2196 0.4627 0.9961;0.2118 0.4667
0.9961;0.2078 0.4745 0.9922;0.2 0.4784 0.9922;0.1961 0.4863 0.9882;0.1922
0.4902 0.9882;0.1882 0.498 0.9843;0.1843 0.502 0.9804;0.1843 0.5098
0.9804;0.1804 0.5137 0.9765;0.1804 0.5176 0.9725;0.1804 0.5255 0.9725;0.1804
0.5294 0.9686;0.1765 0.5333 0.9647;0.1765 0.5412 0.9608;0.1765 0.5451
0.9569;0.1765 0.549 0.9529;0.1765 0.5569 0.949;0.1725 0.5608 0.9451;0.1725
0.5647 0.9412;0.1686 0.5686 0.9373;0.1647 0.5765 0.9333;0.1608 0.5804
0.9294;0.1569 0.5843 0.9255;0.1529 0.5922 0.9216;0.1529 0.5961 0.9176;0.149
0.6 0.9137;0.149 0.6039 0.9098;0.1451 0.6078 0.9098;0.1451 0.6118
0.9059;0.1412 0.6196 0.902;0.1412 0.6235 0.898;0.1373 0.6275 0.898;0.1373
0.6314 0.8941;0.1333 0.6353 0.8941;0.1294 0.6392 0.8902;0.1255 0.6471
0.8902;0.1216 0.651 0.8863;0.1176 0.6549 0.8824;0.1137 0.6588 0.8824;0.1137
0.6627 0.8784;0.1098 0.6667 0.8745;0.1059 0.6706 0.8706;0.102 0.6745
0.8667;0.098 0.6784 0.8627;0.0902 0.6824 0.8549;0.0863 0.6863 0.851;0.0784
0.6902 0.8471;0.0706 0.6941 0.8392;0.0627 0.698 0.8353;0.0549 0.702
0.8314;0.0431 0.702 0.8235;0.0314 0.7059 0.8196;0.0235 0.7098 0.8118;0.0157
0.7137 0.8078;0.0078 0.7176 0.8;0.0039 0.7176 0.7922;0 0.7216 0.7882;0 0.7255
0.7804;0 0.7294 0.7765;0.0039 0.7294 0.7686;0.0078 0.7333 0.7608;0.0157 0.7333
0.7569;0.0235 0.7373 0.749;0.0353 0.7412 0.7412;0.051 0.7412 0.7373;0.0627
0.7451 0.7294;0.0784 0.7451 0.7216;0.0902 0.749 0.7137;0.102 0.7529
0.7098;0.1137 0.7529 0.702;0.1255 0.7569 0.6941;0.1373 0.7569 0.6863;0.1451
0.7608 0.6824;0.1529 0.7608 0.6745;0.1608 0.7647 0.6667;0.1686 0.7647
0.6588;0.1725 0.7686 0.651;0.1804 0.7686 0.6471;0.1843 0.7725 0.6392;0.1922
0.7725 0.6314;0.1961 0.7765 0.6235;0.2 0.7804 0.6157;0.2078 0.7804
0.6078;0.2118 0.7843 0.6;0.2196 0.7843 0.5882;0.2235 0.7882 0.5804;0.2314
0.7882 0.5725;0.2392 0.7922 0.5647;0.251 0.7922 0.5529;0.2588 0.7922
0.5451;0.2706 0.7961 0.5373;0.2824 0.7961 0.5255;0.2941 0.7961 0.5176;0.3059
0.8 0.5059;0.3176 0.8 0.498;0.3294 0.8 0.4863;0.3412 0.8 0.4784;0.3529 0.8
0.4667;0.3686 0.8039 0.4549;0.3804 0.8039 0.4471;0.3922 0.8039 0.4353;0.4039
0.8039 0.4235;0.4196 0.8039 0.4118;0.4314 0.8039 0.4;0.4471 0.8039
0.3922;0.4627 0.8 0.3804;0.4745 0.8 0.3686;0.4902 0.8 0.3569;0.5059 0.8
0.349;0.5176 0.8 0.3373;0.5333 0.7961 0.3255;0.5451 0.7961 0.3176;0.5608
0.7961 0.3059;0.5765 0.7922 0.2941;0.5882 0.7922 0.2824;0.6039 0.7882
0.2745;0.6157 0.7882 0.2627;0.6314 0.7843 0.251;0.6431 0.7843 0.2431;0.6549
0.7804 0.2314;0.6706 0.7804 0.2235;0.6824 0.7765 0.2157;0.698 0.7765
0.2078;0.7098 0.7725 0.2;0.7216 0.7686 0.1922;0.7333 0.7686 0.1843;0.7451
0.7647 0.1765;0.7608 0.7647 0.1725;0.7725 0.7608 0.1647;0.7843 0.7569
0.1608;0.7961 0.7569 0.1569;0.8078 0.7529 0.1529;0.8157 0.749 0.1529;0.8275
0.749 0.1529;0.8392 0.7451 0.1529;0.851 0.7451 0.1569;0.8588 0.7412
0.1569;0.8706 0.7373 0.1608;0.8824 0.7373 0.1647;0.8902 0.7373 0.1686;0.902
0.7333 0.1765;0.9098 0.7333 0.1804;0.9176 0.7294 0.1882;0.9255 0.7294
0.1961;0.9373 0.7294 0.2078;0.9451 0.7294 0.2157;0.9529 0.7294 0.2235;0.9608
0.7294 0.2314;0.9686 0.7294 0.2392;0.9765 0.7294 0.2431;0.9843 0.7333
0.2431;0.9882 0.7373 0.2431;0.9961 0.7412 0.2392;0.9961 0.7451 0.2353;0.9961

```

0.7529 0.2314;0.9961 0.7569 0.2275;0.9961 0.7608 0.2235;0.9961 0.7686
0.2196;0.9961 0.7725 0.2157;0.9961 0.7804 0.2078;0.9961 0.7843 0.2039;0.9961
0.7922 0.2;0.9922 0.7961 0.1961;0.9922 0.8039 0.1922;0.9922 0.8078
0.1922;0.9882 0.8157 0.1882;0.9843 0.8235 0.1843;0.9843 0.8275 0.1804;0.9804
0.8353 0.1804;0.9765 0.8392 0.1765;0.9765 0.8471 0.1725;0.9725 0.851
0.1686;0.9686 0.8588 0.1647;0.9686 0.8667 0.1647;0.9647 0.8706 0.1608;0.9647
0.8784 0.1569;0.9608 0.8824 0.1569;0.9608 0.8902 0.1529;0.9608 0.898
0.149;0.9608 0.902 0.149;0.9608 0.9098 0.1451;0.9608 0.9137 0.1412;0.9608
0.9216 0.1373;0.9608 0.9255 0.1333;0.9608 0.9333 0.1294;0.9647 0.9373
0.1255;0.9647 0.9451 0.1216;0.9647 0.949 0.1176;0.9686 0.9569 0.1098;0.9686
0.9608 0.1059;0.9725 0.9686 0.102;0.9725 0.9725 0.0941;0.9765 0.9765
0.0863;0.9765 0.9843 0.0824;1 1 1];
app.UIFigure.Position = [100 100 424 426];
app.UIFigure.Name = 'MATLAB App';

```

% Create SCIENTIFICCALCULATORLabel

```

app.SCIENTIFICCALCULATORLabel = uilabel(app.UIFigure);
app.SCIENTIFICCALCULATORLabel.BackgroundColor = [1 0.6588 0.6588];
app.SCIENTIFICCALCULATORLabel.HorizontalAlignment = 'center';
app.SCIENTIFICCALCULATORLabel.WordWrap = 'on';
app.SCIENTIFICCALCULATORLabel.FontSize = 16;
app.SCIENTIFICCALCULATORLabel.FontWeight = 'bold';
app.SCIENTIFICCALCULATORLabel.Position = [92 399 218 22];
app.SCIENTIFICCALCULATORLabel.Text = 'SCIENTIFIC CALCULATOR';

```

% Create Button

```

app.Button = uibutton(app.UIFigure, 'push');
app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button.WordWrap = 'on';
app.Button.BackgroundColor = [0.8667 0.7922 0.9294];
app.Button.FontSize = 14;
app.Button.FontWeight = 'bold';
app.Button.FontAngle = 'italic';
app.Button.Position = [23 162 42 27];
app.Button.Text = '1';

```

% Create Button_2

```

app.Button_2 = uibutton(app.UIFigure, 'push');
app.Button_2.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_2.WordWrap = 'on';
app.Button_2.BackgroundColor = [0.8667 0.7922 0.9294];
app.Button_2.FontSize = 14;
app.Button_2.FontWeight = 'bold';
app.Button_2.FontAngle = 'italic';
app.Button_2.Position = [75 162 38 27];
app.Button_2.Text = '2';

```

% Create Button_3

```

app.Button_3 = uibutton(app.UIFigure, 'push');
app.Button_3.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_3.WordWrap = 'on';
app.Button_3.BackgroundColor = [0.8667 0.7922 0.9294];

```

```

app.Button_3.FontSize = 14;
app.Button_3.FontWeight = 'bold';
app.Button_3.FontAngle = 'italic';
app.Button_3.Position = [127 162 39 27];
app.Button_3.Text = '3';

% Create Button_4
app.Button_4 = uibutton(app.UIFigure, 'push');
app.Button_4.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_4.WordWrap = 'on';
app.Button_4.BackgroundColor = [0.8667 0.7922 0.9294];
app.Button_4.FontSize = 14;
app.Button_4.FontWeight = 'bold';
app.Button_4.FontAngle = 'italic';
app.Button_4.Position = [23 117 42 27];
app.Button_4.Text = '4';

% Create Button_5
app.Button_5 = uibutton(app.UIFigure, 'push');
app.Button_5.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_5.WordWrap = 'on';
app.Button_5.BackgroundColor = [0.8667 0.7922 0.9294];
app.Button_5.FontSize = 14;
app.Button_5.FontWeight = 'bold';
app.Button_5.FontAngle = 'italic';
app.Button_5.Position = [75 117 38 27];
app.Button_5.Text = '5';

% Create Button_6
app.Button_6 = uibutton(app.UIFigure, 'push');
app.Button_6.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_6.WordWrap = 'on';
app.Button_6.BackgroundColor = [0.8667 0.7922 0.9294];
app.Button_6.FontSize = 14;
app.Button_6.FontWeight = 'bold';
app.Button_6.FontAngle = 'italic';
app.Button_6.Position = [127 117 39 27];
app.Button_6.Text = '6';

% Create Button_7
app.Button_7 = uibutton(app.UIFigure, 'push');
app.Button_7.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_7.WordWrap = 'on';
app.Button_7.BackgroundColor = [0.8667 0.7922 0.9294];
app.Button_7.FontSize = 14;
app.Button_7.FontWeight = 'bold';
app.Button_7.FontAngle = 'italic';
app.Button_7.Position = [23 70 42 27];
app.Button_7.Text = '7';

% Create Button_8

```



```

app.Button_8 = uibutton(app.UIFigure, 'push');
app.Button_8.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_8.WordWrap = 'on';
app.Button_8.BackgroundColor = [0.8667 0.7922 0.9294];
app.Button_8.FontSize = 14;
app.Button_8.FontWeight = 'bold';
app.Button_8.FontAngle = 'italic';
app.Button_8.Position = [74 70 39 27];
app.Button_8.Text = '8';

```

% Create Button_9

```

app.Button_9 = uibutton(app.UIFigure, 'push');
app.Button_9.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_9.WordWrap = 'on';
app.Button_9.BackgroundColor = [0.8667 0.7922 0.9294];
app.Button_9.FontSize = 14;
app.Button_9.FontWeight = 'bold';
app.Button_9.FontAngle = 'italic';
app.Button_9.Position = [127 70 39 27];
app.Button_9.Text = '9';

```

% Create Button_10

```

app.Button_10 = uibutton(app.UIFigure, 'push');
app.Button_10.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_10.WordWrap = 'on';
app.Button_10.BackgroundColor = [0.4431 0.8588 0.8902];
app.Button_10.FontSize = 14;
app.Button_10.FontWeight = 'bold';
app.Button_10.FontAngle = 'italic';
app.Button_10.Position = [74 24 39 27];
app.Button_10.Text = '0';

```

% Create CButton

```

app.CButton = uibutton(app.UIFigure, 'push');
app.CButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonDelete, true);
app.CButton.WordWrap = 'on';
app.CButton.BackgroundColor = [0.5137 0.5686 0.2275];
app.CButton.FontSize = 14;
app.CButton.FontWeight = 'bold';
app.CButton.FontAngle = 'italic';
app.CButton.FontColor = [1 1 1];
app.CButton.Position = [135 260 83 25];
app.CButton.Text = 'C';

```

% Create Button_23

```

app.Button_23 = uibutton(app.UIFigure, 'push');
app.Button_23.ButtonPushedFcn = createCallbackFcn(app, @ButtonMulti, true);
app.Button_23.WordWrap = 'on';
app.Button_23.BackgroundColor = [0.3529 0.7686 0.0549];
app.Button_23.FontSize = 14;
app.Button_23.FontWeight = 'bold';
app.Button_23.FontAngle = 'italic';

```

```

app.Button_23.Position = [178 117 40 27];
app.Button_23.Text = '*';

% Create Button_13
app.Button_13 = uibutton(app.UIFigure, 'push');
app.Button_13.ButtonPushedFcn = createCallbackFcn(app, @ButtonMulti, true);
app.Button_13.WordWrap = 'on';
app.Button_13.BackgroundColor = [0.3529 0.7686 0.0549];
app.Button_13.FontSize = 14;
app.Button_13.FontWeight = 'bold';
app.Button_13.FontAngle = 'italic';
app.Button_13.Position = [178 70 40 27];
app.Button_13.Text = '/';

% Create Button_14
app.Button_14 = uibutton(app.UIFigure, 'push');
app.Button_14.ButtonPushedFcn = createCallbackFcn(app, @ButtonMulti, true);
app.Button_14.WordWrap = 'on';
app.Button_14.BackgroundColor = [0.3529 0.7686 0.0549];
app.Button_14.FontSize = 14;
app.Button_14.FontWeight = 'bold';
app.Button_14.FontAngle = 'italic';
app.Button_14.Position = [179 162 39 27];
app.Button_14.Text = '+';

% Create Button_15
app.Button_15 = uibutton(app.UIFigure, 'push');
app.Button_15.ButtonPushedFcn = createCallbackFcn(app, @ButtonMulti, true);
app.Button_15.WordWrap = 'on';
app.Button_15.BackgroundColor = [0.3529 0.7686 0.0549];
app.Button_15.FontSize = 14;
app.Button_15.FontWeight = 'bold';
app.Button_15.FontAngle = 'italic';
app.Button_15.Position = [178 24 40 27];
app.Button_15.Text = '-';

% Create sindButton
app.sindButton = uibutton(app.UIFigure, 'push');
app.sindButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonSin, true);
app.sindButton.WordWrap = 'on';
app.sindButton.BackgroundColor = [0.4078 0.8902 0.7294];
app.sindButton.FontSize = 14;
app.sindButton.FontWeight = 'bold';
app.sindButton.FontAngle = 'italic';
app.sindButton.Position = [250 163 44 25];
app.sindButton.Text = 'sind';

% Create Button_16
app.Button_16 = uibutton(app.UIFigure, 'push');
app.Button_16.ButtonPushedFcn = createCallbackFcn(app, @ButtonEqual, true);
app.Button_16.WordWrap = 'on';

```

```

app.Button_16.BackgroundColor = [0.8588 0.7569 0.0863];
app.Button_16.FontSize = 14;
app.Button_16.FontWeight = 'bold';
app.Button_16.FontAngle = 'italic';
app.Button_16.Position = [250 259 157 27];
app.Button_16.Text = '=';

% Create TextArea
app.TextArea = uitextarea(app.UIFigure);
app.TextArea.HorizontalAlignment = 'right';
app.TextArea.FontSize = 14;
app.TextArea.FontWeight = 'bold';
app.TextArea.FontAngle = 'italic';
app.TextArea.FontColor = [1 1 1];
app.TextArea.BackgroundColor = [1 0.9843 0.7882];
app.TextArea.Position = [21 301 386 87];

% Create Button_17
app.Button_17 = uibutton(app.UIFigure, 'push');
app.Button_17.ButtonPushedFcn = createCallbackFcn(app, @Button123, true);
app.Button_17.WordWrap = 'on';
app.Button_17.BackgroundColor = [0 0.4471 0.7412];
app.Button_17.FontSize = 14;
app.Button_17.FontWeight = 'bold';
app.Button_17.FontAngle = 'italic';
app.Button_17.Position = [24 207 41 27];
app.Button_17.Text = '(';

% Create Button_18
app.Button_18 = uibutton(app.UIFigure, 'push');
app.Button_18.ButtonPushedFcn = createCallbackFcn(app, @Button123, true);
app.Button_18.WordWrap = 'on';
app.Button_18.BackgroundColor = [0 0.4471 0.7412];
app.Button_18.FontSize = 14;
app.Button_18.FontWeight = 'bold';
app.Button_18.FontAngle = 'italic';
app.Button_18.Position = [74 206 39 27];
app.Button_18.Text = ')';

% Create cosdButton
app.cosdButton = uibutton(app.UIFigure, 'push');
app.cosdButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonSin, true);
app.cosdButton.WordWrap = 'on';
app.cosdButton.BackgroundColor = [0.4078 0.8902 0.7294];
app.cosdButton.FontSize = 14;
app.cosdButton.FontWeight = 'bold';
app.cosdButton.FontAngle = 'italic';
app.cosdButton.Position = [309 162 42 27];
app.cosdButton.Text = 'cosd';

% Create tandButton

```

```

app.tandButton = uibutton(app.UIFigure, 'push');
app.tandButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonSin, true);
app.tandButton.WordWrap = 'on';
app.tandButton.BackgroundColor = [0.4078 0.8902 0.7294];
app.tandButton.FontSize = 14;
app.tandButton.FontWeight = 'bold';
app.tandButton.FontAngle = 'italic';
app.tandButton.Position = [368 162 41 27];
app.tandButton.Text = 'tand';

% Create cotButton
app.cotButton = uibutton(app.UIFigure, 'push');
app.cotButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonSin, true);
app.cotButton.WordWrap = 'on';
app.cotButton.BackgroundColor = [0.4078 0.8902 0.7294];
app.cotButton.FontSize = 14;
app.cotButton.FontWeight = 'bold';
app.cotButton.FontAngle = 'italic';
app.cotButton.Position = [368 118 41 27];
app.cotButton.Text = 'cot';

% Create sqrtButton
app.sqrtButton = uibutton(app.UIFigure, 'push');
app.sqrtButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonSqrt, true);
app.sqrtButton.WordWrap = 'on';
app.sqrtButton.BackgroundColor = [0 0.4471 0.7412];
app.sqrtButton.FontSize = 14;
app.sqrtButton.FontWeight = 'bold';
app.sqrtButton.FontAngle = 'italic';
app.sqrtButton.Position = [127 206 39 27];
app.sqrtButton.Text = 'sqrt';

% Create Button_20
app.Button_20 = uibutton(app.UIFigure, 'push');
app.Button_20.ButtonPushedFcn = createCallbackFcn(app, @ButtonDecimal, true);
app.Button_20.WordWrap = 'on';
app.Button_20.BackgroundColor = [0.4431 0.8588 0.8902];
app.Button_20.FontSize = 14;
app.Button_20.FontWeight = 'bold';
app.Button_20.FontAngle = 'italic';
app.Button_20.Position = [23 23 42 27];
app.Button_20.Text = '.';

% Create logButton
app.logButton = uibutton(app.UIFigure, 'push');
app.logButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonLog, true);
app.logButton.WordWrap = 'on';
app.logButton.BackgroundColor = [0.6392 0.2863 0.651];
app.logButton.FontSize = 14;
app.logButton.FontWeight = 'bold';
app.logButton.FontAngle = 'italic';
app.logButton.FontColor = [1 1 1];

```

```

app.logButton.Position = [26 259 41 27];
app.logButton.Text = 'log';

% Create eButton
app.eButton = uibutton(app.UIFigure, 'push');
app.eButton.ButtonPushedFcn = createCallbackFcn(app, @Buttonvalue, true);
app.eButton.WordWrap = 'on';
app.eButton.BackgroundColor = [0 0 1];
app.eButton.FontSize = 14;
app.eButton.FontWeight = 'bold';
app.eButton.FontAngle = 'italic';
app.eButton.Position = [252 207 69 27];
app.eButton.Text = 'e';

% Create Button_21
app.Button_21 = uibutton(app.UIFigure, 'push');
app.Button_21.ButtonPushedFcn = createCallbackFcn(app, @Buttonvalue, true);
app.Button_21.WordWrap = 'on';
app.Button_21.BackgroundColor = [0.3529 0.7686 0.0549];
app.Button_21.FontSize = 14;
app.Button_21.FontWeight = 'bold';
app.Button_21.FontAngle = 'italic';
app.Button_21.Position = [180 206 38 27];
app.Button_21.Text = '^';

% Create factorialButton
app.factorialButton = uibutton(app.UIFigure, 'push');
app.factorialButton.ButtonPushedFcn = createCallbackFcn(app, @Buttonfactorial,
true);
app.factorialButton.WordWrap = 'on';
app.factorialButton.BackgroundColor = [0.7647 0.8902 0.6235];
app.factorialButton.FontSize = 14;
app.factorialButton.FontWeight = 'bold';
app.factorialButton.FontAngle = 'italic';
app.factorialButton.Position = [76 259 37 27];
app.factorialButton.Text = 'factorial';

% Create piButton
app.piButton = uibutton(app.UIFigure, 'push');
app.piButton.ButtonPushedFcn = createCallbackFcn(app, @Buttonvalue, true);
app.piButton.WordWrap = 'on';
app.piButton.BackgroundColor = [0.4431 0.8588 0.8902];
app.piButton.FontSize = 14;
app.piButton.FontWeight = 'bold';
app.piButton.FontAngle = 'italic';
app.piButton.Position = [126 23 40 27];
app.piButton.Text = 'pi';

% Create asindButton
app.asindButton = uibutton(app.UIFigure, 'push');
app.asindButton.ButtonPushedFcn = createCallbackFcn(app, @Buttoncosec, true);

```

```

app.asindButton.WordWrap = 'on';
app.asindButton.BackgroundColor = [0.9412 0.7255 0.7255];
app.asindButton.FontSize = 14;
app.asindButton.FontWeight = 'bold';
app.asindButton.FontAngle = 'italic';
app.asindButton.Position = [250 72 44 27];
app.asindButton.Text = 'asind';

% Create Button_24
app.Button_24 = uibutton(app.UIFigure, 'push');
app.Button_24.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button_24.WordWrap = 'on';
app.Button_24.BackgroundColor = [0.0588 1 1];
app.Button_24.FontSize = 14;
app.Button_24.FontWeight = 'bold';
app.Button_24.FontAngle = 'italic';
app.Button_24.Position = [339 207 68 27];
app.Button_24.Text = '%';

% Create cscdButton
app.cscdButton = uibutton(app.UIFigure, 'push');
app.cscdButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonSin, true);
app.cscdButton.BackgroundColor = [0.4078 0.8902 0.7294];
app.cscdButton.FontSize = 14;
app.cscdButton.FontWeight = 'bold';
app.cscdButton.Position = [250 119 44 26];
app.cscdButton.Text = 'cscd';

% Create secdbutton
app.secdButton = uibutton(app.UIFigure, 'push');
app.secdButton.ButtonPushedFcn = createCallbackFcn(app, @ButtonSin, true);
app.secdButton.BackgroundColor = [0.4078 0.8902 0.7294];
app.secdButton.FontSize = 14;
app.secdButton.FontWeight = 'bold';
app.secdButton.Position = [309 118 42 26];
app.secdButton.Text = 'secd';

% Create acosdbutton
app.acosdButton = uibutton(app.UIFigure, 'push');
app.acosdButton.ButtonPushedFcn = createCallbackFcn(app, @Buttoncosec, true);
app.acosdButton.BackgroundColor = [0.9412 0.7255 0.7255];
app.acosdButton.FontSize = 14;
app.acosdButton.FontWeight = 'bold';
app.acosdButton.Position = [309 71 42 28];
app.acosdButton.Text = 'acosd';

% Create atandbutton
app.atandButton = uibutton(app.UIFigure, 'push');
app.atandButton.ButtonPushedFcn = createCallbackFcn(app, @Buttoncosec, true);
app.atandButton.BackgroundColor = [0.9412 0.7255 0.7255];
app.atandButton.FontSize = 14;

```

```

app.atandButton.FontWeight = 'bold';
app.atandButton.Position = [368 71 41 28];
app.atandButton.Text = 'atand';

% Create acscdButton
app.acscdButton = uibutton(app.UIFigure, 'push');
app.acscdButton.ButtonPushedFcn = createCallbackFcn(app, @Buttoncosec, true);
app.acscdButton.BackgroundColor = [0.9412 0.7255 0.7255];
app.acscdButton.FontSize = 14;
app.acscdButton.FontWeight = 'bold';
app.acscdButton.Position = [250 25 44 28];
app.acscdButton.Text = 'acscd';

% Create asecdButton
app.asecdButton = uibutton(app.UIFigure, 'push');
app.asecdButton.ButtonPushedFcn = createCallbackFcn(app, @Buttoncosec, true);
app.asecdButton.BackgroundColor = [0.9412 0.7255 0.7255];
app.asecdButton.FontSize = 14;
app.asecdButton.FontWeight = 'bold';
app.asecdButton.Position = [309 25 42 28];
app.asecdButton.Text = 'asecd';

% Create acotdButton
app.acotdButton = uibutton(app.UIFigure, 'push');
app.acotdButton.ButtonPushedFcn = createCallbackFcn(app, @Buttoncosec, true);
app.acotdButton.BackgroundColor = [0.9412 0.7255 0.7255];
app.acotdButton.FontSize = 14;
app.acotdButton.FontWeight = 'bold';
app.acotdButton.Position = [368 25 41 28];
app.acotdButton.Text = 'acotd';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = Scientificcalculator

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargin == 0
clear app

```

```

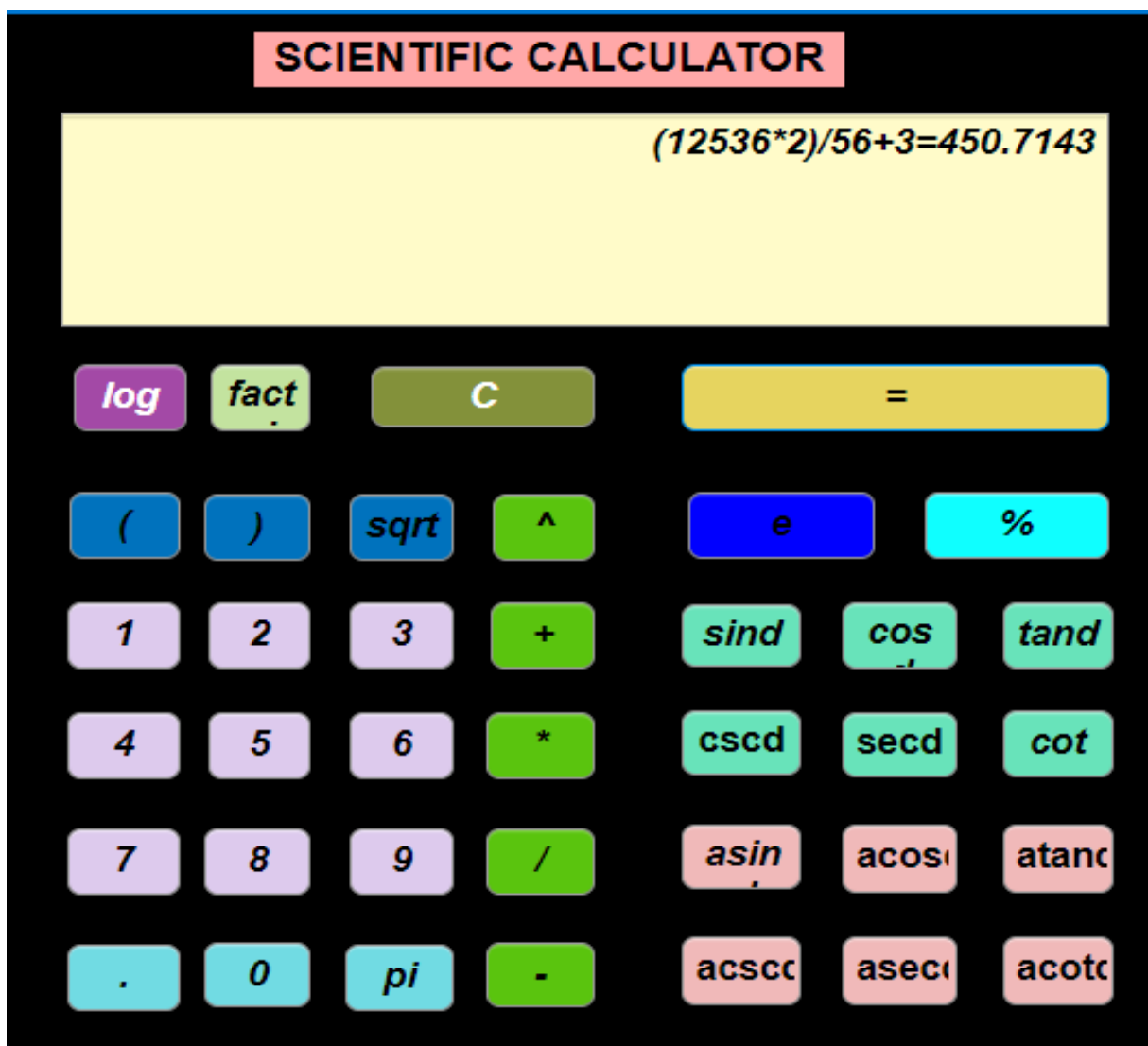
end
end
% Code that executes before app deletion
function delete(app)

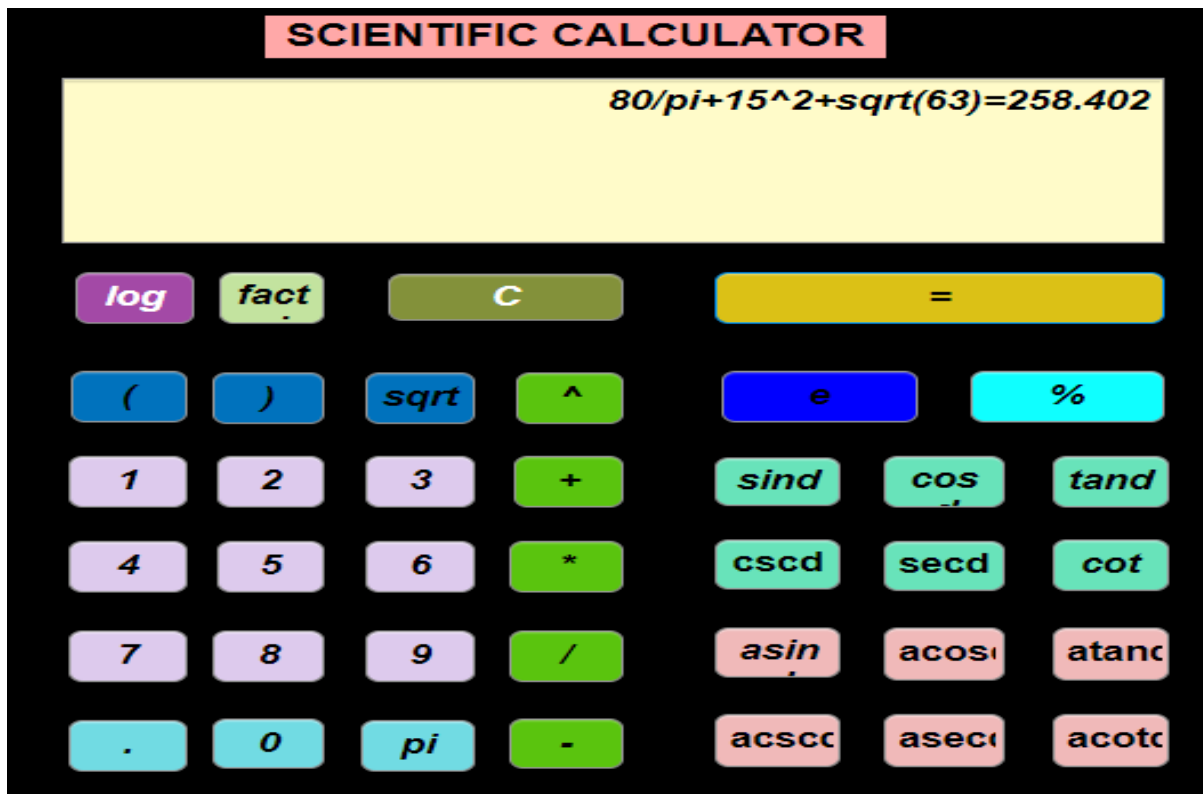
% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end

```

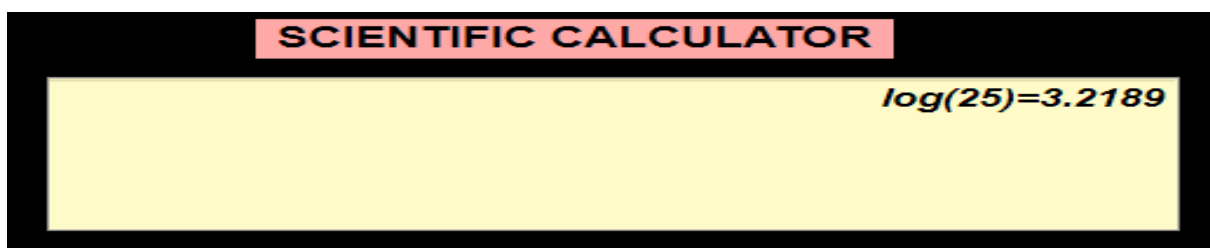
Tasks –

Simple Calculations –

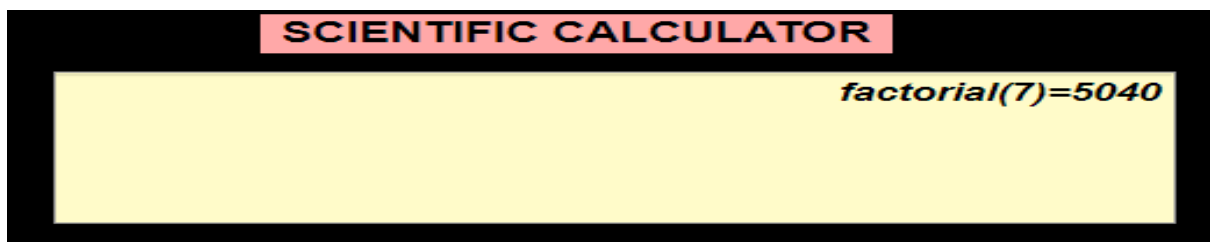




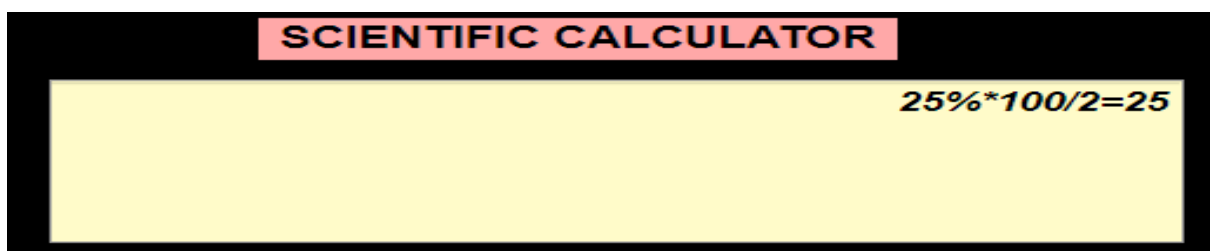
Logarithm –



Factorial –



Percentage –



Trigonometry –

SCIENTIFIC CALCULATOR

$\text{sind}(\log(2))=0.012097$

SCIENTIFIC CALCULATOR

$\text{tand}(45)=1$

Inverse Trigonometry –

SCIENTIFIC CALCULATOR

$\text{asind}(60)=90-274.2991i$

SCIENTIFIC CALCULATOR

$\text{acscd}(45)=1.2733$

Image Processing –



In our daily life we use phone camera for different uses like clicking selfie, photo, barcode scanning for different things like making payments, gathering the information about the objects, face recognition, thumb impression for unlocking phones etc., OCR and many more. Do you know how they do all such things by just looking at the image?

All these things are nothing but image processing. Image Processing gives the computer a vision to detect and recognize the things and take the action accordingly.

A scene, a view we see with our eyes, is actually a continuous signal obtained with electromagnetic energy spectra. The value of this signal perceived by the receptors in our eye is basically determined by two main factors: the amount of light that falls into the environment and the amount of light reflected back from the object into our eyes.

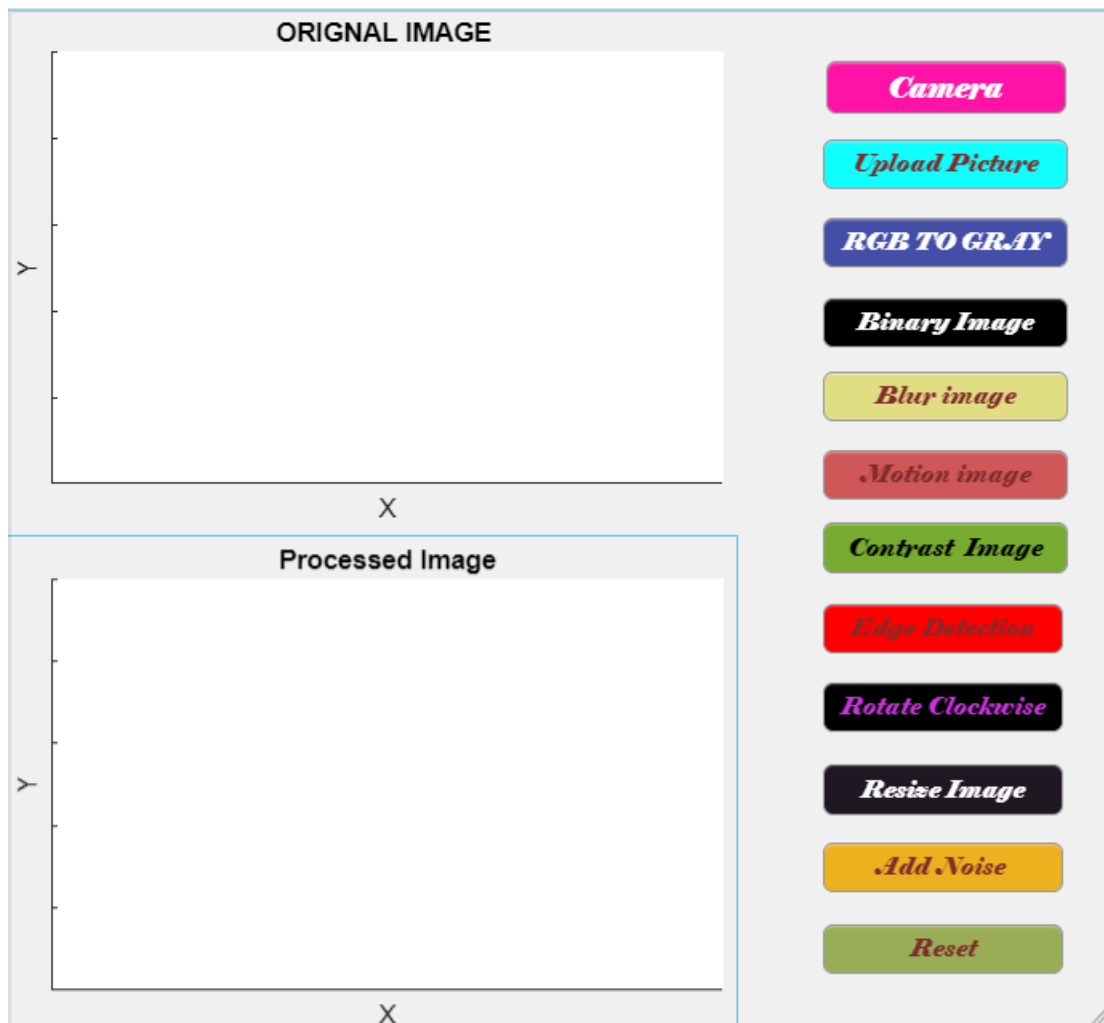
As we know, there are various light sources with different wavelengths on the earth, and we can only distinguish light with a certain wavelength with our eyes. Here, we can call all the scenes that we can perceive thanks to the light in this range, as “**pictures**”. We said that the information reaching our eyes by this light source is “**continuous**”. When we want to transfer this information to digital media and store it in this environment, it means that we are talking about the concept of digital picture, namely “**discrete**” signals.

There are lots of software available for image processing, among them **MATLAB** is the most suitable to start with. MATLAB can perform many advanced image processing operations, but for getting started with image processing in MATLAB, there are some basic operations to perform like RGB to GRAY, Rotate the image, Binary conversion etc.

We can do image processing in MATLAB by 3 different ways, one of them is by directly entering the commands in the editor/command window, second one is by creating a GUI and

the 3rd one is by designing a MATLAB App using App Designer for same. In future GUI interface will not be there, so it is a better practice to use mlApp.

Interface –



How to start – Start the MATLAB, go to apps in upper section, go to Design App and click on new blank app.

Components Used –

Axes –

We are using 2 axes buttons for displaying input image and output image which is processed image.

Camera –

This button is used to take input as a camera picture at instant.

Upload Picture –

This button is used to take the input image from the memory location or storage of the device.

RGB to GRAY –

Using this button, we can convert the image into Grayscale image.

Binary Image –

Using this button image can be converted into Binary image.

Blur Image –

Using this button image can be blurred.

Histogram –

Histogram of the image can be plotted using this function.

Motion Image –

Motional Image can be created using this.

Contrast image –

Image can be contrasted at different level using this button.

Edge Detection –

Edge of the image can be detected using this button.

Resize Image –

Image can be resized into different scale using this button.

Add Noise –

Noisy Image can be created using this button.

Reset –

Axes can be cleared using this button.

Basic commands used-**imread –**

For reading the image.

imshow-

For displaying the image.

rgb2gray –

Converting the image into grayscale.

im2bw –

Converting the image in black and white.

imhist –

For histogram of the image.

imrotate-

For rotating the image.

imedge –

For edge detection.

size –

For display the dimensions of the image.

Code –

```
Classdef ImageProcessor < matlab.apps.AppBase

    % Properties that correspond to app components
    Properties (Access = private)
        UIFigure          matlab.ui.Figure
        CameraButton      matlab.ui.control.Button
        ResetButton       matlab.ui.control.Button
        AddNoiseButton    matlab.ui.control.Button
    end
end
```

```

    ResizeImageButton          matlab.ui.control.Button
    RotateClockwiseButton      matlab.ui.control.Button
    EdgeDetectionButton        matlab.ui.control.Button
    ContrastImageButton        matlab.ui.control.Button
    MotionImageButton          matlab.ui.control.Button
    BlurImageButton            matlab.ui.control.Button
    BinaryImageButton          matlab.ui.control.Button
    RGBTOGRAYButton           matlab.ui.control.Button
    UploadPictureButton        matlab.ui.control.Button
    UIAxes2                    matlab.ui.control.Button
    UIAxes                      matlab.ui.control.Button
end
Properties (Access = private)
    img % Description
    img2 % Description
end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: UploadPictureButton
    function UploadPicture(app, event)
        [file, path] = uigetfile('*.');
        if isequal(file,0)
            figure(app.UIFigure);
            return;
        end
        figure(app.UIFigure);
        app.img = imread(fullfile(path,file));
        imshow(app.img, 'parent', app.UIAxes);
    end

    % Button pushed function: RGBTOGRAYButton
    function RGBTOGRAYButtonPushed(app, event)
        app.img2 = rgb2gray(app.img);
        imshow(app.img2, 'Parent', app.UIAxes2);
    end

    % Button pushed function: AddNoiseButton
    function AddNoiseButtonPushed(app, event)
        app.img2 = imnoise(app.img, 'salt & pepper', 0.4);
        imshow(app.img2, 'Parent', app.UIAxes2);
    end

    % Button pushed function: ResetButton
    function ResetButtonPushed(app, event)
        cla(app.UIAxes, 'reset')
        cla(app.UIAxes2, 'reset')
    end

    % Button pushed function: BinaryImageButton

```

```

function BinaryImageButtonPushed(app, event)
    app.img2 = im2bw(app.img);
    imshow(app.img2, 'Parent', app.UIAxes2);
end

```

```

function
I = app.img;
I=rgb2gray(I);
A=imhist(I);
imshow(A, 'Parent', app.UIAxes2);

```

```

% Button pushed function: BlurImageButton
function BlurImageButtonPushed(app, event)
H = fspecial('disk',5);
app.img2 = imfilter(app.img,H,'replicate');
imshow(app.img2, 'Parent', app.UIAxes2);
end

```

```

% Button pushed function: MotionImageButton
function MotionImageButtonPushed(app, event)
H = fspecial('motion',40,30);
app.img2 = imfilter(app.img,H,'replicate');
imshow(app.img2, 'Parent', app.UIAxes2);
end

```

```

% Button pushed function: EdgeDetectionButton
function EdgeDetectionButtonPushed(app, event)
I = rgb2gray(app.img);
app.img2 = edge(I,"canny");
imshow(app.img2, 'Parent', app.UIAxes2);
end

```

```

% Button pushed function: RotateClockwiseButton
function RotateClockwiseButtonPushed(app, event)
app.img2 = imrotate(app.img,90);
imshow(app.img2, 'Parent', app.UIAxes2);
end

```

```

% Button pushed function: ResizeImageButton
function ResizeImageButtonPushed(app, event)
app.img2 = imresize(app.img,0.33);
imshow(app.img2, 'Parent', app.UIAxes2);
end

```

```

% Button pushed function: ContrastImageButton
function ContrastImageButtonPushed(app, event)
app.img2 = (app.img)*2;
imshow(app.img2, 'Parent', app.UIAxes2);
end

```

```

% Button pushed function: CameraButton
function CameraButtonPushed(app, event)
cam = webcam('HP TrueVision HD Camera');
cam.Resolution = '320x240';
closePreview(cam)
app.img = snapshot(cam);
imshow(app.img, 'Parent', app.UIAxes);
clear('cam');
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 554 508];
app.UIFigure.Name = 'MATLAB App';

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'ORIGINAL IMAGE ');
xlabel(app.UIAxes, 'X')
ylabel(app.UIAxes, 'Y')
zlabel(app.UIAxes, 'Z')
app.UIAxes.XTick = [];
app.UIAxes.XTickLabel = '';
app.UIAxes.YTick = [0 0.2 0.4 0.6 0.8 1];
app.UIAxes.YTickLabel = '';
app.UIAxes.Position = [1 254 361 255];

% Create UIAxes2
app.UIAxes2 = uiaxes(app.UIFigure);
title(app.UIAxes2, 'Processed Image')
xlabel(app.UIAxes2, 'X')
ylabel(app.UIAxes2, 'Y')
zlabel(app.UIAxes2, 'Z')
app.UIAxes2.XTick = [];
app.UIAxes2.YTickLabel = '';
app.UIAxes2.Position = [1 1 361 245];

% Create UploadPictureButton
app.UploadPictureButton = uibutton(app.UIFigure, 'push');
app.UploadPictureButton.ButtonPushedFcn = createCallbackFcn(app,
@UploadPicture, true);
app.UploadPictureButton.BackgroundColor = [0.0588 1 1];
app.UploadPictureButton.FontName = 'Elephant';

```



```

app.UploadPictureButton.FontAngle = 'italic';
app.UploadPictureButton.FontColor = [0.5098 0.1725 0.1725];
app.UploadPictureButton.Position = [407 421 122 25];
app.UploadPictureButton.Text = 'Upload Picture';

% Create RGBTOGRAYButton
app.RGBTOGRAYButton = uibutton(app.UIFigure, 'push');
app.RGBTOGRAYButton.ButtonPushedFcn = createCallbackFcn(app,
@RGBTOGRAYButtonPushed, true);
app.RGBTOGRAYButton.BackgroundColor = [0.2667 0.3059 0.651];
app.RGBTOGRAYButton.FontName = 'Elephant';
app.RGBTOGRAYButton.FontAngle = 'italic';
app.RGBTOGRAYButton.FontColor = [1 1 1];
app.RGBTOGRAYButton.Position = [407 382 122 25];
app.RGBTOGRAYButton.Text = 'RGB TO GRAY';

% Create BinaryImageButton
app.BinaryImageButton = uibutton(app.UIFigure, 'push');
app.BinaryImageButton.ButtonPushedFcn = createCallbackFcn(app,
@BinaryImageButtonPushed, true);
app.BinaryImageButton.BackgroundColor = [0 0 0];
app.BinaryImageButton.FontName = 'Elephant';
app.BinaryImageButton.FontAngle = 'italic';
app.BinaryImageButton.FontColor = [1 1 1];
app.BinaryImageButton.Position = [407 342 122 25];
app.BinaryImageButton.Text = 'Binary Image';

% Create BlurImageButton
app.BlurImageButton = uibutton(app.UIFigure, 'push');
app.BlurImageButton.ButtonPushedFcn = createCallbackFcn(app,
@BlurImageButtonPushed, true);
app.BlurImageButton.BackgroundColor = [0.8784 0.8667 0.5098];
app.BlurImageButton.FontName = 'Elephant';
app.BlurImageButton.FontAngle = 'italic';
app.BlurImageButton.FontColor = [0.5098 0.1725 0.1725];
app.BlurImageButton.Position = [407 305 122 25];
app.BlurImageButton.Text = 'Blur image';

% Create MotionImageButton
app.MotionImageButton = uibutton(app.UIFigure, 'push');
app.MotionImageButton.ButtonPushedFcn = createCallbackFcn(app,
@MotionImageButtonPushed, true);
app.MotionImageButton.BackgroundColor = [0.8118 0.3412 0.3412];
app.MotionImageButton.FontName = 'Elephant';
app.MotionImageButton.FontAngle = 'italic';
app.MotionImageButton.FontColor = [0.5098 0.1725 0.1725];
app.MotionImageButton.Position = [407 266 122 25];
app.MotionImageButton.Text = 'Motion image';

% Create ContrastImageButton
app.ContrastImageButton = uibutton(app.UIFigure, 'push');

```

```

app.ContrastImageButton.ButtonPushedFcn = createCallbackFcn(app,
@ContrastImageButtonPushed, true);
app.ContrastImageButton.BackgroundColor = [0.4667 0.6745 0.1882];
app.ContrastImageButton.FontName = 'Elephant';
app.ContrastImageButton.FontAngle = 'italic';
app.ContrastImageButton.Position = [407 229 122 26];
app.ContrastImageButton.Text = 'Contrast Image';

```

% Create EdgeDetectionButton

```

app.EdgeDetectionButton = uibutton(app.UIFigure, 'push');
app.EdgeDetectionButton.ButtonPushedFcn = createCallbackFcn(app,
@EdgeDetectionButtonPushed, true);
app.EdgeDetectionButton.BackgroundColor = [1 0 0];
app.EdgeDetectionButton.FontName = 'Elephant';
app.EdgeDetectionButton.FontAngle = 'italic';
app.EdgeDetectionButton.FontColor = [0.5098 0.1725 0.1725];
app.EdgeDetectionButton.Position = [407 189 120 25];
app.EdgeDetectionButton.Text = 'Edge Detection';

```

% Create RotateClockwiseButton

```

app.RotateClockwiseButton = uibutton(app.UIFigure, 'push');
app.RotateClockwiseButton.ButtonPushedFcn = createCallbackFcn(app,
@RotateClockwiseButtonPushed, true);
app.RotateClockwiseButton.BackgroundColor = [0 0 0];
app.RotateClockwiseButton.FontName = 'Elephant';
app.RotateClockwiseButton.FontAngle = 'italic';
app.RotateClockwiseButton.FontColor = [0.7765 0.2118 0.851];
app.RotateClockwiseButton.Position = [407 150 120 25];
app.RotateClockwiseButton.Text = 'Rotate Clockwise';

```

% Create ResizeImageButton

```

app.ResizeImageButton = uibutton(app.UIFigure, 'push');
app.ResizeImageButton.ButtonPushedFcn = createCallbackFcn(app,
@ResizeImageButtonPushed, true);
app.ResizeImageButton.BackgroundColor = [0.1216 0.0902 0.1294];
app.ResizeImageButton.FontName = 'Elephant';
app.ResizeImageButton.FontAngle = 'italic';
app.ResizeImageButton.FontColor = [1 1 1];
app.ResizeImageButton.Position = [407 108 120 26];
app.ResizeImageButton.Text = 'Resize Image';

```

% Create AddNoiseButton

```

app.AddNoiseButton = uibutton(app.UIFigure, 'push');
app.AddNoiseButton.ButtonPushedFcn = createCallbackFcn(app,
@AddNoiseButtonPushed, true);
app.AddNoiseButton.BackgroundColor = [0.9294 0.6941 0.1255];
app.AddNoiseButton.FontName = 'Elephant';
app.AddNoiseButton.FontAngle = 'italic';
app.AddNoiseButton.FontColor = [0.5098 0.1725 0.1725];
app.AddNoiseButton.Position = [407 70 120 25];
app.AddNoiseButton.Text = 'Add Noise ';

```

```

% Create ResetButton
app.ResetButton = uibutton(app.UIFigure, 'push');
app.ResetButton.ButtonPushedFcn = createCallbackFcn(app, @ResetButtonPushed,
true);
app.ResetButton.BackgroundColor = [0.6 0.6784 0.3412];
app.ResetButton.FontName = 'Elephant';
app.ResetButton.FontAngle = 'italic';
app.ResetButton.FontColor = [0.5098 0.1725 0.1725];
app.ResetButton.Position = [407 29 120 25];
app.ResetButton.Text = 'Reset';

% Create CameraButton
app.CameraButton = uibutton(app.UIFigure, 'push');
app.CameraButton.ButtonPushedFcn = createCallbackFcn(app, @CameraButtonPushed,
true);
app.CameraButton.BackgroundColor = [1 0.0745 0.651];
app.CameraButton.FontName = 'Elephant';
app.CameraButton.FontSize = 14;
app.CameraButton.FontAngle = 'italic';
app.CameraButton.FontColor = [1 1 1];
app.CameraButton.Position = [408 459 120 26];
app.CameraButton.Text = 'Camera';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = ImageProcessor

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargin == 0
clear app
end
end

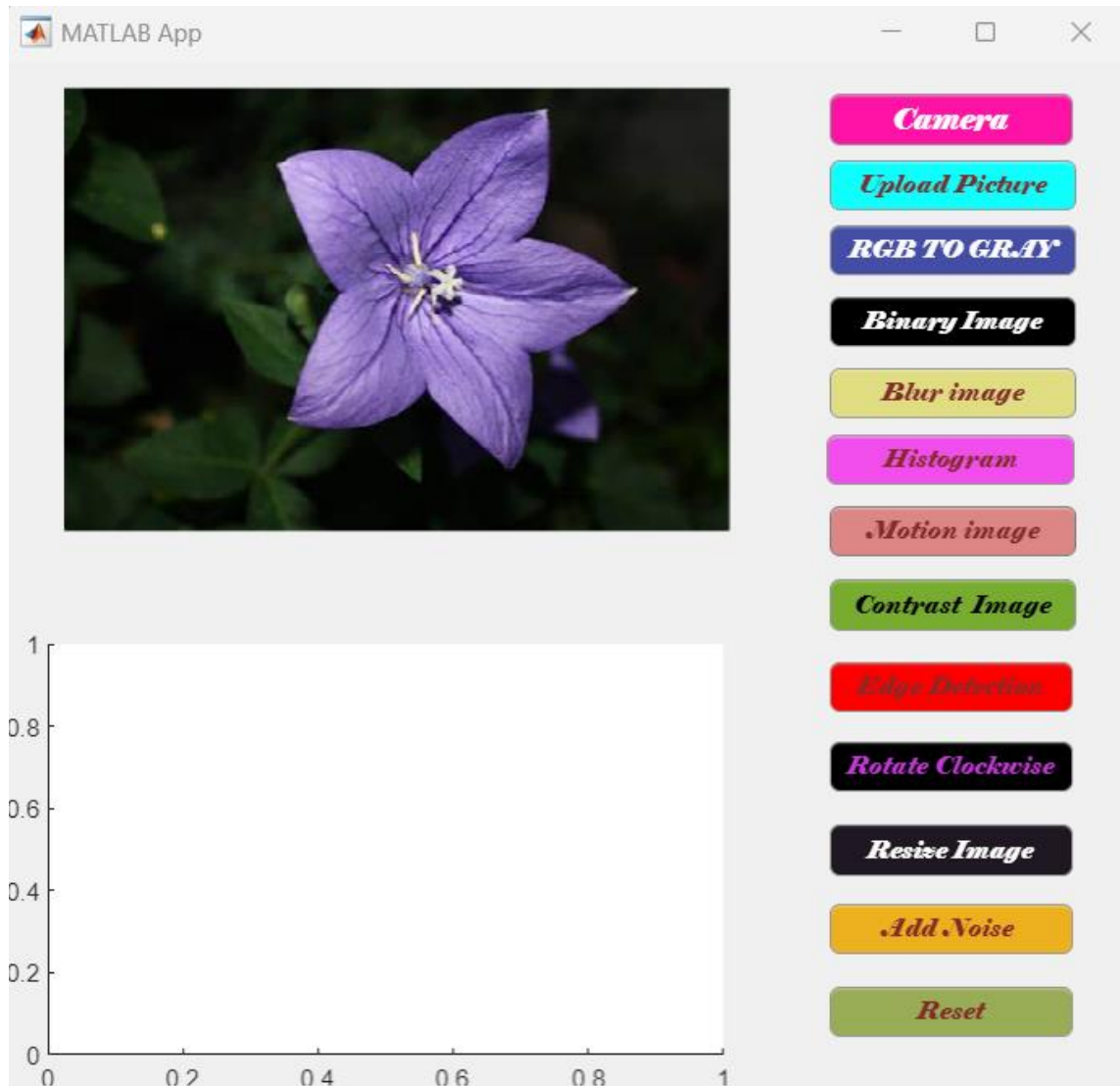
% Code that executes before app deletion
function delete(app)

```

```
% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```


Tasks –

Picture upload from device memory –




RGB to GRAY image of the picture –

ORIGINAL IMAGE



Processed Image




Y

Y

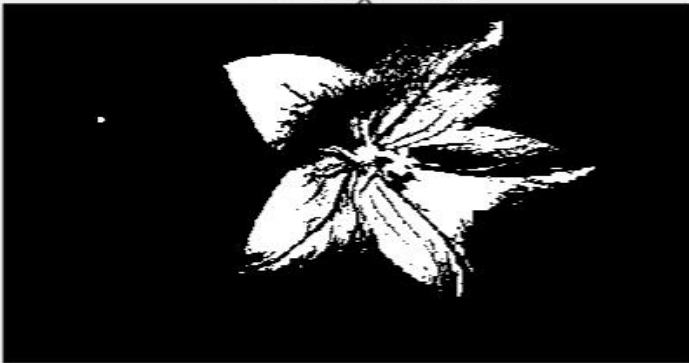
- Camera
- Upload Picture
- RGB TO GRAY**
- Binary Image
- Blur image
- Motion image
- Contrast Image
- Edge Detection
- Rotate Clockwise
- Resize Image
- Add Noise
- Reset

Binary Image –

ORIGINAL IMAGE



Processed Image



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise


Resize Image

Add Noise


Reset

Blur Image –

ORIGINAL IMAGE



Processed Image



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise


Resize Image

Add Noise


Reset

Motion Image –

ORIGINAL IMAGE



Processed Image



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise


Resize Image

Add Noise


Reset

Contrast Image –

ORIGINAL IMAGE



Processed Image



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise


Resize Image

Add Noise

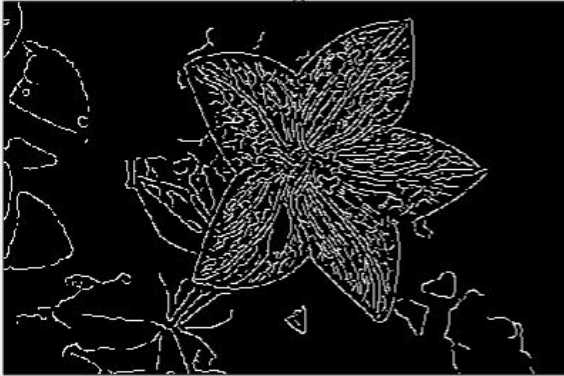
Reset

Edge Detection –

ORIGINAL IMAGE



Processed Image



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise


Resize Image

Add Noise


Reset

Rotated Image (clockwise) –

ORIGINAL IMAGE



Processed Image



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise


Resize Image






Add Noise


Reset

Resize Image –

ORIGINAL IMAGE




Processed Image     



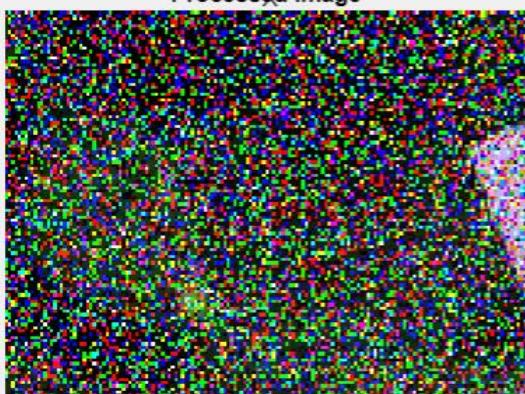
Camera
Upload Picture
RGB TO GRAY
Binary Image
Blur image
Motion image
Contrast Image
Edge Detection
Rotate Clockwise
Resize Image
Add Noise
Reset

Noised Image –

ORIGINAL IMAGE



Processed Image



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise

Resize Image

Add Noise

Reset

Reset (UIAxes cleared) –



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise

Resize Image

Add Noise

Reset

Real Time Picture Captured by device camera at the moment –



Camera

Upload Picture

RGB TO GRAY

Binary Image

Blur image

Motion image

Contrast Image

Edge Detection

Rotate Clockwise

Resize Image

Add Noise

Reset

Chapter – 4

IMAGE PROCESSING WITH PYTHON

INTRODUCTION:

We now already know about the Image Processing in preceding chapters, but for this chapter our main focus is on “How can we process a Image in Python”. So before moving to Image Processing we will start our introduction with Python..

Python- **Python** is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

Python consistently ranks as one of the most popular programming languages.

To make things short and clear about “Why to use Python”

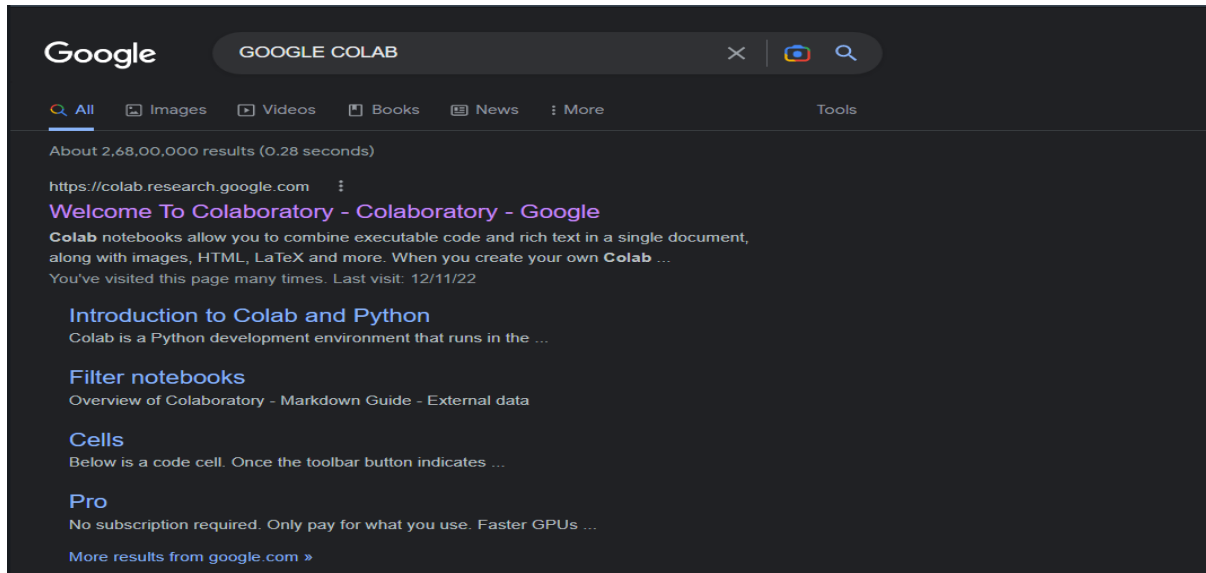
- It is easy for beginners to understand.
- It has a lot of libraries and inbuilt functions which make things easier to user.
- It is able to compute a bigger data in seconds.
- It is easier to transport, (for example in above chapter we had write a code for image processing in MATLAB, but the disadvantage of that code is it can only run in MATLAB compiler and there are only two alternatives of MATLAB that are MATLAB Online, Octave Compiler. Whereas the python have many alternatives and that code can be easily transported to Colab notebook, Jupiter notebook, Spyder notebook and more.

For Image Processing we will be using Google Colab notebook which is free for user. While the reader can use any python compiler that is available to them.

GETTING FAMILIER WITH GOOGLE COLAB

Step-1

We need to search GOOGLE COLAB in any Browser.

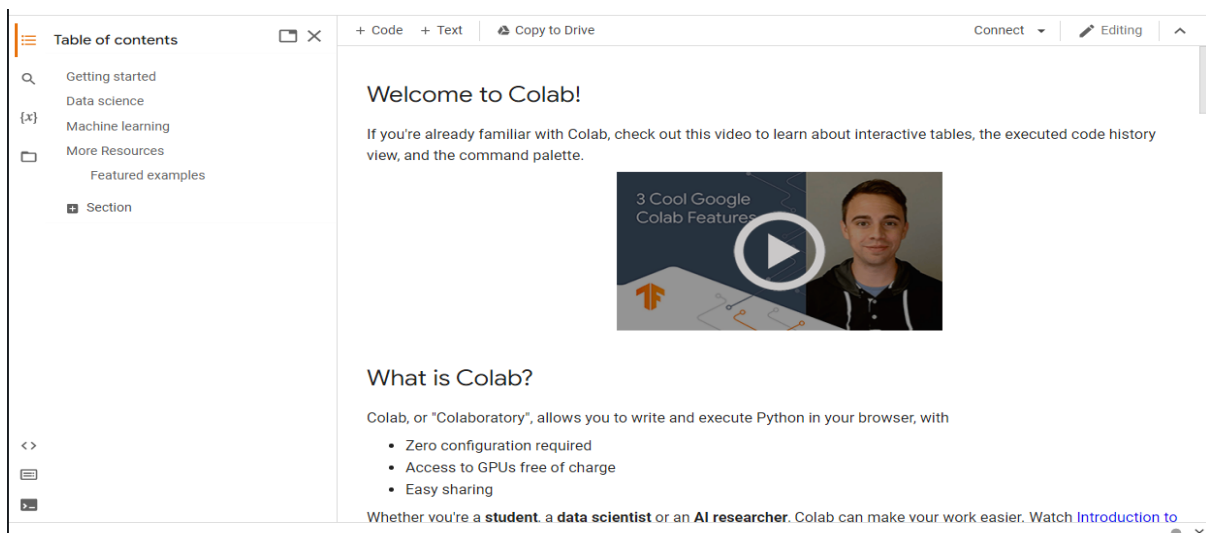


We need to select the first option which is titled as :

”Welcome To Colaboratory – Colaboratory - Google”

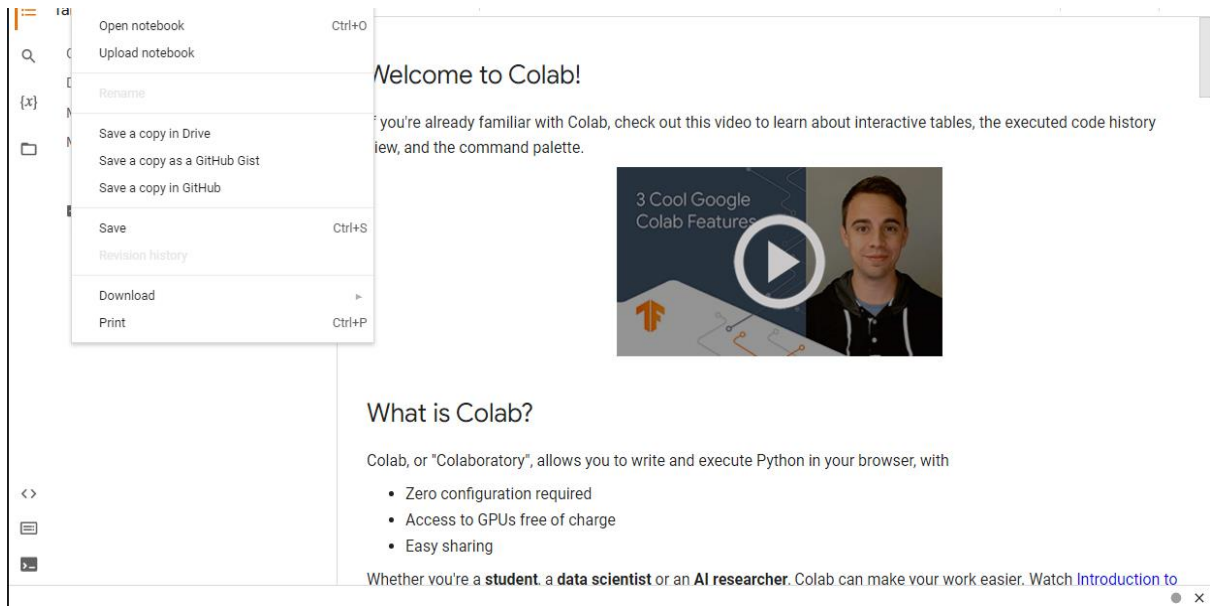
Step-2

After selecting the first option Google Colab is open and its Interface will look like.



Step-3

We need to go to file section and select New Notebook option, to write a new program.



Step-4 -

After selecting the New notebook option a new tab is open where we can write our program.



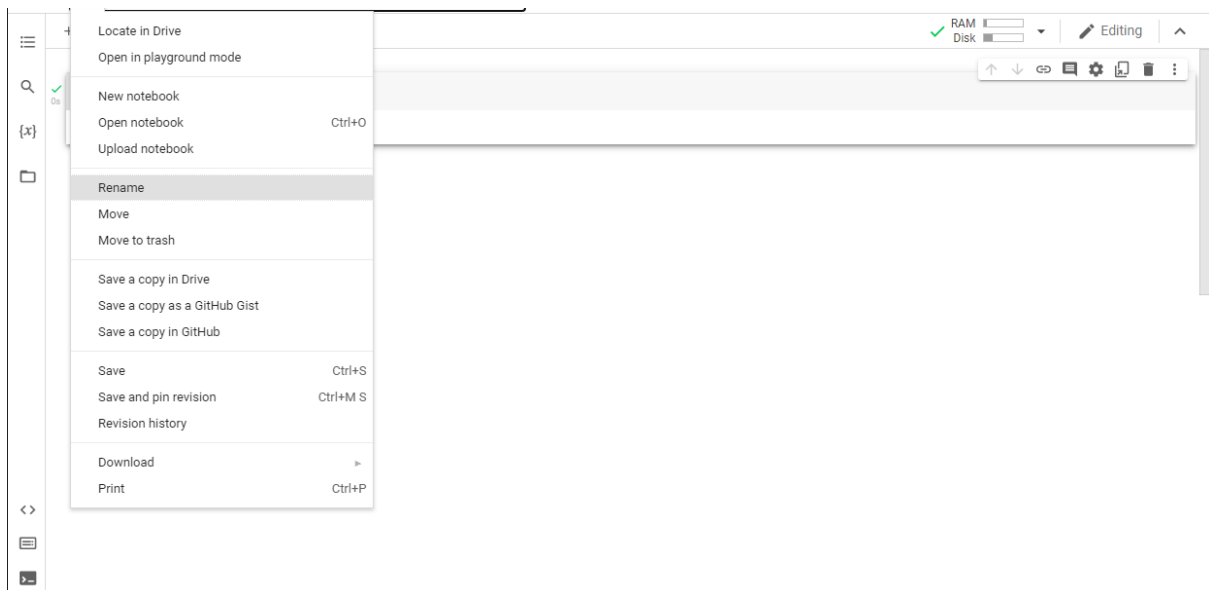
Step-5

We can write our program in the given cells.



Step-6

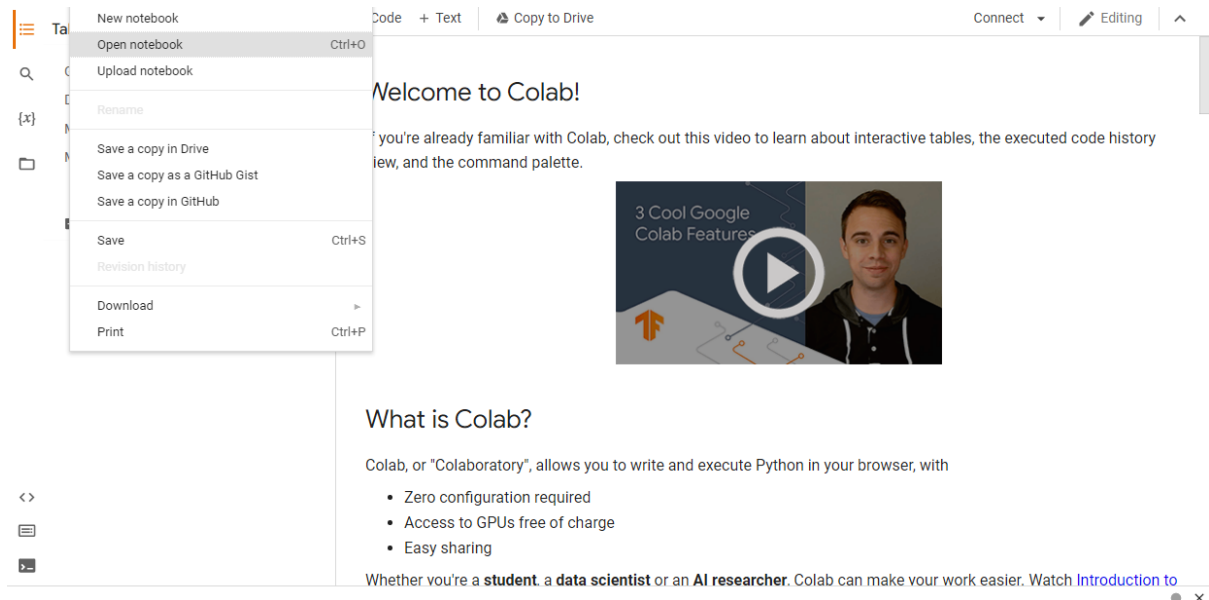
After running the program, the program is automatically saved in your drive, we can also rename the file from file section.



Opening the Saved File:

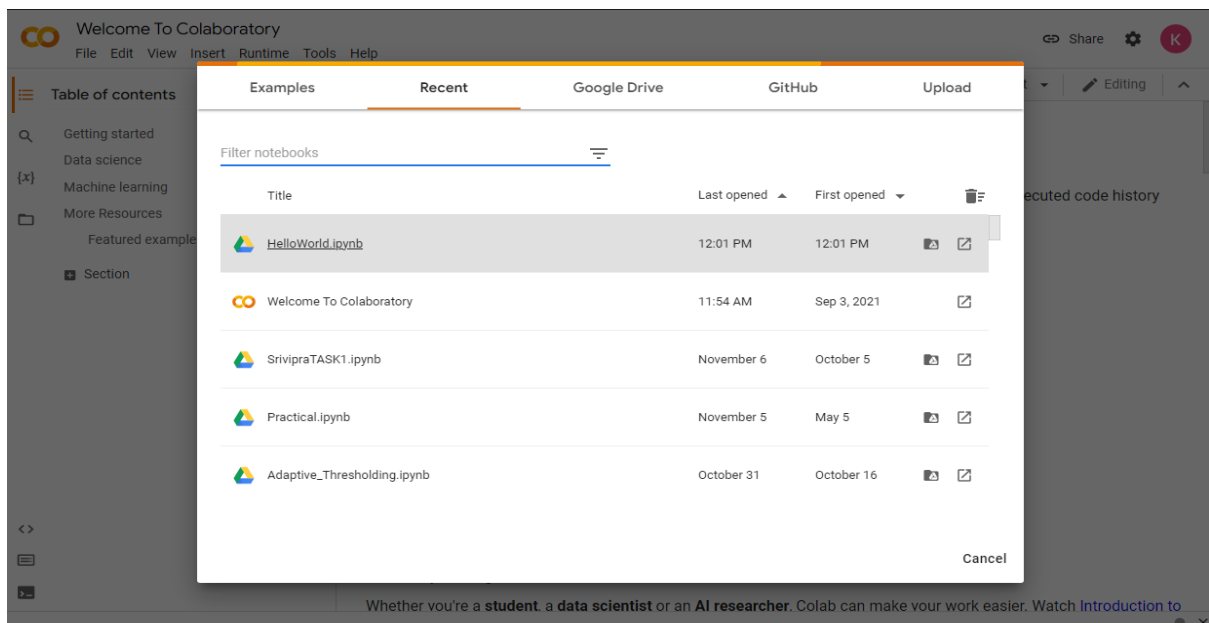
Step-1

To open the saved file, we need to go to the file section and then select Open Notebook option.



Step-2

After selecting the Open Notebook Option we will get our saved notebook Interface where we can select which Notebook we need to open. After selecting a particular notebook the notebook get opened in a new tab.

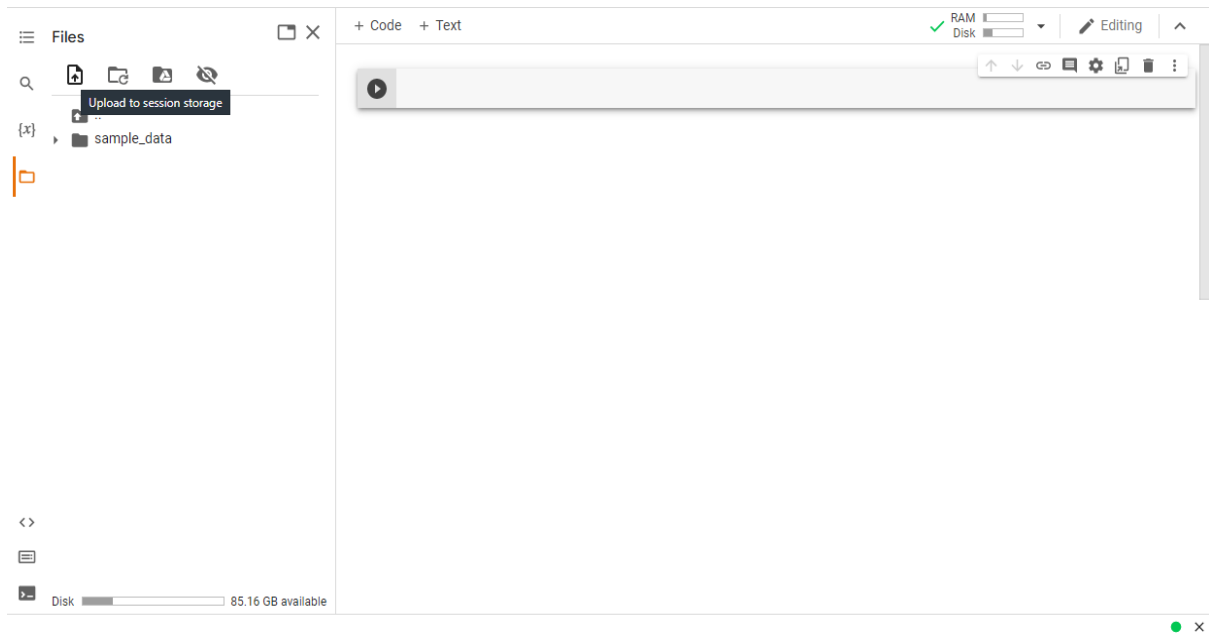
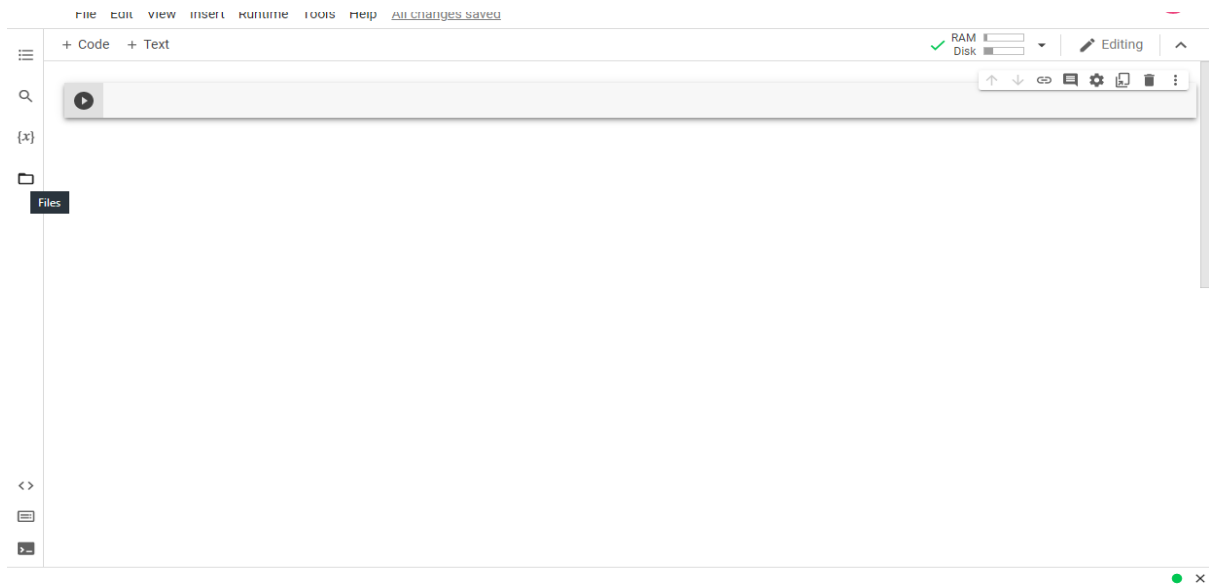


GETTING STARTED WITH PYTHON IMAGE PROCESSING:

Uploading the Image:

To start Image Processing we need the image that is to be processed, we have to first save/upload the image to the folder where our python program is saved.

Meanwhile in Google Colab we only need to upload the image in the Files section represented below.

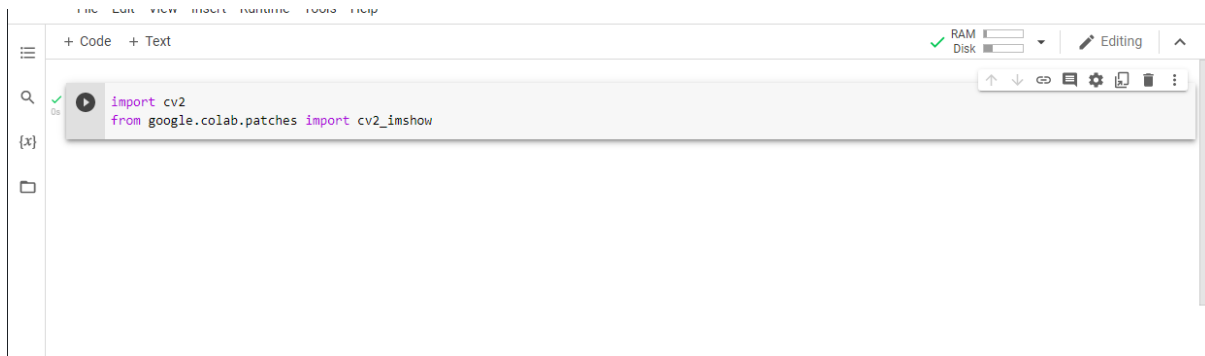


Importing the Libraries:

We cannot open an Image directly in Python we need a library to read and show image.

In python we have OpenCv that corresponds to “Open Computer Vision”, through this library we can read as well as show the image.

So to use this library we need to import this library to the program after importing we can use this library.



```
import cv2
from google.colab.patches import cv2_imshow
```

In Google Colab after importing the ‘cv2’ we need to import ‘cv2_imshow’ as in Google Colab cv2_imshow is not present in cv2 library therefore we have to import cv2_imshow individually.

Meanwhile in other python notebooks the ‘cv2’ library consist the ‘cv2.imshow’ with it.

Reading the Image:

We have uploaded the image in Files Section, Now we will read the image using ‘cv2.imread’ command.

In files section we can see the image named Tiger is uploaded in files section.

cv2.imread() command takes the image address as input



```
[13] import cv2
      from google.colab.patches import cv2_imshow

Tiger1=cv2.imread("/content/Tiger.jpg")
```

If we print tiger1 it will show us the 3D Matrix as output.

The screenshot shows a Google Colab notebook interface. On the left, the 'Files' pane shows a folder named 'sample_data' containing a file 'Tiger.jpg'. The main code area contains the following code:

```
[13] import cv2
      from google.colab.patches import cv2_imshow

[16] Tiger1=cv2.imread("/content/Tiger.jpg")

      print(Tiger1)
```

The output of the code is a large NumPy array representing the image data:

```
[[[ 80 110 127]
   [ 80 110 127]
   [ 79 109 126]
   ...
   [ 63 120 135]
   [ 63 120 135]
   [ 65 120 135]]

  [[ 80 110 127]
   [ 80 110 127]
   [ 79 109 126]
   ...
   [ 63 120 135]
   [ 65 120 135]
   [ 65 120 135]]

  [[ 80 110 129]
   [ 80 110 129]
   [ 80 110 129]
   ...
   [ 65 120 135]
   [ 65 120 135]
   [ 65 120 135]]]
```

The screenshot shows a Google Colab notebook interface. On the left, the 'Files' pane shows a folder named 'sample_data' containing a file 'Tiger.jpg'. The main code area contains the following code:

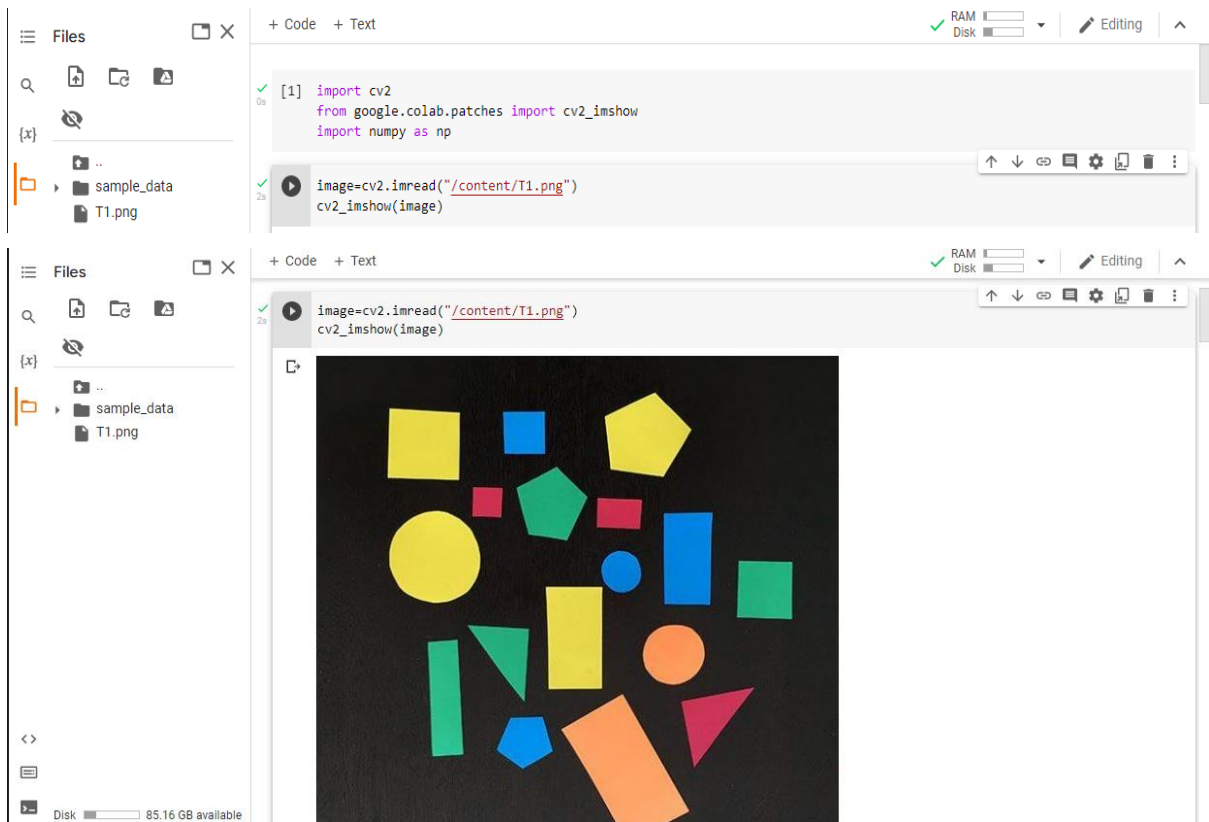
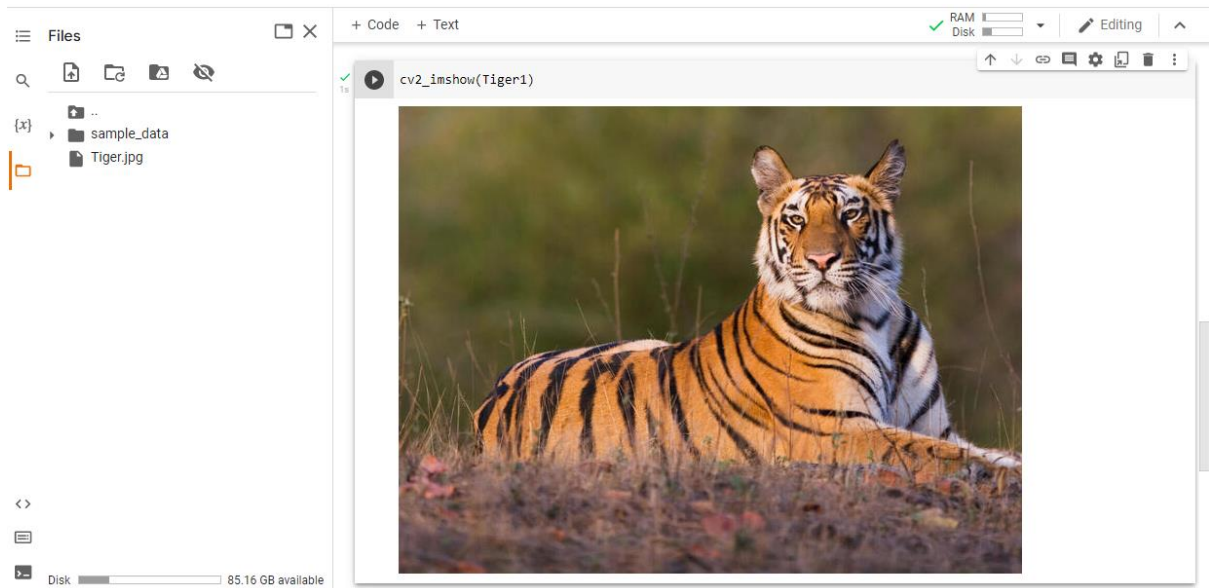
```
[[ 60 60 74]
 [ 59 59 73]
 [ 56 57 71]
 ...
 [ 83 78 99]
 [ 81 76 97]
 [ 81 73 96]]

[[ 63 61 73]
 [ 63 61 73]
 [ 59 59 73]
 ...
 [ 80 78 98]
 [ 80 75 96]
 [ 78 72 95]]

[[ 67 63 75]
 [ 65 63 75]
 [ 63 60 75]
 ...
 [ 79 77 97]
 [ 78 73 94]
 [ 77 71 94]]]
```

Showing the Image:

To show the Image in python we have to first read that image first then using cv2_imshow() Command (“for Google Colab”), otherwise use cv2.imshow() command for other python notebooks.



Converting Image from (RGB to Gray) and (RGB to HSV) :

In Opencv library we can also convert the color of the image from (RGB to GrayScale), (RGB to HSV) and many others. The most useful conversion is the RGB to Gray and RGB to HSV for Image Processing purposes.

In Python there is a command for converting a Image profile:

```
gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
```

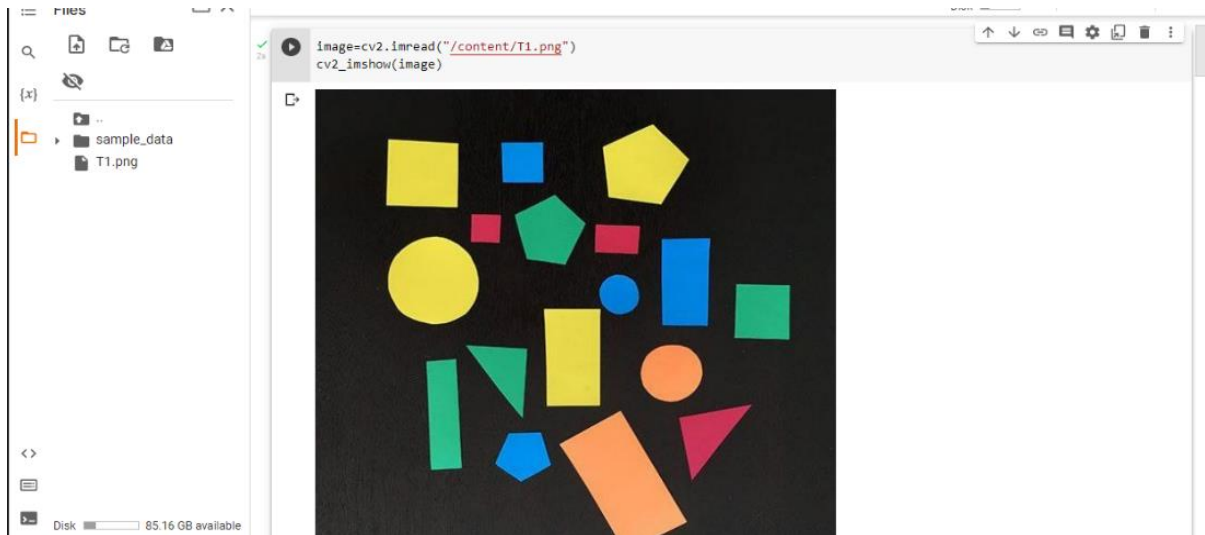
The above command will convert the RGB Image to Gray Image.

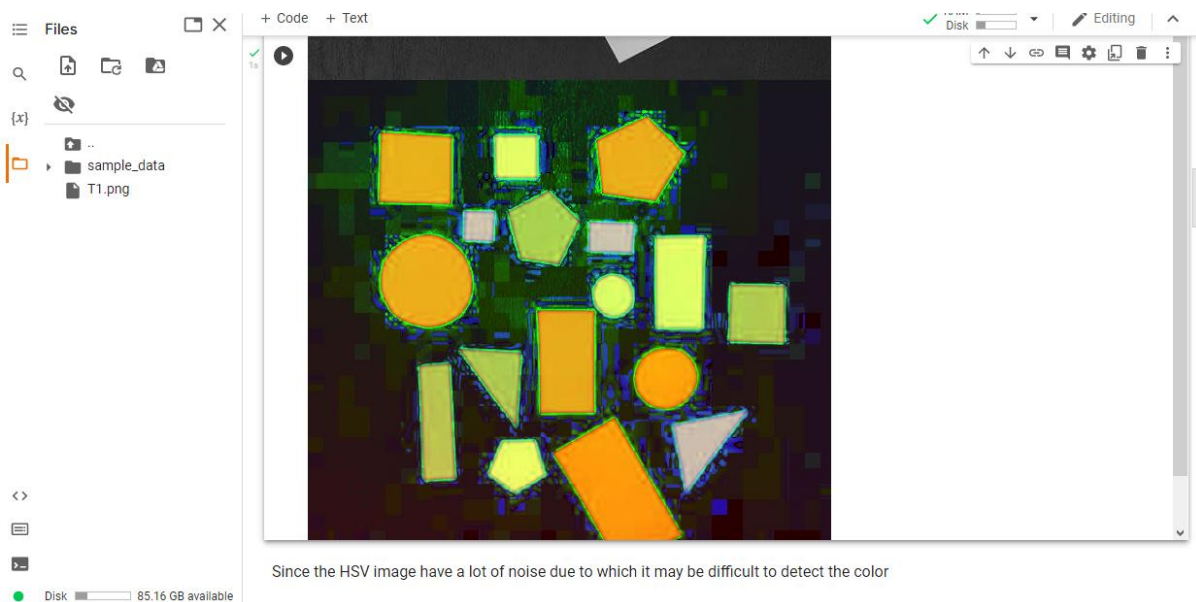
```
hsv=cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
```

The above command will convert the RGB Image to HSV Image.

The HSV profile of Image helps us in detecting the color of the object in a particular Image.

As HSV profile consists the color of the original image. HSV is helpful if we have detected the object in an image and we need to tell the color of that image then HSV profile is a miracle for us.





Since in the above HSV Image we can see that there are lots of noises outside the object. So in such type of HSV image it is difficult to detect the color of a particular object, if we have to detect the color of a particular object we need to curb the noises of this HSV image it can be done using erode or dilating the Image.

DETECTING THE OBJECTS WITHIN THE IMAGE:

To detect the objects within the Image we have to first convert the Gray Image into Binary Image. We can convert the Gray Scale Image to Binary Image by using the Threshold function in Opencv library.

Goal

- In this tutorial, you will learn simple thresholding, adaptive thresholding and Otsu's thresholding.
- You will learn the functions `cv.threshold` and `cv.adaptiveThreshold`.

Simple Thresholding

Here, the matter is straight-forward. For every pixel, the same threshold value is applied. If the pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value. The function `cv.threshold` is used to apply the thresholding. The first argument is the source image, which **should be a grayscale image**. The second argument is the threshold value which is used to classify the pixel values. The third argument is the maximum value which is assigned to pixel values exceeding the threshold. OpenCV provides different types of thresholding which is given by the fourth parameter of the function. Basic thresholding as described above is done by using the type `cv.THRESH_BINARY`. All simple thresholding types are:

- `cv.THRESH_BINARY`
- `cv.THRESH_BINARY_INV`
- `cv.THRESH_TRUNC`
- `cv.THRESH_TOZERO`
- `cv.THRESH_TOZERO_INV`

```

import numpy as np
from matplotlib import pyplot as plt

img = cv.imread('gradient.png',0)
ret,thresh1 = cv.threshold(img,127,255,cv.THRESH_BINARY)
ret,thresh2 = cv.threshold(img,127,255,cv.THRESH_BINARY_INV)
ret,thresh3 = cv.threshold(img,127,255,cv.THRESH_TRUNC)
ret,thresh4 = cv.threshold(img,127,255,cv.THRESH_TOZERO)
ret,thresh5 = cv.threshold(img,127,255,cv.THRESH_TOZERO_INV)

titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in range(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray',vmin=0,vmax=255)
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])

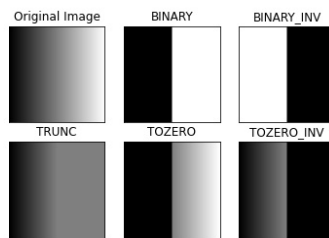
plt.show()

```

Note

To plot multiple images, we have used the plt.subplot() function. Please checkout the matplotlib docs for more details.

The code yields this result:



We will start first detecting the shapes

To detect the shapes we have to change the gray image to Binary Image using Threshold
ret,thresh=cv2.threshold(gray,75,255,0)

#cv2_imshow(thresh)

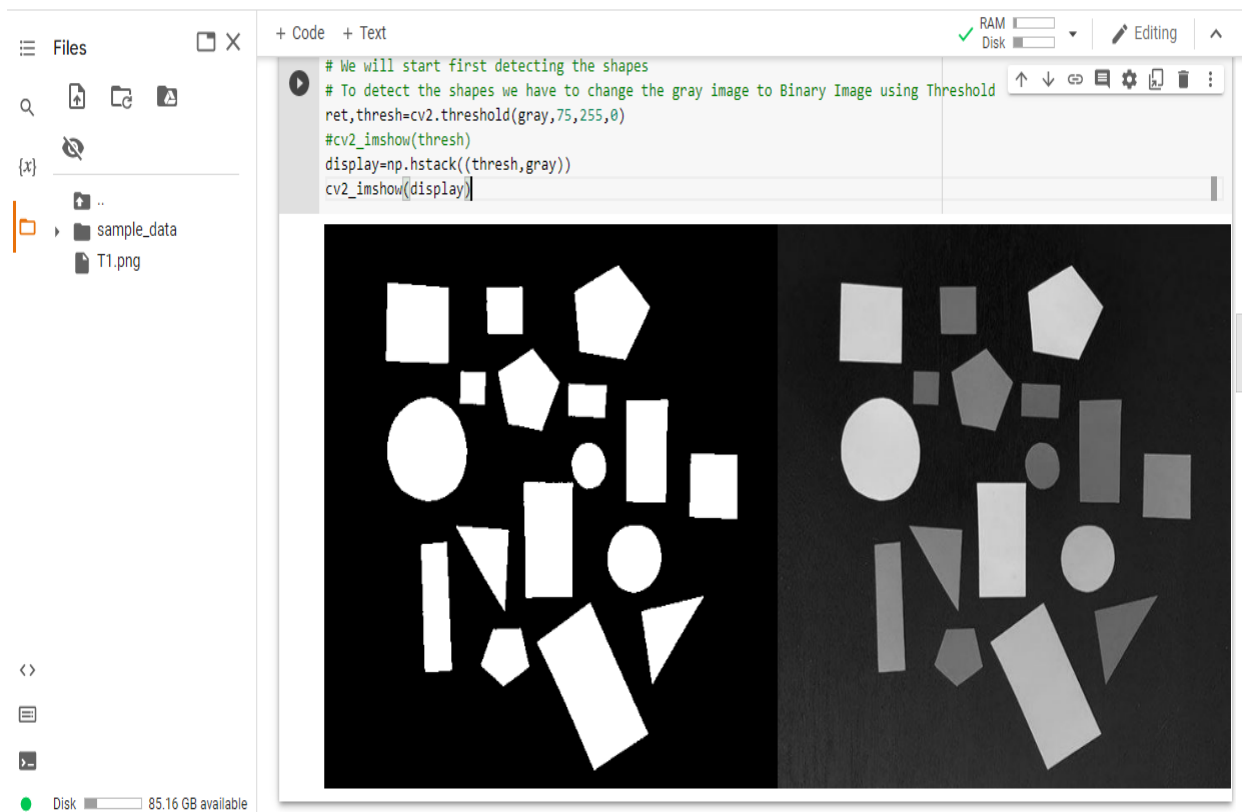
display=np.hstack((thresh,gray))

cv2_imshow(display)

To use the Threshold command we need to specify the limit that after which limit of Grayscale image the image pixel is set to be white or black.

In above example we have set the limit as 75, we can also choose other limits but limit at 75 gives the perfect Binary Image of the GrayScale Image. This Limit will vary to Image to Image.

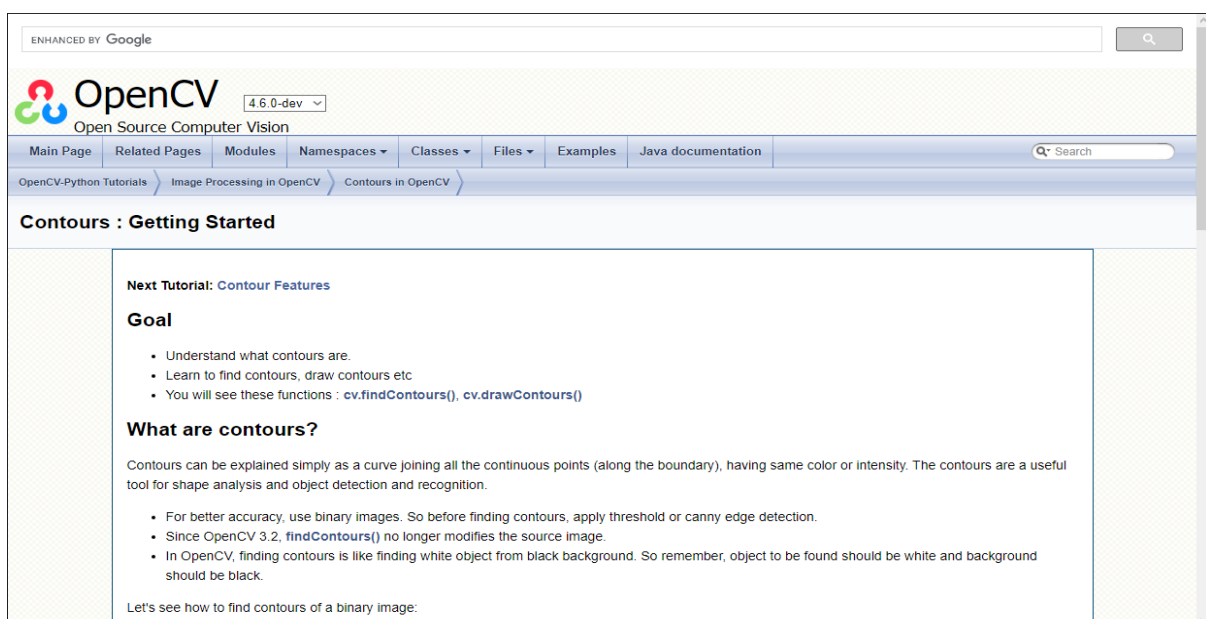
Using the above command we can convert the Grayscale Image into Binary Image.0



findContours():

To find the information about the shapes present in the image we use 'findcontour' function as this function will scan the binary image whose background is black and the object present in the image is white and calculate coordinates of the white object present in the Image. The findcontour function needed two functions that are cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE

to get information about the boundary points.



- Learn to find contours, draw contours etc
- You will see these functions : `cv.findContours()`, `cv.drawContours()`

What are contours?

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition.

- For better accuracy, use binary images. So before finding contours, apply threshold or canny edge detection.
- Since OpenCV 3.2, `findContours()` no longer modifies the source image.
- In OpenCV, finding contours is like finding white object from black background. So remember, object to be found should be white and background should be black.

Let's see how to find contours of a binary image:

```
import numpy as np
import cv2 as cv

im = cv.imread('test.jpg')
imggray = cv.cvtColor(im, cv.COLOR_BGR2GRAY)
ret, thresh = cv.threshold(imggray, 127, 255, 0)
contours, hierarchy = cv.findContours(thresh, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)
```

See, there are three arguments in `cv.findContours()` function, first one is source image, second is contour retrieval mode, third is contour approximation method. And it outputs the contours and hierarchy. Contours is a Python list of all the contours in the image. Each individual contour is a Numpy array of (x,y) coordinates of boundary points of the object.

Note

We will discuss second and third arguments and about hierarchy in details later. Until then, the values given to them in code sample will work fine for all images.

approxPolyDP:

After getting the perimeter of the objects specified in Image from `findcontour()` function, we need to find the vertices of the object present in the Image so to determine the shape of the object.

The `approxPolyDP` function helps us to find the vertices of the object from the information of perimeters that comes from the `findcontours` function.

◆ `approxPolyDP()`

```
void cv::approxPolyDP ( InputArray  curve,
                      OutputArray  approxCurve,
                      double        epsilon,
                      bool          closed
                      )
```

Python:

```
cv.approxPolyDP( curve, epsilon, closed[, approxCurve] ) -> approxCurve
```

```
#include <opencv2/imgproc.hpp>
```

Approximates a polygonal curve(s) with the specified precision.

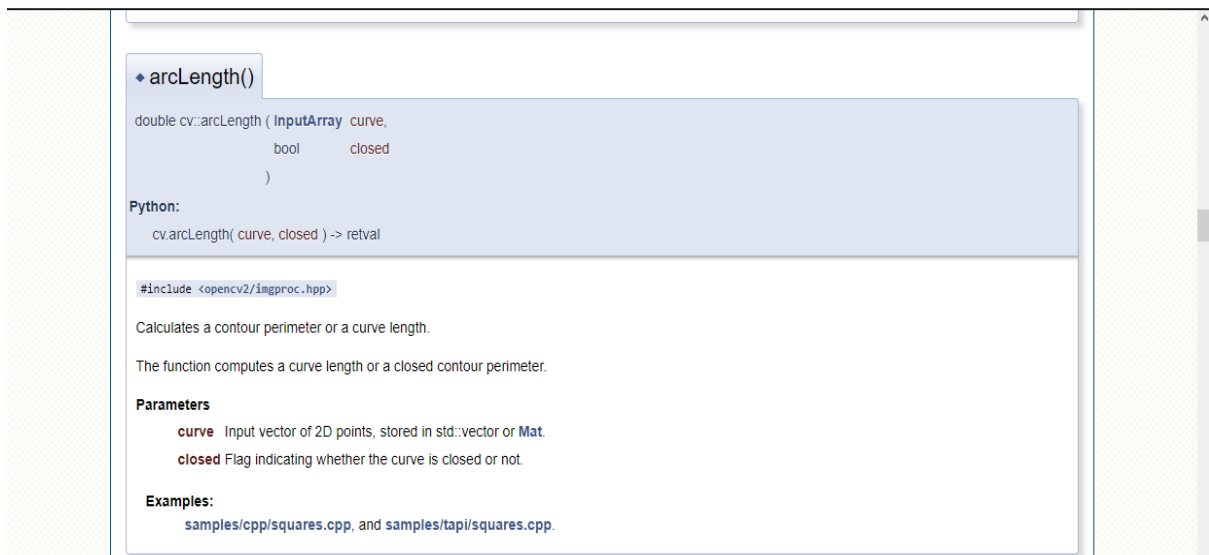
The function `cv::approxPolyDP` approximates a curve or a polygon with another curve/polygon with less vertices so that the distance between them is less or equal to the specified precision. It uses the Douglas-Peucker algorithm http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm

Parameters

- curve** Input vector of a 2D point stored in `std::vector` or `Mat`
- approxCurve** Result of the approximation. The type should match the type of the input curve.
- epsilon** Parameter specifying the approximation accuracy. This is the maximum distance between the original curve and its approximation.
- closed** If true, the approximated curve is closed (its first and last vertices are connected). Otherwise, it is not closed.

Examples:

`samples/cpp/contours2.cpp`, `samples/cpp/squares.cpp`, and `samples/tapi/squares.cpp`.



```

contour,h=cv2.findContours(thresh,cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
t=[];
for cnt in contour:
    approx=cv2.approxPolyDP(cnt,0.01*cv2.arcLength(cnt,True),True)
    print(len(approx))
    t.append(len(approx))
print(t)

```

The above code will find the vertices of the object from the large information about the object perimeter that is gathered from the find contours function. Meanwhile the approxPolyDP function will calculate the vertices of the object present in Image. In the above code 't' is the array which will store the vertices of the Object present in the Image.

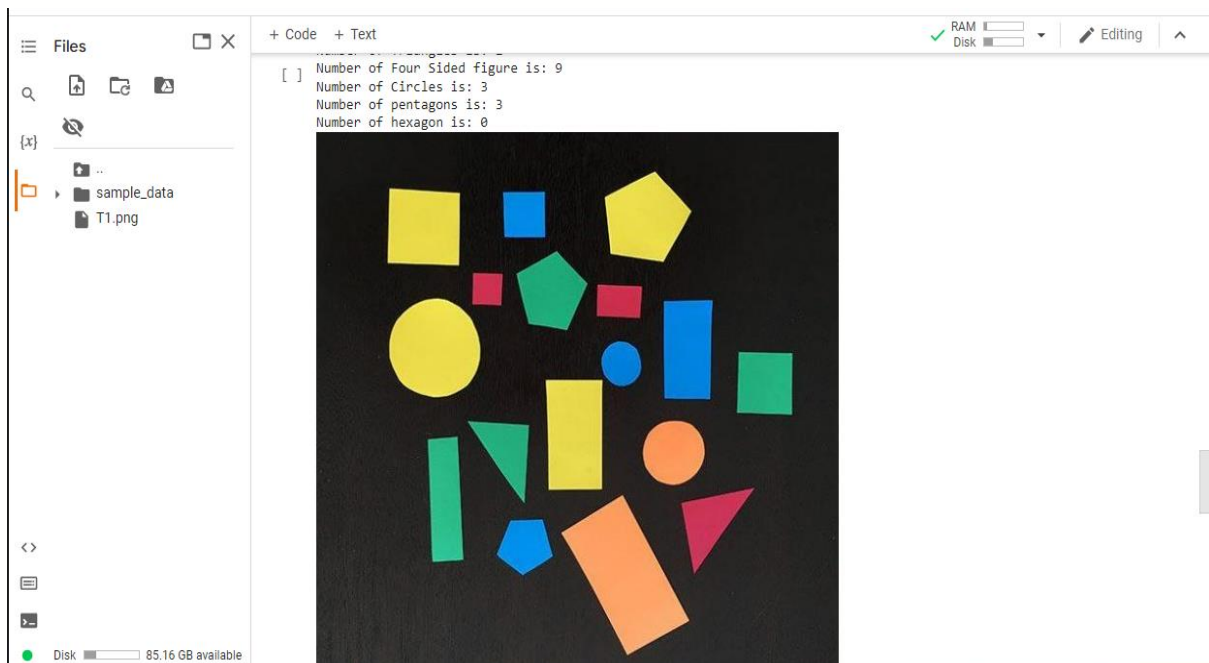


Now after this step the most of our job is done as now we will use the if-else ladder for determining the number of object present in the Image.



```
[ ] p=0;
tr=0;
c=0;
s=0;
he=0;
for i in range(len(t)):
    if (t[i]==5):
        p=p+1;
    elif (t[i]==3):
        tr=tr+1;
    elif (t[i]==4):
        s=s+1;
    elif (t[i]==6):
        he=he+1;
    elif (t[i]>=9):
        c=c+1;
print("Number of Triangles is:",tr)
print("Number of Four Sided figure is:",s)
print("Number of Circles is:",c)
print("Number of pentagons is:",p)
print("Number of hexagon is:",he)
cv2_imshow(image)

Number of Triangles is: 2
Number of Four Sided figure is: 9
Number of Circles is: 3
Number of pentagons is: 3
Number of hexagon is: 0
```



Highlighting The 2-D Object Present In Image:

For highlighting the 2-D Objects present in Images.

We will first use the findcontours() function to determine the information about perimeters and Using the for loop for each object we will draw the contour using drawcontours() function for highlighting the objects present in Image.

How to draw the contours?

To draw the contours, `cv.drawContours` function is used. It can also be used to draw any shape provided you have its boundary points. Its first argument is source image, second argument is the contours which should be passed as a Python list, third argument is index of contours (useful when drawing individual contour. To draw all contours, pass -1) and remaining arguments are color, thickness etc.

- To draw all the contours in an image:

```
cv.drawContours(img, contours, -1, (0,255,0), 3)
```

- To draw an individual contour, say 4th contour:

```
cv.drawContours(img, contours, 3, (0,255,0), 3)
```

- But most of the time, below method will be useful:

```
cnt = contours[4]
cv.drawContours(img, [cnt], 0, (0,255,0), 3)
```

```
contour,h=cv2.findContours(thresh,cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
for cnt in contour:
    approx=cv2.approxPolyDP(cnt,0.01*cv2.arcLength(cnt,True),True)
    n=len(approx);
    if n==3:
        cv2.drawContours(image,[cnt],-1,(10,100,69),3)
        #cv2.putText(image,"Triangle",cnt[0,0],cv2.FONT_HERSHEY_COMPLEX,1,(255,250,205),1)
    elif n==4:
        cv2.drawContours(image,[cnt],-1,(238,232,170),6)
        #cv2.putText(image,"Quadilateral",cnt[0,0],cv2.FONT_HERSHEY_COMPLEX,1,(255,250,205),6)
    elif n==5:
        cv2.drawContours(image,[cnt],-1,(188,143,143),6)
        #cv2.putText(image,"Pentagon",cnt[0,0],cv2.FONT_HERSHEY_COMPLEX,1,(255,250,205),4)
    elif n==6:
        cv2.drawContours(image,[cnt],-1,(255,255,255),6)
        #cv2.putText(image,"Hexagon",cnt[0,0],cv2.FONT_HERSHEY_COMPLEX,1,(255,250,205),4)
    elif n>=9:
        cv2.drawContours(image,[cnt],-1,(205,133,90),3)
        #cv2.putText(image,"Circle",cnt[0,0],cv2.FONT_HERSHEY_COMPLEX,1,(255,250,205),1)
cv2.imshow(image)
```

In the above code every object present in Image will be highlighted and the color for different shapes is different so to differentiate between the shapes.

Colab interface for `SrivipraTASK1.ipynb`. The main display area shows a black image with various colored geometric shapes (squares, circles, triangles, pentagons) overlaid with white outlines, representing object detection results. The interface includes a file browser on the left showing a folder named `sample_data` containing `T1.png`. The top right shows RAM and Disk usage, and the bottom right shows a toolbar with navigation and editing icons.

Conclusion

To conclude our report we have divided our project in four different chapters. In first chapter, we discuss about MATLAB and APP designer, its basics and how we are going to use these in our project. In second chapter, we start our project with some simple tasks like simple calculator then we design a Scientific Calculator and then perform some tasks on them illustrating the operations that designed calculator can do. In third chapter, we design an app for Image Processing. In which we converted an image into Binary Image, Gray Scale Image, Motion, Blur, Rotation etc. We also created a function to take real time picture using Webcam of laptop. In fourth chapter, we discussed the Image Processing using python through Opencv Library. We also used the Image Processing techniques to detect the 2-D shapes and highlighting those shapes that are present in the given image.

References

<https://www.myassignmentservices.com/images/matlab-assignment-expert.jpg>

<https://www.mathworks.com/help/matlab/>

<https://www.youtube.com/watch?v=huWmCqaeXvE>

<https://www.geeksforgeeks.org/digital-image-processing-algorithms-using-matlab/>

<https://youtu.be/H7aWsR46i4>

<https://cdn.educba.com/academy/wp-content/uploads/2019/12/digital-image-processing-1.jpg>

<https://cdn.educba.com/academy/wp-content/uploads/2019/12/digital-image-processing-1.jpg>

https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html)

https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html)

https://docs.opencv.org/4.x/d3/dc0/group_imgproc_shape.html#ga0012a5fdaea70b8a9970165d98722b4c)