

[Blog Home](#)[Data Science](#)[Categories](#)[Free Courses](#)

Data Science Tutorials

[Machine Learning](#)[Machine Learning Tutorials](#)[Deep Learning](#)[Big Data](#)[Big Data Tutorials](#)[Python Tutorials](#)[Python Data Science Tutorials](#)[Blockchain Tutorials](#)[Apache Spark Tutorials](#)[SQL & NoSQLs](#)[R Programming Tutorials](#)[Java Script Tutorials](#)[AngularJS Tutorials](#)[MongoDB Tutorials](#)[SAS Tutorials](#)[SAP HANA Tutorials](#)[AI Tutorials](#)

Python is the pool of libraries. It has numerous libraries for real-world applications. One such library is OpenCV.

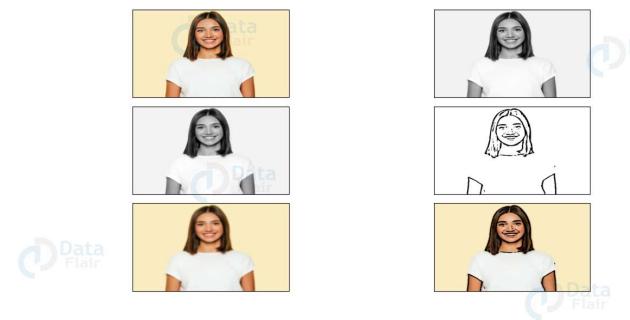
-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +
-

OpenCV is a cross-platform library used for Computer Vision. It includes applications for video and image capturing and processing. It is majorly used in image transformation, object detection, face recognition, and many other stunning applications.

What are we going to build?

At the end of this article, we aim to transform images into its cartoon. Yes, we will CARTOONIFY the images. Thus, we will build a python application that will transform an image into its cartoon using OpenCV.

We aim to build an application which looks like:



- Python Projects +
- Python Django Projects +
- Machine Learning Proj... +
- Deep Learning Projects +
- AI Projects +
- Python Interview Questi... +
- Python Quiz +

```

 7.5
76 fig, axes = plt.subplots(3,2, figsize=(8,8), subplot_kw={'xticks':[], 'yticks':[]}, gridspec_kw={'wspace':0, 'hspace':0})
77 for i, ax in enumerate(axes.flat):
78     ax.imshow(images[i], cmap='gray')
79
80 save1=Button(top,text="Save cartoon image",command=lambda:saveImage())
81 save1.configure(background="#364156", foreground="white", font=('calibri',10,'bold'))
82 save1.pack(side=TOP,pady=5)
83
84 plt.show()
85
86 def save(ReSized6,ImagePath):
87     #saving an image using cv2
88     newName="cartoonified_"
89     path1 = os.path.dirname(ImagePath)
90     extension=os.path.splitext(ImagePath)[1]
91     path = os.path.join(path1, newName+extension)
92     cv2.imwrite(path, cv2.cvtColor(ReSized6, cv2.COLOR_RGB2BGR))
93     I= "Image saved by name " + newName + " at "+ path
94     tk.messagebox.showinfo(title=None, message=I)
95
96 uploadButton=top,text="Cartoonify an Image",command=upload,padx=10,pady=5
97 upload.configure(background="#364156", foreground="white", font=('calibri',10,'bold'))

```

Let's head toward creating something
CREATIVE.

Original Image:



Steps to develop Image Cartoonifier

Download Image
Cartoonifies Code

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

Please download source code of Image
Cartoonfier Project: [Cartoonify an Image in Python](#)

Step 1: Importing the required modules

We will import the following modules:

- CV2: Imported to use OpenCV for image processing
- easygui: Imported to open a file box. It allows us to select any file from our system.
- Numpy: Images are stored and processed as numbers. These are taken as arrays. We use NumPy to deal with arrays.
- Imageio: Used to read the file which is chosen by file box using a path.
- Matplotlib: This library is used for visualization and plotting. Thus, it is imported to form the plot of images.
- OS: For OS interaction. Here, to read the path and save images to that path.

Code:

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

```

1. import cv2 #for image processing
2. import easygui #to open the
filebox
3. import numpy as np #to store
image
4. import imageio #to read image
stored at particular path
5. import sys
6. import matplotlib.pyplot as plt
7. import os
8. import tkinter as tk
9. from tkinter import filedialog
10. from tkinter import *
11. from PIL import ImageTk, Image

```

Step 2: Building a File Box to choose a particular file

In this step, we will build the main window of our application, where the buttons, labels, and images will reside. We also give it a title by title() function.

Code:

```

1. """ fileopenbox opens the box to
choose file
2. and help us store file path as
string """
3. def upload():
4.
ImagePath=easygui.fileopenbox()
cartoonify(ImagePath)

```

Explanation:

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

The above code opens the file box, i.e the pop up box to choose the file from the device, which opens every time you run the code. fileopenbox() is the method in easyGUI module which returns the path of the chosen file as a string.

NOTE: Now, all the operation will be done on the button click, thus all the below steps are the part of function cartoonify (ImagePath)

```
1. def cartoonify(ImagePath):
```

Step 3: How is an image stored?

Now, just think, how will a program read an image? For a computer, everything is just numbers. Thus, in the below code, we will convert our image into a numpy array.

Code:

```
1. #read the image
2.         originalimage =
cv2.imread(ImagePath)
```

```
3.     originalimage =
4.         cv2.cvtColor(originalimage,
5.                     cv2.COLOR_BGR2RGB)
6.         #print(image) # image is stored
7.         #in form of numbers
8.         # confirm that image is chosen
9.         if originalimage is None:
10.             print("Can not find any
11.                   image. Choose appropriate file")
12.             sys.exit()
13.
14.             ReSized1 =
15.                 cv2.resize(originalimage, (960,
16.                                         540))
17.                 #plt.imshow(ReSized1,
18.                           cmap='gray')
```

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

Explanation:

Imread is a method in cv2 which is used to store images in the form of numbers. This helps us to perform operations according to our needs. The image is read as a numpy array, in which cell values depict R, G, and B values of a pixel.

NOTE: We resize the image after each transformation to display all the images on a similar scale at last.

Beginning with image transformations:

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

To convert an image to a cartoon, multiple transformations are done. Firstly, an image is converted to a Grayscale image. Yes, similar to the old day's pictures.! Then, the Grayscale image is smoothed, and we try to extract the edges in the image. Finally, we form a color image and mask it with edges. This creates a beautiful cartoon image with edges and lightened color of the original image.

Let's start with these transformations to convert an image to its cartoon image.

Step 4: Transforming an image to grayscale

Code:

```
1. #converting an image to grayscale
2. grayScaleImage =
   cv2.cvtColor(originalImage,
   cv2.COLOR_BGR2GRAY)
3. ReSized2 =
   cv2.resize(grayScaleImage, (960,
   540))
4. #plt.imshow(ReSized2,
   cmap='gray')
```

Explanation:

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

cvtColor(image, flag) is a method in cv2 which is used to transform an image into the color space mentioned as ‘flag’. Here, our first step is to convert the image into grayscale. Thus, we use the BGR2GRAY flag. This returns the image in grayscale. A grayscale image is stored as grayScaleImage.

After each transformation, we resize the resultant image using the resize() method in cv2 and display it using imshow() method. This is done to get more clear insights into every single transformation step.

The above code will generate the following output:

-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +
-

Step 5: Smoothening a grayscale image

Code:

```
1. #applying median blur to smoothen  
an image  
2. smoothGrayScale =  
cv2.medianBlur(grayScaleImage, 5)  
3. ReSized3 =  
cv2.resize(smoothGrayScale, (960,  
540))  
4. #plt.imshow(ReSized3, cmap='gray')
```

Explanation:

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

To smoothen an image, we simply apply a blur effect. This is done using medianBlur() function. Here, the center pixel is assigned a mean value of all the pixels which fall under the kernel. In turn, creating a blur effect.

The above code generates the following output:

Step 6: Retrieving the edges of an image

Code:

```
1. #retrieving the edges for cartoon  
2. effect  
3. #by using thresholding technique  
4. getEdge =  
5. cv2.adaptiveThreshold(smoothGraySca  
255,  
6. cv2.ADAPTIVE_THRESH_MEAN_C,  
7. cv2.THRESH_BINARY, 9, 9)  
8. ReSized4 = cv2.resize(getEdge, (96  
540))  
9. #plt.imshow(ReSized4, cmap='gray')
```

Python Projects +

Python Django Projects +

Machine Learning Proj... +

Deep Learning Projects +

AI Projects +

Python Interview Questi... +

Python Quiz +

Explanation:

Cartoon effect has two specialties:

1. Highlighted Edges
2. Smooth colors

In this step, we will work on the first specialty.

Here, we will try to retrieve the edges and highlight them. This is attained by the adaptive thresholding technique. The threshold value is the mean of the neighborhood pixel values area minus the constant C. C is a constant that is subtracted from the mean or weighted sum of the neighborhood pixels. Thresh_binary is the type of threshold applied, and the remaining parameters determine the block size.

The above code will generate output like
below:

-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +
-

Step 7: Preparing a Mask Image

Code:

```
1. #applying bilateral filter to
   remove noise
2. #and keep edge sharp as required
3. colorImage =
   cv2.bilateralFilter(originalImage,
   9, 300, 300)
4. ReSized5 = cv2.resize(colorImage,
   (960, 540))
5. #plt.imshow(ReSized5, cmap='gray')
```

Explanation:

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

In the above code, we finally work on the second specialty. We prepare a lightened color image that we mask with edges at the end to produce a cartoon image. We use bilateralFilter which removes the noise. It can be taken as smoothening of an image to an extent.

The third parameter is the diameter of the pixel neighborhood, i.e, the number of pixels around a certain pixel which will determine its value. The fourth and Fifth parameter defines sigmaColor and sigmaSpace. These parameters are used to give a sigma effect, i.e make an image look vicious and like water paint, removing the roughness in colors.

Yes, it's similar to BEAUTIFY or AI effect in cameras of modern mobile phones.

The above code generates the following output:

-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +
-

Step 8: Giving a Cartoon Effect

Code:

```
1. #masking edged image with our  
"BEAUTIFY" image  
2. cartoonImage =  
cv2.bitwise_and(colorImage,  
colorImage, mask=getEdge)  
3.  
4. ReSized6 =  
cv2.resize(cartoonImage, (960,  
540))  
5. #plt.imshow(ReSized6, cmap='gray')
```

Explanation:

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

So, let's combine the two specialties. This will be done using MASKING. We perform bitw^{ing} and on two images to mask them. Remember, images are just numbers?

Yes, so that's how we mask edged image on our "BEAUTIFY" image.

This finally CARTOONIFY our image!

The above code will generate output like below:

Step 9: Plotting all the transitions together

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

Code:

```
1. # Plotting the whole transition
2. images=[ReSized1, ReSized2,
   ReSized3, ReSized4, ReSized5,
   ReSized6]
3. fig, axes = plt.subplots(3,2,
   figsize=(8,8), subplot_kw=
   {'xticks':[], 'yticks':[]},
   gridspec_kw=dict(hspace=0.1,
   wspace=0.1))
4. for i, ax in enumerate(axes.flat):
5.     ax.imshow(images[i],
   cmap='gray')
6. //save button code
7. plt.show()
```

Explanation:

To plot all the images, we first make a list of all the images. The list here is named “images” and contains all the resized images. Now, we create axes like subl=plots in a plot and display one-one images in each block on the axis using imshow() method.

plt.show() plots the whole plot at once after we plot on each subplot.

The above code will generate output like
below:

-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +
-

Step 10: Functionality of save button

```
1. def save(ReSized6, ImagePath):  
2.     #saving an image using imwrite  
3.     newName="cartoonified_Image"  
4.     path1 =  
5.     os.path.dirname(ImagePath)  
6.  
7.     extension=os.path.splitext(ImagePat  
[1]
```

```
6.         path = os.path.join(path1,
7.                               newName+extension)
8.         cv2.imwrite(path,
9.                     cv2.cvtColor(ReSized6,
10.                               cv2.COLOR_RGB2BGR))
11.         I = "Image saved by name " +
12.             newName +" at "+ path
13.
14.         tk.messagebox.showinfo(title=None,
15.                                 message=I)
```

Python Projects +

Python Django Projects +

Machine Learning Proj... +

Deep Learning Projects +

AI Projects +

Python Interview Questi... +

Python Quiz +

Explanation:

Here, the idea is to save the resultant image. For this, we take the old path, and just change the tail (name of the old file) to a new name and store the cartoonified image with a new name in the same folder by appending the new name to the head part of the file.

For this, we extract the head part of the file path by `os.path.dirname()` method. Similarly, `os.path.splitext(ImagePath)[1]` is used to extract the extension of the file from the path.

Here, `newName` stores “Cartoonified_Image” as the name of a new file. `os.path.join(path1, newName + extension)` joins the head of path to the newname and extension. This forms the complete path for the new file.

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

imwrite() method of cv2 is used to save the file at the path mentioned. cv2.cvtColor(ReSize^cv2.COLOR_RGB2BGR) is used to assure that no color get extracted or highlighted while we save our image. Thus, at last, the user is given confirmation that the image is saved with the name and path of the file.

Once executed with button, this function will result in below output:

Step 11: Making the main window

-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +

```
1. top=tk.Tk()
2. top.geometry('400x400')
3. top.title('Cartoonify Your Image !')
4. top.configure(background='white')
5.
label=Label(top,background="#CDCDCD"
font=('calibri',20,'bold'))
```

Step 12: Making the Cartoonify button in the main window

```
1. upload=Button(top,text="Cartoonify
Image",command=upload,padx=10,pady=
2.
upload.configure(background='#36415
foreground='white',font=
('calibri',10,'bold'))
3. upload.pack(side=TOP,pady=50)
```

-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +
-

Step 13: Making a Save button in the main window

```
1. save1=Button(top,text="Save cartoonified image",command=lambda:  
    save(ImagePath,  
    ReSized6),padx=30,pady=5)  
2.  
    save1.configure(background='#364156'  
    foreground='white',font=  
    ('calibri',10,'bold'))  
3. save1.pack(side=TOP,pady=50)
```

The above code makes a button as soon as the image transformation is done. It gives an

option to the user to save cartoonified image.

-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +
-

Step 14: Main function to build the tkinter window

```
1. top.mainloop()
```

The Final Result:

-
- Python Projects +
 - Python Django Projects +
 - Machine Learning Proj... +
 - Deep Learning Projects +
 - AI Projects +
 - Python Interview Questi... +
 - Python Quiz +
-

Summary

Yes, now you have a reason to tease your sibling by saying “You look like a cartoon”. Just cartoonify his/ her image, and show it!

We have successfully developed Image Cartoonifier with OpenCV in Python. This is the magic of openCV which let us do miracles. We suggest you make a photo editor of your own and try different effects.

Your opinion matters

Please write your valuable feedback about
DataFlair on [Google](#) | [Facebook](#)

Python Projects +

Python Django Projects +

Machine Learning Proj... +

Deep Learning Projects +

AI Projects +

Python Interview Questi... +

Python Quiz +

Tags: cartoonify image deep learning project

machine learning project opencv project Python project

15 RESPONSES

[Comments 15](#)

[Pingbacks 0](#)

Rajatha S ⌂ [October 8, 2020 at 11:31 pm](#)

Can we use visual studio code to run the code?

Do we require any database?

[Reply](#)

Amit Baap ⌂ [January 10, 2021 at 12:15 pm](#)

omg what a question

[Reply](#)

Jcak ⌂ [June 13, 2022 at 2:12 pm](#)

Good question, you are good question

[Reply](#)

jasmeen kaur ⌂ [November 17, 2020 at 12:16 pm](#)

the top.mainloop() function taking too long in
the jupyter notebook to execute, the pop

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

window for selecting an image is not showingng
.. i tried the other method too to use the source code and directly select the image the it is showing no result pls conntact me as soon as possible

[Reply](#)

Jyothi Panuganti Ⓜ December 2, 2020 at 5:46 pm

NameError: name 'originalimage' is not defined
i got this error before that easygui is not popping up any dialog box to upload the file

[Reply](#)

Jyothi Panuganti Ⓜ December 4, 2020 at 3:16 pm

```
TclError Traceback (most recent call last)
in ()
2 matplotlib.use('Agg')
3 import matplotlib.pyplot as plt
--> 4 top=tk.Tk()
5 top.geometry('400×400')
6 top.title('Cartoonify Your Image !')

/usr/lib/python3.6/tkinter/__init__.py in
__init__(self, screenName, baseName,
className, useTk, sync, use)
2021 baseName = baseName + ext
2022 interactive = 0
-> 2023 self.tk = _tkinter.create(screenName,
baseName, className, interactive,
wantobjects, useTk, sync, use)
```

2024 if useTk:

2025 self._loadtk()

TclError: no display name and no \$DISPLAY
environment variable

Please help me with these errors

[Reply](#)

Md. Rakibul Hasan

⌚ [December 18, 2020 at 2:33 pm](#)

Yes you can use visual studio code. No
database is required.

[Reply](#)

uzumaki ⌚ [February 18, 2021 at 4:59 pm](#)

thank you for sharing this!
how does this come under machine learning
project ?

[Reply](#)

tilak ved ⌚ [March 26, 2021 at 11:22 am](#)

where to get cv2 and all libraries in
anaconda??

[Reply](#)

imegos ⌚ [May 8, 2021 at 12:01 am](#)

cv2.error: OpenCV(4.5.2)
C:\Users\runneradmin\AppData\Local\Temp\pip-
req-build-
vi271kac\opencv\modules\imgproc\src\color.cpp:182:

| | |
|----------------------------|---|
| Python Projects | + |
| Python Django Projects | + |
| Machine Learning Proj... | + |
| Deep Learning Projects | + |
| AI Projects | + |
| Python Interview Questi... | + |
| Python Quiz | + |

error: (-215:Assertion failed) !_src.empty() in
function ‘cv::cvtColor’
help pleass :/

[Reply](#)

manjiri ⏴ July 23, 2021 at 4:40 pm

did you get the solution for that?

[Reply](#)

sree ⏴ August 1, 2021 at 11:12 am

how to run the code????

[Reply](#)

Cris D ⏴ January 10, 2022 at 5:28 pm

sure Rajatha you can use visual studio code
and a database its not necessary 😊

[Reply](#)

sai ⏴ November 23, 2022 at 10:53 pm

same error bro do you have solution

[Reply](#)

Aman ⏴ January 3, 2023 at 6:30 pm

GOOD WORK can easily be implemented in
PyCharm and Jupyter Notebook without any
error.

[Reply](#)

LEAVE A REPLY

Comment *

Python Projects +

Python Django Projects +

Machine Learning Proj... +

Deep Learning Projects +

AI Projects +

Python Interview Questi... +

Name ***Email *****Post Comment**

[Home](#) [About us](#) [Contact us](#) [Terms and Conditions](#) [Privacy Policy](#) [Disclaimer](#) [Write For Us](#) [Success Stories](#)



DataFlair © 2023. All Rights Reserved.

