</>

cartoonifier project

Team:

MOSTAFA MAZEN

NAL KHALED

ALI ELSAMAHY

AHEMED ELSHOURA

JANA AYMAN

The problems we encountered:

WHAT IS THE MOST SUITABLE WAY TO CONVERT THE IMAGE?

We start searching to find a suitable methods to give us the cartoon effect and to work efficiently with variety of images, then we figured out the k-means Color quantization algorithm

HOW TO GET USE OF GUI IN THE RIGHT WAY?

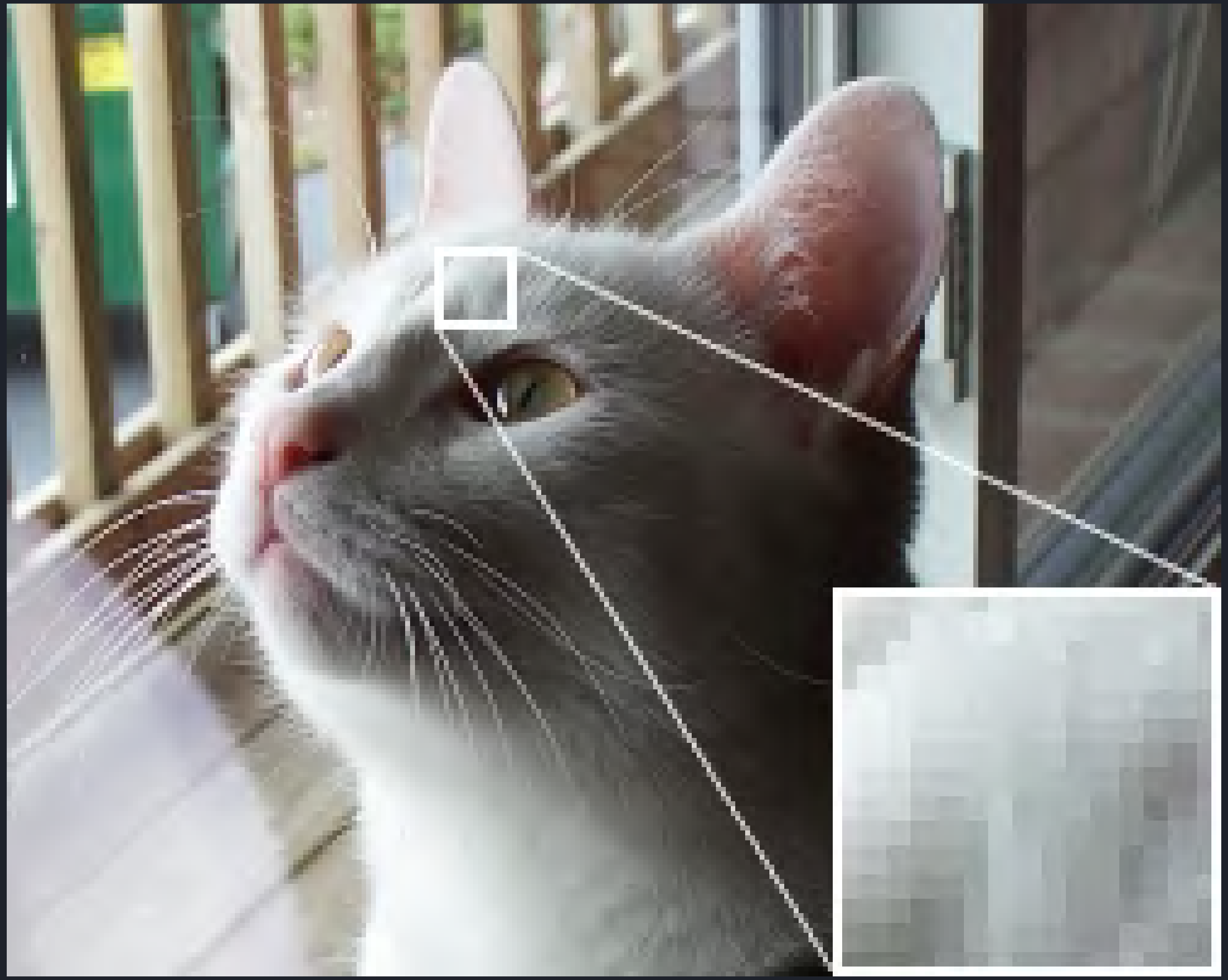
uploading image ,displaying it ,converting it then displaying it after converting and finally saving it ,these all are the problems we faced adding to that the live camera convertor ,which was a grate thing to add

Color quantization:

It is a process that reduces the number of distinct colors used in an image, usually with the intention that the new image should be as visually similar as possible to the original image. Color quantization is critical for displaying images with many colors on devices that can only display a limited number of colors, usually due to memory limitations, and enables efficient compression of certain types of images.

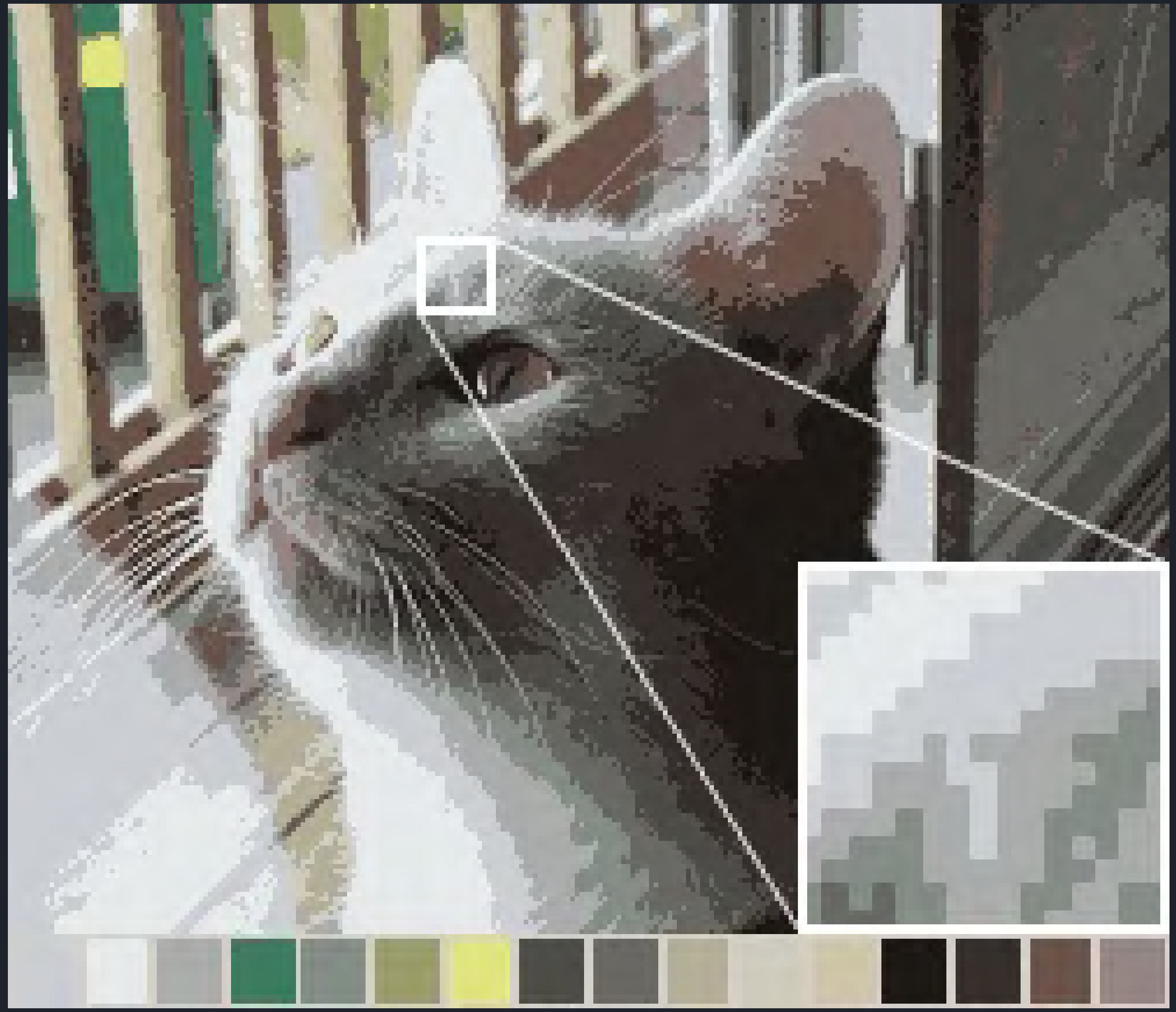
Pitch

COLOR QUANTIZATION



An example image in 24-bit RGB color

Each pixel can have integer values from 0 to 255



The same image reduced to a palette of 16 colors

colors specifically chosen to best represent the image.
chosen colors are represented in the palette.

A code snippet

```
def color_quantization(img, k):
    data=np.float32(img).reshape((-1, 3))
    criteria=(cv2.TERM_CRITERIA_EPS+cv2.TERM_CRITERIA_MAX_ITER,20,0.001)
    ret, label, center=cv2.kmeans(data, k , None, criteria, 10, cv2.KMEANS_RANDOM_CENTERS)
```

it's important to turn the image array into a float array of size 32 bits so the k-means algorithm can deal with it ,then we create the iteration termination criteria ,and finally we use the CV2 k-means algorithm adding the data and the criteria we created before to get the centroids and label

The value of the algorithm:

- Classify the colors according to the nearest centroid
- The ability to reshape pixels as image again

GUI

We used the tkinter library in building the GUI ,because it has the most appropriate components for our project from frames ,buttons and labels ,but We faced two great problems in the GUI.

IMAGE DISPLAY SECTION:

- How to display the image in the GUI after and before converting.
- How to display the different images with different sizes in the GUI?

LIVE CAMERA SECTION:

- How to employ camera in the project?
- How to display the live photo in the GUI after converting it?

solution of the problem:

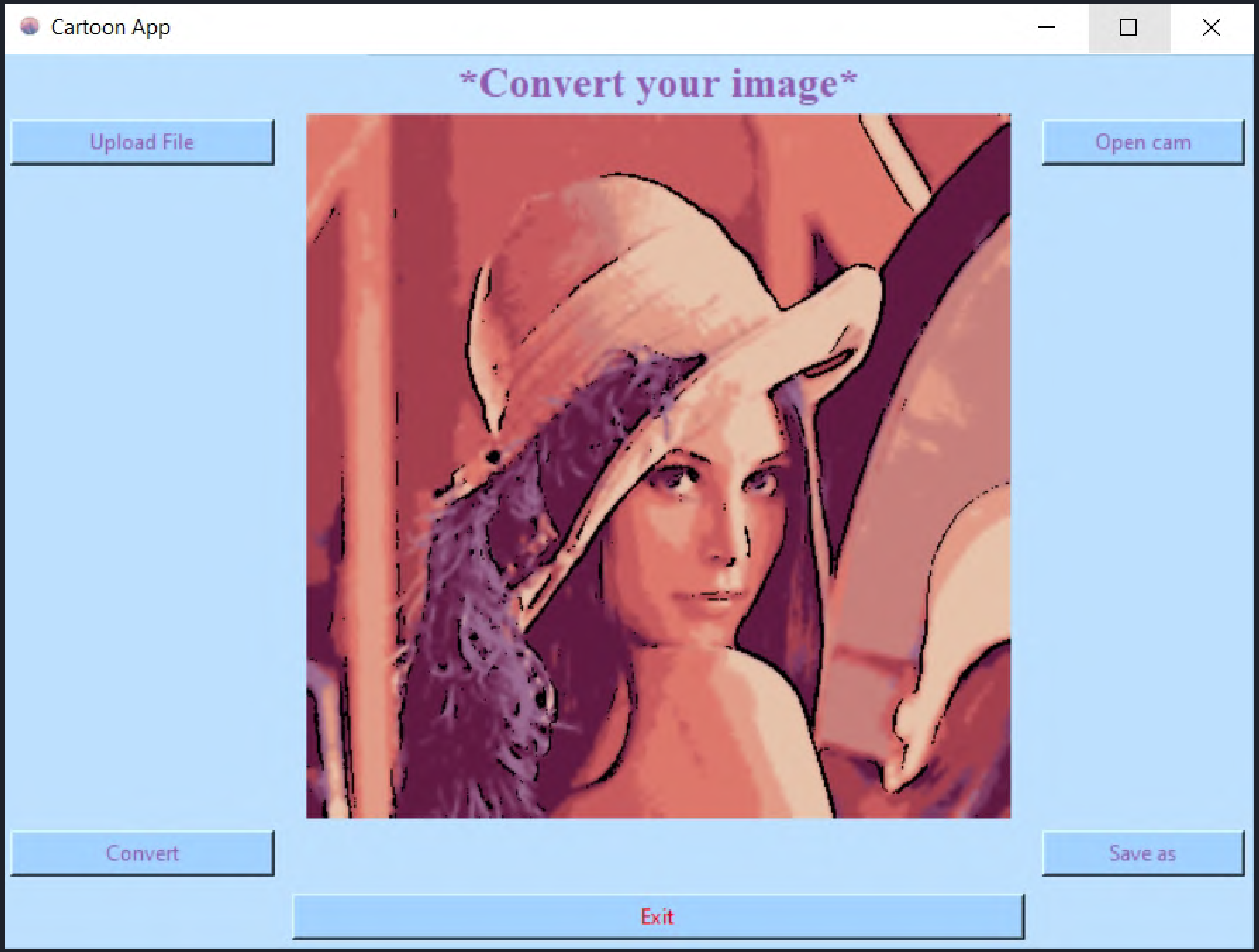
IMAGE DISPLAY SECTION:

We add upload ,convert and save buttons to get the image ,display it to the user ,convert it, display it again ,then save it.
We built a special function to resize the image without any distortion.

LIVE CAMERA SECTION:

We used the open cv library to access the camera and to convert the live photo we get from it ,then we used the tkinter library to display the converted result

final result:



</// >

Thank you

Pitch