In [14]:
```python
import numpy as np
import matplotlib.pyplot as plt
from skimage.io import imread, imshow, imsave
import cv2
```

In [15]:
```python
def viewImage(image, name_of_window):
    cv2.namedWindow(name_of_window, cv2.WINDOW_NORMAL)
    cv2.imshow(name_of_window, image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

In [16]:
```python
face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml
eye_cascade = cv2.CascadeClassifier("haarcascade_eye.xml")
image = cv2.imread("love.jpg")
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(
    gray,
    scaleFactor= 1.1,
    minNeighbors= 5,
    minSize=(35, 35)
)
```

In [17]:
```python
faces_detected = "Лиц обнаружено: " + format(len(faces))
print(faces_detected)
index = 1
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (255, 255, 0), 1)

    center = (x+int(w/2), y+int(h/2))
    radius = int(w/2)
    cv2.circle(image, center, radius, (0,255, 0), 2)

    a = image[y:y+h, x:x+w]
    b = gray[y:y+h, x:x+w]

    eyes = eye_cascade.detectMultiScale(b)
    for (x1, y1, w1, h1) in eyes:
        cv2.rectangle(a, (x1, y1), (x1+w1, y1+h1), (0, 0, 255), 1)
    index += 1

viewImage(image,faces_detected)
```

Лиц обнаружено: 1

In [29]:
```python
img= imread("cameo-apple-tree-1.jpg");
flood = cv2.imread("cameo-apple-tree-1.jpg");
flood=cv2.cvtColor(flood,cv2.COLOR_BGR2RGB)
seed = (300, 300)
cv2.floodFill(flood, None, seedPoint=seed, newVal=(255, 0, 0), loDiff=(1
plt.figure(figsize=(figure_size,figure_size))
plt.subplot(1,2,1),plt.imshow(img)
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(1,2,2),plt.imshow(flood)
plt.title('Segmented flood Fill' ), plt.xticks([]), plt.yticks([])
plt.show()
```
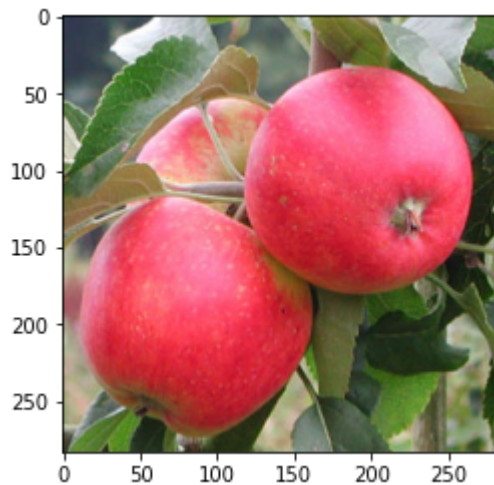
Original Image　　　　　　　　　　　　　　　　Segmented flood Fill

In [28]:
```python
%matplotlib inline
image = cv2.imread('pink-lady.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image)
```
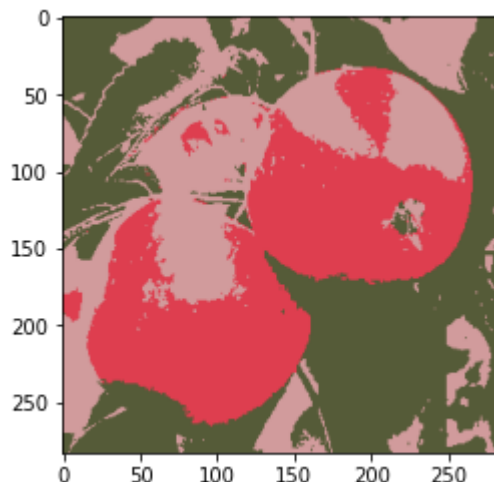
Out[28]: <matplotlib.image.AxesImage at 0x7fd806804490>



In [23]:
```python
pixel_vals = image.reshape((-1,3))
pixel_vals = np.float32(pixel_vals)
```

In [24]:
```python
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.8

k = 3
retval, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria, 10,

centers = np.uint8(centers)
segmented_data = centers[labels.flatten()]

segmented_image = segmented_data.reshape((image.shape))
plt.imshow(segmented_image)
```

Out[24]: <matplotlib.image.AxesImage at 0x7fd8066ebdc0>



In [ ]: