

Задание 6

В ходе выполнения работы использовалась библиотека Pillow PIL

```
from PIL import Image
```

Обучение происходило путем «скармливания» RGB пикселей с кожей, для это загружалась и конвертировалась в многомерный массив, а потом и в list картинка кожи на белом фоне

```
skin = Image.open('Skin.png').convert('RGB')
dataskin = np.array(skin)
dataskin = dataskin.reshape(skin.size[0] * skin.size[1], 3)
```



Рис 1. Фрагмент картинки с кожей

Если пиксель не белый, то добавляем его RGB- значения и метку кожа (1) в списки

```
for item in dataskin:
    if (item != listBack).any():
        listR.append(item.tolist()[0])
        listG.append(item.tolist()[1])
        listB.append(item.tolist()[2])
        listLabel.append(1)
```

Аналогично делаем с не кожей, присваивая метку «0».

Создаем датафрейм из этих листов и удаляем дубликаты

```
df = pd.DataFrame({'R':listR,'G':listG,'B':listB, 'label':listLabel})
print(df)

df = df.drop_duplicates(subset=['R', 'G', 'B'], keep=False)
print(df)
```

При подборе обучающих картинок я старался, чтобы количество меток кожи не кожи были примерно равны.

```
print(len(df[df["label"]==0]))
print(len(df[df["label"]==1]))
```

Создаем и обучаем классификатор

```

points_train, points_test, labels_train, labels_test =
train_test_split(df.iloc[:, :-1], df['label'],

test_size=0.25, random_state=0)
gnb = GaussianNB()
gnb.fit(points_train, labels_train)
prediction = gnb.predict(points_test)

```

Предсказываем значения меток для каждого пикселя картинки и создаем лист меток

```
listpredict = gnb.predict(datatest.reshape(test.size[0]*test.size[1],3))
```

Перекрашиваем все пиксели с меткой 1 в красный цвет

```

img = np.array(test)
row = 0
column = 0
for item in img:
    for i in item:
        if listpredict[row* len(item) + column] == 1:
            img[row][column] = replacement_color
            column += 1
        column=0
        row+=1

img2 = Image.fromarray(img, mode='RGB')
img2.show()

```

Получившиеся результаты:





