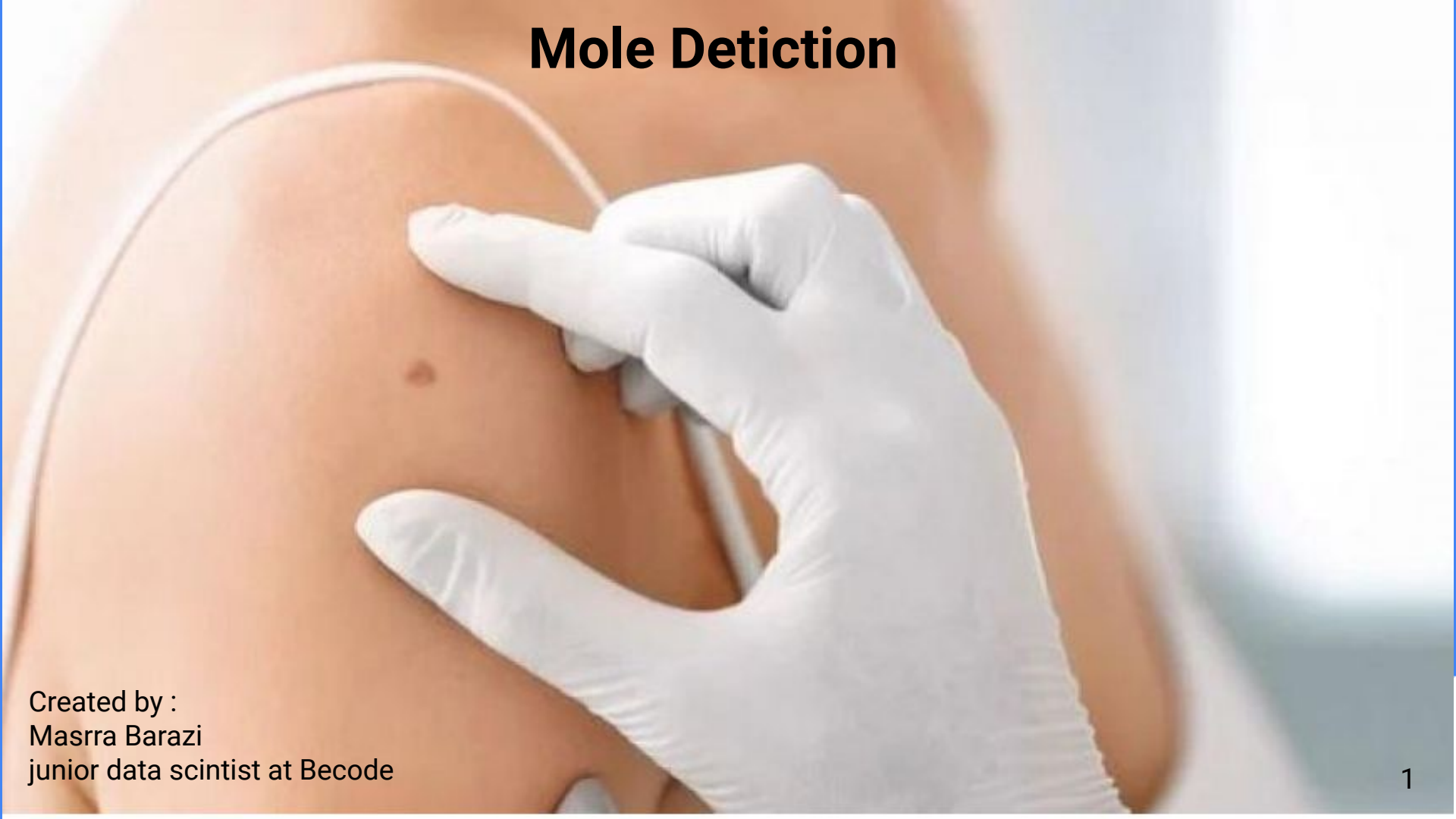
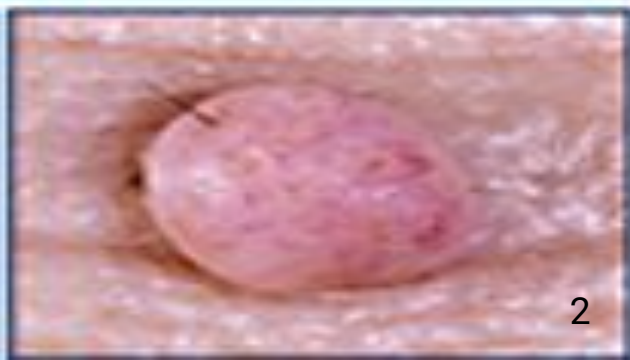
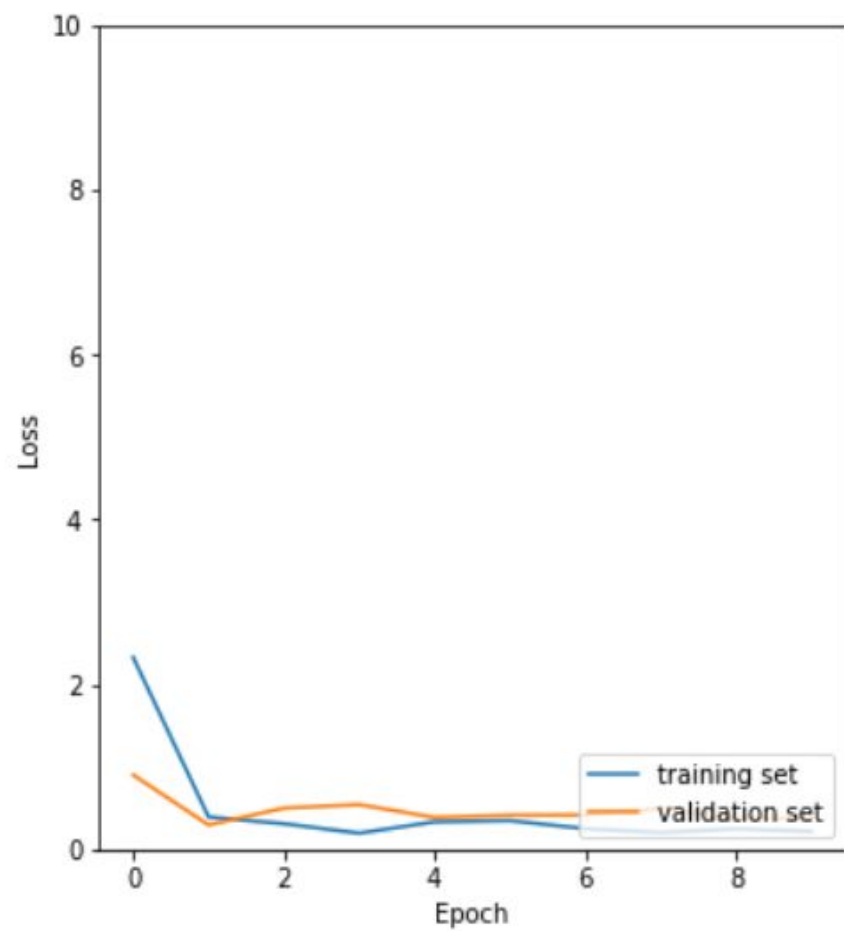
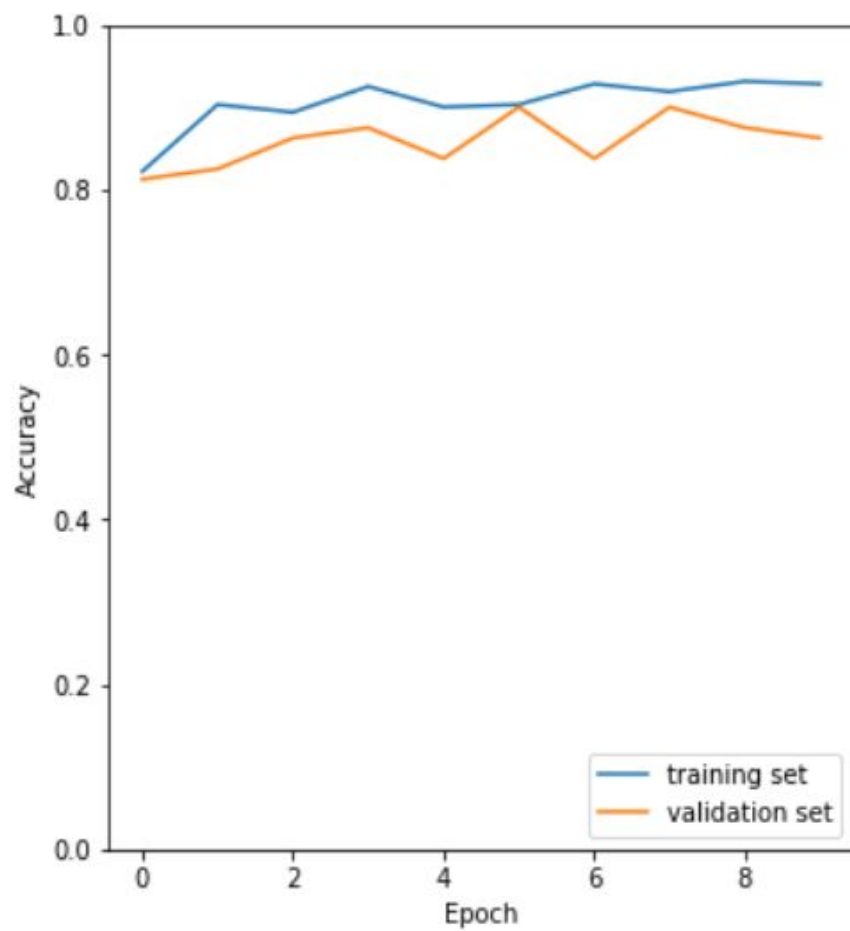


# Mole Detiction



Created by :  
Masrra Barazi  
junior data scintist at Bencode





# Steps to do the project

- 1-Import libraries and metadata.
- 2-Build a mapping between image ids and types of cancer.
- 3-Get the images and clean it .
- 4-Determain X and y(x is the pixells of the images and y is the types of the cancer)
- 5-Label y from (0 to 6)because we have 7 types of the cancer.
- 6-Split the data to x\_train,x\_val,x\_test,y\_train,y\_val,y\_test.
- 7-Import the model Resnet50.

# Steps to do the project

8-Set the input shape of the model to 224x224 pixels, with 3 color channels.

9-Freeze the imported layers so they cannot be retrained.

10-Adding flattening and dense layers.

11-Compile and fit the model.

12-Save the model.

13-Plot the training and the accuracy of the model.

14-Build the app to detect the type of the cancer.

# How to build an app

- 1- import the necessary libraries like(flask,request,render\_template)
- 2-put the name of the app
- 3-load the model
- 4-get the images and clean them
- 5-Convert your image pixels to a numpy array of values .

# How to build an app

6-Reshape your image dimensions so that the colour channels correspond to what your model expects.

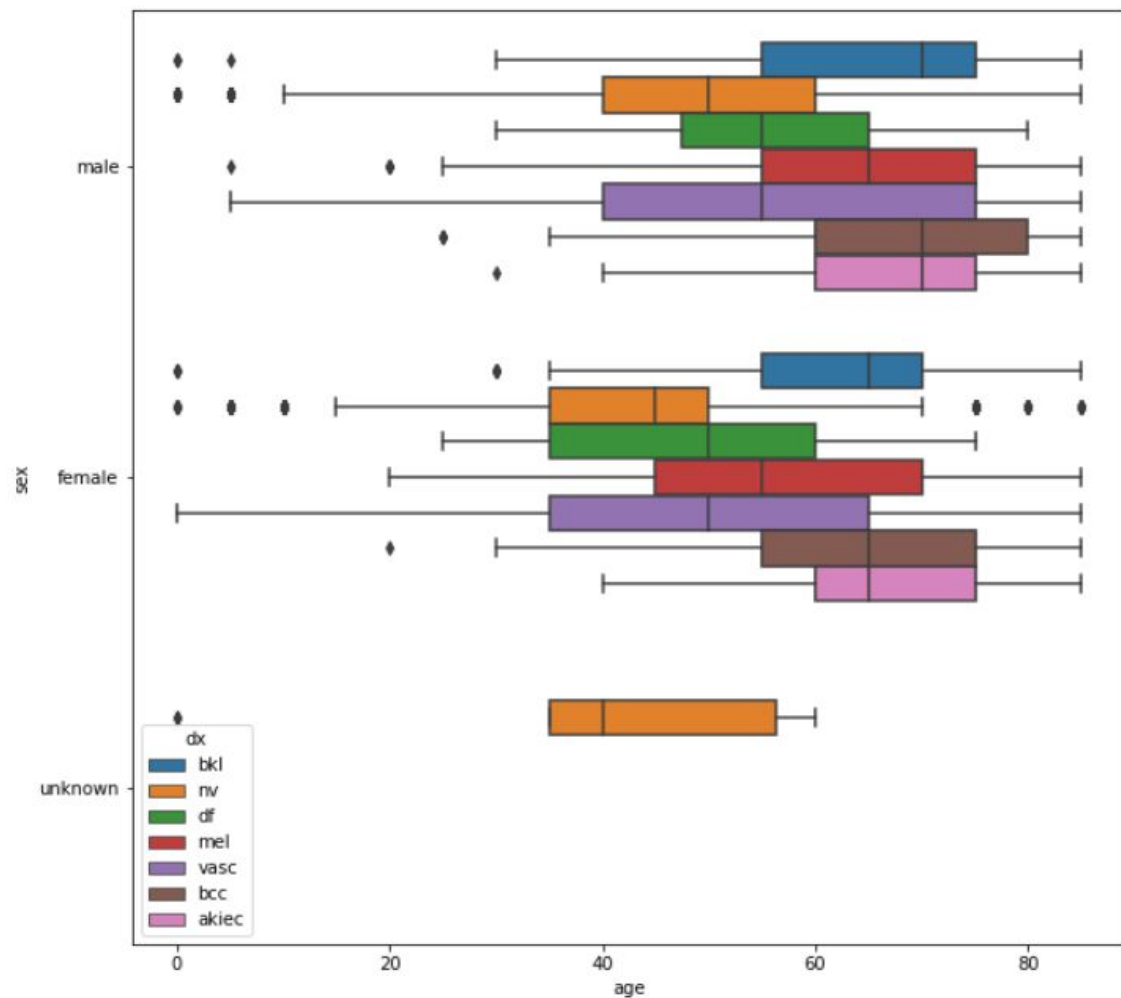
7-Preprocess your image with `preprocess_input`.

8-Apply the model and get the prediction.

9-Show the prediction to the user.

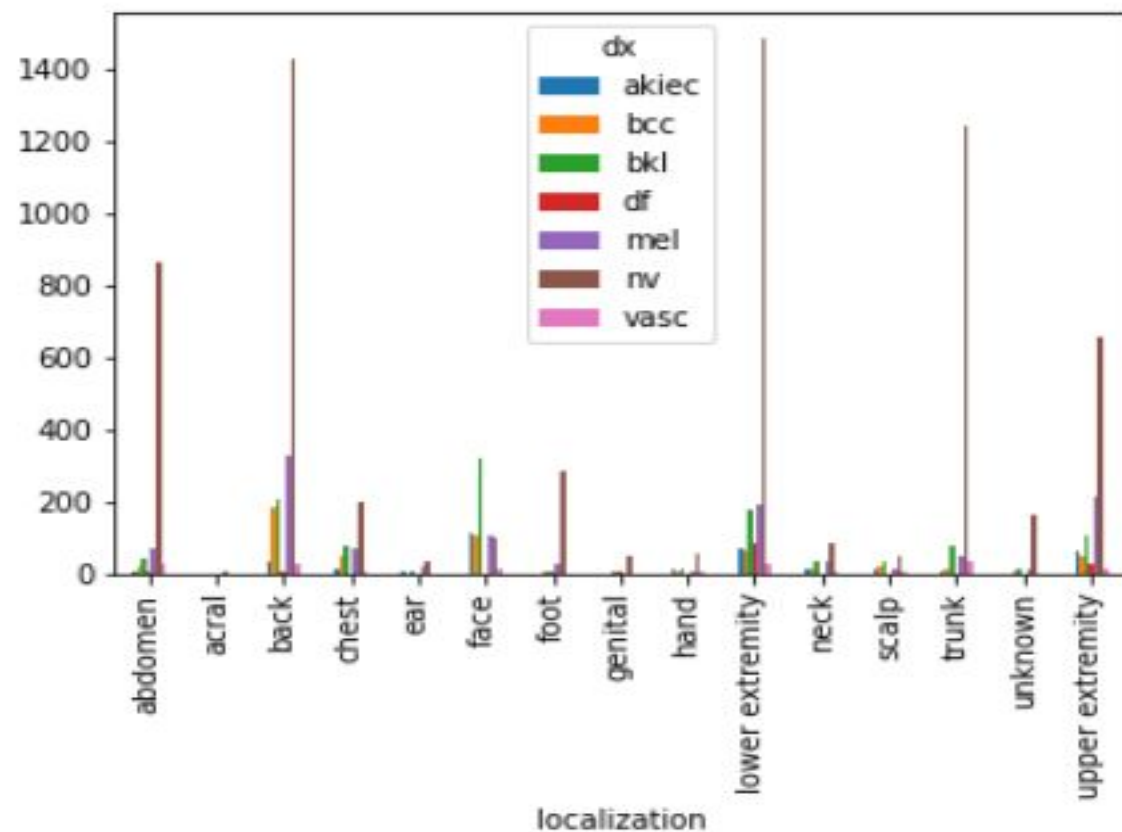
10-pass the information from python file to the html file.

```
<AxesSubplot:xlabel='age', ylabel='sex'>
```





```
: df.groupby('localization')['dx'].value_counts().unstack().plot(kind='bar')  
:  
: <AxesSubplot:xlabel='localization'>
```



# Problems :

- Technical issue
- The app couldn't work efficiently

Upload an Image:

file1

submit

The result is:

# Thank you for listening

