skincancer

March 31, 2024

[1]:

```
from sklearn.metrics import confusion_matrix
```

[2]:

```
from sklearn.metrics import confusion_matrix

# Example true labels and predicted labels
y_true = [0, 1, 0, 1, 0, 1]
y_pred = [0, 1, 1, 1, 0, 0]

# Compute confusion matrix
cm = confusion_matrix(y_true, y_pred)

print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[2 1]
 [1 2]]
```

[3]:

```
pip install plot_confusion_matrix
```

```
Requirement already satisfied: plot_confusion_matrix in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages
(0.0.2)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: matplotlib in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
plot_confusion_matrix) (3.8.3)
Requirement already satisfied: numpy in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
plot_confusion_matrix) (1.26.4)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (4.50.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (24.0)
Requirement already satisfied: pillow>=8 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
python-dateutil>=2.7->matplotlib->plot_confusion_matrix) (1.16.0)
```

[6]:

```python
from sklearn.model_selection import train_test_split
```

[9]:

```
pip install plot_confusion_matrix
```

```
Requirement already satisfied: plot_confusion_matrix in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages
(0.0.2)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: matplotlib in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
plot_confusion_matrix) (3.8.3)
Requirement already satisfied: numpy in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
plot_confusion_matrix) (1.26.4)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (4.50.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (24.0)
Requirement already satisfied: pillow>=8 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->plot_confusion_matrix) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
python-dateutil>=2.7->matplotlib->plot_confusion_matrix) (1.16.0)
```

[10]:

```python
import plot_confusion_matrix
```

[15]:

```
pip install --upgrade scikit-learn
```

```
Requirement already satisfied: scikit-learn in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages
(1.4.1.post1)
Requirement already satisfied: numpy<2.0,>=1.19.5 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (1.12.0)
Requirement already satisfied: joblib>=1.2.0 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\users\theja\appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (3.4.0)
Note: you may need to restart the kernel to use updated packages.
```

[ ]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import os
import itertools

from glob import glob
from PIL import Image
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
from tensorflow.keras import backend as K
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from keras.utils.vis_utils import plot_model
```

[4]:
```
os.getcwd()
```

[4]:
```
'C:\\Users\\theja'
```

[5]:
```
os.chdir("C:\\Users\\theja\\Downloads\\skin cancer.csv")
```

[6]:
```
df = pd.read_csv('HAM10000_metadata.csv')
df.head()
```

[6]:
```
     lesion_id      image_id   dx dx_type   age   sex localization
0  HAM_0000118  ISIC_0027419  bkl   histo  80.0  male        scalp
1  HAM_0000118  ISIC_0025030  bkl   histo  80.0  male        scalp
2  HAM_0002730  ISIC_0026769  bkl   histo  80.0  male        scalp
3  HAM_0002730  ISIC_0025661  bkl   histo  80.0  male        scalp
4  HAM_0001466  ISIC_0031633  bkl   histo  75.0  male          ear
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10015 entries, 0 to 10014
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   lesion_id     10015 non-null  object
 1   image_id      10015 non-null  object
 2   dx            10015 non-null  object
 3   dx_type       10015 non-null  object
 4   age           9958 non-null   float64
 5   sex           10015 non-null  object
 6   localization  10015 non-null  object
dtypes: float64(1), object(6)
memory usage: 547.8+ KB
```

[8]:
```
df.shape
```

[8]:
```
(10015, 7)
```

[9]:

```
lesion_type_dict = {
    'nv': 'Melanocytic nevi',
    'mel': 'Melanoma',
    'bkl': 'Benign keratosis-like lesions ',
    'bcc': 'Basal cell carcinoma',
    'akiec': 'Actinic keratoses',
    'vasc': 'Vascular lesions',
    'df': 'Dermatofibroma'
}
base_skin_dir = 'C:\\Users\\theja\\Downloads\\skin cancer.csv'
```

[10]:

```
imageid_path_dict = {os.path.splitext(os.path.basename(x))[0]: x
                     for x in glob(os.path.join(base_skin_dir, '*', '*.jpg'))}
```

[11]:

```
df['path'] = df['image_id'].map(imageid_path_dict.get)
df['cell_type'] = df['dx'].map(lesion_type_dict.get)
df['cell_type_idx'] = pd.Categorical(df['cell_type']).codes

df.head()
```

[11]:

```
     lesion_id      image_id   dx dx_type   age   sex localization  \
0  HAM_0000118  ISIC_0027419  bkl   histo  80.0  male        scalp
1  HAM_0000118  ISIC_0025030  bkl   histo  80.0  male        scalp
2  HAM_0002730  ISIC_0026769  bkl   histo  80.0  male        scalp
3  HAM_0002730  ISIC_0025661  bkl   histo  80.0  male        scalp
4  HAM_0001466  ISIC_0031633  bkl   histo  75.0  male          ear

                                               path  \
0  C:\Users\theja\Downloads\skin cancer.csv\HAM10…
1  C:\Users\theja\Downloads\skin cancer.csv\HAM10…
2  C:\Users\theja\Downloads\skin cancer.csv\HAM10…
3  C:\Users\theja\Downloads\skin cancer.csv\HAM10…
4  C:\Users\theja\Downloads\skin cancer.csv\HAM10…

                        cell_type  cell_type_idx
0  Benign keratosis-like lesions               2
1  Benign keratosis-like lesions               2
2  Benign keratosis-like lesions               2
3  Benign keratosis-like lesions               2
4  Benign keratosis-like lesions               2
```

[12]:

```
df.isna().sum()
```

[12]:

```
lesion_id         0
image_id          0
dx                0
dx_type           0
```

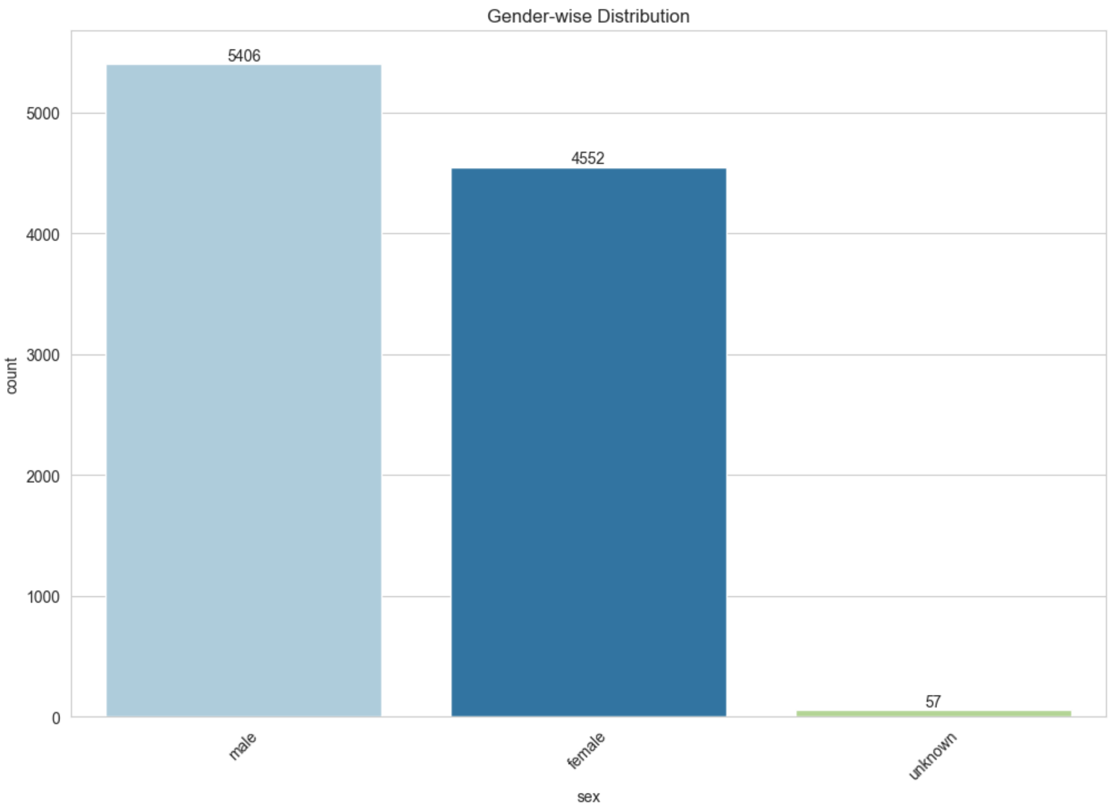
```
age               57
sex                0
localization       0
path               0
cell_type          0
cell_type_idx      0
dtype: int64
```

[13]:
```
df.isna().sum().sum()
```

[13]:
```
57
```

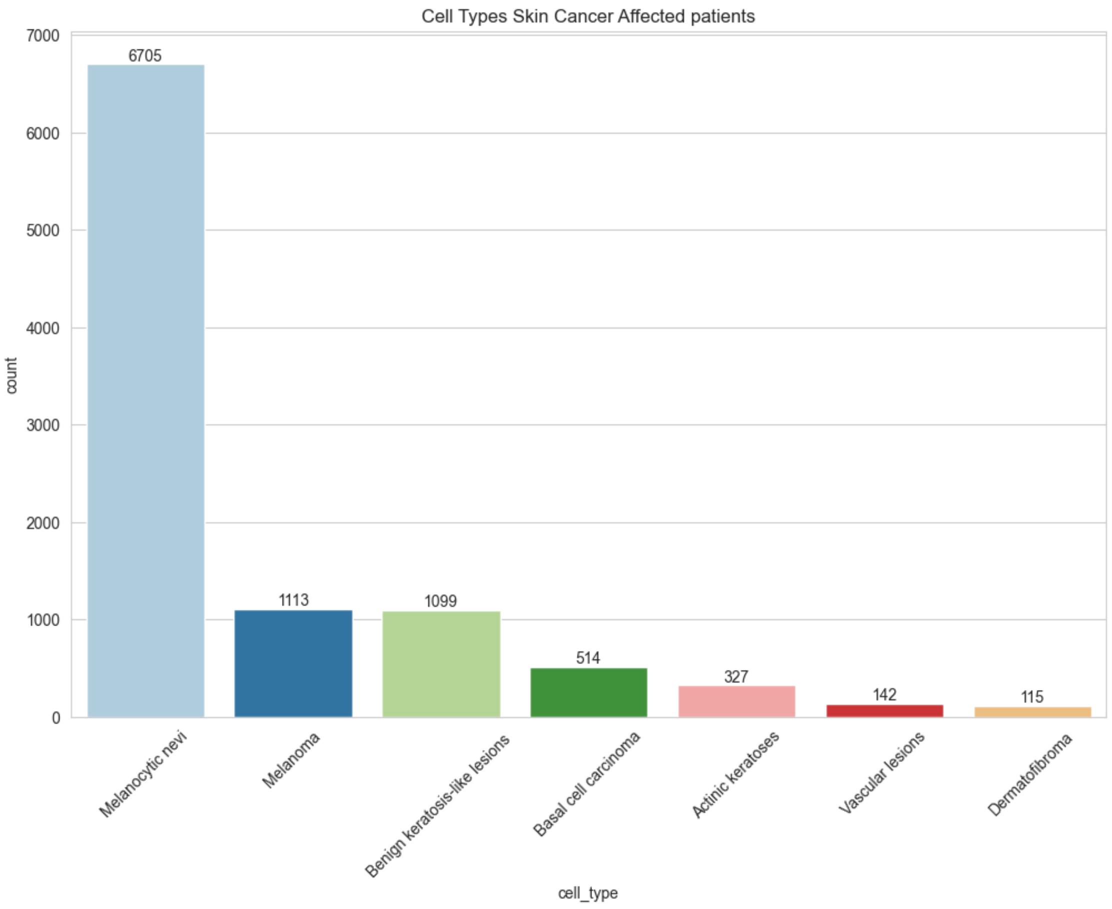
[14]:
```
sns.set_style('whitegrid')
colors = ['#87ace8','#e3784d', 'green']
fig,axes = plt.subplots(figsize=(12,8))

ax = sns.countplot(x='sex',data=df, palette = 'Paired')
for container in ax.containers:
    ax.bar_label(container)
plt.title('Gender-wise Distribution')
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\theja\AppData\Local\Temp\ipykernel_22352\1013143007.py:5:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  ax = sns.countplot(x='sex',data=df, palette = 'Paired')
```

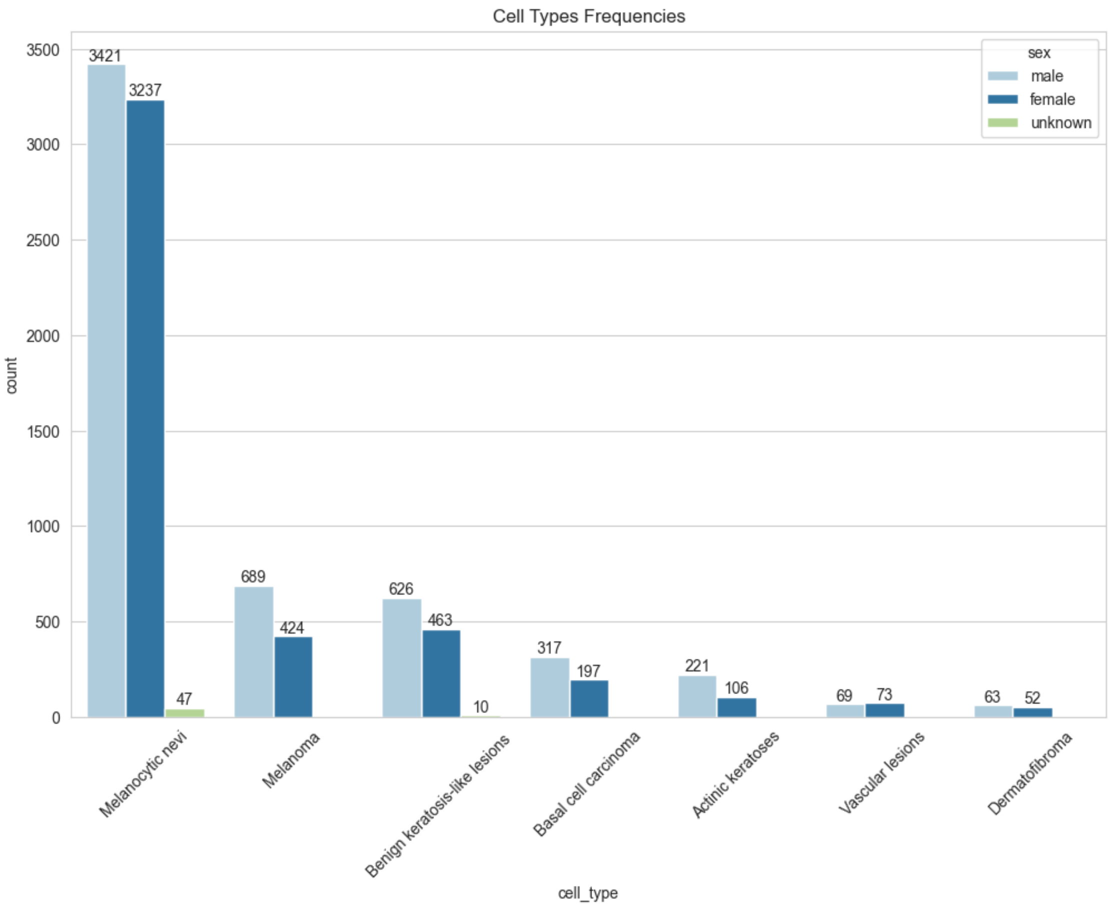


[15]:
```
sns.set_style('whitegrid')
fig,axes = plt.subplots(figsize=(12,8))
ax = sns.countplot(x='cell_type',data=df, order = df['cell_type'].
 ↪value_counts().index, palette = 'Paired')
for container in ax.containers:
    ax.bar_label(container)
plt.title('Cell Types Skin Cancer Affected patients')
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\theja\AppData\Local\Temp\ipykernel_22352\2900061788.py:3:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  ax = sns.countplot(x='cell_type',data=df, order =
df['cell_type'].value_counts().index, palette = 'Paired')
```

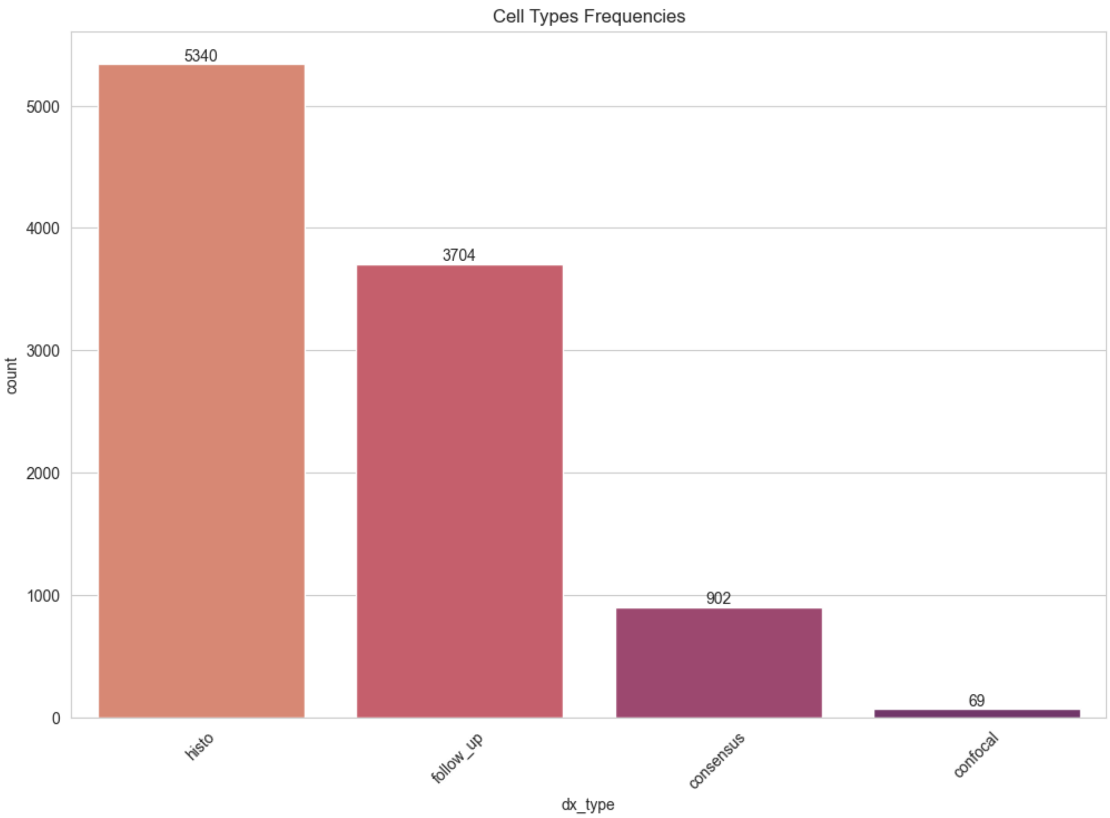


[16]:
```
sns.set_style('whitegrid')
fig,axes = plt.subplots(figsize=(12,8))
ax = sns.countplot(x='cell_type',hue='sex', data=df, order = df['cell_type'].
 ↪value_counts().index, palette = 'Paired')
for container in ax.containers:
    ax.bar_label(container)
plt.title('Cell Types Frequencies')
plt.xticks(rotation=45)
plt.show()
```

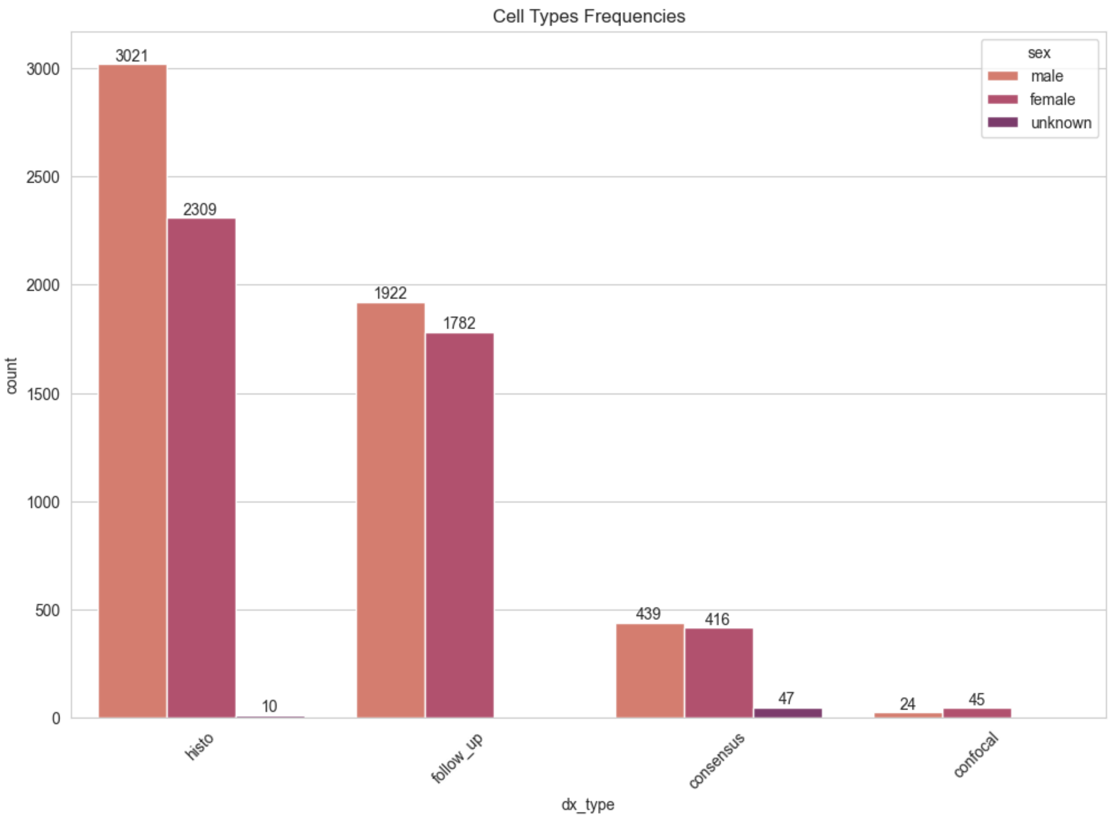


```
[17]: sns.set_style('whitegrid')
      fig,axes = plt.subplots(figsize=(12,8))
      ax = sns.countplot(x='dx_type',data=df, order = df['dx_type'].value_counts().
        ↪index, palette = 'flare')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Cell Types Frequencies')
      plt.xticks(rotation=45)
      plt.show()
```

```
C:\Users\theja\AppData\Local\Temp\ipykernel_22352\4047707394.py:3:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  ax = sns.countplot(x='dx_type',data=df, order =
df['dx_type'].value_counts().index, palette = 'flare')
```



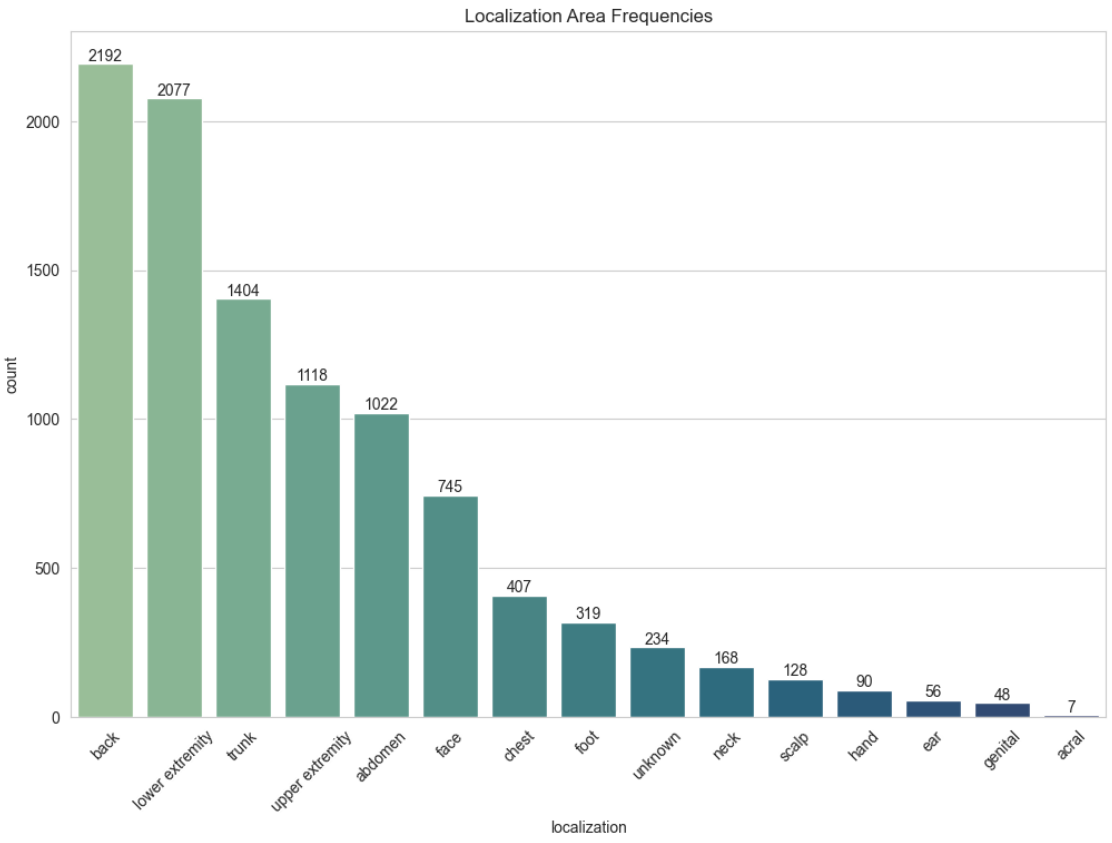
```
[18]: sns.set_style('whitegrid')
      fig,axes = plt.subplots(figsize=(12,8))
      ax = sns.countplot(x='dx_type',hue='sex', data=df, order = df['dx_type'].
       ↪value_counts().index, palette = 'flare')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Cell Types Frequencies')
      plt.xticks(rotation=45)
      plt.show()
```



[19]:

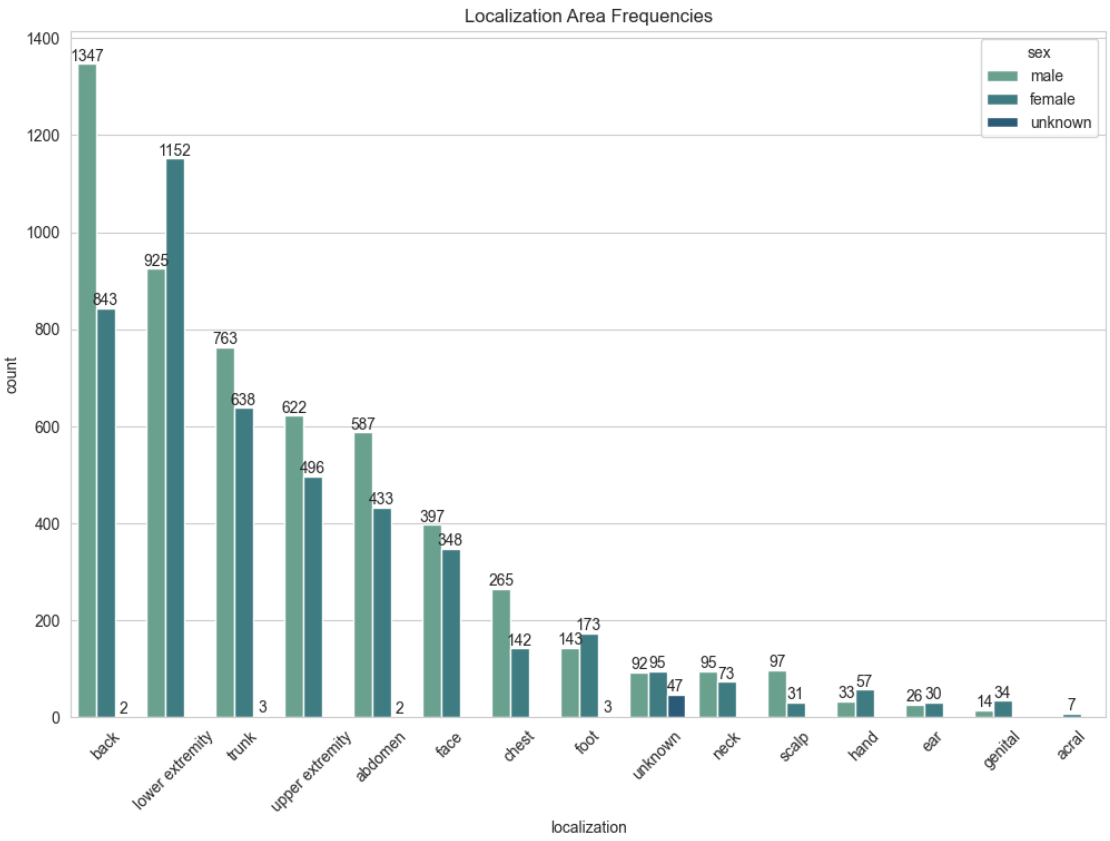
```
sns.set_style('whitegrid')
fig,axes = plt.subplots(figsize=(12,8))
ax = sns.countplot(x='localization',data=df, order = df['localization'].
  ↪value_counts().index, palette = 'crest')
for container in ax.containers:
    ax.bar_label(container)
plt.title('Localization Area Frequencies')
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\theja\AppData\Local\Temp\ipykernel_22352\1969139068.py:3:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  ax = sns.countplot(x='localization',data=df, order =
df['localization'].value_counts().index, palette = 'crest')
```

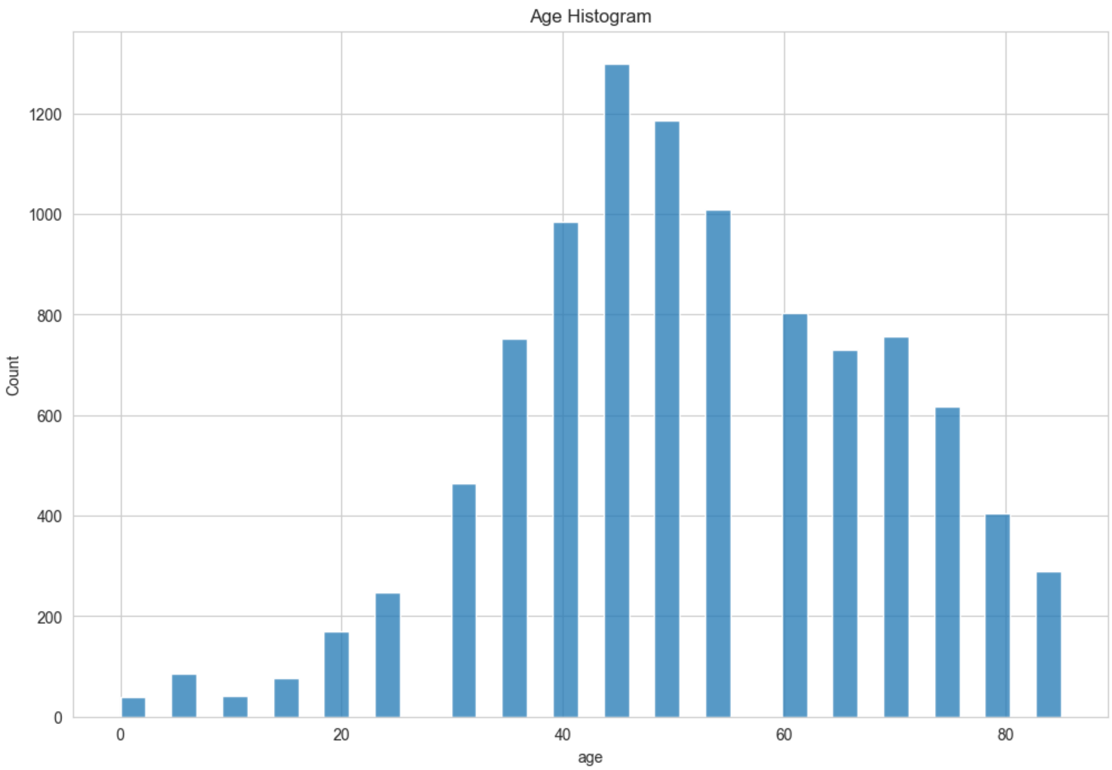


[20]:

```
sns.set_style('whitegrid')
fig,axes = plt.subplots(figsize=(12,8))
ax = sns.countplot(x='localization',hue='sex', data=df, order =␣
 ↪df['localization'].value_counts().index, palette = 'crest')
for container in ax.containers:
    ax.bar_label(container)
plt.title('Localization Area Frequencies')
plt.xticks(rotation=45)
plt.show()
```

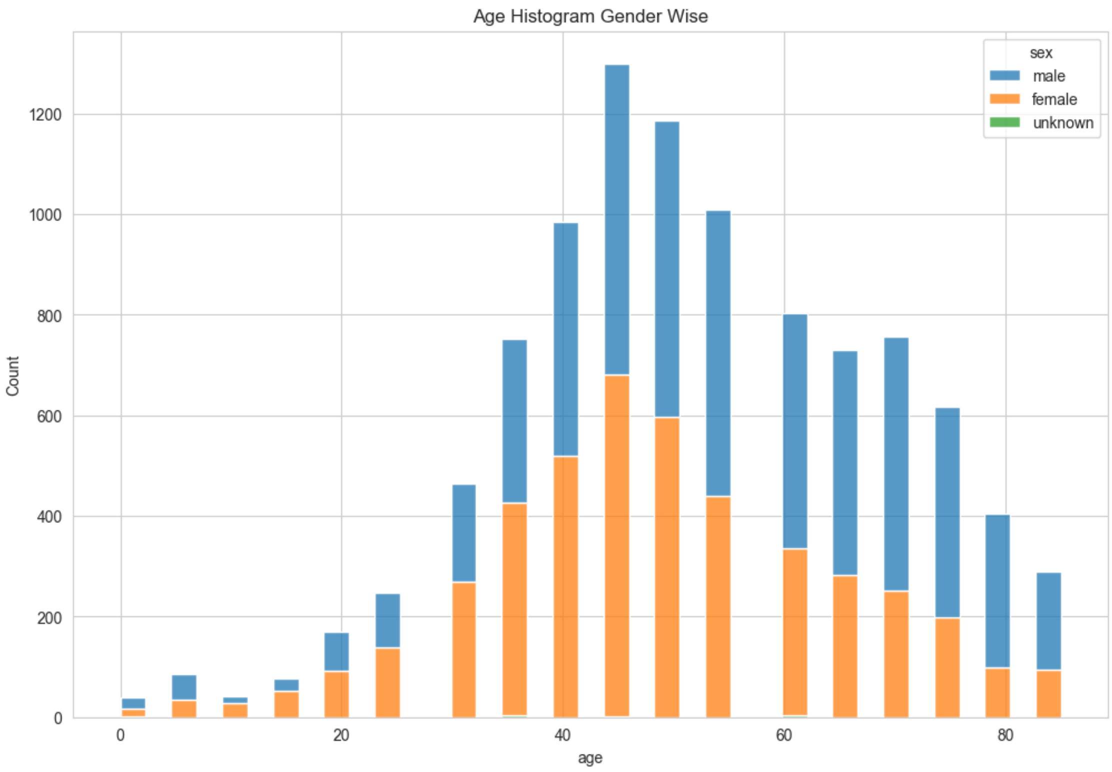


```
[21]: sns.set_style('whitegrid')
      fig,axes = plt.subplots(figsize=(12,8))
      ax = sns.histplot(data=df, x='age')
      plt.title('Age Histogram')
      plt.show()
```

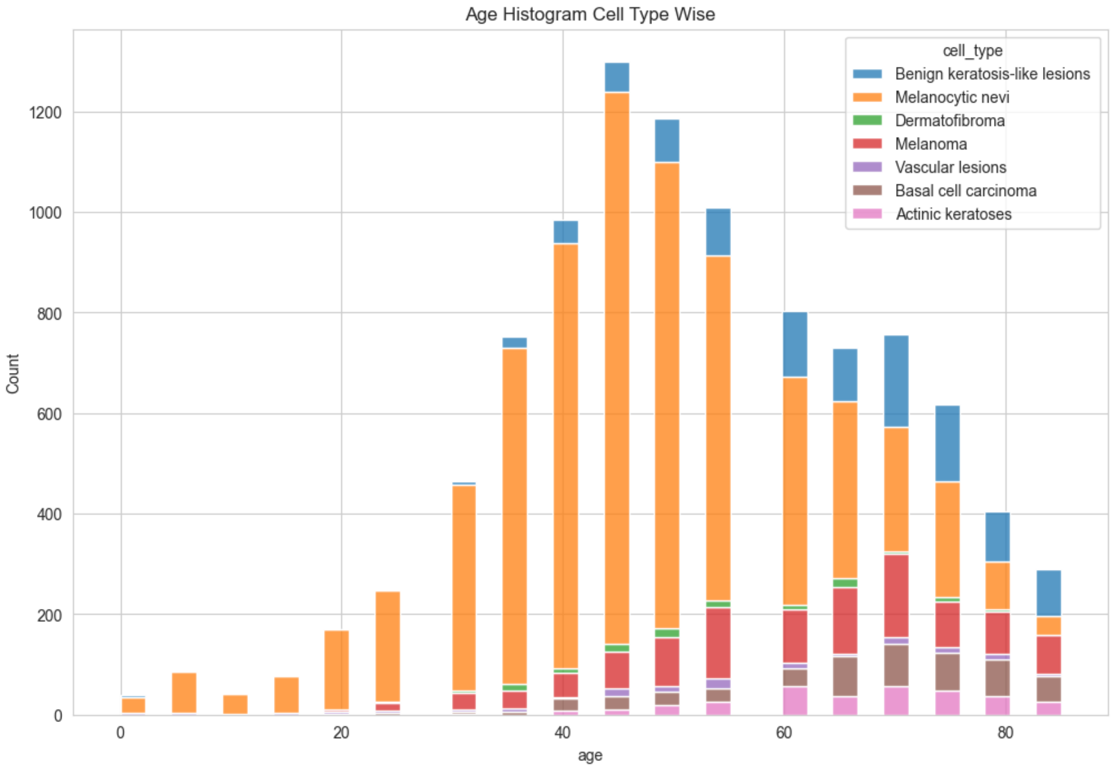


[22]:

```
sns.set_style('whitegrid')
fig,axes = plt.subplots(figsize=(12,8))
ax = sns.histplot(data=df, x='age',hue='sex',multiple='stack')
plt.title('Age Histogram Gender Wise')
plt.show()
```

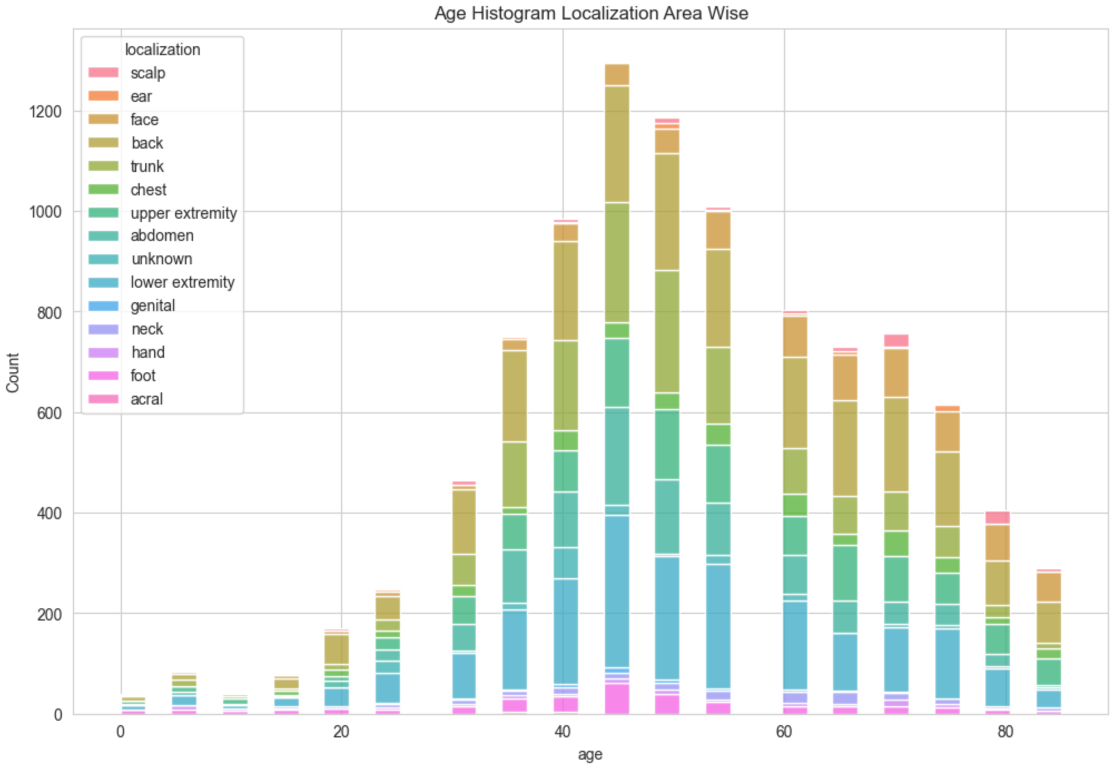


[23]:
```
sns.set_style('whitegrid')
fig,axes = plt.subplots(figsize=(12,8))
ax = sns.histplot(data=df, x='age',hue='cell_type',multiple='stack')
plt.title('Age Histogram Cell Type Wise')
plt.show()
```



[24]:

```
sns.set_style('whitegrid')
fig,axes = plt.subplots(figsize=(12,8))
ax = sns.histplot(data=df, x='age',hue='localization',multiple='stack')
plt.title('Age Histogram Localization Area Wise')
plt.show()
```

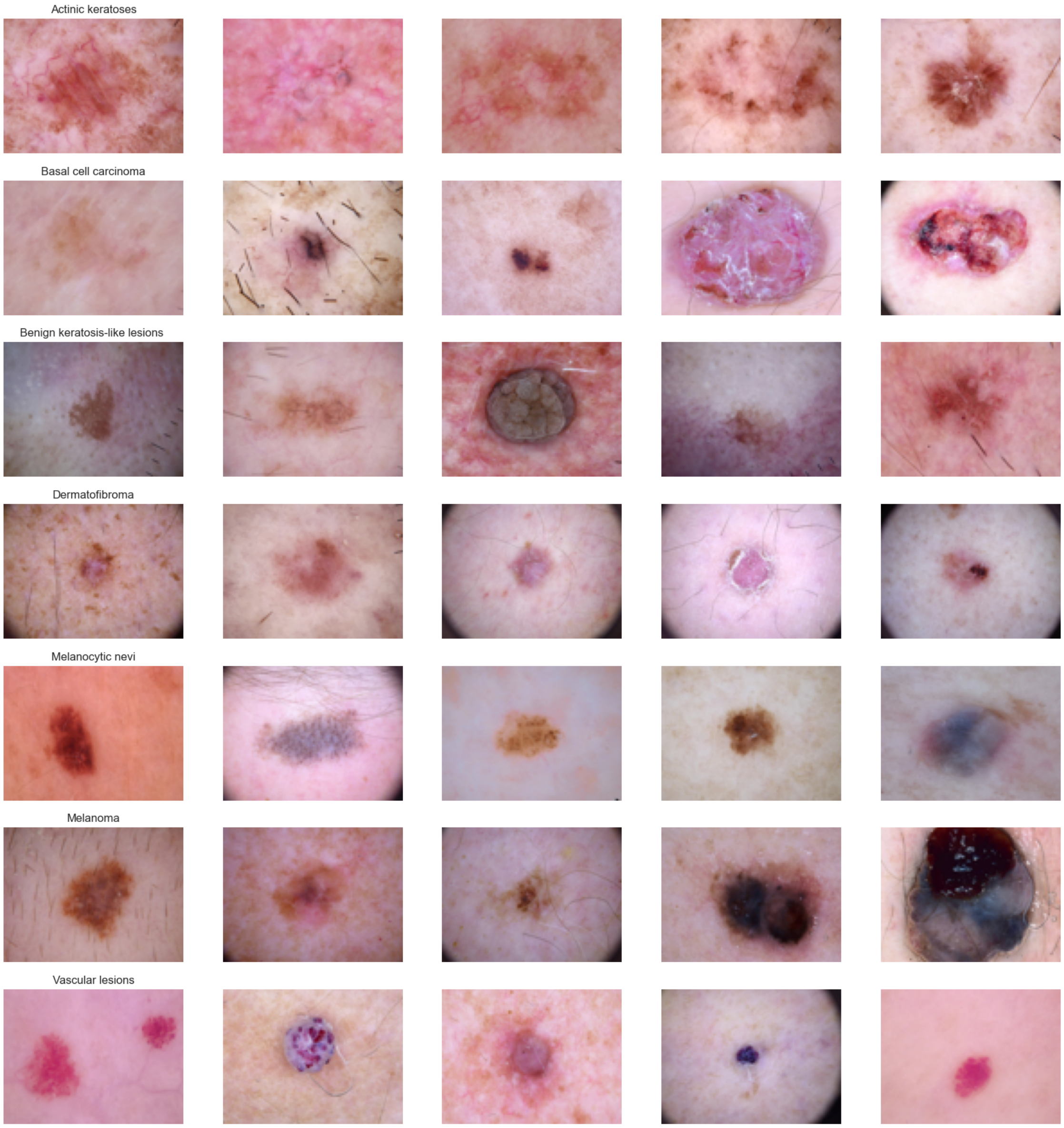


[30]:

```
df['image'] = df['path'].map(lambda x: np.asarray(Image.open(x).
 ↪resize((100,75))))
```

[34]:

```
n_samples = 5
fig, m_axs = plt.subplots(7, n_samples, figsize = (4*n_samples, 3*7))
for n_axs, (type_name, type_rows) in zip(m_axs, df.sort_values(['cell_type']).
 ↪groupby('cell_type')):
    n_axs[0].set_title(type_name)
    for c_ax, (_, c_row) in zip(n_axs, type_rows.sample(n_samples,␣
 ↪random_state=1234).iterrows()):
        c_ax.imshow(c_row['image'])
        c_ax.axis('off')
fig.savefig('category_samples.png', dpi=300)
```



[28]:
```
df['image'].map(lambda x: x.shape).value_counts()
```

[28]:
```
image
(75, 100, 3)    10015
Name: count, dtype: int64
```

[36]:
```
features=df.drop(columns=['cell_type_idx'],axis=1)
target=df['cell_type_idx']
```

[37]:
```
x_train_o, x_test_o, y_train_o, y_test_o = train_test_split(features, target,␣
  ↪test_size=0.20,random_state=1234)
```

[39]:
```
x_train = np.asarray(x_train_o['image'].tolist())
x_test = np.asarray(x_test_o['image'].tolist())

x_train_mean = np.mean(x_train)
x_train_std = np.std(x_train)

x_test_mean = np.mean(x_test)
x_test_std = np.std(x_test)

x_train = (x_train - x_train_mean)/x_train_std
x_test = (x_test - x_test_mean)/x_test_std
```

[40]:
```
y_train = to_categorical(y_train_o, num_classes = 7)
y_test = to_categorical(y_test_o, num_classes = 7)
```

[41]:
```
x_train, x_validate, y_train, y_validate = train_test_split(x_train, y_train,␣
 ↪test_size = 0.1, random_state = 2)
```

[42]:
```
x_train, x_validate, y_train, y_validate = train_test_split(x_train, y_train,␣
 ↪test_size = 0.1, random_state = 2)
```

[43]:
```
# my CNN architechture is In -> [[Conv2D->relu]*2 -> MaxPool2D -> Dropout]*2 ->␣
 ↪Flatten -> Dense -> Dropout -> Out
input_shape = (75, 100, 3)
num_classes = 7

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',padding =␣
 ↪'Same',input_shape=input_shape))
model.add(Conv2D(32,kernel_size=(3, 3), activation='relu',padding = 'Same',))
model.add(MaxPool2D(pool_size = (2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), activation='relu',padding = 'Same'))
model.add(Conv2D(64, (3, 3), activation='relu',padding = 'Same'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.40))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
model.summary()
```

```
C:\Users\theja\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\layers\convolutional\base_conv.py:99: UserWarning: Do not
pass an `input_shape`/`input_dim` argument to a layer. When using Sequential
```

```
models, prefer using an `Input(shape)` object as the first layer in the model
instead.
  super().__init__(
```

```
Model: "sequential"


 Layer (type)                          Output Shape                        ␣
↪Param #

 conv2d (Conv2D)                       (None, 75, 100, 32)                     ␣
↪896

 conv2d_1 (Conv2D)                     (None, 75, 100, 32)                   ␣
↪9,248

 max_pooling2d (MaxPooling2D)          (None, 37, 50, 32)                      ␣
↪   0

 dropout (Dropout)                     (None, 37, 50, 32)                      ␣
↪   0

 conv2d_2 (Conv2D)                     (None, 37, 50, 64)                   ␣
↪18,496

 conv2d_3 (Conv2D)                     (None, 37, 50, 64)                   ␣
↪36,928

 max_pooling2d_1 (MaxPooling2D)        (None, 18, 25, 64)                      ␣
↪   0

 dropout_1 (Dropout)                   (None, 18, 25, 64)                      ␣
↪   0

 flatten (Flatten)                     (None, 28800)                           ␣
↪   0

 dense (Dense)                         (None, 128)                          ␣
↪3,686,528

 dropout_2 (Dropout)                   (None, 128)                             ␣
↪   0

 dense_1 (Dense)                       (None, 7)                               ␣
↪903
```

```
Total params: 3,752,999 (14.32 MB)

Trainable params: 3,752,999 (14.32 MB)

Non-trainable params: 0 (0.00 B)
```

```
[ ]:
```