



```
[2]: import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import cv2
import matplotlib.patches as patches
import tensorflow as tf
from keras.layers import Flatten, Dense, Conv2D, MaxPooling2D, Dropout
from keras.models import Sequential
```

Using TensorFlow backend.

```
[3]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

import tensorflow as tf
from keras.preprocessing.image import img_to_array, load_img
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential

import os
from bs4 import BeautifulSoup
```

## Loading datasets

+ Code

+ Markdown

```
[4]: images=os.path.join("/kaggle/input/face-mask-detection-dataset/Medical mask/Medical mask/Medical Mask/images")
      annotations=os.path.join("/kaggle/input/face-mask-detection-dataset/Medical mask/Medical mask/Medical Mask/annotatic
      train=pd.read_csv(os.path.join("/kaggle/input/face-mask-detection-dataset/train.csv"))
      submission=pd.read_csv(os.path.join("/kaggle/input/face-mask-detection-dataset/submission.csv"))
```

```
[5]: img_folder = '/kaggle/input/face-mask-detection/images/'
      annot_folder = '/kaggle/input/face-mask-detection/annotations/'
```

```
[6]: #Dataset folder path
      data = "/kaggle/input/face-mask-dataset/data"
```

[7]:

```
print(len(train))
train.head()
```

15412

[7]:

	name	x1	x2	y1	y2	classname
0	2756.png	69	126	294	392	face_with_mask
1	2756.png	505	10	723	283	face_with_mask
2	2756.png	75	252	264	390	mask_colorful
3	2756.png	521	136	711	277	mask_colorful
4	6098.jpg	360	85	728	653	face_no_mask

```
[8]: print(len(submission))
      submission.head()
```

8142

```
[8]:
```

	name	x1	x2	y1	y2	classname
0	1800.jpg	NaN	NaN	NaN	NaN	NaN
1	1800.jpg	NaN	NaN	NaN	NaN	NaN
2	1800.jpg	NaN	NaN	NaN	NaN	NaN
3	1799.jpg	NaN	NaN	NaN	NaN	NaN
4	1799.jpg	NaN	NaN	NaN	NaN	NaN

```
[9]: len(os.listdir(images))
```

[9]: 6024

We are having 6024 images.

```
[10]: a=os.listdir(images)
      b=os.listdir(annotations)
      a.sort()
      b.sort()
```

```
[11]: print(len(b),len(a))
```

```
4326 6024
```

```
[12]: train_images=a[1698:]
      test_images=a[:1698]
```

```
[13]: test_images[0]
```

```
[13]: '0001.jpg'
```

Let's see some of the images.

```
[14]: img=plt.imread(os.path.join(images,test_images[0]))  
      plt.imshow(img)  
      plt.show()
```



```
[15]: img=plt.imread(os.path.join(images,train_images[1]))  
      plt.imshow(img)  
      plt.show()
```



```
[16]: options=['face_with_mask','face_no_mask']  
      train= train[train['classname'].isin(options)]  
      train.sort_values('name',axis=0,inplace=True)
```

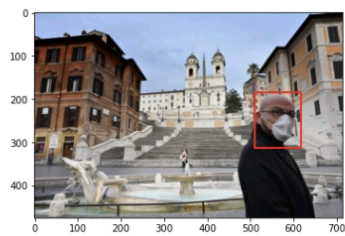


```
[16]: options=['face_with_mask', 'face_no_mask']  
      train= train[train['classname'].isin(options)]  
      train.sort_values('name', axis=0, inplace=True)
```

```
[18]: image=train_images[5]

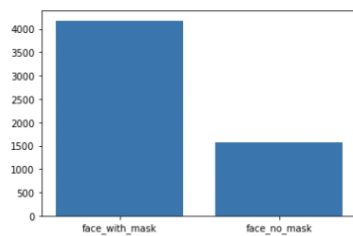
img=plt.imread(os.path.join(images,image))

fig,ax = plt.subplots(1)
ax.imshow(img)
boxes=get_boxes(image)
for box in boxes:
    rect = patches.Rectangle((box[0],box[1]),box[2]-box[0],box[3]-box[1],linewidth=2,edgecolor='r',facecolor='none')
    ax.add_patch(rect)
plt.show()
```



```
[19]: plt.bar(['face_with_mask', 'face_no_mask'], train.classname.value_counts())
```

```
[19]: <BarContainer object of 2 artists>
```

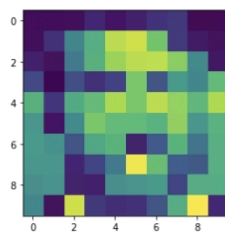


## Creating training data

```
[24]: img_size=10
data=[]
path="/kaggle/input/face-mask-detection-dataset/Medical mask/Medical mask/Medical Mask/images/"
def create_data():
    for i in range(len(train)):
        arr=[]
        for j in train.iloc[i]:
            arr.append(j)
        img_array=cv2.imread(os.path.join(images,arr[0]),cv2.IMREAD_GRAYSCALE)
        crop_image = img_array[arr[2]:arr[4],arr[1]:arr[3]]
        new_img_array=cv2.resize(crop_image,(img_size,img_size))
        data.append([new_img_array,arr[5]])
create_data()
```

```
[21]: data[0][0]  
      plt.imshow(data[0][0])
```

```
[21]: <matplotlib.image.AxesImage at 0x7f4999031310>
```



```
[22]: x=[]
      y=[]
      for features, labels in data:
          x.append(features)
          y.append(labels)
      from sklearn.preprocessing import LabelEncoder
      lbl=LabelEncoder()
      y=lbl.fit_transform(y)
```

+ Code

+ Markdown

```
[23]: x=np.array(x).reshape(-1,10,10,1)
      x=tf.keras.utils.normalize(x,axis=1)
      from keras.utils import to_categorical
      y = to_categorical(y)
```

```
[ ]:
```