

mineracao_dados_complexos_imagens_coloridas

November 27, 2019

```
[7]: import os
# os.environ['PATH'].split(';')
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
plt.style.use('seaborn-white')
import pandas as pd
import re
import cv2

import skimage.io
from skimage import color

from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix

%matplotlib inline
```

0.0.1 Dados para treinamento dos modelos

```
[8]: df_treino = pd.DataFrame(columns=['name','data','classe'])
i = 0
for filename in os.listdir("dataset/treino"):
    path = os.path.join('dataset','treino', filename)
    img = skimage.io.imread(path)
    classe = re.search('floresta|incendio', filename)[0]
    if classe == 'floresta':
        classe = 1
    else:
        classe = 2
    df_treino.loc[i]=[filename,np.ravel(img),classe]
    i = i+1
df_treino
```

```
[8]:          name                                     data classe
0    floresta7.jpg [22, 32, 41, 37, 47, 56, 40, 50, 60, 34, 44, 5...     1
1    floresta50.jpg [179, 182, 105, 108, 112, 25, 115, 122, 27, 10...     1
2   incendio44.jpg [68, 31, 23, 68, 31, 23, 68, 31, 23, 68, 31, 2...     2
3   floresta40.jpg [28, 31, 12, 17, 25, 2, 23, 33, 8, 23, 32, 5, ...     1
4   floresta43.jpg [255, 254, 234, 255, 254, 234, 255, 254, 234, ...     1
..      ...
95  incendio30.jpg [203, 209, 207, 203, 209, 207, 203, 209, 207, ...     2
96  incendio14.jpg [194, 179, 182, 202, 187, 190, 195, 179, 180, ...     2
97  incendio18.jpg [216, 220, 219, 216, 220, 221, 216, 220, 221, ...     2
98  floresta21.jpg [19, 3, 4, 27, 15, 17, 20, 15, 12, 15, 15, 3, ...     1
99  floresta14.jpg [45, 41, 29, 36, 34, 21, 25, 23, 8, 46, 44, 29...     1
```

[100 rows x 3 columns]

```
[9]: df1 = df_treino[df_treino['classe'] == 1]
df2 = df_treino[df_treino['classe'] == 2]
df1 = df1.reset_index(drop=True)
df2 = df2.reset_index(drop=True)
```

0.1 Dados para teste/validation dos modelos

```
[10]: df_teste = pd.DataFrame(columns=['name','data','classe_real','classe_predita'])
i = 0
for filename in os.listdir("dataset/teste"):
    path = os.path.join('dataset','teste', filename)
    img = skimage.io.imread(path)
    classe = re.search('floresta|incendio', filename)[0]
    if classe == 'floresta':
        classe = 1
    else:
        classe = 2
    df_teste.loc[i]=[filename,np.ravel(img),classe,0]
    i = i+1
df_teste
```

```
[10]:          name                                     data \
0    floresta49.jpg [47, 76, 0, 91, 124, 37, 79, 112, 31, 51, 79, ...
1    floresta2.jpg [0, 3, 8, 1, 2, 7, 2, 3, 7, 0, 0, 2, 2, 2, 4, ...
2   incendio33.jpg [128, 39, 0, 129, 40, 0, 129, 40, 0, 129, 40, ...
3   floresta35.jpg [148, 175, 60, 134, 155, 50, 128, 147, 66, 139...
4   incendio20.jpg [98, 138, 164, 97, 137, 163, 97, 137, 163, 99, ...
5   incendio54.jpg [15, 3, 3, 15, 3, 3, 15, 3, 3, 15, 3, 3, 15, 3...
6   floresta5.jpg [77, 91, 12, 146, 161, 78, 150, 167, 75, 155, ...
7   floresta38.jpg [5, 30, 1, 19, 44, 15, 19, 44, 14, 18, 44, 7, ...
8   floresta4.jpg [2, 8, 0, 2, 8, 0, 1, 6, 0, 0, 5, 1, 0, 4, 3, ...
```

```

9  incendio15.jpg  [16, 5, 1, 17, 6, 2, 17, 6, 2, 19, 8, 4, 18, 7...
10 incendio12.jpg  [255, 244, 188, 255, 244, 188, 255, 245, 191, ...
11 floresta24.jpg  [103, 158, 139, 113, 155, 171, 116, 141, 171, ...
12 floresta3.jpg   [28, 64, 0, 16, 50, 0, 18, 53, 11, 14, 50, 2, ...
13 incendio21.jpg  [90, 123, 140, 107, 143, 159, 119, 158, 175, 9...
14 incendio10.jpg  [5, 12, 64, 5, 12, 64, 3, 13, 64, 2, 14, 66, 2...
15 incendio24.jpg  [61, 15, 0, 61, 15, 2, 58, 14, 1, 57, 13, 2, 5...
16 floresta1.jpg   [91, 97, 71, 91, 94, 75, 72, 75, 56, 52, 61, 3...
17 incendio56.jpg  [32, 2, 4, 32, 2, 4, 32, 2, 4, 32, 2, 4, 32, 2...
18 floresta28.jpg  [44, 48, 15, 39, 45, 7, 49, 57, 16, 54, 65, 25...
19 incendio19.jpg  [91, 25, 13, 92, 26, 14, 93, 27, 15, 94, 28, 1...

```

	classe_real	classe_predita
0	1	0
1	1	0
2	2	0
3	1	0
4	2	0
5	2	0
6	1	0
7	1	0
8	1	0
9	2	0
10	2	0
11	1	0
12	1	0
13	2	0
14	2	0
15	2	0
16	1	0
17	2	0
18	1	0
19	2	0

```
[11]: df_results = pd.DataFrame(columns=['iteracao','floresta','incendio','erro'])
```

0.1.1 Clustering - Agrupamento de Dados (Aprendizado não supervisionado) - Os dados não tem rótulo/classe

```
[12]: from sklearn.cluster import KMeans

for iteracao in range(0,30):

    kmeans = KMeans(n_clusters=2, random_state=0).fit( list(df_teste['data']) )

    dados_agrupados = kmeans.labels_
```

```

centroids = kmeans.cluster_centers_
df_teste['cluster_group'] = dados_agrupados

florestas = []
incendios = []
erros = []
for i in df_teste.index:
    if (df_teste.iloc[i,2] == 1 and df_teste.iloc[i,4] == 0):
        florestas.append(df_teste.loc[i])
    elif (df_teste.iloc[i,2] == 2 and df_teste.iloc[i,4] == 1):
        incendios.append(df_teste.loc[i])
    else:
        erros.append(df_teste.loc[i])
#----- ↵ □

print("florestas_agrupadas_corretamente:")
fig1, axes1 = plt.subplots(nrows=1, ncols=len(florestas), figsize=(18, 10))
cont = 0
for f in florestas:
    #print(df_teste.iloc[f.name,0])
    path = os.path.join('dataset','teste', df_teste.iloc[f.name,0])
    img = skimage.io.imread(path)
    axes1[cont].imshow(img) # mpimg.imread( )
    axes1[cont].set_title(df_teste.iloc[f.name,0])
    cont = cont+1
fig1.tight_layout()
plt.show()
print("total_florestas_agrupadas_corretamente:",len(florestas))
print(70*"#")
print("incendios_agrupados_corretamente:")
fig2, axes2 = plt.subplots(nrows=1, ncols=len(incendios), figsize=(18, 10))
cont2 = 0
for i in incendios:
    #print(df_teste.iloc[i.name,0])
    path = os.path.join('dataset','teste', df_teste.iloc[i.name,0])
    img = skimage.io.imread(path)
    axes2[cont2].imshow(img) # mpimg.imread( )
    axes2[cont2].set_title(df_teste.iloc[i.name,0])
    cont2 = cont2+1
fig2.tight_layout()
plt.show()
print("total_incendios_agrupados_corretamente:",len(incendios))
print(70*"#")
print("florestas_e_incendios_agrupados_incorretamente:")
fig3, axes3 = plt.subplots(nrows=1, ncols=len(erros), figsize=(18, 10))
cont3 = 0
for e in erros:

```

```

#print(df_teste.iloc[e.name,0])
path = os.path.join('dataset','teste', df_teste.iloc[e.name,0])
img = skimage.io.imread(path)
axes3[cont3].imshow(img) # mpimg.imread( )
axes3[cont3].set_title(df_teste.iloc[e.name,0])
cont3 = cont3+1
fig3.tight_layout()
plt.show()
print("total_florestas_e_incendios_agrupados_incorretamente:",len(erros))
print("Iteração: ",(iteracao+1) )
new_row = {'iteracao' : iteracao+1, 'floresta' : len(florestas), 'incendio' :
↪: len(incendios), 'erro' : len(erros)}
#append row to the dataframe
df_results = df_results.append(new_row, ignore_index=True)
print(70*"=")

df_results.to_excel('results_30_runs_kmeans_df_teste_coloridas.xlsx',index=False)

```

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

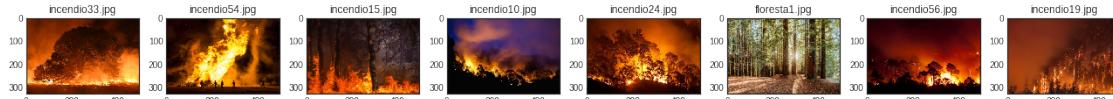
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#####

florestas_e_incendios_agrupados_incorretamente:

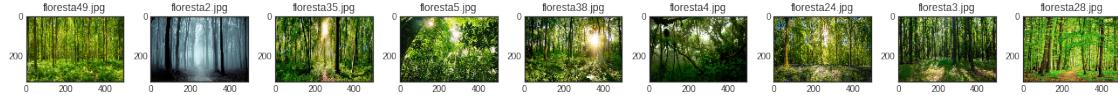


total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 1

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

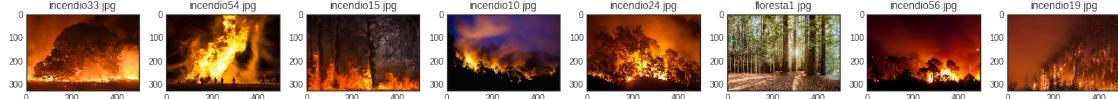
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#####

florestas_e_incendios_agrupados_incorretamente:

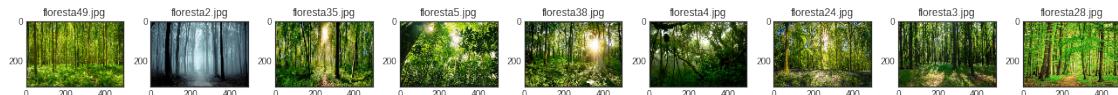


total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 2

=====

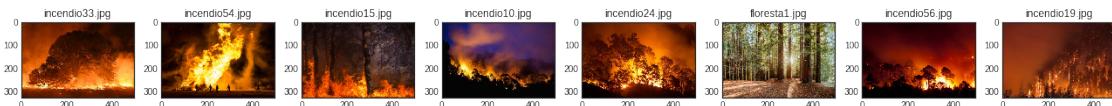
florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9
#####
incendios_agrupados_corretamente:

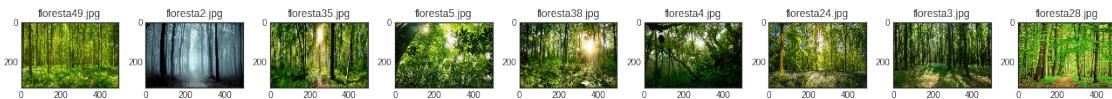


total_incendios_agrupados_corretamente: 3
#####
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8
Iteração: 3
=====

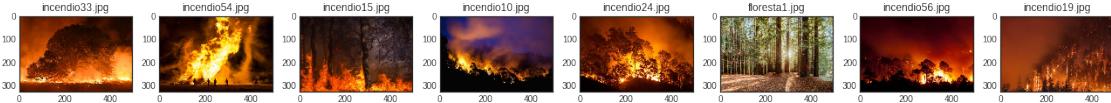
florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9
#####
incendios_agrupados_corretamente:



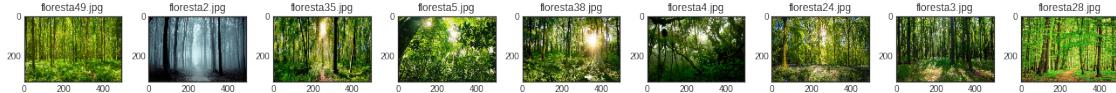
```
total_incendios_agrupados_corretamente: 3
#####
florestas_e_incendios_agrupados_incorretamente:
```



```
total_florestas_e_incendios_agrupados_incorretamente: 8
```

```
Iteração: 4
```

```
=====
florestas_agrupadas_corretamente:
```



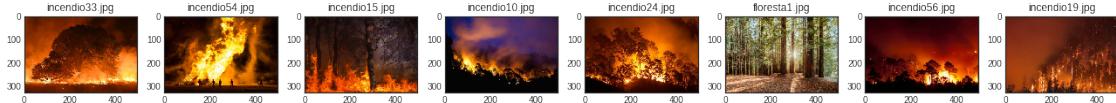
```
total_florestas_agrupadas_corretamente: 9
```

```
#####
incendios_agrupados_corretamente:
```



```
total_incendios_agrupados_corretamente: 3
```

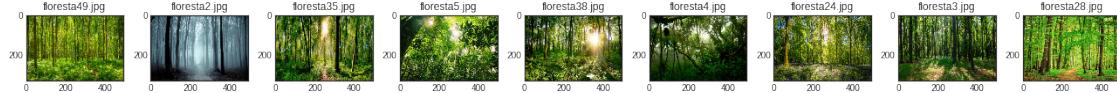
```
#####
florestas_e_incendios_agrupados_incorretamente:
```



`total_florestas_e_incendios_agrupados_incorretamente: 8`

`Iteração: 5`

`=====`
`florestas_agrupadas_corretamente:`



`total_florestas_agrupadas_corretamente: 9`

`#####
#####`

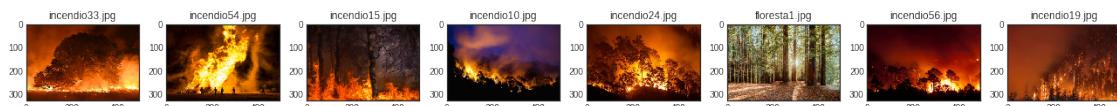
`incendios_agrupados_corretamente:`



`total_incendios_agrupados_corretamente: 3`

`#####
#####`

`florestas_e_incendios_agrupados_incorretamente:`



`total_florestas_e_incendios_agrupados_incorretamente: 8`

`Iteração: 6`

`=====`
`florestas_agrupadas_corretamente:`



`total_florestas_agrupadas_corretamente: 9`

`#####
#####`

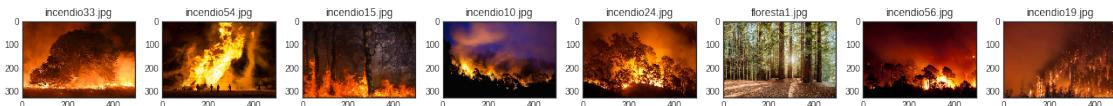
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 7

=====
florestas_agrupadas_corretamente:



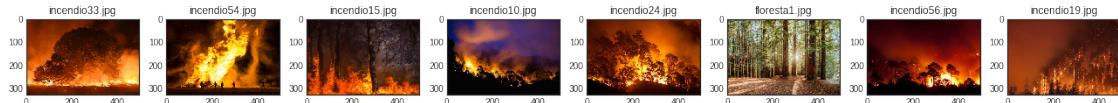
total_florestas_agrupadas_corretamente: 9

#####
#

incendios_agrupados_corretamente:



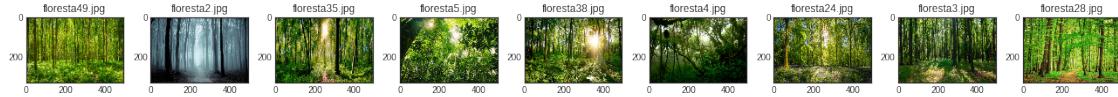
total_incendios_agrupados_corretamente: 3
#####
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 8

=====
florestas_agrupadas_corretamente:



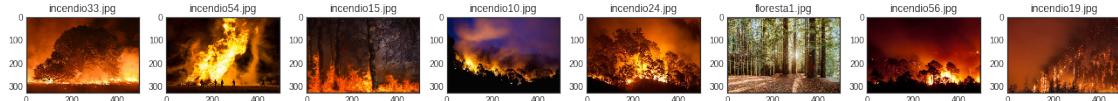
total_florestas_agrupadas_corretamente: 9

#####
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
florestas_e_incendios_agrupados_incorretamente:

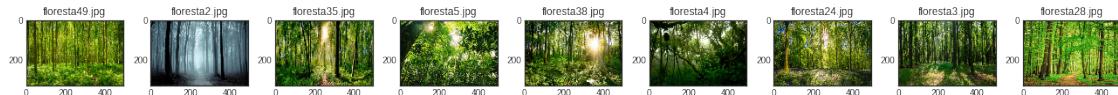


total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 9

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

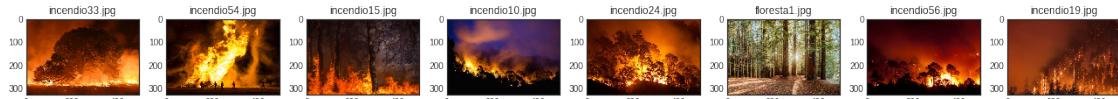
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 10

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

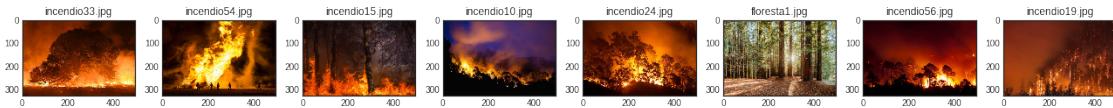
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 11

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

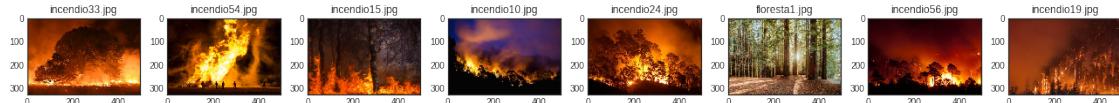
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

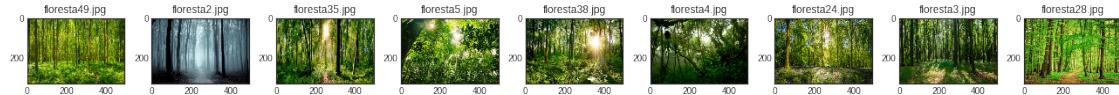
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 12

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

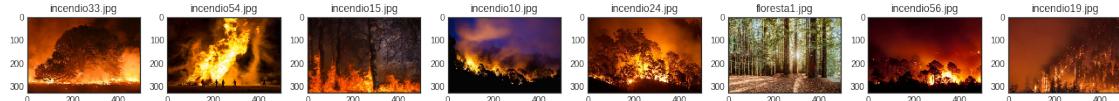
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#####

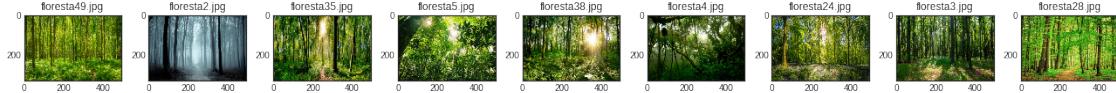
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 13

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

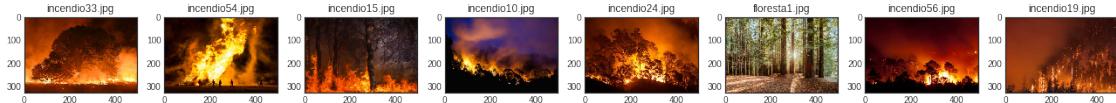
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 14

=====
=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

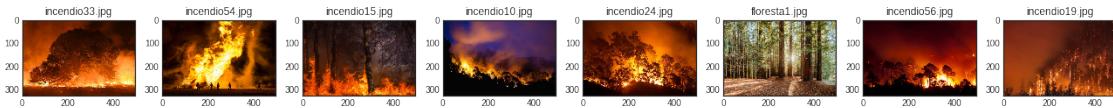
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 15

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

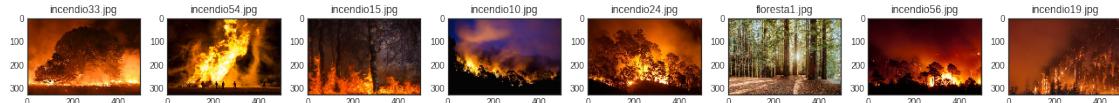
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

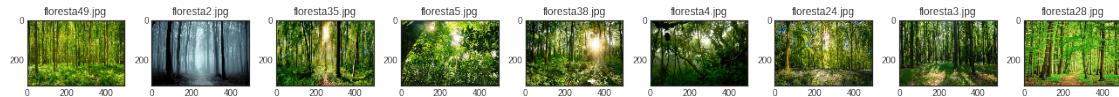
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 16

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

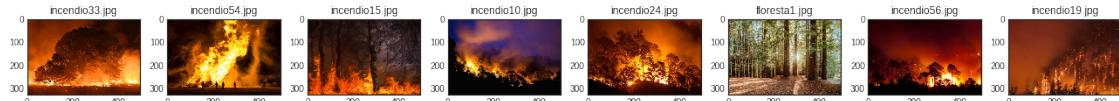
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#####

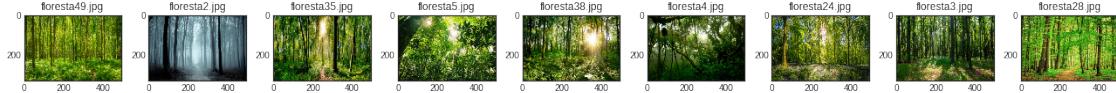
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 17

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

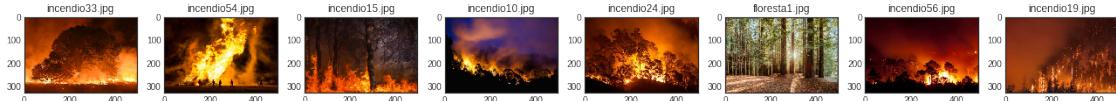
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 18

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

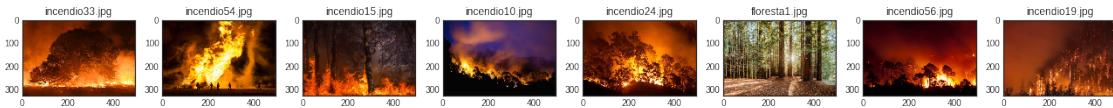
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 19

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

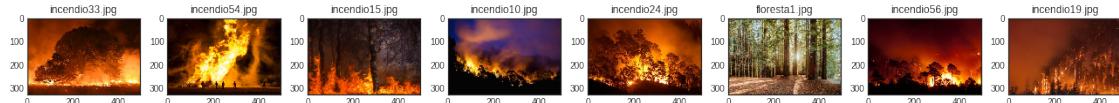
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

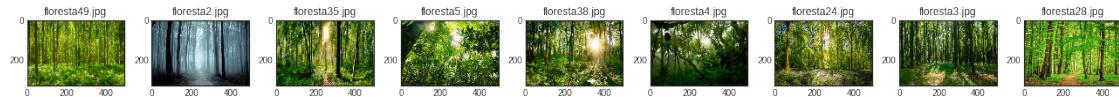
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 20

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

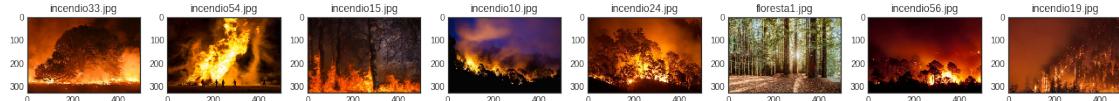
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#####

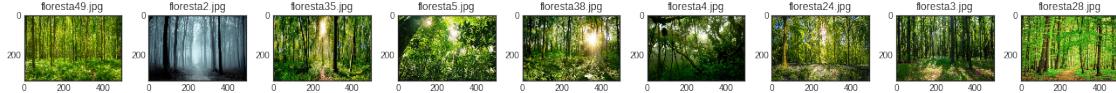
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 21

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

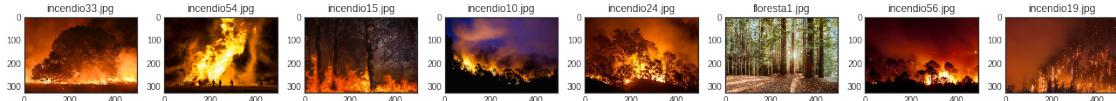
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 22

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

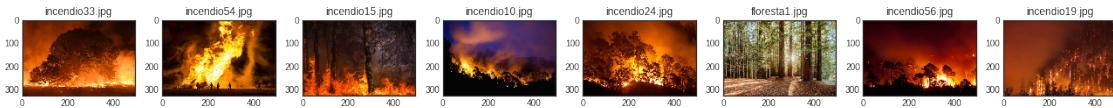
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 23

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

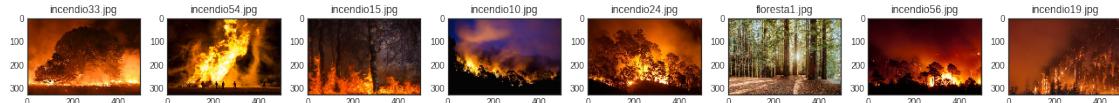
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

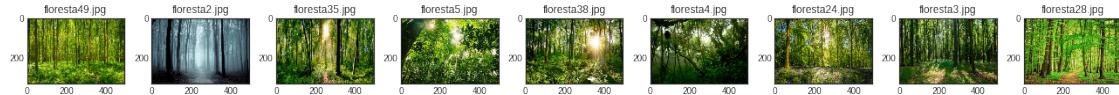
`florestas_e_incendios_agrupados_incorretamente:`



`total_florestas_e_incendios_agrupados_incorretamente: 8`

`Iteração: 24`

`florestas_agrupadas_corretamente:`



`total_florestas_agrupadas_corretamente: 9`

`#####
#####`

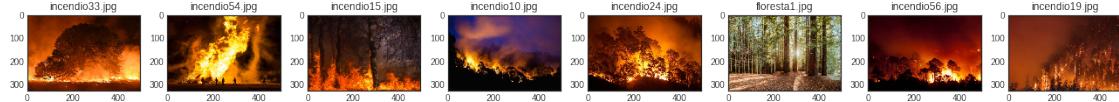
`incendios_agrupados_corretamente:`



`total_incendios_agrupados_corretamente: 3`

`#####
#####`

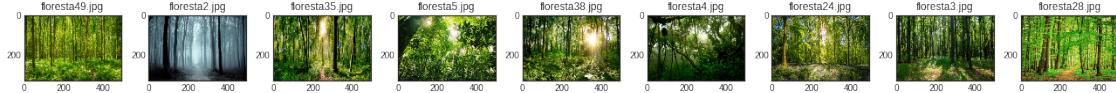
`florestas_e_incendios_agrupados_incorretamente:`



`total_florestas_e_incendios_agrupados_incorretamente: 8`

`Iteração: 25`

`florestas_agrupadas_corretamente:`



total_florestas_agrupadas_corretamente: 9

#####

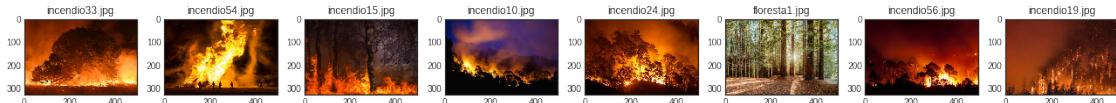
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 26

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

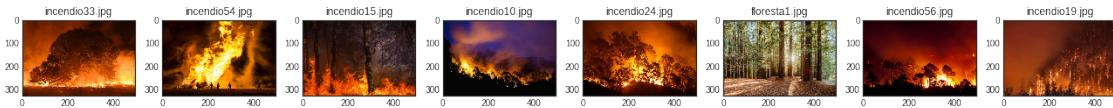
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 27

=====

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####

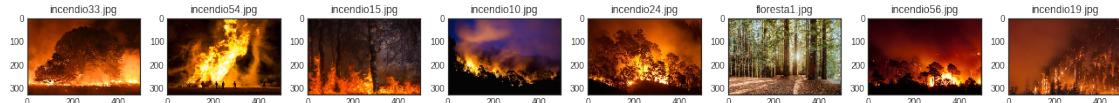
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####

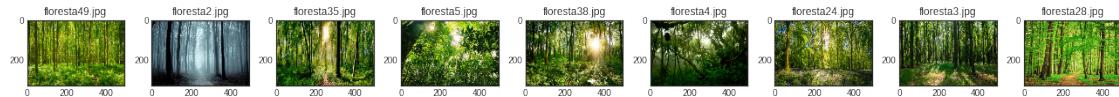
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 28

florestas_agrupadas_corretamente:



total_florestas_agrupadas_corretamente: 9

#####
#####

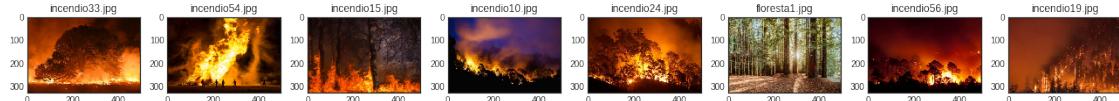
incendios_agrupados_corretamente:



total_incendios_agrupados_corretamente: 3

#####
#####

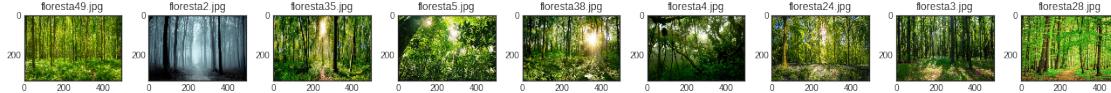
florestas_e_incendios_agrupados_incorretamente:



total_florestas_e_incendios_agrupados_incorretamente: 8

Iteração: 29

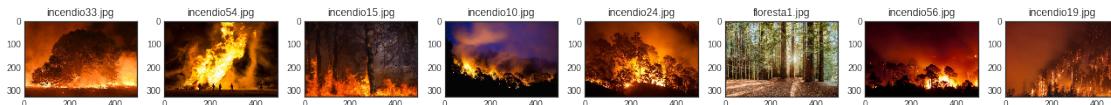
florestas_agrupadas_corretamente:



```
total_florestas_agrupadas_corretamente: 9
#####
incendios_agrupados_corretamente:
```



```
total_incendios_agrupados_corretamente: 3
#####
florestas_e_incendios_agrupados_incorretamente:
```



```
total_florestas_e_incendios_agrupados_incorretamente: 8
Iteração: 30
=====
```

0.1.2 Classification - Classificação de Dados (Aprendizado supervisionado) - Os dados tem um rótulo/classe

```
[13]: # classificação
from sklearn import svm
clf1 = svm.SVC(gamma='scale')
clf1.fit(list(df_treino['data']), list(df_treino['classe']))
y_pred = clf1.predict( list(df_teste['data']) )
df_teste['classe_predita'] = y_pred
# df_teste
print("Matriz de confusão:\n",confusion_matrix( list(df_teste['classe_real']), list(df_teste['classe_predita']) ) )
```

```

print("Classification Report:\n",classification_report(
    list(df_teste['classe_real']), list(df_teste['classe_predita']) ))
print("Accuracy:",accuracy_score(list(df_teste['classe_real']), y_pred))

```

Matriz de confusão:

```

[[9 1]
 [2 8]]

```

Classification Report:

	precision	recall	f1-score	support
1	0.82	0.90	0.86	10
2	0.89	0.80	0.84	10
accuracy			0.85	20
macro avg	0.85	0.85	0.85	20
weighted avg	0.85	0.85	0.85	20

Accuracy: 0.85

```

[14]: from sklearn.naive_bayes import GaussianNB
clf2 = GaussianNB()
clf2.fit(list(df_treino['data']), list(df_treino['classe']))
y_pred = clf2.predict(list(df_teste['data']))
df_teste['classe_predita'] = y_pred
print("Matriz de confusão:\n",confusion_matrix( list(df_teste['classe_real']),list(df_teste['classe_predita']) ))
print("Classification Report:\n",classification_report(
    list(df_teste['classe_real']), list(df_teste['classe_predita']) ))
print("Accuracy:",accuracy_score(list(df_teste['classe_real']), y_pred))

```

Matriz de confusão:

```

[[8 2]
 [6 4]]

```

Classification Report:

	precision	recall	f1-score	support
1	0.57	0.80	0.67	10
2	0.67	0.40	0.50	10
accuracy			0.60	20
macro avg	0.62	0.60	0.58	20
weighted avg	0.62	0.60	0.58	20

Accuracy: 0.6

```

[15]: from sklearn.tree import DecisionTreeClassifier
clf3 = DecisionTreeClassifier(random_state=0)

```

```

clf3.fit(list(df_treino['data']), list(df_treino['classe']))
y_pred = clf3.predict(list(df_teste['data']))
df_teste['classe_predita'] = y_pred
print("Matriz de confusão:\n",confusion_matrix( list(df_teste['classe_real']),list(df_teste['classe_predita']) ) )
print("Classification Report:\n",classification_report(list(df_teste['classe_real']), list(df_teste['classe_predita']) ))
print("Accuracy:",accuracy_score(list(df_teste['classe_real']), y_pred))

```

Matriz de confusão:

```

[[6 4]
 [2 8]]

```

Classification Report:

	precision	recall	f1-score	support
1	0.75	0.60	0.67	10
2	0.67	0.80	0.73	10
accuracy			0.70	20
macro avg	0.71	0.70	0.70	20
weighted avg	0.71	0.70	0.70	20

Accuracy: 0.7

```
[16]: from sklearn.linear_model import LogisticRegression
clf4 = LogisticRegression(solver='lbfgs',multi_class='multinomial',random_state=1)
clf4.fit(list(df_treino['data']), list(df_treino['classe']))
y_pred = clf4.predict(list(df_teste['data']))
df_teste['classe_predita'] = y_pred
print("Matriz de confusão:\n",confusion_matrix( list(df_teste['classe_real']),list(df_teste['classe_predita']) ) )
print("Classification Report:\n",classification_report(list(df_teste['classe_real']), list(df_teste['classe_predita']) ))
print("Accuracy:",accuracy_score(list(df_teste['classe_real']), y_pred))
```

Matriz de confusão:

```

[[9 1]
 [2 8]]

```

Classification Report:

	precision	recall	f1-score	support
1	0.82	0.90	0.86	10
2	0.89	0.80	0.84	10
accuracy			0.85	20
macro avg	0.85	0.85	0.85	20

weighted avg 0.85 0.85 0.85 20

Accuracy: 0.85

```
[17]: from sklearn.ensemble import RandomForestClassifier, VotingClassifier
clf5 = RandomForestClassifier(n_estimators=50, random_state=1)
clf5.fit(list(df_treino['data']), list(df_treino['classe']))
y_pred = clf5.predict(list(df_teste['data']))
df_teste['classe_predita'] = y_pred
print("Matriz de confusão:\n",confusion_matrix( list(df_teste['classe_real']),list(df_teste['classe_predita']) ) )
print("Classification Report:\n",classification_report(list(df_teste['classe_real']), list(df_teste['classe_predita']) ))
print("Accuracy:",accuracy_score(list(df_teste['classe_real']), y_pred))
```

Matriz de confusão:

```
[[8 2]
 [2 8]]
```

Classification Report:

	precision	recall	f1-score	support
1	0.80	0.80	0.80	10
2	0.80	0.80	0.80	10
accuracy			0.80	20
macro avg	0.80	0.80	0.80	20
weighted avg	0.80	0.80	0.80	20

Accuracy: 0.8

```
[18]: eclf1 = VotingClassifier(estimators=[('svm', clf1), ('nb', clf2), ('dt',clf3),('lr', clf4), ('rf', clf5)], voting='hard')
eclf1 = eclf1.fit(list(df_treino['data']), list(df_treino['classe']))
y_pred = eclf1.predict(list(df_teste['data']))
df_teste['classe_predita'] = y_pred
print("Matriz de confusão:\n",confusion_matrix( list(df_teste['classe_real']),list(df_teste['classe_predita']) ) )
print("Classification Report:\n",classification_report(list(df_teste['classe_real']), list(df_teste['classe_predita']) ))
print("Accuracy:",accuracy_score(list(df_teste['classe_real']), y_pred))
```

Matriz de confusão:

```
[[9 1]
 [2 8]]
```

Classification Report:

	precision	recall	f1-score	support
1	0.82	0.90	0.86	10

2	0.89	0.80	0.84	10
accuracy			0.85	20
macro avg	0.85	0.85	0.85	20
weighted avg	0.85	0.85	0.85	20

Accuracy: 0.85

```
[19]: from sklearn.ensemble import AdaBoostClassifier
clf6 = AdaBoostClassifier(n_estimators=100, random_state=0)
# clf6.feature_importances_
clf6.fit(list(df_treino['data']), list(df_treino['classe']))
y_pred = clf6.predict(list(df_teste['data']))
df_teste['classe_predita'] = y_pred
print("Matriz de confusão:\n",confusion_matrix( list(df_teste['classe_real']),list(df_teste['classe_predita']) ))
print("Classification Report:\n",classification_report(list(df_teste['classe_real']), list(df_teste['classe_predita']) ))
print("Accuracy:",accuracy_score(list(df_teste['classe_real']), y_pred))
```

Matriz de confusão:

```
[[9 1]
 [2 8]]
```

Classification Report:

	precision	recall	f1-score	support
1	0.82	0.90	0.86	10
2	0.89	0.80	0.84	10
accuracy			0.85	20
macro avg	0.85	0.85	0.85	20
weighted avg	0.85	0.85	0.85	20

Accuracy: 0.85

```
[20]: from sklearn.neural_network import MLPClassifier
clf7 = MLPClassifier(hidden_layer_sizes=(2, ), activation='relu',solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant',learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True,random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1,beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10)
clf7.fit(list(df_treino['data']), list(df_treino['classe']))
y_pred = clf7.predict(list(df_teste['data']))
df_teste['classe_predita'] = y_pred
print("Matriz de confusão:\n",confusion_matrix( list(df_teste['classe_real']),list(df_teste['classe_predita']) ))
```

```

print("Classification Report:\n",classification_report(
    list(df_teste['classe_real']), list(df_teste['classe_predita']) ))
print("Accuracy:",accuracy_score(list(df_teste['classe_real']), y_pred))

```

Matriz de confusão:

```

[[10  0]
 [10  0]]

```

Classification Report:

	precision	recall	f1-score	support
1	0.50	1.00	0.67	10
2	0.00	0.00	0.00	10
accuracy			0.50	20
macro avg	0.25	0.50	0.33	20
weighted avg	0.25	0.50	0.33	20

Accuracy: 0.5

```

/home/rafael-sanches/Documentos/Doutorado/Disciplinas/Mineracao_de_Dados_Complexos/venv_python3/lib/python3.6/site-
packages/sklearn/metrics/classification.py:1437: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples.
'precision', 'predicted', average, warn_for)

```

[21]:

```

'''
# df_teste[ df_teste['classe_real'] != df_teste['classe_predita']]
from __future__ import absolute_import, division, print_function,
__unicode_literals

# TensorFlow e tf.keras
import tensorflow as tf
from tensorflow import keras

print(tf.__version__)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout,
    MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
'''
```

[21]:

```

"\n# df_teste[ df_teste['classe_real'] != df_teste['classe_predita']]"
from __future__ import absolute_import, division, print_function,
unicode_literals\n\n# TensorFlow e tf.keras\nimport tensorflow as tf\nfrom tensorflow import keras\nprint(tf.__version__)\n\nfrom tensorflow.keras.models
```

```
import Sequential\nfrom tensorflow.keras.layers import Dense, Conv2D, Flatten,\nDropout, MaxPooling2D\nfrom tensorflow.keras.preprocessing.image import\nImageDataGenerator\n\n[22]: '''\nPATH = 'datasetcnn'\ntrain_dir = os.path.join(PATH, 'treino')\nvalidation_dir = os.path.join(PATH, 'teste')\n\ntrain_floresta_dir = os.path.join(train_dir, 'floresta') # directory with our\n→training cat pictures\ntrain_incendio_dir = os.path.join(train_dir, 'incendio') # directory with our\n→training dog pictures\nvalidation_floresta_dir = os.path.join(validation_dir, 'floresta') # directory\n→with our validation cat pictures\nvalidation_incendio_dir = os.path.join(validation_dir, 'incendio') # directory\n→with our validation dog pictures\n\nnum_floresta_tr = len(os.listdir(train_floresta_dir))\nnum_incendio_tr = len(os.listdir(train_incendio_dir))\n\nnum_floresta_val = len(os.listdir(validation_floresta_dir))\nnum_incendio_val = len(os.listdir(validation_incendio_dir))\n\ntotal_train = num_floresta_tr + num_incendio_tr\ntotal_val = num_floresta_val + num_incendio_val\n\nprint('total training floresta images:', num_floresta_tr)\nprint('total training incendio images:', num_incendio_tr)\n\nprint('total validation floresta images:', num_floresta_val)\nprint('total validation incendio images:', num_incendio_val)\nprint("--")\nprint("Total training images:", total_train)\nprint("Total validation images:", total_val)\n\nbatch_size = 128\nepochs = 15\nIMG_HEIGHT = 512#150\nIMG_WIDTH = 512#150\n\ntrain_image_generator = ImageDataGenerator(rescale=1./255) # Generator for our\n→training data\nvalidation_image_generator = ImageDataGenerator(rescale=1./255) # Generator for\n→our validation data
```

```

train_data_gen = train_image_generator.  

    ↪flow_from_directory(batch_size=batch_size,  

                        directory=train_dir,  

                        shuffle=True,  

                        □  

    ↪target_size=(IMG_HEIGHT, IMG_WIDTH),  

                        class_mode='binary')  

val_data_gen = validation_image_generator.  

    ↪flow_from_directory(batch_size=batch_size,  

                        □  

    ↪directory=validation_dir,  

                        □  

    ↪target_size=(IMG_HEIGHT, IMG_WIDTH),  

                        □  

    ↪class_mode='binary')  

x_train, y_train = next(train_data_gen)  

x_test, y_test = next(val_data_gen)  

# This function will plot images in the form of a grid with 1 row and 5 columns  

    ↪where images are placed in each column.  

def plotImages(images_arr):  

    fig, axes = plt.subplots(1, 5, figsize=(20,20))  

    axes = axes.flatten()  

    for img, ax in zip(images_arr, axes):  

        ax.imshow(img)  

        ax.axis('off')  

    plt.tight_layout()  

    plt.show()  

#plotImages(x_train[:5])  

model = Sequential([  

    Conv2D(16, 3, padding='same', activation='relu', input_shape=(IMG_HEIGHT,  

    ↪IMG_WIDTH ,3)),  

    MaxPooling2D(),  

    Conv2D(32, 3, padding='same', activation='relu'),  

    MaxPooling2D(),  

    Conv2D(64, 3, padding='same', activation='relu'),  

    MaxPooling2D(),  

    Flatten(),  

    Dense(512, activation='relu'),  

    Dense(1, activation='sigmoid')

```

```

])
```

```

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

```

model.summary()
```

```

history = model.fit_generator(
    train_data_gen,
    steps_per_epoch=total_train // batch_size,
    epochs=4,
    validation_data=val_data_gen,
    validation_steps=total_val // batch_size
)
```

```

# usage in model
#model.fit(ds_train, epochs=10, verbose=True)

'''
```

```

'''
```

```

history = model.fit_generator(
    x_train,
    steps_per_epoch=total_train,
    epochs=4,
    validation_data=y_train,
    validation_steps=total_val
)
model.fit(train_data_gen, epochs=10, steps_per_epoch=118,
          validation_data=val_data_gen, validation_steps=(total_val/batch_size)
'''
```

[22]: '\nhistory = model.fit_generator(\n x_train,\n steps_per_epoch=total_train, \n epochs=4, \n validation_data=y_train,\n validation_steps=total_val\n)\nmodel.fit(train_data_gen, epochs=10,\n steps_per_epoch=118,\n validation_data=val_data_gen, validation_steps=(total_val/batch_size))\n'