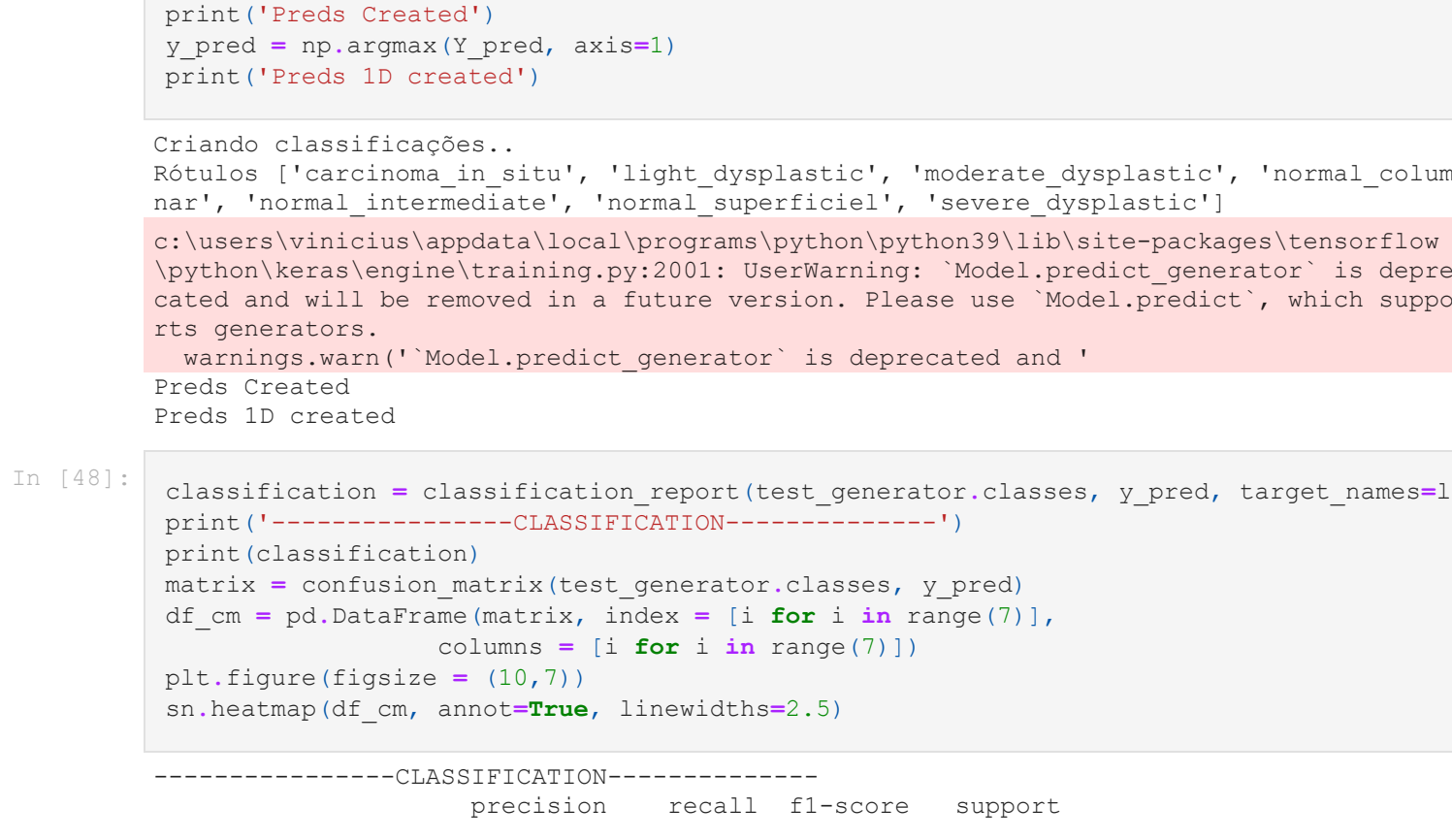



```
45/45 [=====] - 69s 2s/step - loss: 1.4906 - accuracy: 0.3789
45/45 [=====] - 69s 2s/step - loss: 1.4602 - accuracy: 0.4602
Epoch 5/50
45/45 [=====] - 69s 2s/step - loss: 1.4066 - accuracy: 0.4413
45/45 [=====] - 69s 2s/step - loss: 1.4886 - accuracy: 0.4886
Epoch 6/50
45/45 [=====] - 69s 2s/step - loss: 1.3309 - accuracy: 0.4878
45/45 [=====] - 69s 2s/step - loss: 1.5057 - accuracy: 0.5057
45/45 [=====] - 69s 2s/step - loss: 1.3096 - accuracy: 0.4865
45/45 [=====] - 69s 2s/step - loss: 1.4943 - accuracy: 0.4943
Epoch 7/50
45/45 [=====] - 69s 2s/step - loss: 1.2366 - accuracy: 0.4961
45/45 [=====] - 69s 2s/step - loss: 1.5511 - accuracy: 0.5511
45/45 [=====] - 69s 2s/step - loss: 1.1755 - accuracy: 0.5264
45/45 [=====] - 69s 2s/step - loss: 1.4886 - accuracy: 0.4886
Epoch 10/50
45/45 [=====] - 69s 2s/step - loss: 1.1954 - accuracy: 0.5031
45/45 [=====] - 69s 2s/step - loss: 1.4773 - accuracy: 0.4773
Epoch 11/50
45/45 [=====] - 69s 2s/step - loss: 1.1251 - accuracy: 0.5332
45/45 [=====] - 69s 2s/step - loss: 1.365 - accuracy: 0.5000
Epoch 12/50
45/45 [=====] - 69s 2s/step - loss: 1.1158 - accuracy: 0.5529
45/45 [=====] - 69s 2s/step - loss: 1.5227 - accuracy: 0.5227
Epoch 13/50
45/45 [=====] - 69s 2s/step - loss: 1.1358 - accuracy: 0.5423
45/45 [=====] - 69s 2s/step - loss: 1.5114 - accuracy: 0.5114
Epoch 14/50
45/45 [=====] - 69s 2s/step - loss: 1.1067 - accuracy: 0.5813
45/45 [=====] - 69s 2s/step - loss: 1.0598 - accuracy: 0.5455
Epoch 15/50
45/45 [=====] - 69s 2s/step - loss: 1.0481 - accuracy: 0.5884
45/45 [=====] - 69s 2s/step - loss: 1.0423 - accuracy: 0.5704
45/45 [=====] - 69s 2s/step - loss: 1.0933 - accuracy: 0.5114
45/45 [=====] - 69s 2s/step - loss: 0.9862 - accuracy: 0.6403
45/45 [=====] - 69s 2s/step - loss: 1.0513 - accuracy: 0.5170
45/45 [=====] - 69s 2s/step - loss: 0.9523 - accuracy: 0.6272
45/45 [=====] - 69s 2s/step - loss: 1.0561 - accuracy: 0.5114
45/45 [=====] - 69s 2s/step - loss: 1.0024 - accuracy: 0.6098
45/45 [=====] - 69s 2s/step - loss: 1.0608 - accuracy: 0.5170
45/45 [=====] - 69s 2s/step - loss: 0.9424 - accuracy: 0.6272
45/45 [=====] - 69s 2s/step - loss: 1.0652 - accuracy: 0.5170
Epoch 21/50
45/45 [=====] - 69s 2s/step - loss: 0.9156 - accuracy: 0.6682
45/45 [=====] - 69s 2s/step - loss: 1.0312 - accuracy: 0.5170
Epoch 22/50
45/45 [=====] - 69s 2s/step - loss: 0.9247 - accuracy: 0.6377
45/45 [=====] - 69s 2s/step - loss: 1.0523 - accuracy: 0.5284
Epoch 23/50
45/45 [=====] - 69s 2s/step - loss: 0.8900 - accuracy: 0.6619
45/45 [=====] - 69s 2s/step - loss: 1.0828 - accuracy: 0.4716
Epoch 24/50
45/45 [=====] - 69s 2s/step - loss: 0.8829 - accuracy: 0.6756
45/45 [=====] - 69s 2s/step - loss: 1.0488 - accuracy: 0.5284
Epoch 25/50
45/45 [=====] - 69s 2s/step - loss: 0.8544 - accuracy: 0.6783
45/45 [=====] - 69s 2s/step - loss: 1.0523 - accuracy: 0.5227
Epoch 26/50
45/45 [=====] - 68s 2s/step - loss: 0.8658 - accuracy: 0.6761
45/45 [=====] - 68s 2s/step - loss: 1.0555 - accuracy: 0.5200
Epoch 27/50
45/45 [=====] - 71s 2s/step - loss: 0.8149 - accuracy: 0.6606
45/45 [=====] - 69s 2s/step - loss: 0.8152 - accuracy: 0.7062
45/45 [=====] - 69s 2s/step - loss: 1.0523 - accuracy: 0.5227
45/45 [=====] - 69s 2s/step - loss: 0.7377 - accuracy: 0.7334
45/45 [=====] - 69s 2s/step - loss: 1.0519 - accuracy: 0.5114
45/45 [=====] - 69s 2s/step - loss: 0.7096 - accuracy: 0.7418
45/45 [=====] - 69s 2s/step - loss: 1.0309 - accuracy: 0.5284
45/45 [=====] - 69s 2s/step - loss: 0.7416 - accuracy: 0.7069
45/45 [=====] - 70s 2s/step - loss: 0.7224 - accuracy: 0.7552
45/45 [=====] - 69s 2s/step - loss: 1.0803 - accuracy: 0.4886
Epoch 33/50
45/45 [=====] - 69s 2s/step - loss: 0.7301 - accuracy: 0.7601
45/45 [=====] - 69s 2s/step - loss: 1.0695 - accuracy: 0.5284
Epoch 34/50
45/45 [=====] - 69s 2s/step - loss: 0.6834 - accuracy: 0.7277
45/45 [=====] - 69s 2s/step - loss: 1.0834 - accuracy: 0.5114
Epoch 35/50
45/45 [=====] - 69s 2s/step - loss: 0.6751 - accuracy: 0.7485
45/45 [=====] - 69s 2s/step - loss: 1.0895 - accuracy: 0.5170
Epoch 36/50
45/45 [=====] - 69s 2s/step - loss: 0.6780 - accuracy: 0.7548
45/45 [=====] - 69s 2s/step - loss: 1.0504 - accuracy: 0.5284
Epoch 37/50
45/45 [=====] - 69s 2s/step - loss: 0.6469 - accuracy: 0.7456
45/45 [=====] - 69s 2s/step - loss: 1.1093 - accuracy: 0.5300
Epoch 38/50
45/45 [=====] - 69s 2s/step - loss: 0.5993 - accuracy: 0.8144
45/45 [=====] - 69s 2s/step - loss: 1.0682 - accuracy: 0.5227
Epoch 39/50
45/45 [=====] - 69s 2s/step - loss: 0.6231 - accuracy: 0.7856
45/45 [=====] - 69s 2s/step - loss: 1.0682 - accuracy: 0.5227
45/45 [=====] - 69s 2s/step - loss: 0.6192 - accuracy: 0.7827
45/45 [=====] - 69s 2s/step - loss: 1.0678 - accuracy: 0.5114
Epoch 40/50
45/45 [=====] - 69s 2s/step - loss: 0.5944 - accuracy: 0.7945
45/45 [=====] - 69s 2s/step - loss: 1.0807 - accuracy: 0.5511
Epoch 41/50
45/45 [=====] - 69s 2s/step - loss: 0.5996 - accuracy: 0.8080
45/45 [=====] - 69s 2s/step - loss: 1.0706 - accuracy: 0.5114
Epoch 42/50
45/45 [=====] - 69s 2s/step - loss: 0.5357 - accuracy: 0.7958
45/45 [=====] - 69s 2s/step - loss: 1.0871 - accuracy: 0.5398
Epoch 44/50
45/45 [=====] - 69s 2s/step - loss: 0.5224 - accuracy: 0.8251
45/45 [=====] - 69s 2s/step - loss: 1.0693 - accuracy: 0.5739
Epoch 45/50
45/45 [=====] - 69s 2s/step - loss: 0.5467 - accuracy: 0.8271
45/45 [=====] - 69s 2s/step - loss: 1.1028 - accuracy: 0.5511
Epoch 46/50
45/45 [=====] - 69s 2s/step - loss: 0.5019 - accuracy: 0.8169
45/45 [=====] - 69s 2s/step - loss: 1.1278 - accuracy: 0.5284
Epoch 47/50
45/45 [=====] - 69s 2s/step - loss: 0.5060 - accuracy: 0.8508
45/45 [=====] - 69s 2s/step - loss: 1.1076 - accuracy: 0.5455
Epoch 48/50
45/45 [=====] - 69s 2s/step - loss: 0.4981 - accuracy: 0.8243
45/45 [=====] - 69s 2s/step - loss: 1.0860 - accuracy: 0.5455
Epoch 49/50
45/45 [=====] - 69s 2s/step - loss: 0.5554 - accuracy: 0.8165
45/45 [=====] - 69s 2s/step - loss: 1.1376 - accuracy: 0.5170
Epoch 50/50
45/45 [=====] - 69s 2s/step - loss: 0.4959 - accuracy: 0.8373
45/45 [=====] - 69s 2s/step - loss: 1.0867 - accuracy: 0.5568
```

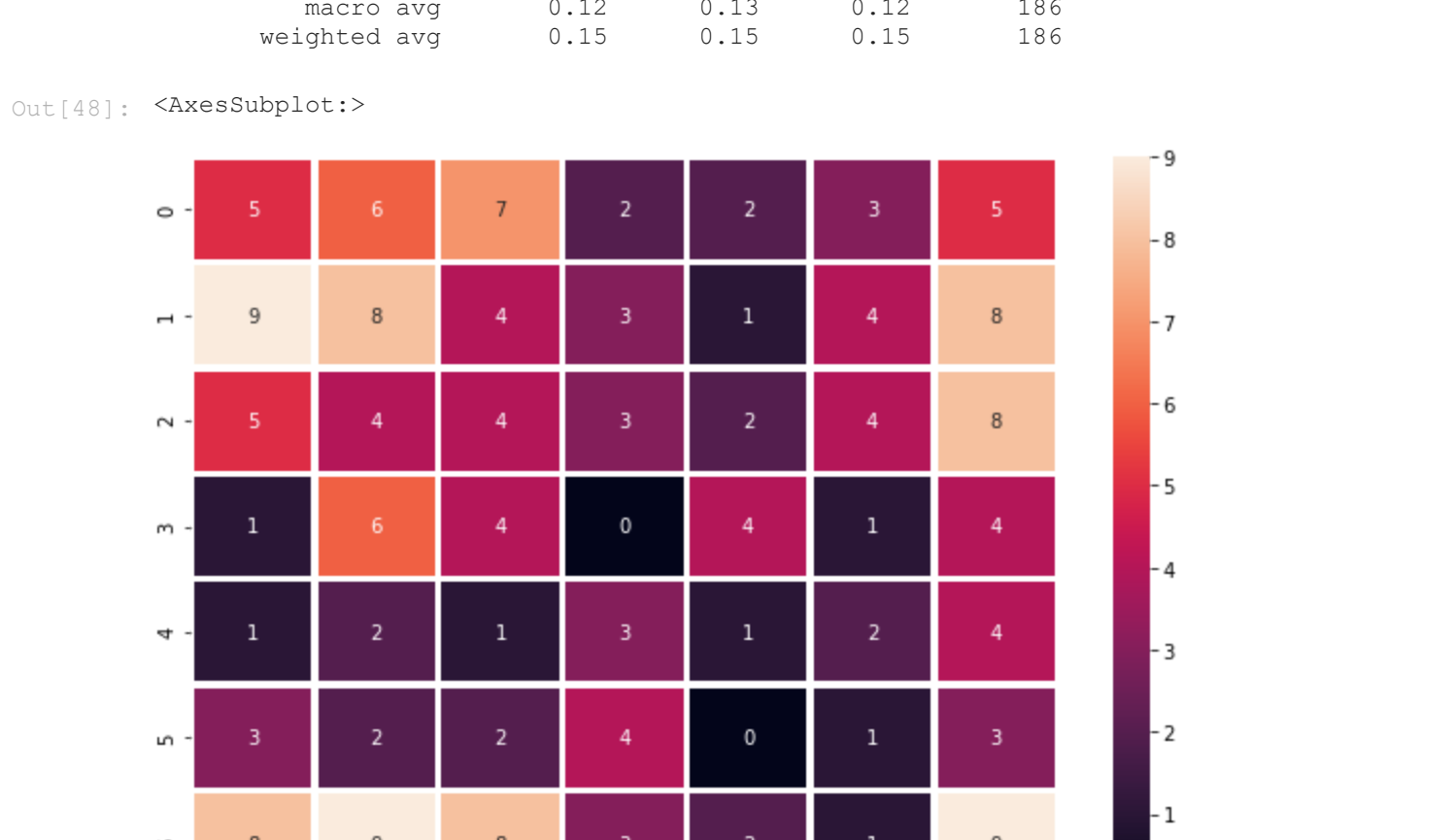
```
In [44]: #Avaliando o modelo
loss_train, test_acc = model.evaluate_generator(train_generator, steps=step_size_train)
loss_test, test_acc = model.evaluate_generator(test_generator, steps=step_size_test)
print('Train: %.3f, Test: %.3f' % (train_acc, test_acc))
```

```
c:\users\vinicius\appdata\local\programs\python\python39\lib\site-packages\tensorflow\python\keras\engine\training.py:1973: UserWarning: 'Model.evaluate_generator' is deprecated and will be removed in a future version. Please use 'Model.evaluate', which supports generators.
warnings.warn('Model.evaluate_generator' is deprecated and '
Train: 0.971, Test: 0.557
```

```
In [45]: #Apresentando resultados em graficos
plt.title('Loss')
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.legend()
plt.show()
```



```
In [46]: # Criando graficos para visualização dos resultados
print()
plt.title('Accuracy')
plt.plot(history.history['accuracy'], label='train')
plt.plot(history.history['val_accuracy'], label='test')
plt.legend()
plt.show()
```



```
In [47]: print('Criando classificações...')
labels = os.listdir('Database')
print('Rotulos', labels)
#criando estruturas para métricas de avaliação, processo um pouco mais demorado
y_pred = model.predict_generator(test_generator)
print('Preds Created')
y_pred = np.argmax(Y_pred, axis=1)
print('Preds 1D created')
```

```
Criando classificações...
Rotulos ['carcinoma in situ', 'light dysplastic', 'moderate dysplastic', 'normal columnar', 'normal intermediate', 'normal superficial', 'severe dysplastic']
c:\users\vinicius\appdata\local\programs\python\python39\lib\site-packages\tensorflow\python\keras\engine\training.py:2001: UserWarning: 'Model.predict_generator' is deprecated and will be removed in a future version. Please use 'Model.predict', which supports generators.
warnings.warn('Model.predict_generator' is deprecated and '
Preds created
Preds 1D created
```

```
In [48]: classification = classification_report(test_generator.classes, y_pred, target_names=labels)
print(classification)
matrix = confusion_matrix(test_generator.classes, y_pred)
df_cm = pd.DataFrame(matrix, index = [i for i in range(7)],
                      columns = [i for i in range(7)])
plt.figure(figsize = (10,7))
sn.heatmap(df_cm, annot=True, linewidth=2.5)
```

		precision	recall	f1-score	support
	carcinoma_in_situ	0.16	0.17	0.16	30
	light_dysplastic	0.22	0.22	0.22	37
	moderate_dysplastic	0.13	0.13	0.13	30
	normal_columnar	0.00	0.00	0.00	20
	normal_intermediate	0.08	0.07	0.08	14
	normal_superficial	0.06	0.07	0.06	15
	severe_dysplastic	0.22	0.23	0.22	40
	accuracy			0.15	186
	macro avg	0.12	0.13	0.12	186
	weighted avg	0.15	0.15	0.15	186



```
In [ ] :
```