Skin lesion segmentation

Robson Novato
Augusto Ferreira
Pedro Lucas

O que é?

- Processo de identificação e isolamento de áreas anormais na pele.
- É uma tarefa importante em dermatologia clínica e pesquisa, pois ajuda a detectar e monitorar condições de pele, como câncer de pele e psoríase.
- Envolve a separação da área da pele afetada do restante da imagem de pele.
- É um desafio devido a variações na cor, tamanho e forma das lesões, bem como a presença de artefatos e ruídos na imagem.
- A precisão é importante para o diagnóstico correto de condições de pele e para o monitoramento da eficácia do tratamento.

Funcionamento do Algoritmo

Dataset:

- Banco de imagens do desafio ISIC_2020
- O conjunto de dados contém 33.126 imagens de treinamento dermatoscópicas de lesões cutâneas benignas e malignas únicas de mais de 2.000 pacientes. Todos os diagnósticos malignos foram confirmados por histopatologia e os diagnósticos benignos foram confirmados usando acordo de especialistas, acompanhamento longitudinal ou histopatologia.

```
image_name,patient_id,lesion_id,sex,age_approx,anatom_site_general_challenge,diagnosis,benign_malignant,target
ISIC_2637011,IP_7279968,IL_7972535,male,45,head/neck,unknown,benign,0
ISIC_0015719,IP_3075186,IL_4649854,female,45,upper extremity,unknown,benign,0
ISIC_0052212,IP_2842074,IL_9087444,female,50,lower extremity,nevus,benign,0
ISIC_0068279,IP_6890425,IL_4255399,female,45,head/neck,unknown,benign,0
ISIC_0074268,IP_8723313,IL_6898037,female,55,upper extremity,unknown,benign,0
ISIC_0074311,IP_2950485,IL_3593551,female,40,lower extremity,unknown,benign,0
ISIC_0074542,IP_4698288,IL_5017890,male,25,lower extremity,unknown,benign,0
ISIC_0075663,IP_6017204,IL_1711395,female,35,torso,unknown,benign,0
ISIC_0075914,IP_7622888,IL_8599857,male,30,torso,unknown,benign,0
ISIC_0076262,IP_5075533,IL_8541111,female,50,lower extremity,unknown,benign,0
ISIC_0076545,IP_9802602,IL_2772532,male,55,upper extremity,unknown,benign,0
ISIC_0076742,IP_2318163,IL_9716539,male,75,upper extremity,unknown,benign,0
ISIC_0076995,IP_2235340,IL_7147389,female,55,torso,nevus,benign,0
ISIC_0077472,IP_3691360,IL_1155814,female,40,torso,unknown,benign,0
ISIC_0077735,IP_1109756,IL_6062320,male,70,torso,unknown,benign,0
ISIC_0078703,IP_7279968,IL_6850356,male,45,torso,unknown,benign,0
ISIC_0078712,IP_2189124,IL_9423574,male,40,lower extremity,unknown,benign,0
ISIC_0079038,IP_5295861,IL_1642984,male,70,torso,unknown,benign,0
ISIC_0080512,IP_1870306,IL_3564480,male,75,torso,unknown,benign,0
ISIC_0080752,IP_2613684,IL_7587923,male,50,torso,unknown,benign,0
ISIC_0080817,IP_7318404,IL_0565635,male,50,lower extremity,unknown,benign,0
ISIC_0081956,IP_2010919,IL_9016564,female,50,upper extremity,unknown,benign,0
ISIC_0082348,IP_7684360,IL_4059094,male,55,torso,unknown,benign,0
ISIC_0082543,IP_9463965,IL_4853191,female,30,torso,unknown,benign,0
ISIC_0082934,IP_6572129,IL_7831003,male,65,torso,unknown,benign,0
ISIC_0083035,IP_5805281,IL_2812726,male,50,torso,unknown,benign,0
```

Treinamento

- Carregamento dos dados
- Treinar fazendo a segmentação ou não.
- Começamos fazendo o split do dataset

```
# ------------ Dataset split ------------ #
X_train, X_test, y_train, y_test = train_test_split(_images, labels, train_size=0.9, random_state=6, stratify=labels)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, train_size=0.9, random_state=6, stratify=y_train)

train_count, val_count, test_count = np.bincount(y_train), np.bincount(y_val), np.bincount(y_test)
print("""Dataset split:
    Train set:      {} benign ({:.2%}), {} malignant ({:.2%})
    Validation set: {} benign ({:.2%}), {} malignant ({:.2%})
    Test set:       {} benign ({:.2%}), {} malignant ({:.2%})
""".format(
    train_count[0], train_count[0]/sum(train_count), train_count[1], train_count[1]/sum(train_count),
    val_count[0],   val_count[0]/sum(val_count),     val_count[1],   val_count[1]/sum(val_count),
    test_count[0],  test_count[0]/sum(test_count),   test_count[1],  test_count[1]/sum(test_count)
))
```

Criação do modelo

- Modelo sequencial EfficientNet
- Estratégia de espelhamento do tensorFlow

```
# ------------ Training ------------ #
print("Starting training model")

mirrored_strategy = tf.distribute.MirroredStrategy()

with mirrored_strategy.scope():
    efficientnet = EfficientNet(weights='imagenet', include_top=False, input_shape=input_shape, classes=2)

    model = keras.models.Sequential()
    model.add(efficientnet)
    model.add(keras.layers.GlobalAveragePooling2D())
    model.add(keras.layers.Dense(1, activation='sigmoid'))

    model.summary()

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

Alguns callbacks do Keras

- Os callbacks são usados para monitorar o progresso do treinamento e tomar atitudes de acordo com os resultados

1) EarlyStopping
2) ReduceLROnPlateau
3) ModelCheckpoint

Treinamento do modelo usando o fit do Keras

```
try:
    history = model.fit(
        X_train,
        y_train,
        validation_data=(X_val, y_val),
        epochs=epochs,
        batch_size=batch_size,
        verbose=2,
        callbacks=callbacks,
        shuffle=True,
    )
except:
    pass
```

Teste de uma imagem

./test.py image model [--segment]

Explicação da segmentação:

```python
# KMeans segmentation
def kmeans_mask(image, return_rgb=False):
    K = 2
    attempts = 1
    _, labels, centers = cv2.kmeans(np.float32(image.reshape((-1, 3))), K, None, None, attempts, cv2.KMEANS_RANDOM_CENTERS)
    centers = np.uint8(centers)
    lesion_cluster = np.argmin(np.mean(centers, axis=1))
    lesion_mask = labels.flatten() == lesion_cluster
    if return_rgb:
        rgb_mask = np.zeros(image.shape)
        rgb_mask[~lesion_mask.reshape(image.shape[:2])] = 255
        return rgb_mask
    return lesion_mask

def kmeans_segmentation(image, force_copy=True, mask=None):
    lesion_mask = mask if mask else kmeans_mask(image)
    segmented_img = image.reshape((-1, 3))
    if force_copy and segmented_img.base is image:
        segmented_img = segmented_img.copy()
    segmented_img[~lesion_mask] = 255
    return segmented_img.reshape(image.shape)
```

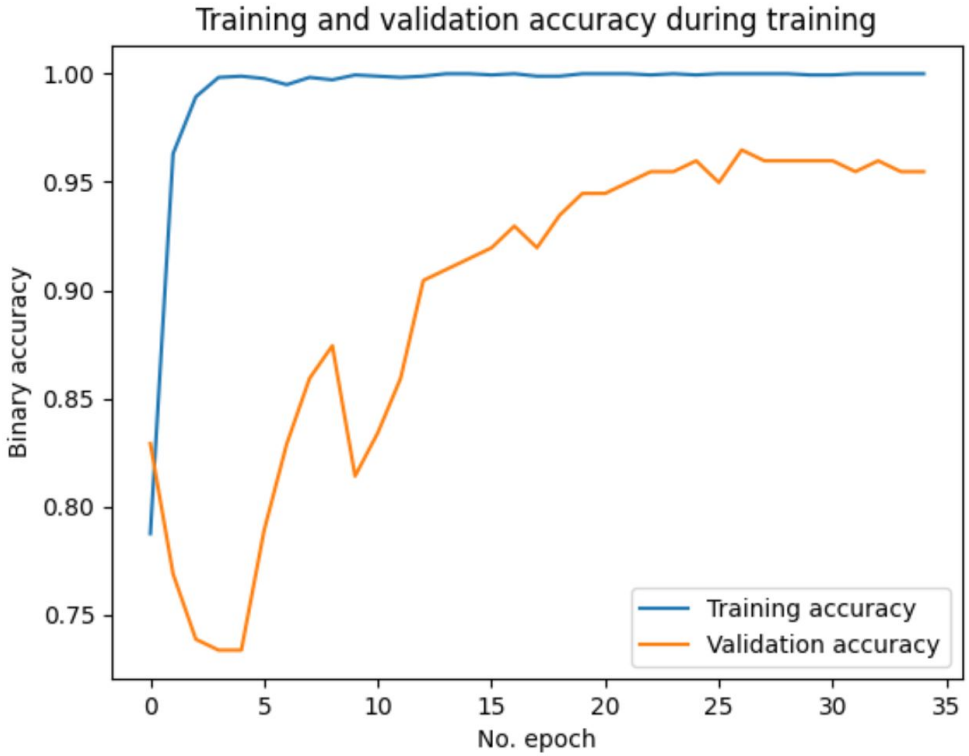
Os testes

```
[→  deteccao-melanoma git:(main) × ./test.py dataset/train_jpeg/ISIC_0298093.jpg modeloTreinado.h5
2023-03-14 00:59:26.945770: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow bi
ions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild Te
Loading model...
Loading image...
1/1 [==============================] - 1s 1s/step
Benign (malignant probability: 3.567799%)
```

```python
prediction = model.predict(image[np.newaxis]).ravel()[0]

print(f"{{}} (malignant probability: {prediction:%})".format(
    'Benign' if prediction < 0.5 else 'Malignant'
))
```

Os resultados - sem segmentação



Os resultados - com segmentação