

# cancer-aprendizagem-profunda

May 15, 2025

```
[1]: # Carrega a extensão
      %load_ext watermark

      # Lista TODOS os pacotes instalados (via pip) + metadados
      %watermark -d -t -m -v
      !pip list # Exibe a lista completa de pacotes
```

Python implementation: CPython  
Python version : 3.10.13  
IPython version : 8.36.0

Compiler : GCC 12.3.0  
OS : Linux  
Release : 6.1.0-34-amd64  
Machine : x86\_64  
Processor :  
CPU cores : 2  
Architecture: 64bit

| Package              | Version |
|----------------------|---------|
| absl-py              | 2.2.0   |
| aiohappyeyeballs     | 2.6.1   |
| aiohttp              | 3.11.18 |
| aiosignal            | 1.3.2   |
| anyio                | 4.9.0   |
| argon2-cffi          | 23.1.0  |
| argon2-cffi-bindings | 21.2.0  |
| arrow                | 1.3.0   |
| asttokens            | 3.0.0   |
| astunparse           | 1.6.3   |
| async-lru            | 2.0.5   |
| async-timeout        | 5.0.1   |
| attrs                | 25.3.0  |
| babel                | 2.17.0  |
| beautifulsoup4       | 4.13.4  |
| bleach               | 6.2.0   |
| blinker              | 1.9.0   |

|                           |           |
|---------------------------|-----------|
| Brotli                    | 1.1.0     |
| cached-property           | 1.5.2     |
| cachetools                | 5.5.2     |
| certifi                   | 2025.4.26 |
| cffi                      | 1.17.1    |
| charset-normalizer        | 3.4.2     |
| click                     | 8.2.0     |
| comm                      | 0.2.2     |
| contourpy                 | 1.3.2     |
| cryptography              | 44.0.3    |
| cycler                    | 0.12.1    |
| debugpy                   | 1.8.14    |
| decorator                 | 5.2.1     |
| defusedxml                | 0.7.1     |
| exceptiongroup            | 1.3.0     |
| executing                 | 2.2.0     |
| fastjsonschema            | 2.21.1    |
| flatbuffers               | 25.2.10   |
| fonttools                 | 4.58.0    |
| fqdn                      | 1.5.1     |
| frozenlist                | 1.6.0     |
| gast                      | 0.6.0     |
| google-auth               | 2.40.1    |
| google-auth-oauthlib      | 1.2.2     |
| google-pasta              | 0.2.0     |
| grpcio                    | 1.54.3    |
| h11                       | 0.16.0    |
| h2                        | 4.2.0     |
| h5py                      | 3.13.0    |
| hpack                     | 4.1.0     |
| httpcore                  | 1.0.9     |
| httpx                     | 0.28.1    |
| hyperframe                | 6.1.0     |
| idna                      | 3.10      |
| importlib_metadata        | 8.6.1     |
| importlib_resources       | 6.5.2     |
| imutils                   | 0.5.4     |
| ipykernel                 | 6.29.5    |
| ipython                   | 8.36.0    |
| ipywidgets                | 8.1.7     |
| isoduration               | 20.11.0   |
| jedi                      | 0.19.2    |
| Jinja2                    | 3.1.6     |
| joblib                    | 1.5.0     |
| json5                     | 0.12.0    |
| jsonpointer               | 3.0.0     |
| jsonschema                | 4.23.0    |
| jsonschema-specifications | 2025.4.1  |

|                          |        |
|--------------------------|--------|
| jupyter                  | 1.0.0  |
| jupyter_client           | 8.6.3  |
| jupyter-console          | 6.6.3  |
| jupyter_core             | 5.7.2  |
| jupyter-events           | 0.12.0 |
| jupyter-lsp              | 2.2.5  |
| jupyter_server           | 2.16.0 |
| jupyter_server_terminals | 0.5.3  |
| jupyterlab               | 4.4.2  |
| jupyterlab_pygments      | 0.3.0  |
| jupyterlab_server        | 2.27.3 |
| jupyterlab_widgets       | 3.0.15 |
| keras                    | 2.15.0 |
| kiwisolver               | 1.4.7  |
| Markdown                 | 3.8    |
| MarkupSafe               | 3.0.2  |
| matplotlib               | 3.8.0  |
| matplotlib-inline        | 0.1.7  |
| mistune                  | 3.1.3  |
| ml-dtypes                | 0.2.0  |
| multidict                | 6.4.3  |
| munkres                  | 1.1.4  |
| nbclient                 | 0.10.2 |
| nbconvert                | 7.16.6 |
| nbformat                 | 5.10.4 |
| nest_asyncio             | 1.6.0  |
| notebook                 | 7.4.2  |
| notebook_shim            | 0.2.4  |
| numpy                    | 1.26.0 |
| oauthlib                 | 3.2.2  |
| opencv-python            | 4.8.1  |
| opt_einsum               | 3.4.0  |
| overrides                | 7.7.0  |
| packaging                | 25.0   |
| pandas                   | 2.2.3  |
| pandocfilters            | 1.5.0  |
| parso                    | 0.8.4  |
| patsy                    | 1.0.1  |
| pexpect                  | 4.9.0  |
| pickleshare              | 0.7.5  |
| pillow                   | 11.2.1 |
| pip                      | 23.3.1 |
| pkgutil_resolve_name     | 1.3.10 |
| platformdirs             | 4.3.8  |
| ply                      | 3.11   |
| prometheus_client        | 0.21.1 |
| prompt_toolkit           | 3.0.51 |
| propcache                | 0.3.1  |

|                         |             |
|-------------------------|-------------|
| protobuf                | 4.21.12     |
| psutil                  | 7.0.0       |
| ptyprocess              | 0.7.0       |
| pure_eval               | 0.2.3       |
| py3nvml                 | 0.2.7       |
| pyasn1                  | 0.6.1       |
| pyasn1_modules          | 0.4.2       |
| pycparser               | 2.22        |
| Pygments                | 2.19.1      |
| PyJWT                   | 2.10.1      |
| pyOpenSSL               | 25.0.0      |
| pyparsing               | 3.2.3       |
| PyQt5                   | 5.15.9      |
| PyQt5-sip               | 12.12.2     |
| PySocks                 | 1.7.1       |
| python-dateutil         | 2.9.0.post0 |
| python-json-logger      | 2.0.7       |
| pytz                    | 2025.2      |
| pyu2f                   | 0.1.5       |
| PyYAML                  | 6.0.2       |
| pyzmq                   | 26.4.0      |
| qtconsole               | 5.6.1       |
| QtPy                    | 2.4.3       |
| referencing             | 0.36.2      |
| requests                | 2.32.3      |
| requests-oauthlib       | 2.0.0       |
| rfc3339_validator       | 0.1.4       |
| rfc3986-validator       | 0.1.1       |
| rpds-py                 | 0.24.0      |
| rsa                     | 4.9.1       |
| scikit-learn            | 1.3.2       |
| scipy                   | 1.15.2      |
| seaborn                 | 0.13.0      |
| Send2Trash              | 1.8.3       |
| setuptools              | 80.1.0      |
| sip                     | 6.7.12      |
| six                     | 1.17.0      |
| sniffio                 | 1.3.1       |
| soupsieve               | 2.7         |
| stack_data              | 0.6.3       |
| statsmodels             | 0.14.4      |
| tensorboard             | 2.15.2      |
| tensorboard_data_server | 0.7.0       |
| tensorflow              | 2.15.0      |
| tensorflow-estimator    | 2.15.0      |
| termcolor               | 3.1.0       |
| terminado               | 0.18.1      |
| threadpoolctl           | 3.6.0       |

|                       |                |
|-----------------------|----------------|
| tinycss2              | 1.4.0          |
| toml                  | 0.10.2         |
| tomli                 | 2.2.1          |
| tornado               | 6.4.2          |
| traitlets             | 5.14.3         |
| types-python-dateutil | 2.9.0.20241206 |
| typing_extensions     | 4.13.2         |
| typing_utils          | 0.1.0          |
| tzdata                | 2025.2         |
| unicodedata2          | 16.0.0         |
| uri-template          | 1.3.0          |
| urllib3               | 2.4.0          |
| watermark             | 2.4.0          |
| wcwidth               | 0.2.13         |
| webcolors             | 24.11.1        |
| webencodings          | 0.5.1          |
| websocket-client      | 1.8.0          |
| Werkzeug              | 3.1.3          |
| wheel                 | 0.45.1         |
| widgetsnbextension    | 4.0.14         |
| wrapt                 | 1.14.1         |
| xmltodict             | 0.14.2         |
| yaml                  | 1.20.0         |
| zipp                  | 3.21.0         |
| zstandard             | 0.23.0         |

```
[2]: # Imports e configurações

import os
import cv2
import random
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Activation, Flatten, Dense, Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator, img_to_array
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from imutils import paths
import imutils
import seaborn as sns
```

```
%matplotlib inline
plt.rcParams['figure.figsize'] = (10, 6)
```

2025-05-15 01:04:29.471199: I tensorflow/core/platform/cpu\_feature\_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.  
To enable the following instructions: SSE4.1 SSE4.2, in other operations, rebuild TensorFlow with the appropriate compiler flags.

```
[3]: # Carregamento das imagens (Label pelo nome do arquivo)

imagens_treino = "./dados/treino/"

imagens = []
labels = []

# Lista todos os caminhos das imagens
imagePaths = sorted(list(paths.list_images(imagens_treino)))
random.seed(42)
random.shuffle(imagePaths)

for imagePath in imagePaths:
    # Leitura e redimensionamento
    image = cv2.imread(imagePath)
    image = cv2.resize(image, (40, 40))
    image = img_to_array(image)
    imagens.append(image)

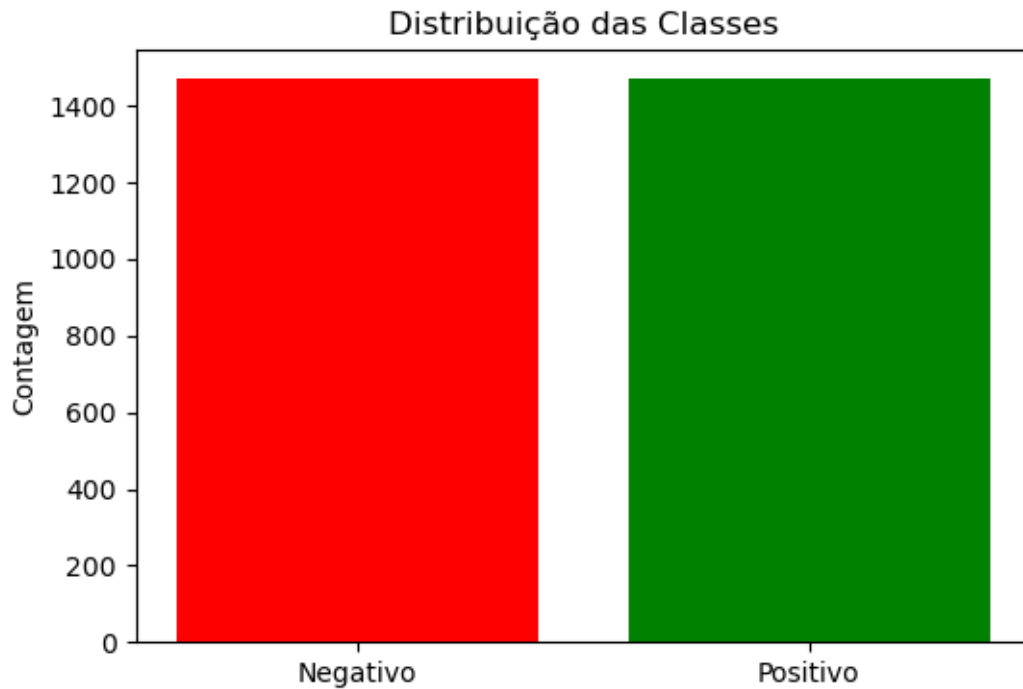
    # Extrai o label do nome do arquivo (ex: "imagem_pos.jpg")
    filename = imagePath.split(os.path.sep)[-1] # Pega o nome do arquivo
    if "pos" in filename.lower():
        label = 1
    elif "neg" in filename.lower():
        label = 0
    else:
        raise ValueError(f"Arquivo {filename} não contém 'pos' ou 'neg' no nome!")
    labels.append(label)

# Normalização
imagens = np.array(imagens, dtype="float32") / 255.0
labels = np.array(labels)
```

```
[4]: # Análise de distribuição das classes
```

```
plt.figure(figsize=(6, 4))
```

```
plt.bar(["Negativo", "Positivo"], [np.sum(labels == 0), np.sum(labels == 1)],  
        color=["red", "green"])  
plt.title("Distribuição das Classes")  
plt.ylabel("Contagem")  
plt.show()
```



```
[5]: # Pré-processamento  
  
# Divisão treino/teste  
(x_treino, x_teste, y_treino, y_teste) = train_test_split(imagens, labels,  
        test_size=0.25, random_state=42)  
  
# One-Hot Encoding  
y_treino = to_categorical(y_treino, num_classes=2)  
y_teste = to_categorical(y_teste, num_classes=2)  
  
# Data Augmentation  
aug = ImageDataGenerator(  
    rotation_range=45,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.3,  
    zoom_range=0.3,  
    horizontal_flip=True,
```

```

        vertical_flip=True,
        fill_mode="nearest"
    )

# Pesos para classes desbalanceadas (ajuste conforme a distribuição)
class_weights = {0: 1, 1: 5} # Exemplo: classe "pos" tem peso 5x maior

```

[6]: *# Construção de modelo*

```

def build_improved_cnn(width, height, depth, classes):
    model = Sequential()
    input_shape = (height, width, depth)

    # Bloco Convolutacional 1
    model.add(Conv2D(32, (3, 3), padding="same", input_shape=input_shape))
    model.add(Activation("relu"))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    # Bloco Convolutacional 2
    model.add(Conv2D(64, (3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    # Bloco Convolutacional 3
    model.add(Conv2D(128, (3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    # Camadas Densas
    model.add(Flatten())
    model.add(Dense(512, activation="relu"))
    model.add(Dropout(0.5))

    # Saída
    model.add(Dense(classes, activation="softmax"))

    return model

# Hyperparâmetros
EPOCHS = 50
LR = 1e-4
BATCH_SIZE = 32

# Compilação
model = build_improved_cnn(40, 40, 3, 2)
model.compile(
    optimizer=Adam(learning_rate=LR),

```



```

        loss="categorical_crossentropy",
        metrics=["accuracy"]
    )
    model.summary()

```

Model: "sequential"

| Layer (type)                        | Output Shape        | Param # |
|-------------------------------------|---------------------|---------|
| conv2d (Conv2D)                     | (None, 40, 40, 32)  | 896     |
| activation (Activation)             | (None, 40, 40, 32)  | 0       |
| max_pooling2d (MaxPooling2D)        | (None, 20, 20, 32)  | 0       |
| conv2d_1 (Conv2D)                   | (None, 20, 20, 64)  | 18496   |
| activation_1 (Activation)           | (None, 20, 20, 64)  | 0       |
| max_pooling2d_1 (MaxPooling2D)      | (None, 10, 10, 64)  | 0       |
| conv2d_2 (Conv2D)                   | (None, 10, 10, 128) | 73856   |
| activation_2 (Activation)           | (None, 10, 10, 128) | 0       |
| max_pooling2d_2 (MaxPooling2D)      | (None, 5, 5, 128)   | 0       |
| flatten (Flatten)                   | (None, 3200)        | 0       |
| dense (Dense)                       | (None, 512)         | 1638912 |
| dropout (Dropout)                   | (None, 512)         | 0       |
| dense_1 (Dense)                     | (None, 2)           | 1026    |
| Total params: 1733186 (6.61 MB)     |                     |         |
| Trainable params: 1733186 (6.61 MB) |                     |         |
| Non-trainable params: 0 (0.00 Byte) |                     |         |

[7]: # Callbacks

```
callbacks = [
```

```

EarlyStopping(monitor="val_loss", patience=10, restore_best_weights=True),
ReduceLROnPlateau(monitor="val_loss", factor=0.1, patience=5)
]

```

[8]: # Treinamento

```

print("[INFO] Treinamento iniciado...")
history = model.fit(
    aug.flow(x_treino, y_treino, batch_size=BATCH_SIZE),
    validation_data=(x_teste, y_teste),
    steps_per_epoch=len(x_treino) // BATCH_SIZE,
    epochs=EPOCHS,
    callbacks=callbacks,
    class_weight=class_weights,
    verbose=1
)
print("[INFO] Treinamento concluído!")

```

[INFO] Treinamento iniciado...

Epoch 1/50

68/68 [=====] - 24s 314ms/step - loss: 1.3587 - accuracy: 0.5037 - val\_loss: 0.7087 - val\_accuracy: 0.5245 - lr: 1.0000e-04

Epoch 2/50

68/68 [=====] - 20s 284ms/step - loss: 1.0373 - accuracy: 0.6076 - val\_loss: 0.6854 - val\_accuracy: 0.6481 - lr: 1.0000e-04

Epoch 3/50

68/68 [=====] - 21s 303ms/step - loss: 1.0250 - accuracy: 0.6297 - val\_loss: 0.7300 - val\_accuracy: 0.6168 - lr: 1.0000e-04

Epoch 4/50

68/68 [=====] - 21s 300ms/step - loss: 1.0022 - accuracy: 0.6371 - val\_loss: 0.8390 - val\_accuracy: 0.5761 - lr: 1.0000e-04

Epoch 5/50

68/68 [=====] - 26s 378ms/step - loss: 1.0082 - accuracy: 0.6348 - val\_loss: 0.6847 - val\_accuracy: 0.6427 - lr: 1.0000e-04

Epoch 6/50

68/68 [=====] - 19s 274ms/step - loss: 0.9558 - accuracy: 0.6596 - val\_loss: 0.7774 - val\_accuracy: 0.5842 - lr: 1.0000e-04

Epoch 7/50

68/68 [=====] - 19s 279ms/step - loss: 0.9330 - accuracy: 0.6762 - val\_loss: 0.5954 - val\_accuracy: 0.6916 - lr: 1.0000e-04

Epoch 8/50

68/68 [=====] - 20s 293ms/step - loss: 0.9344 - accuracy: 0.6822 - val\_loss: 0.6338 - val\_accuracy: 0.6753 - lr: 1.0000e-04

Epoch 9/50

68/68 [=====] - 22s 316ms/step - loss: 0.9326 - accuracy: 0.6812 - val\_loss: 0.6147 - val\_accuracy: 0.6535 - lr: 1.0000e-04

Epoch 10/50

68/68 [=====] - 24s 345ms/step - loss: 0.9109 -

```

accuracy: 0.6780 - val_loss: 0.6553 - val_accuracy: 0.6549 - lr: 1.0000e-04
Epoch 11/50
68/68 [=====] - 20s 285ms/step - loss: 0.8829 -
accuracy: 0.6923 - val_loss: 0.7134 - val_accuracy: 0.6399 - lr: 1.0000e-04
Epoch 12/50
68/68 [=====] - 20s 296ms/step - loss: 0.8627 -
accuracy: 0.6863 - val_loss: 0.5794 - val_accuracy: 0.7147 - lr: 1.0000e-04
Epoch 13/50
68/68 [=====] - 21s 299ms/step - loss: 0.8622 -
accuracy: 0.6996 - val_loss: 0.5709 - val_accuracy: 0.7092 - lr: 1.0000e-04
Epoch 14/50
68/68 [=====] - 21s 302ms/step - loss: 0.8399 -
accuracy: 0.7254 - val_loss: 0.5170 - val_accuracy: 0.7351 - lr: 1.0000e-04
Epoch 15/50
68/68 [=====] - 19s 282ms/step - loss: 0.8387 -
accuracy: 0.7222 - val_loss: 0.5656 - val_accuracy: 0.7065 - lr: 1.0000e-04
Epoch 16/50
68/68 [=====] - 21s 300ms/step - loss: 0.8433 -
accuracy: 0.7240 - val_loss: 0.5760 - val_accuracy: 0.7147 - lr: 1.0000e-04
Epoch 17/50
68/68 [=====] - 19s 281ms/step - loss: 0.8384 -
accuracy: 0.7268 - val_loss: 0.6177 - val_accuracy: 0.6861 - lr: 1.0000e-04
Epoch 18/50
68/68 [=====] - 20s 285ms/step - loss: 0.8266 -
accuracy: 0.7171 - val_loss: 0.6133 - val_accuracy: 0.6984 - lr: 1.0000e-04
Epoch 19/50
68/68 [=====] - 20s 293ms/step - loss: 0.7925 -
accuracy: 0.7378 - val_loss: 0.5635 - val_accuracy: 0.7269 - lr: 1.0000e-04
Epoch 20/50
68/68 [=====] - 21s 302ms/step - loss: 0.7862 -
accuracy: 0.7636 - val_loss: 0.5972 - val_accuracy: 0.7133 - lr: 1.0000e-05
Epoch 21/50
68/68 [=====] - 22s 325ms/step - loss: 0.7778 -
accuracy: 0.7475 - val_loss: 0.5765 - val_accuracy: 0.7215 - lr: 1.0000e-05
Epoch 22/50
68/68 [=====] - 21s 309ms/step - loss: 0.7407 -
accuracy: 0.7585 - val_loss: 0.5416 - val_accuracy: 0.7351 - lr: 1.0000e-05
Epoch 23/50
68/68 [=====] - 21s 302ms/step - loss: 0.7560 -
accuracy: 0.7489 - val_loss: 0.5820 - val_accuracy: 0.7228 - lr: 1.0000e-05
Epoch 24/50
68/68 [=====] - 21s 303ms/step - loss: 0.7603 -
accuracy: 0.7493 - val_loss: 0.6091 - val_accuracy: 0.7160 - lr: 1.0000e-05
[INFO] Treinamento concluído!

```

[14]: # Avaliação

```

# Acurácia
test_loss, test_acc = model.evaluate(x_teste, y_teste, verbose=0)
print(f"Acurácia em teste: {test_acc * 100:.2f}%")

# Matriz de Confusão
y_pred = np.argmax(model.predict(x_teste), axis=1)
y_true = np.argmax(y_teste, axis=1)
cm = confusion_matrix(y_true, y_pred)

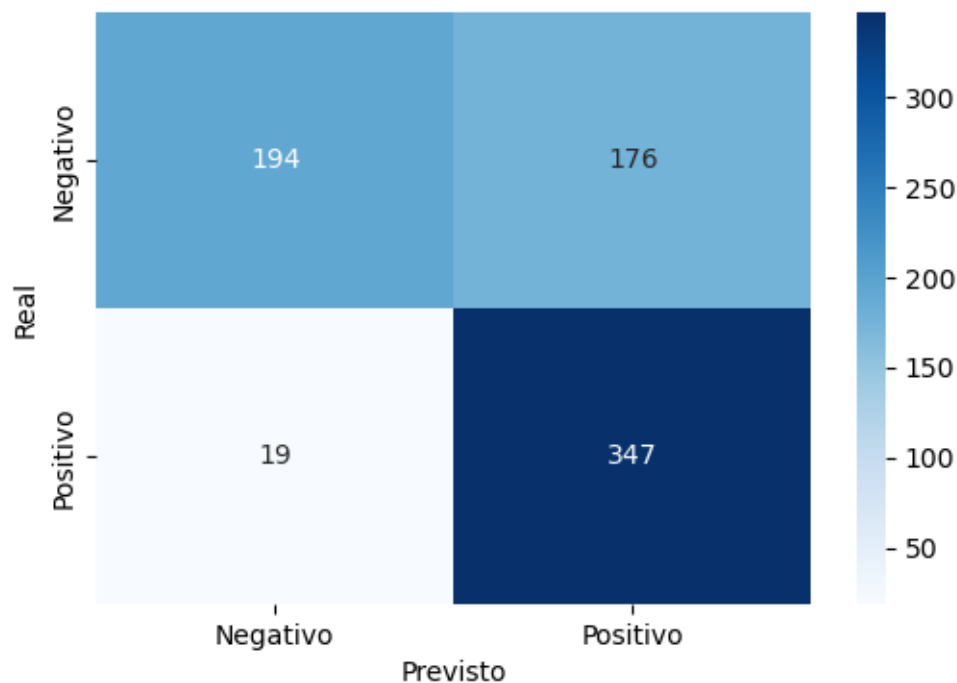
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["Negativo", "Positivo"], yticklabels=["Negativo", "Positivo"])
plt.xlabel("Previsto")
plt.ylabel("Real")
plt.show()

# Relatório de Classificação
print(classification_report(y_true, y_pred, target_names=["Negativo", "Positivo"]))

```

Acurácia em teste: 73.51%

23/23 [=====] - 3s 140ms/step



precision    recall    f1-score    support

|              |      |      |      |     |
|--------------|------|------|------|-----|
| Negativo     | 0.91 | 0.52 | 0.67 | 370 |
| Positivo     | 0.66 | 0.95 | 0.78 | 366 |
| accuracy     |      |      | 0.74 | 736 |
| macro avg    | 0.79 | 0.74 | 0.72 | 736 |
| weighted avg | 0.79 | 0.74 | 0.72 | 736 |

```
[10]: # Teste com imagem 1

image_path = "./dados/teste/imagem1_cancer.png" # Exemplo: nome deve conter
↳ "pos" ou "neg"
image = cv2.imread(image_path)
orig = image.copy()

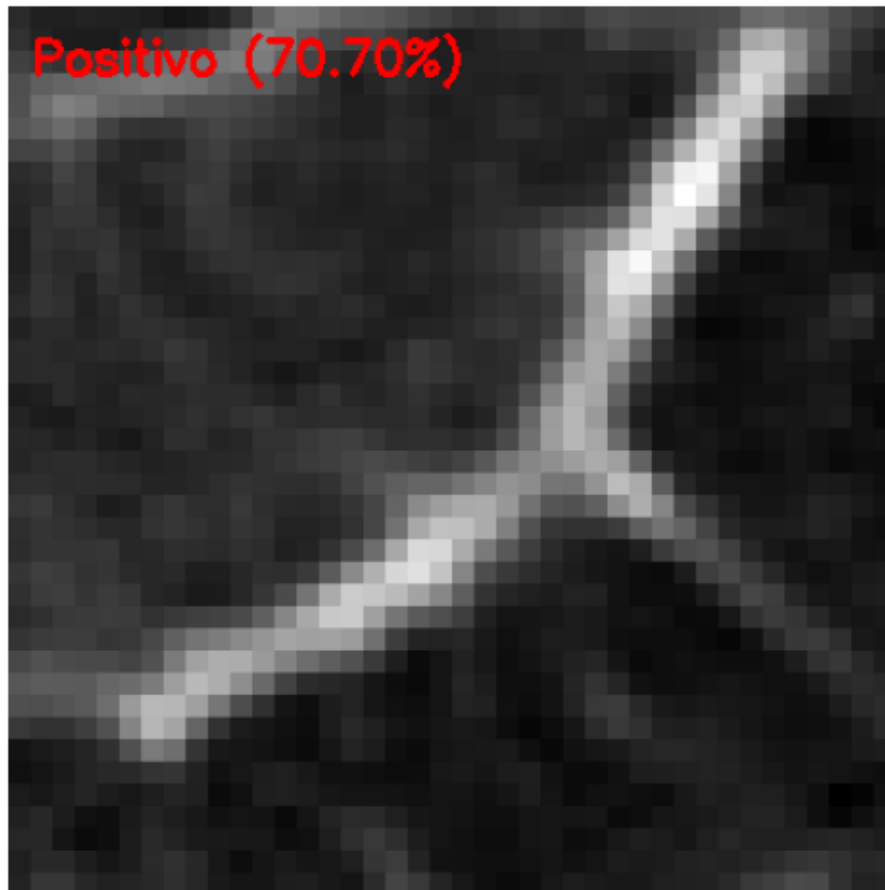
# Pré-processamento
image = cv2.resize(image, (40, 40))
image = image.astype("float32") / 255.0
image = np.expand_dims(image, axis=0)

# Predição
pred = model.predict(image)
label = "Positivo" if np.argmax(pred) == 1 else "Negativo"
proba = np.max(pred) * 100

# Exibição
output = imutils.resize(orig, width=400)
text = f"{label} ({proba:.2f}%)"
color = (0, 255, 0) if proba > 95 else (0, 0, 255)

cv2.putText(output, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)
plt.imshow(cv2.cvtColor(output, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.show()
```

1/1 [=====] - 0s 211ms/step



```
[11]: # Teste com imagem 2

image_path = "./dados/teste/imagem2_cancer.png" # Exemplo: nome deve conter
↳ "pos" ou "neg"
image = cv2.imread(image_path)
orig = image.copy()

# Pré-processamento
image = cv2.resize(image, (40, 40))
image = image.astype("float32") / 255.0
image = np.expand_dims(image, axis=0)

# Predição
pred = model.predict(image)
label = "Positivo" if np.argmax(pred) == 1 else "Negativo"
proba = np.max(pred) * 100

# Exibição
```

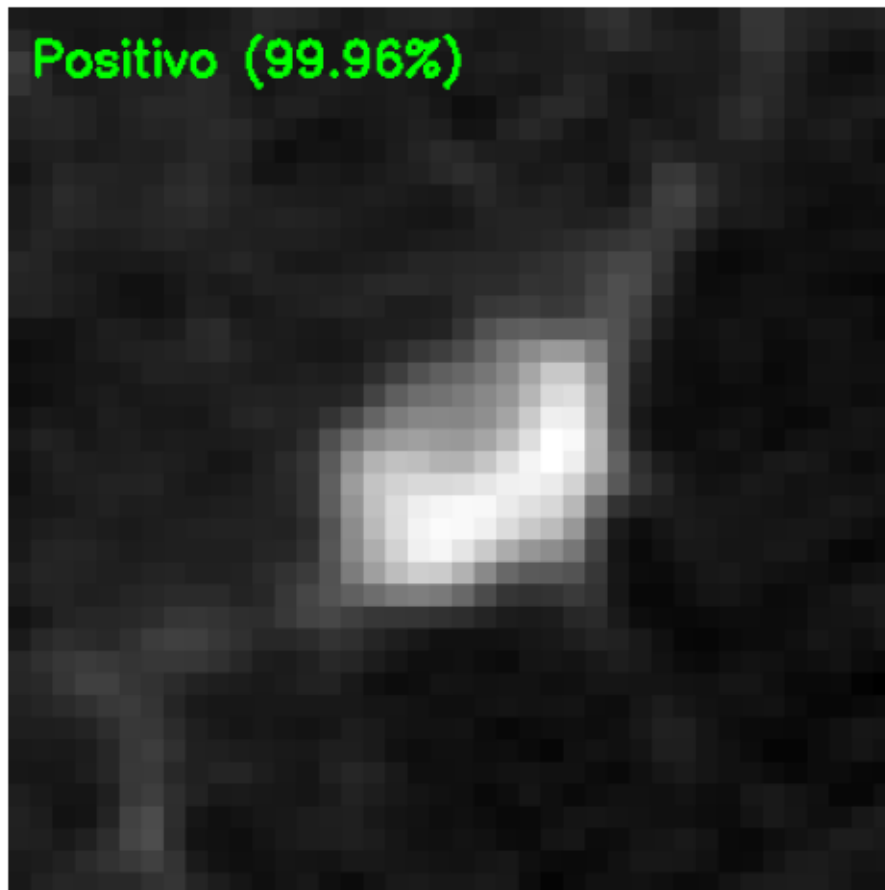
```

output = imutils.resize(orig, width=400)
text = f"{label} ({proba:.2f}%)"
color = (0, 255, 0) if proba > 95 else (0, 0, 255)

cv2.putText(output, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)
plt.imshow(cv2.cvtColor(output, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.show()

```

1/1 [=====] - 0s 55ms/step



[12]: # Teste com imagem 3

```

image_path = "./dados/teste/imagen3_no_cancer.png" # Exemplo: nome deve conter
↳ "pos" ou "neg"
image = cv2.imread(image_path)
orig = image.copy()

```

```

# Pré-processamento
image = cv2.resize(image, (40, 40))
image = image.astype("float32") / 255.0
image = np.expand_dims(image, axis=0)

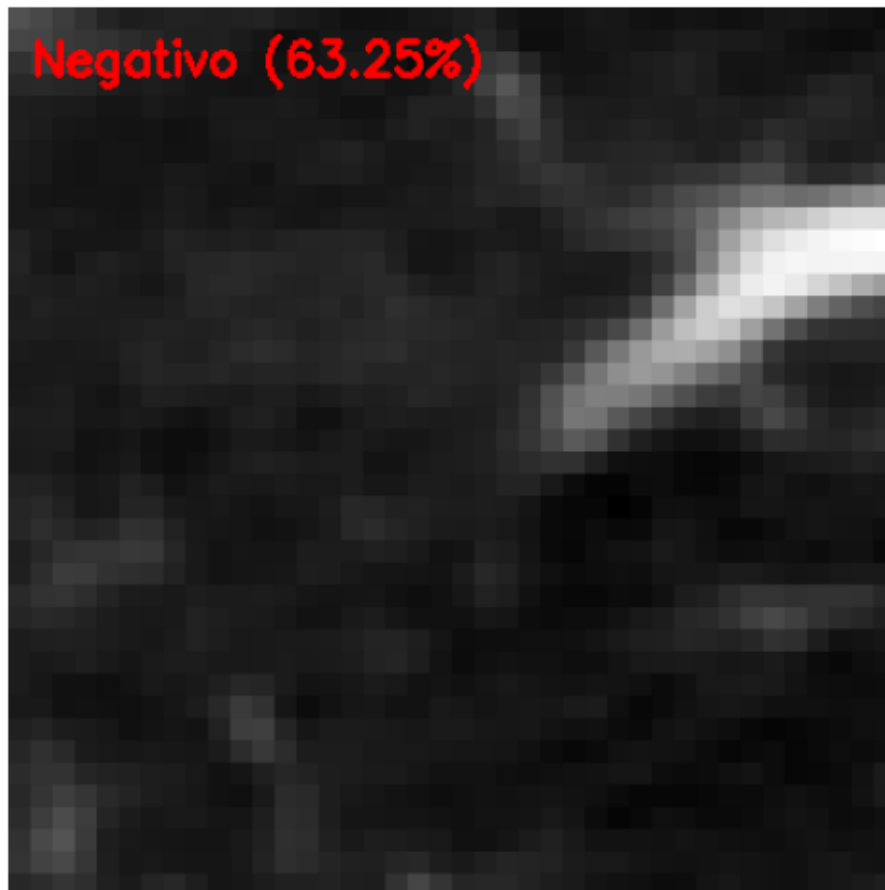
# Predição
pred = model.predict(image)
label = "Positivo" if np.argmax(pred) == 1 else "Negativo"
proba = np.max(pred) * 100

# Exibição
output = imutils.resize(orig, width=400)
text = f"{label} ({proba:.2f}%)"
color = (0, 255, 0) if proba > 95 else (0, 0, 255)

cv2.putText(output, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)
plt.imshow(cv2.cvtColor(output, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.show()

```

1/1 [=====] - 0s 90ms/step





```
[13]: # Teste com imagem 4

image_path = "./dados/teste/imagem4_no_cancer.png" # Exemplo: nome deve conter
↳ "pos" ou "neg"
image = cv2.imread(image_path)
orig = image.copy()

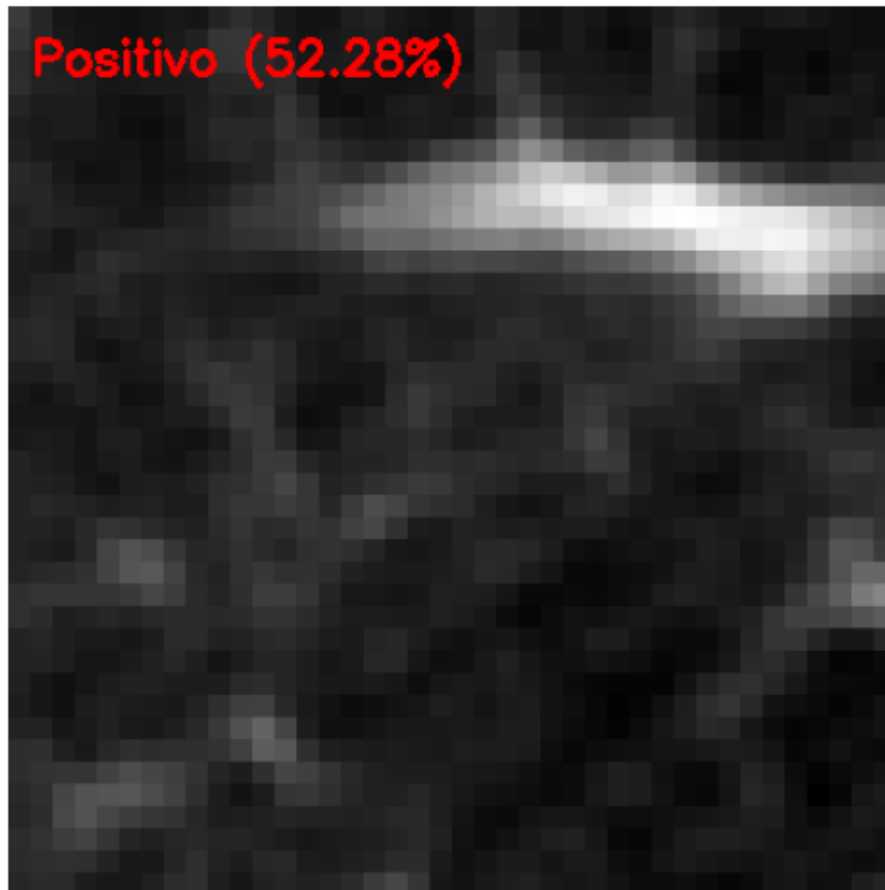
# Pré-processamento
image = cv2.resize(image, (40, 40))
image = image.astype("float32") / 255.0
image = np.expand_dims(image, axis=0)

# Predição
pred = model.predict(image)
label = "Positivo" if np.argmax(pred) == 1 else "Negativo"
proba = np.max(pred) * 100

# Exibição
output = imutils.resize(orig, width=400)
text = f"{label} ({proba:.2f}%)"
color = (0, 255, 0) if proba > 95 else (0, 0, 255)

cv2.putText(output, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)
plt.imshow(cv2.cvtColor(output, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.show()
```

1/1 [=====] - 0s 50ms/step



```
[17]: # Gerar arquivo de modelo

from tensorflow.keras.models import save_model

# Define o caminho do arquivo
model_path = "./modelo/modelo_treinado.keras"

# Salva o modelo
save_model(model, model_path)

# Verifica se o arquivo foi criado
import os
if os.path.exists(model_path):
    print(f"Modelo salvo com sucesso em: {model_path}")
else:
    print("Erro ao salvar o modelo!")
```

Modelo salvo com sucesso em: ./modelo/modelo\_treinado.keras

[ ]: