



Universidade do Minho
Aprendizagem Profunda

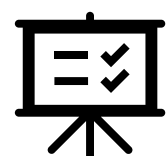
Segmentação **Do Cérebro Humano:** Arquitetura U-Net

Daniel Xavier PG50310

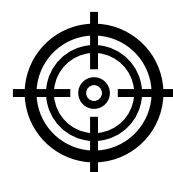
Diogo Rebelo PG50327



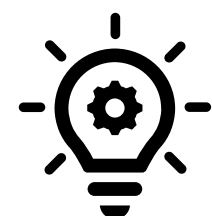
Motivação e Objetivos



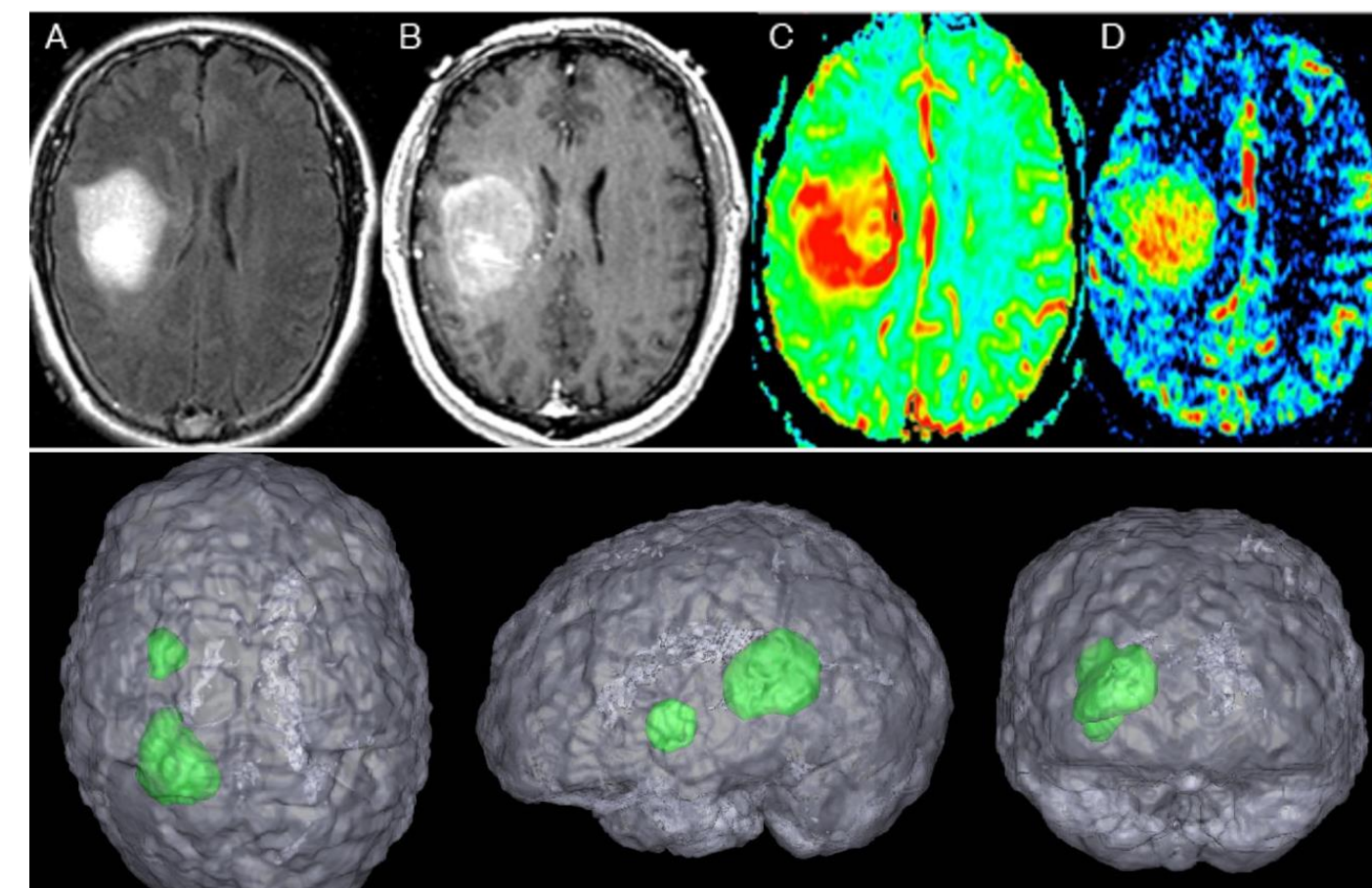
Temática: Desenvolvimento de um modelo capaz de sintetizar imagens de RM e identificar a existência de tumores;



Objetivo: conceber modelos otimizados para a previsão



Porquê ? Tema de extrema importância visa melhorar a compreensão e o diagnóstico de doenças neurológicas

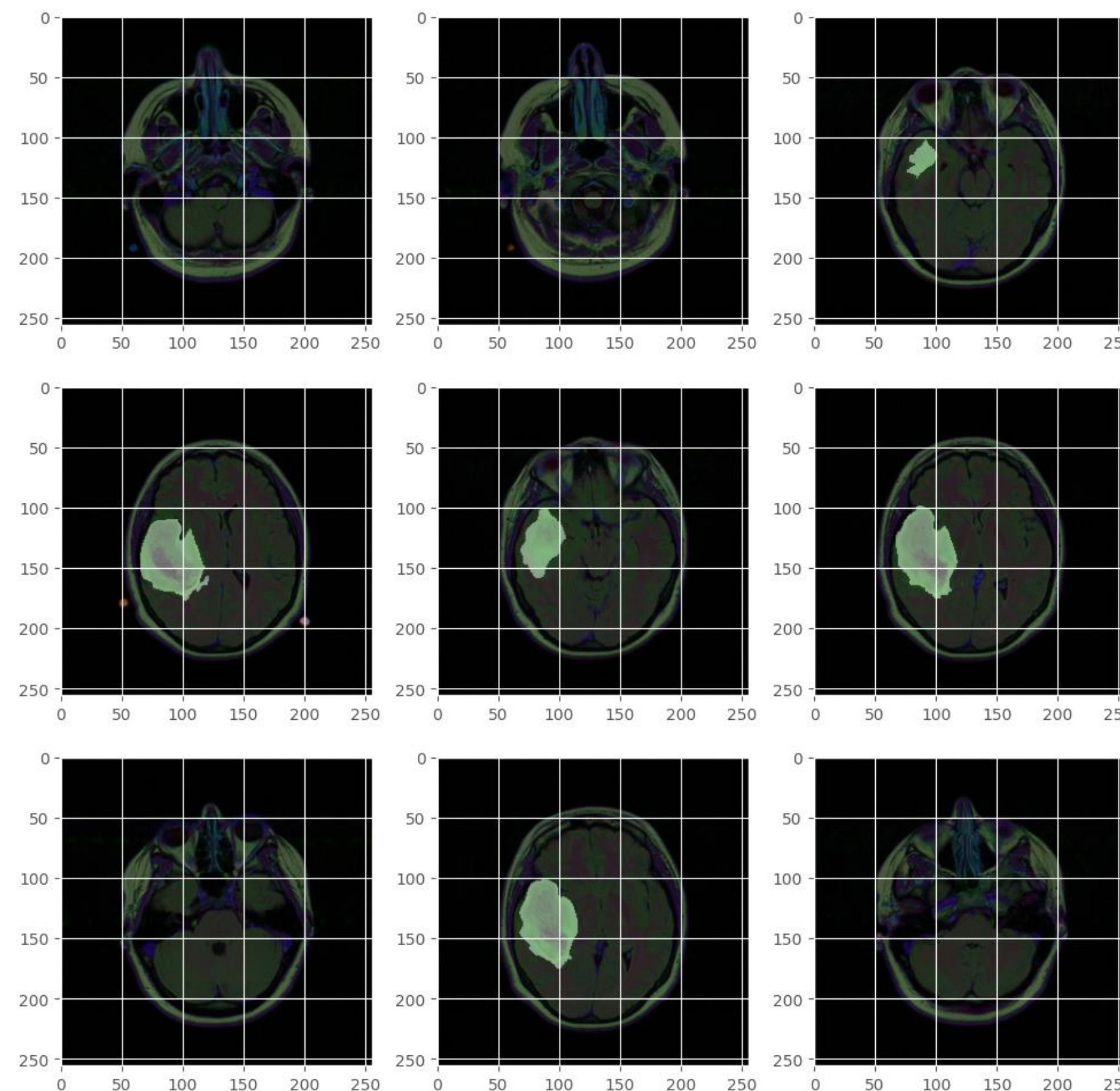




Dataset

Retirado do Kaggle: Brain MRI segmentation

- Imagens de Ressonância Magnética do Cérebro com máscaras manuais de segmentação de anomalias FLAIR (1.06 GB)
- As imagens foram obtidas do The Cancer Imaging Archive (TCIA)
- Correspondem a 110 doentes incluídos na coleção de gliomas de grau inferior do The Cancer Genome Atlas (TCGA) com dados de sequência de recuperação de inversão atenuada por fluido (FLAIR) e de agrupamento genómico disponíveis





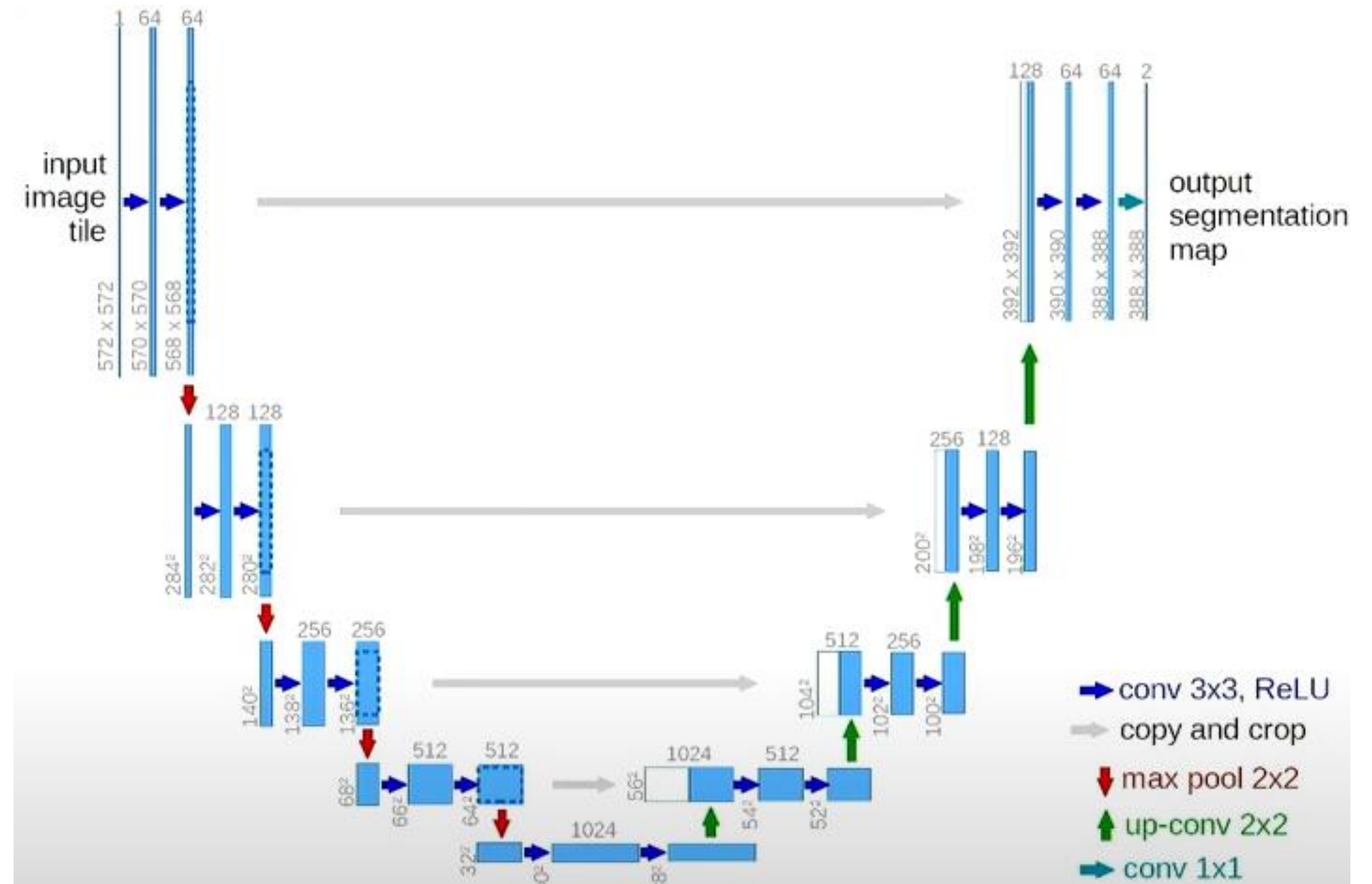
Arquitetura U-NET

Duas partes principais: **Encoder e Decoder**

Ao contrário de um **VAE**, não aprende a representação latente dos dados. Propósitos diferentes.

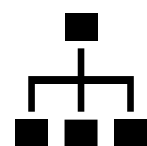
Encoder: Composto por várias camadas convolucionais, cada uma seguida por uma função de ativação não linear, como ReLU. À medida que as camadas convolucionais são empilhadas, a resolução espacial das características é reduzida por meio de operações de pooling.

Decoder: de modo inverso, cada camada no caminho de expansão é composta por uma camada de convolução transposta, que realiza a operação de upsampling para aumentar a resolução espacial, seguida por uma concatenação com as características correspondentes do caminho de contração.





Treino do Modelo



Assegurado por uma função que gera dados. É fornecido:

- Batch_size: número de amostras a serem geradas por iteração;
- Augmentation_dict: parâmetros do gerador do keras;
- Efetua-se a configuração de origem através do dataframe;
- Normalização do par de máscaras e imagens, com atribuição da classificação ($> 0.5 \rightarrow 1$; $< 0.5 \rightarrow 0$)

Configuração dos parâmetros da rede:

```
model = unet(input_size=(im_height, im_width, 3))

decay_rate = learning_rate / EPOCHS

opt = tf.keras.optimizers.legacy.Adam(learning_rate=learning_rate, beta_1=0.9, beta_2=0.999, epsilon=None, decay=decay_rate, amsgrad=False)

model.compile(optimizer=opt, loss=dice_coefficients_loss, metrics=["binary_accuracy", iou, dice_coefficients])

callbacks = [ModelCheckpoint('unet.hdf5', verbose=1, save_best_only=True)]

history = model.fit(train_gen,
                    steps_per_epoch=len(df_train) / BATCH_SIZE,
                    epochs=EPOCHS,
                    callbacks=callbacks,
                    validation_data=test_gen,
                    validation_steps=len(df_val) / BATCH_SIZE)
```



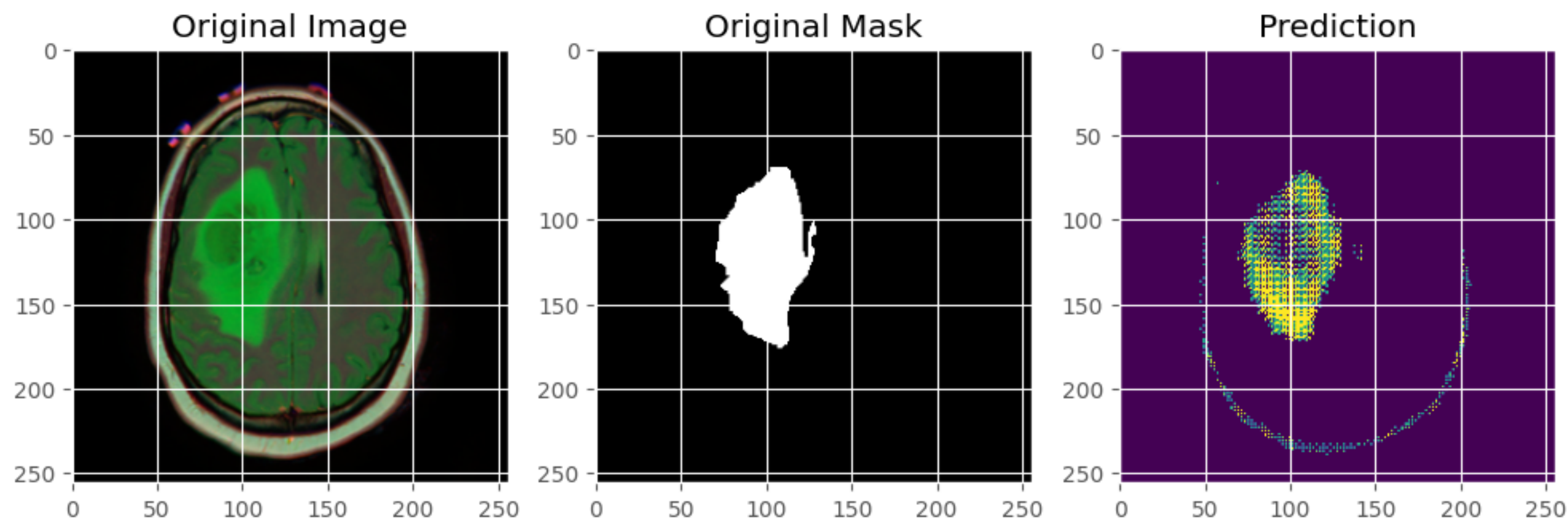

Resultados e Trabalho Futuro

Análise do processo de treino

- Completar a realização do processo de treino;
- Analisar a evolução de parâmetros;

Otimização

- Efetuar o processo de tuning do modelo com otimização dos hiperparâmetros;





Universidade do Minho
Aprendizagem Profunda

Segmentação **Do Cérebro Humano:** Arquitetura U-Net

Daniel Xavier PG50310

Diogo Rebelo PG50327