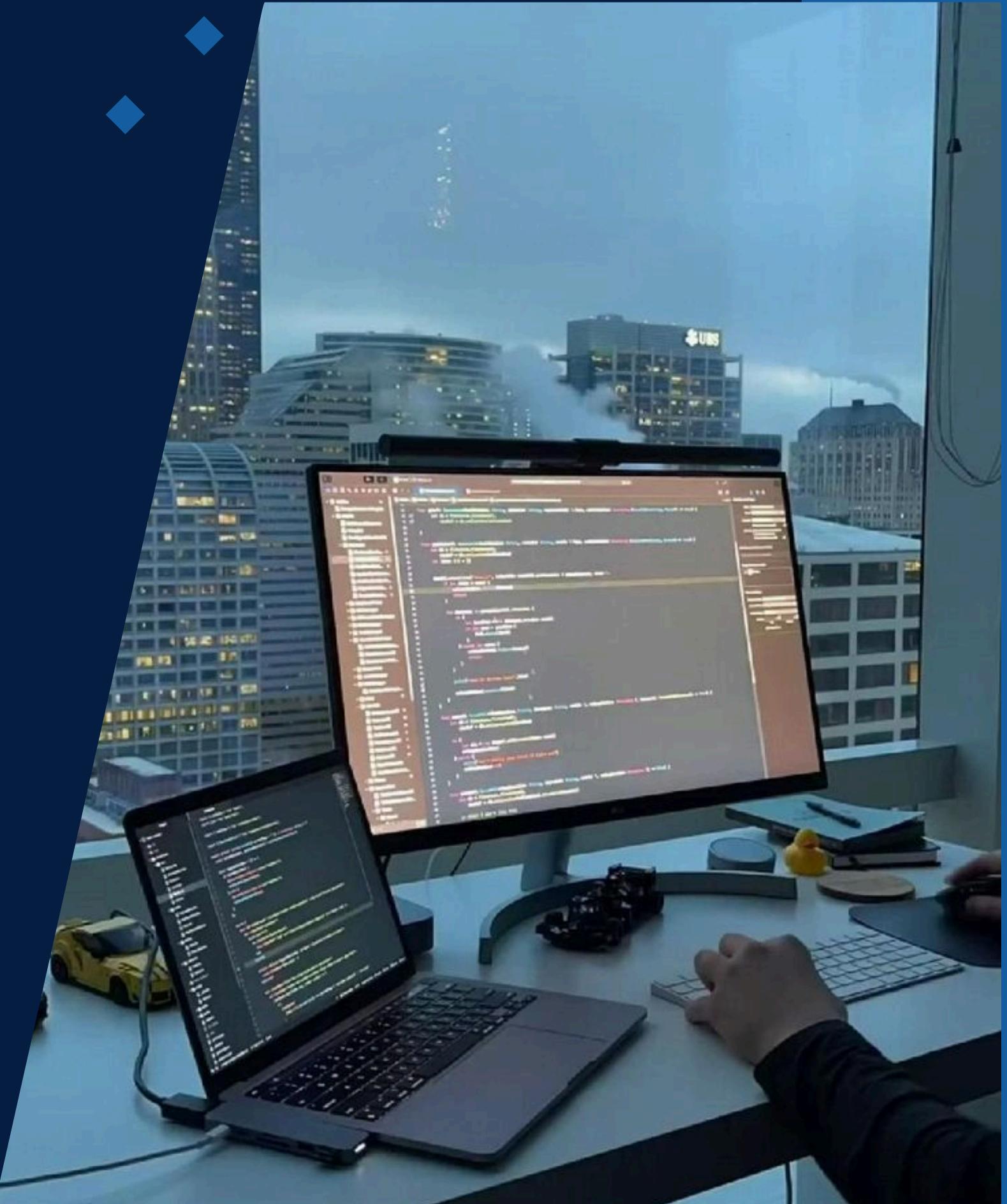


COMPUTER VISION PROJECT

Presented by: Acyr Eduardo Marconato



CAPITULOS

01

Problema/Dataset

02

Criação de máscaras

03

Arquitetura

04

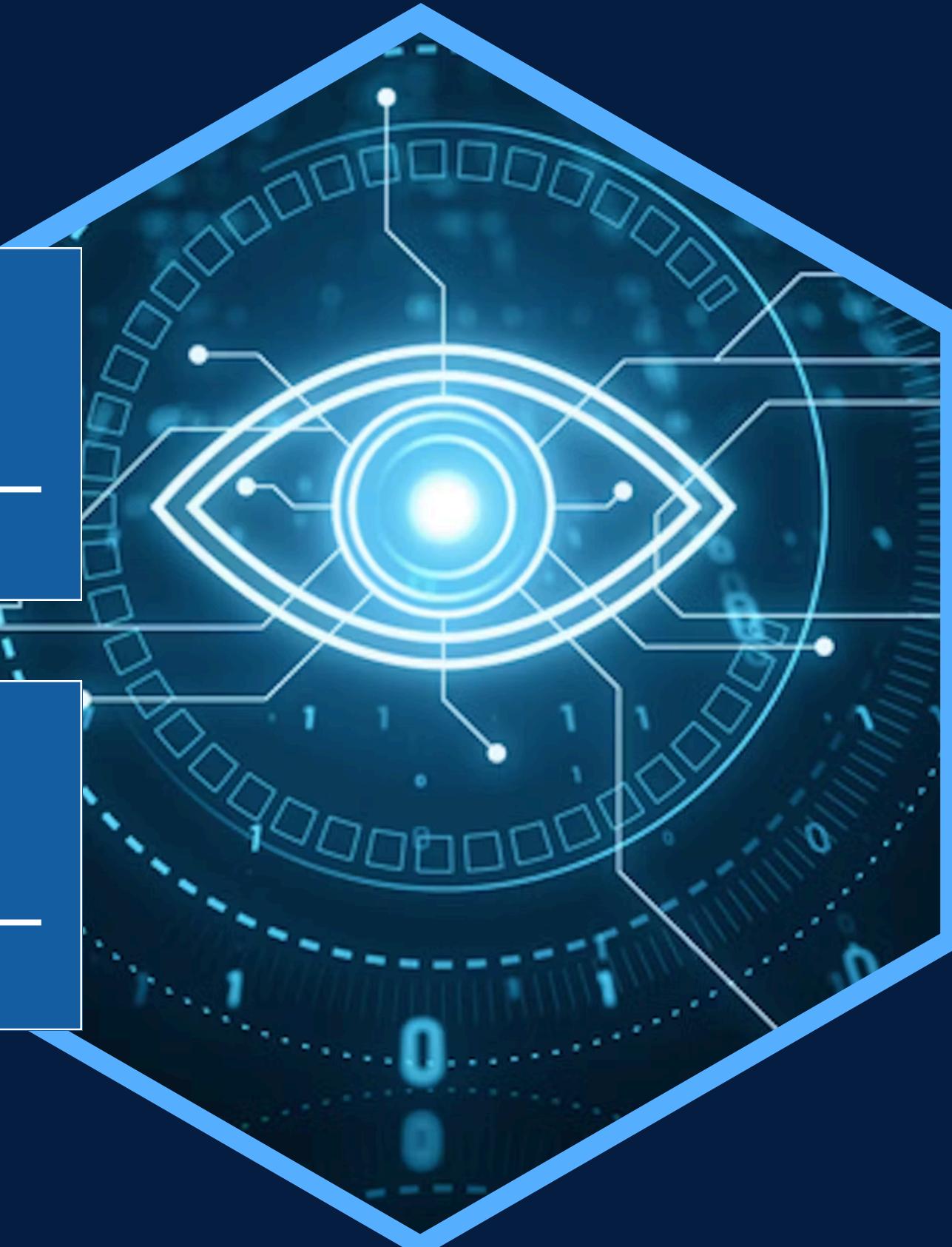
Treinamento

05

Resultados

06

Conclusão





PROBLEMA

Criar um segmentador de imagens para definir quais tipos de alimentos estão em um prato

DATASET

O dataset fornecido pelo professor possui dois tipos de imagens, uma com apenas um alimento no prato, e outra com varios, em ambas multiplos backgrounds foram utilizados



Imagen de 1 alimento

2011 imagens de pratos com alimentos individuais, com um total de 16 tipos de alimentos distintos



Imagenes de vários alimentos

289 imagens com cerca de 30 pratos distintos



Backgrounds

cerca de 5 backgrounds (mesas) são usados pelas imagens, com variedade de iluminação e objetos próximos



MAIORES DIFICULDADES

Similaridade entre classes

Multiplas Classes são extremamente similares, ao ponto de ser difícil identificar elas mesmo

Ambiguidade

Em outros casos, molhos podem dificultar a classificação do que realmente é considerado comida;

CRIAÇÃO DE MASCARAS

Com base nos valores hsv, uma máscara para o prato é adicionada

Com base no nome do alimento na imagem, uma nova máscara é criada usando os valores hsv para um dado alimento

Após a criação da máscara, duas cópias são criadas com alterações e adição de ruídos e imperfeições a imagem original, triplicando o dataset inicial

Resize

Todas as imagens são redimensionadas para as proporções 256x256

Pratos

Buracos

Alimento

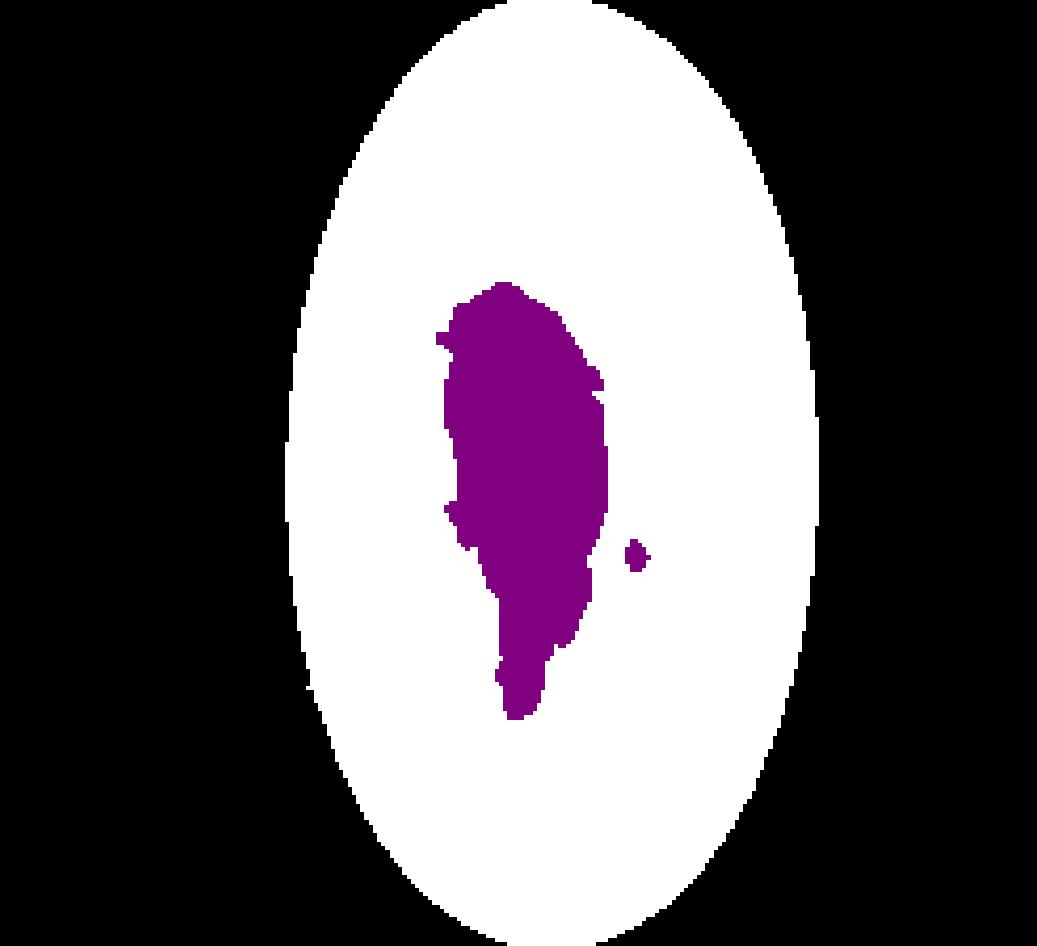
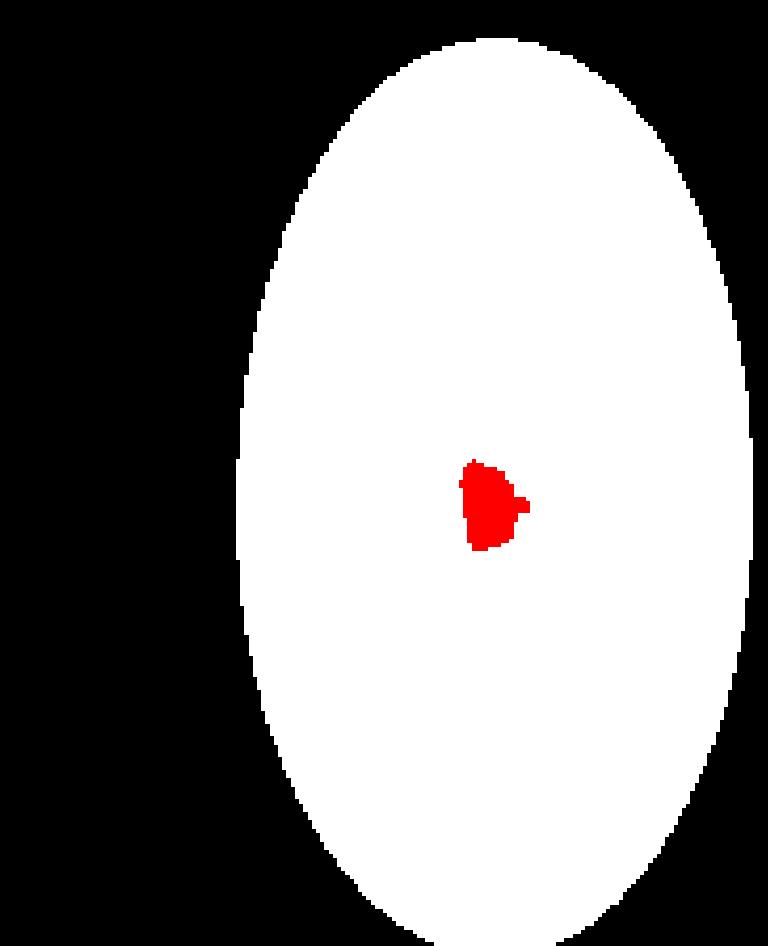
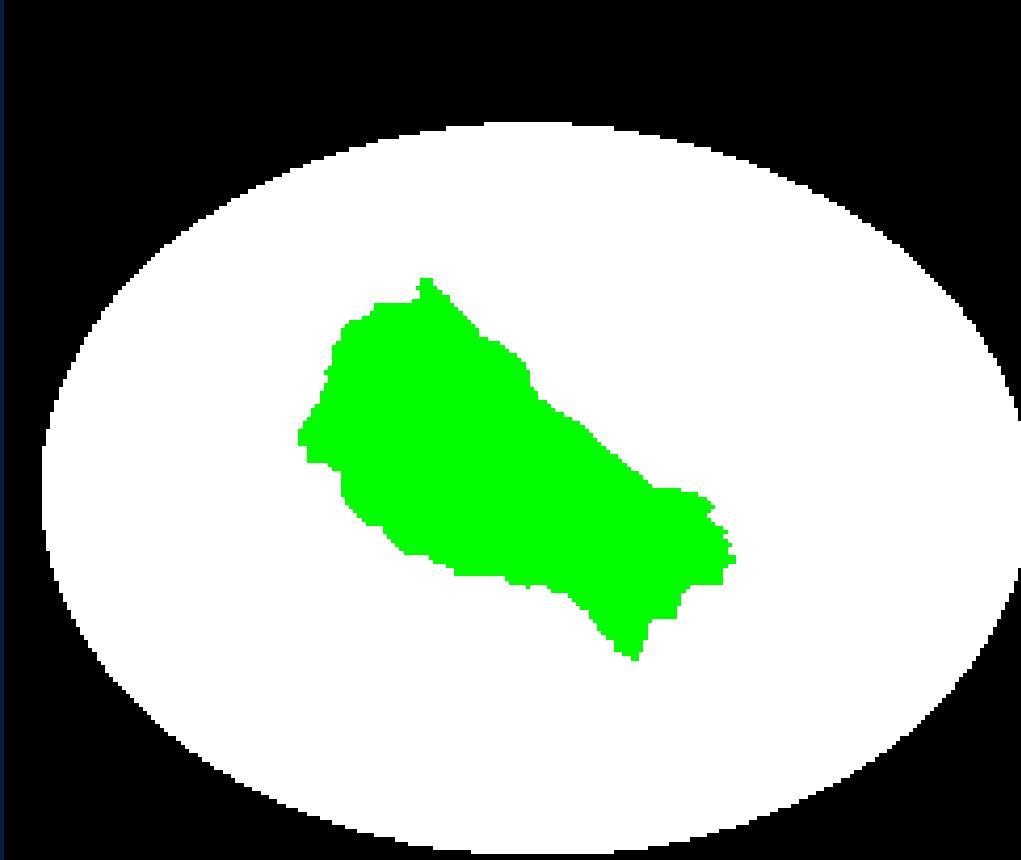
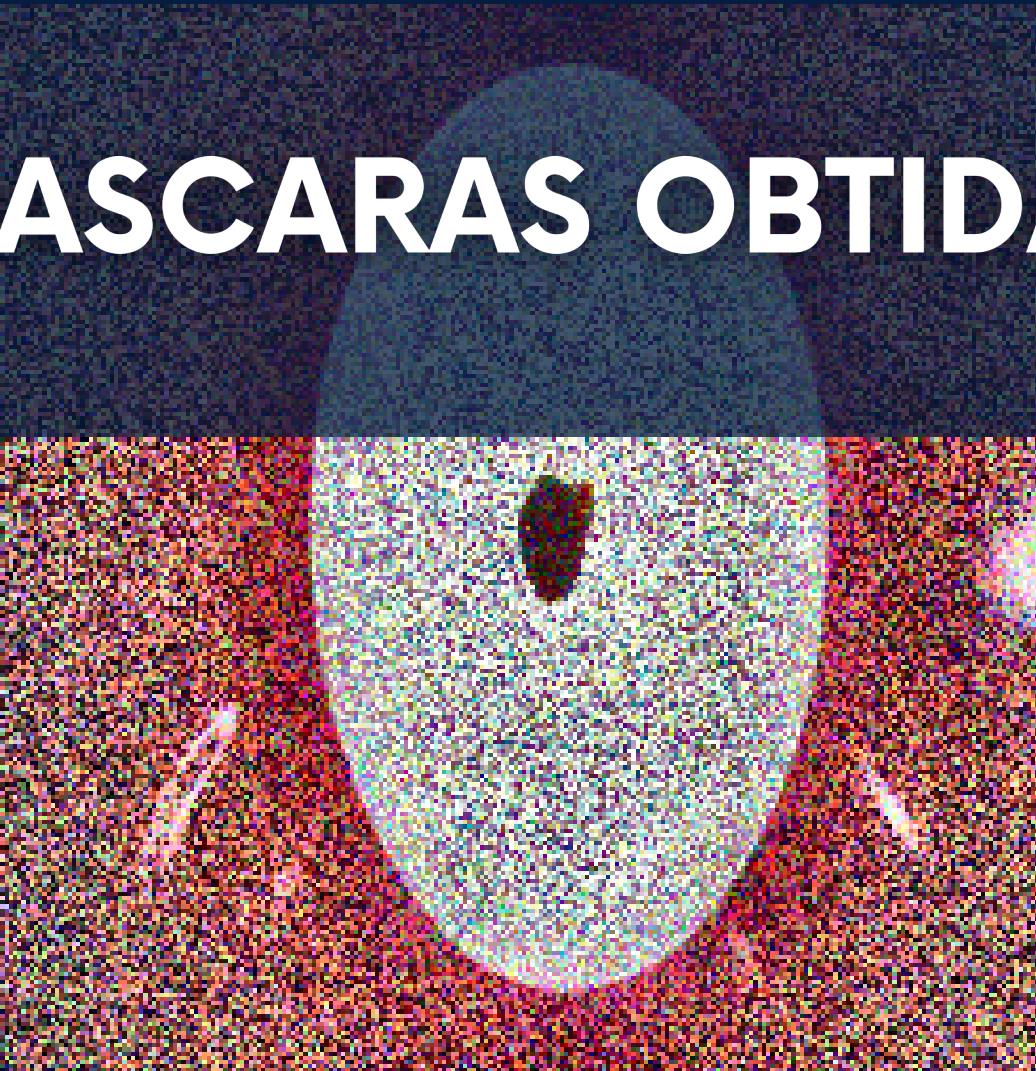
Buracos

Augmentation

Buracos, ou manchas de uma classe em outra são preenchidos, usando um limite de tamanho para não danificar a forma principal

Qualquer pequeno buraco dentro da máscara de alimento é preenchido

MASCARAS OBTIDAS





ARQUITETURA

Qual a melhor arquitetura de segmentação para
este dataset? E qual backbone?

ESTRUTURA DA UNET

A U-Net é uma rede voltada para segmentação de imagens médicas (inicialmente), combinando um encoder para capturar o contexto e um decoder para reconstruir os detalhes. Também está presente em muitos algoritmos de geração de imagem.

Encoder (Contrátil)

- Sequência de blocos **Conv2D + ReLU**
- **MaxPooling** reduz progressivamente a resolução
- Captura características contextuais globais, ou seja, **qual classe está naquela região**

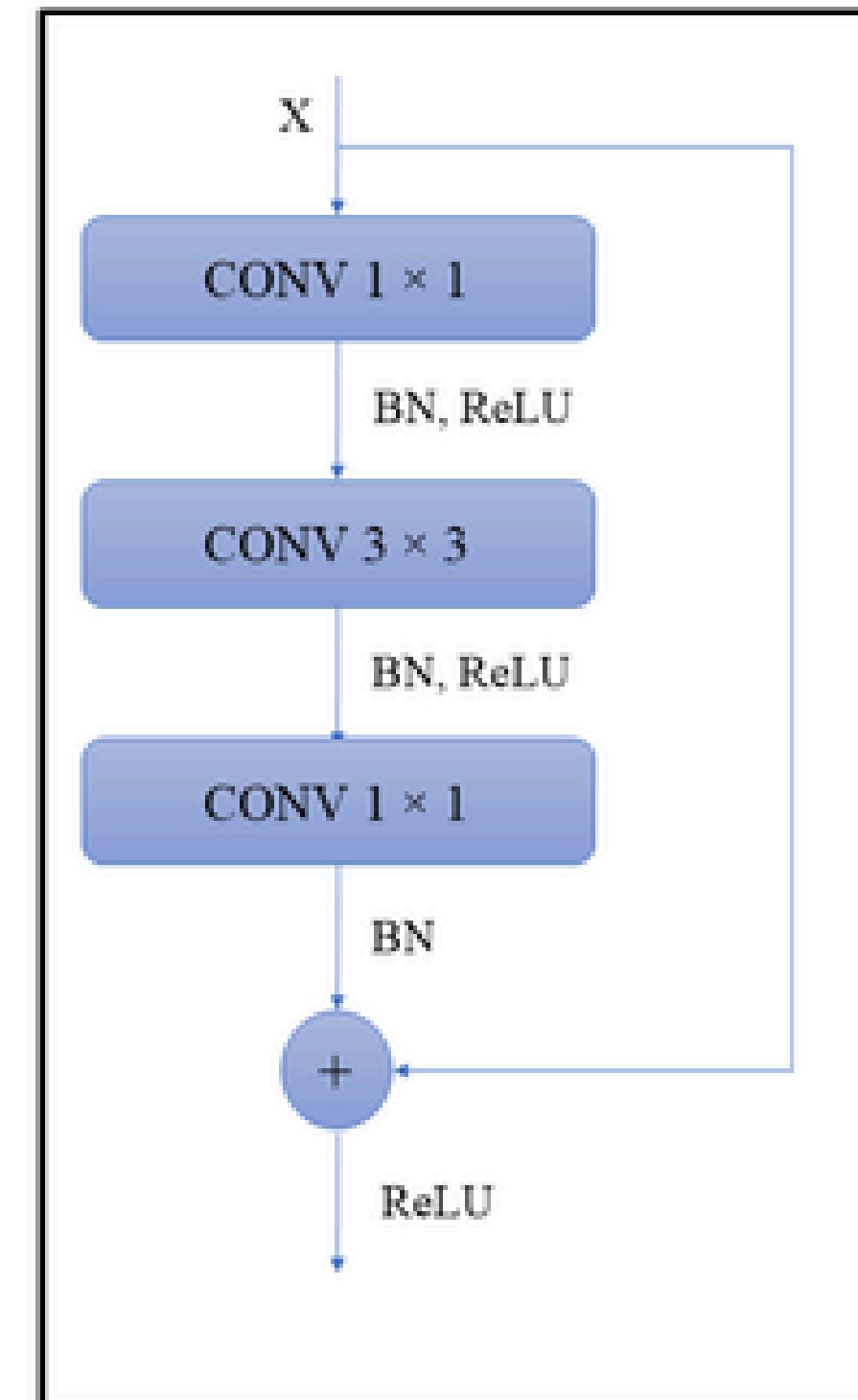
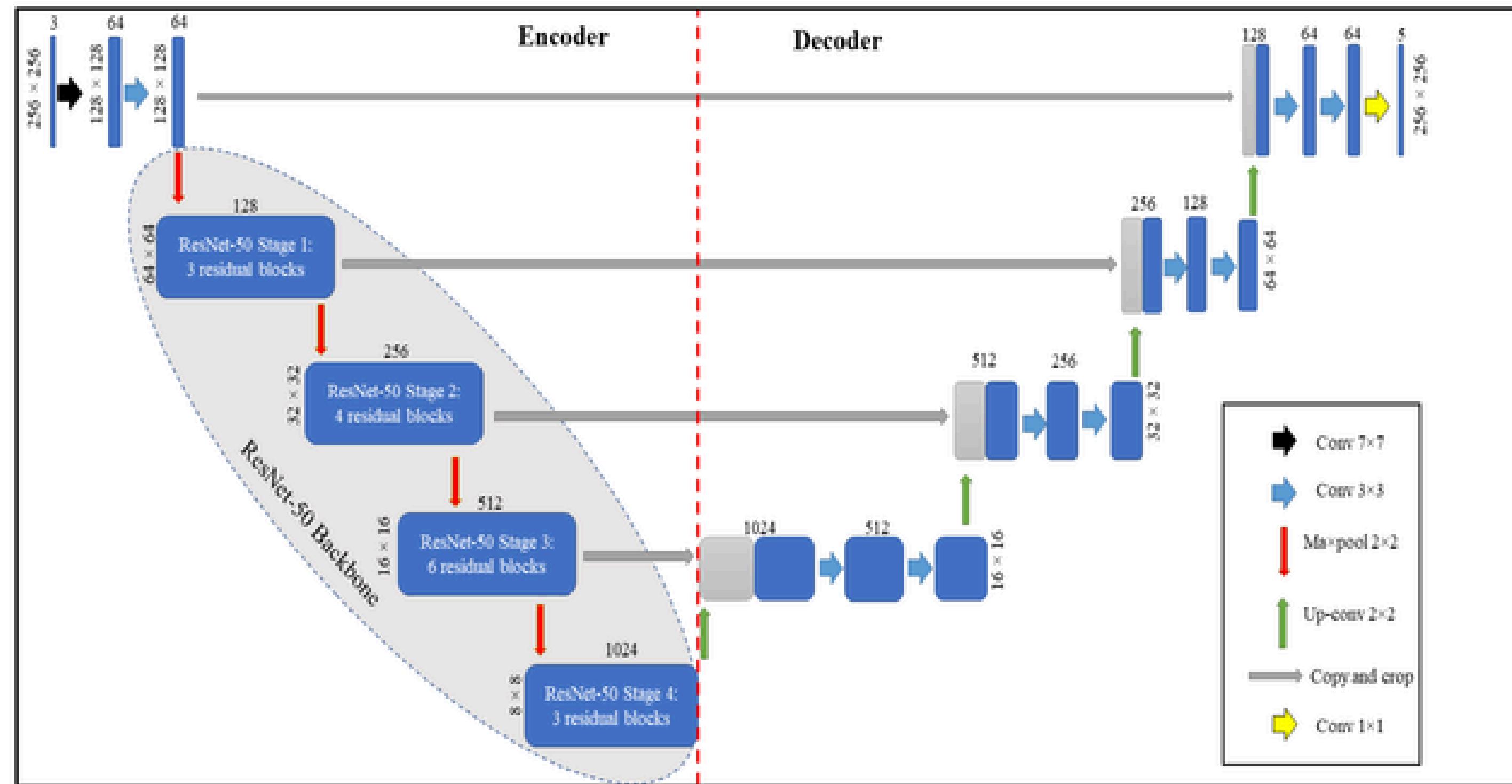
Decoder (Expansivo)

- Blocos **Conv2D + ReLU** para refinar e reconstruir os detalhes espaciais
- **UpSampling** ou **Transposed Convolution** para aumentar a resolução
- Recupera a forma original com a segmentação pixel a pixel, ou seja, **quais pixels pertencem a classe**

Concatenação

- Ligações entre camadas do encoder e decoder no mesmo nível de profundidade
- Permite o reuso de características espaciais finas
- Ajuda na precisão dos contornos e na localização exata das estruturas
- **Combina o "contexto global"** (decoder) **com o "detalhe local"** (encoder)

U-NET MODEL COM RESNET BACKBONE



ARQUITETURA COMPLETA

Arquitetura do Modelo:

- Base: U-Net com codificador ResNet50
- Encoder pré-treinado no ImageNet (ResNet50)
- Extração de features em diferentes escalas

Backbone: ResNet50 (Encoder):

- Entradas: imagens RGB ($256 \times 256 \times 3$)
- Camadas extraídas:
 - conv1_relu (128×128)
 - conv2_block3_out (64×64)
 - conv3_block4_out (32×32)
 - conv4_block6_out (16×16)
 - conv5_block3_out (8×8 , bottleneck)

Decoder com Mecanismos de Atenção:

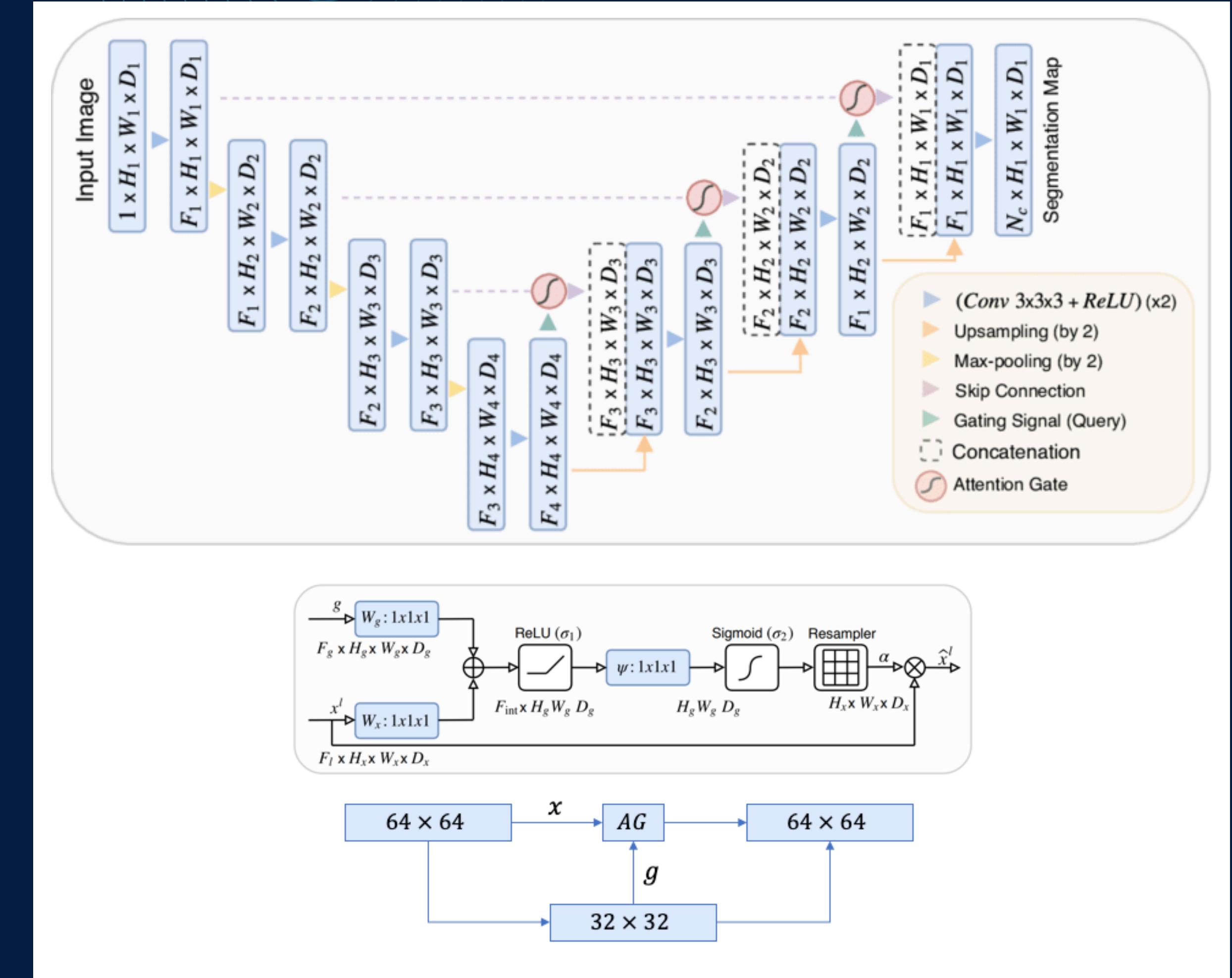
- Upsampling + concatenação com skip connections
- Attention Gate: foca nas regiões relevantes antes da concatenação
- CBAM (Channel & Spatial Attention): recalibra as features no encoder para destacar o que e onde é mais importante

Saída Final:

- Conv2D 1×1 com softmax
- Gera mapa de probabilidade por classe (por pixel)

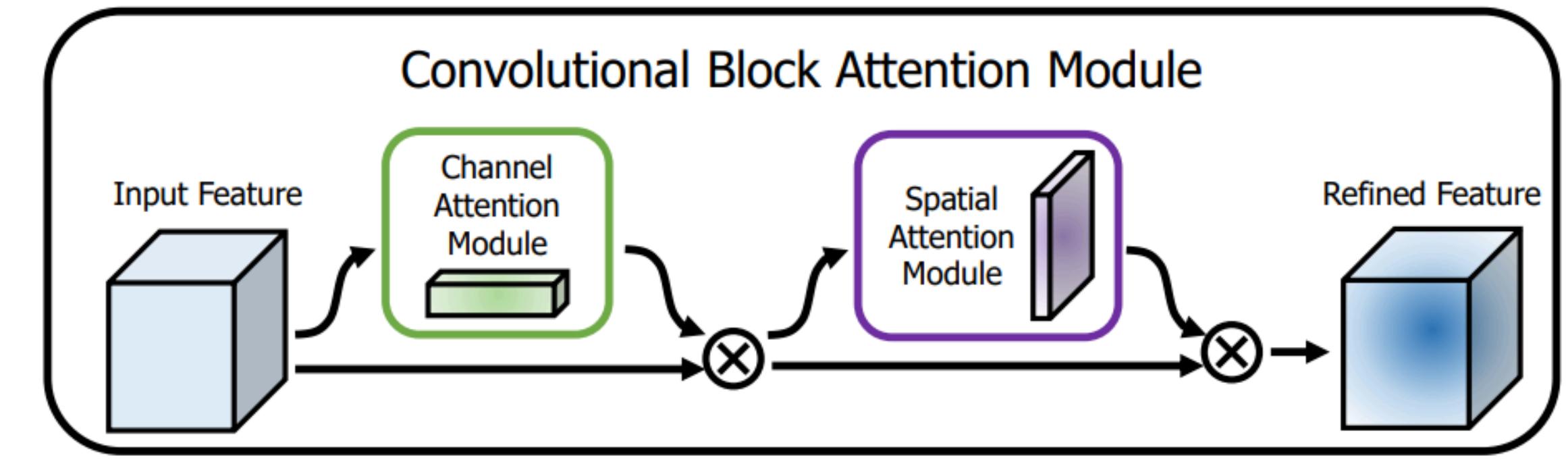
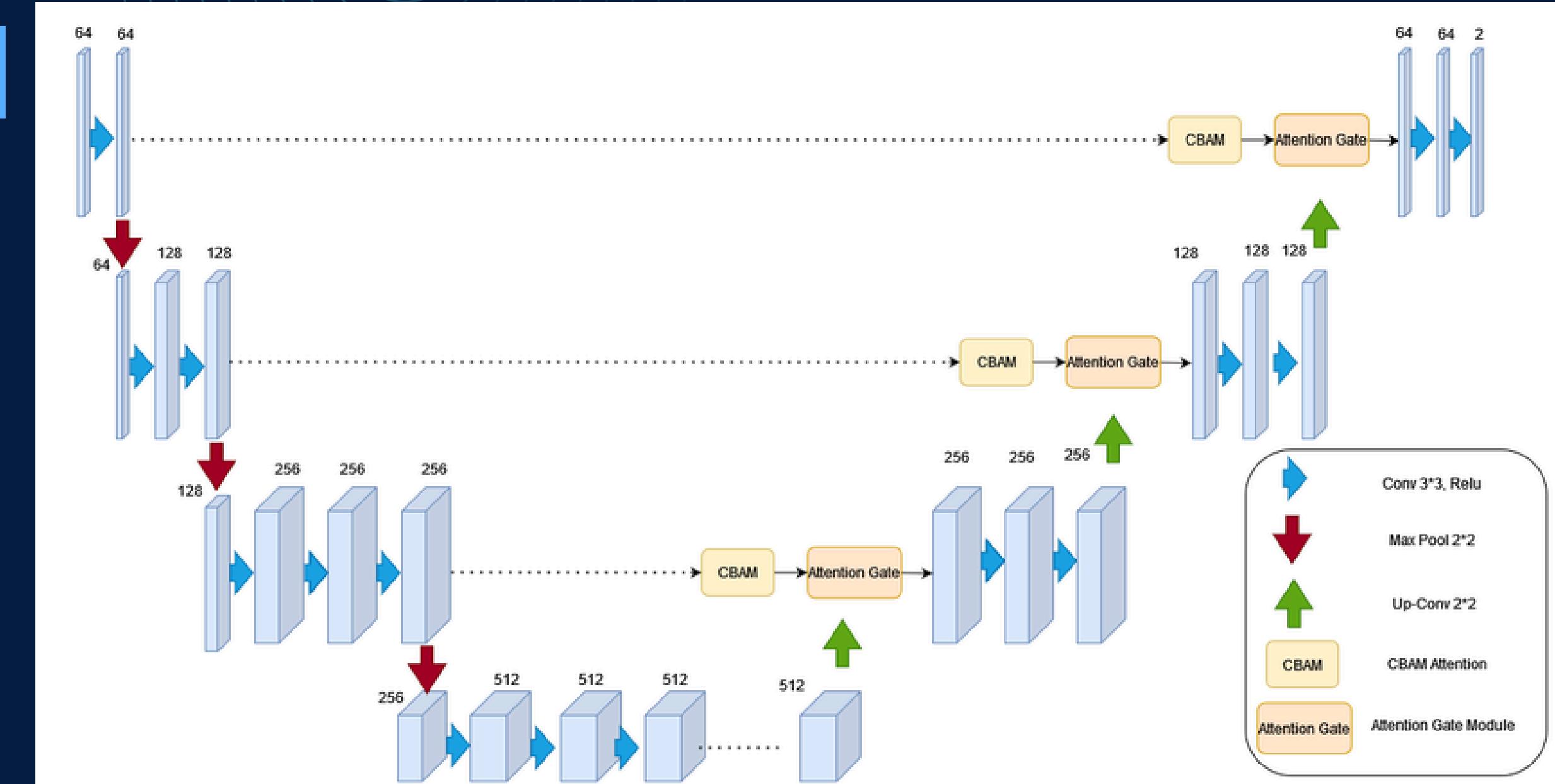
Attention gate

- Melhora a atenção do modelo para as áreas de interesse;
- Combina os dados espaciais e os dados de features em uma função ReLU para definir locais de maior interesse

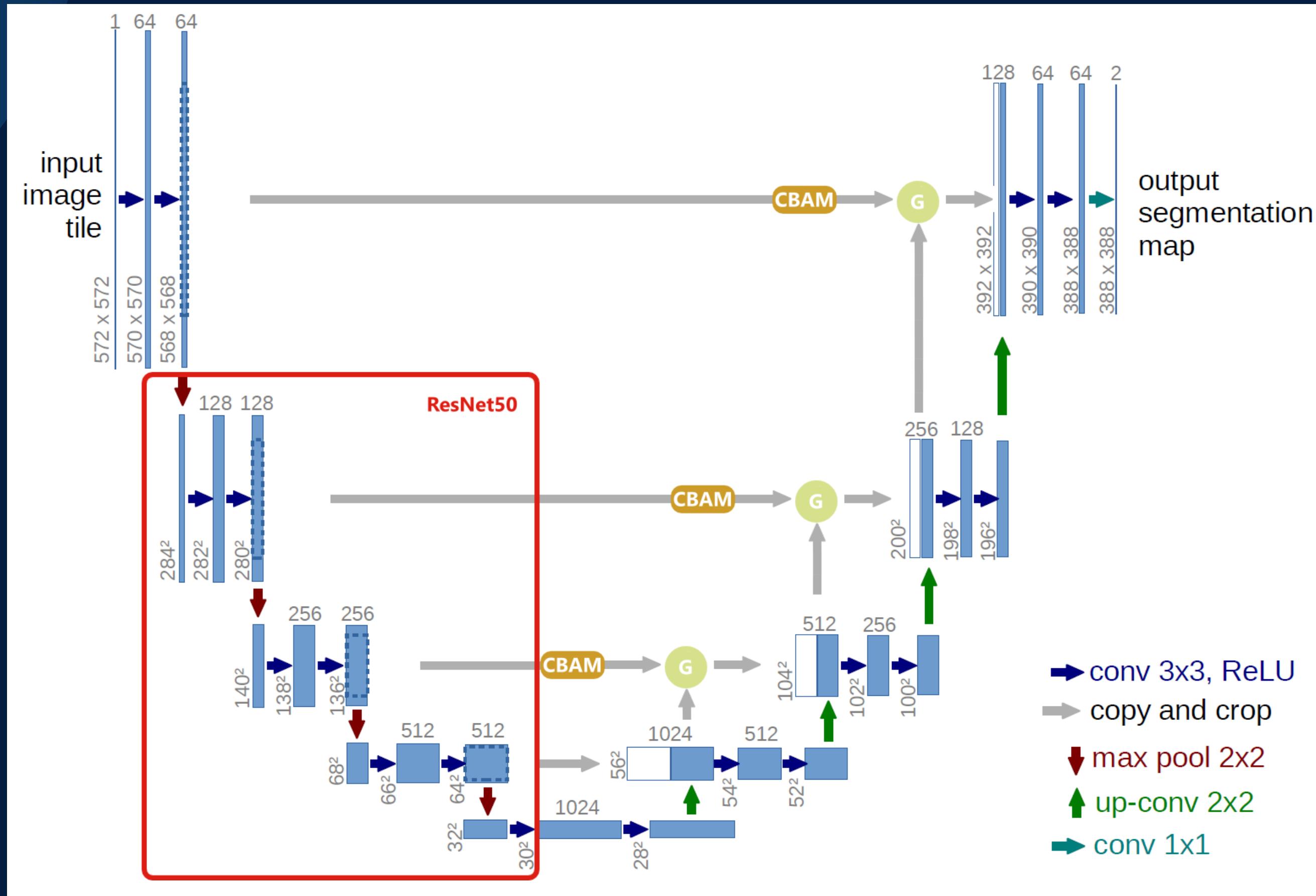


Convolutional Block Attention Module (CBAM)

- Aplica um modulo de atenção de canais e um modulo de atenção espacial para refinar as features de interesse



ARQUITETURA



TREINAMENTO

Pré-Treinamento com 3 Classes:

- Classes: fundo, prato e comida
- Treinamento completo da U-Net com ResNet50 usando imagens anotadas com rótulos simples
- Objetivo: ensinar a estrutura básica da imagem e separar regiões principais

Treinamento com Encoder Congelado:

- Carregamento dos pesos do modelo 3 classes
- Camadas do encoder congeladas
- Treinamento por 3 épocas apenas no decoder
- Ajuste inicial para nova tarefa com 18 classes

Fine-Tuning com Encoder Descongelado:

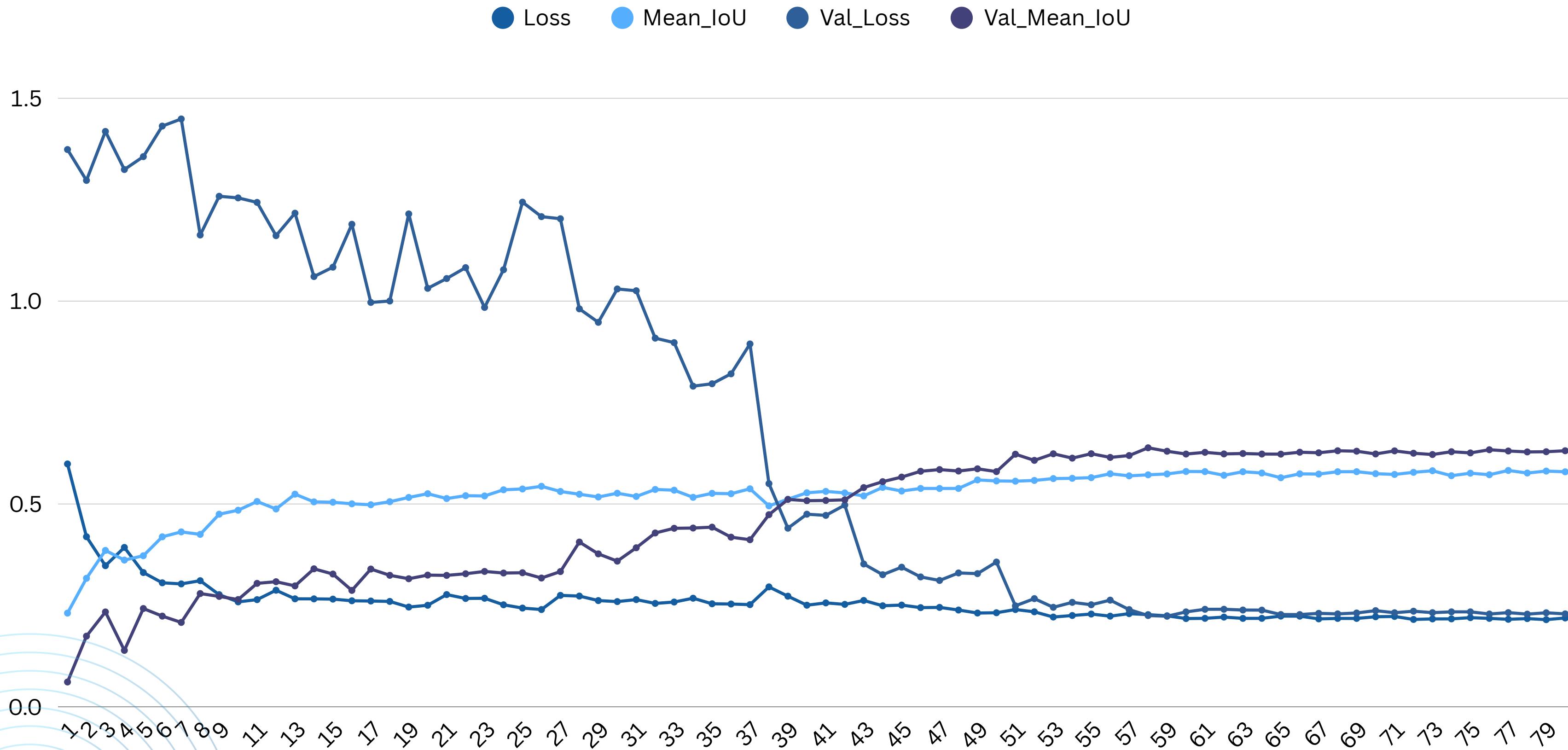
- Descongelamento de todas as camadas
- Recompilação com Adam($2e-3$) e retomada do treinamento (initial_epoch=3)
- Aprendizado completo da rede para o novo conjunto de 18 classes

Função de Perda (Loss Function):

- 0.75* Sparse cathegorical crossentropy com pesos +
- 0.15* Dice
- 0.15* focal_loss



Dados de treinamento

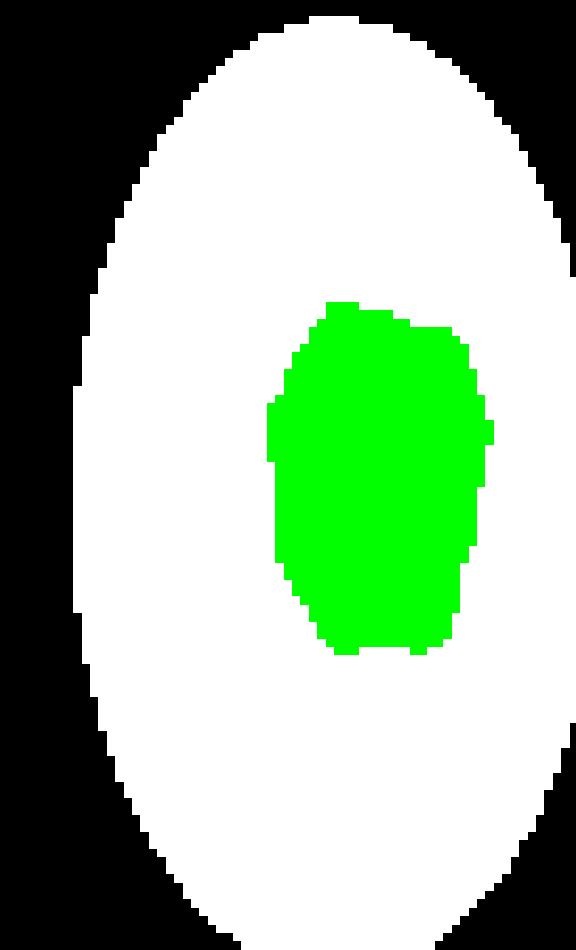
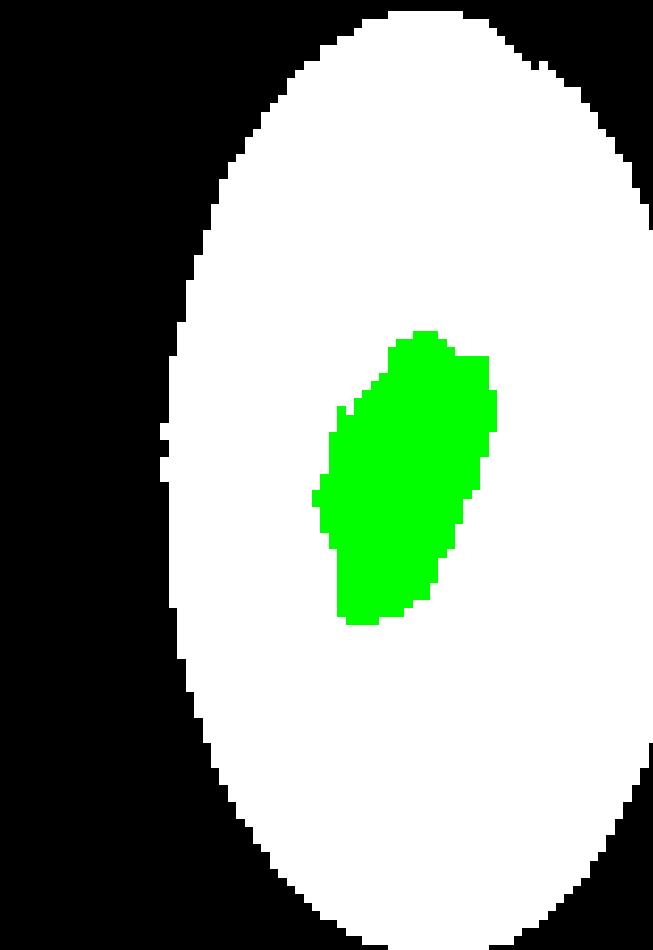
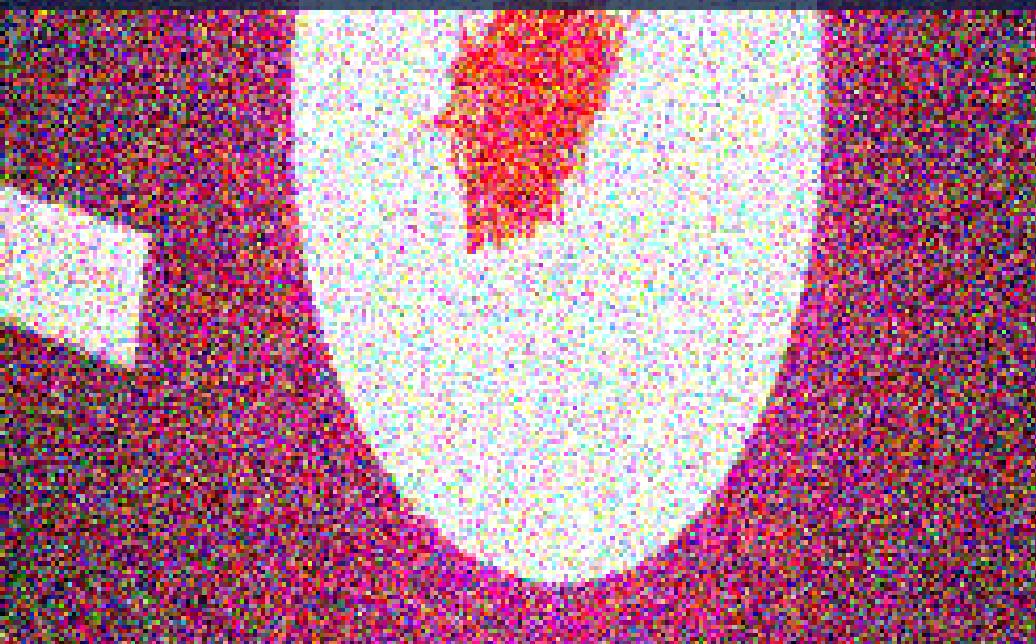




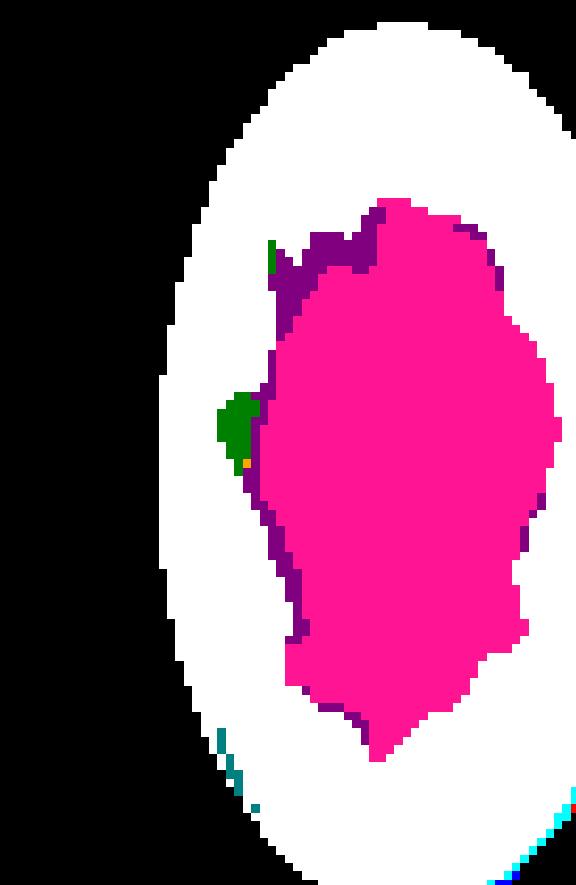
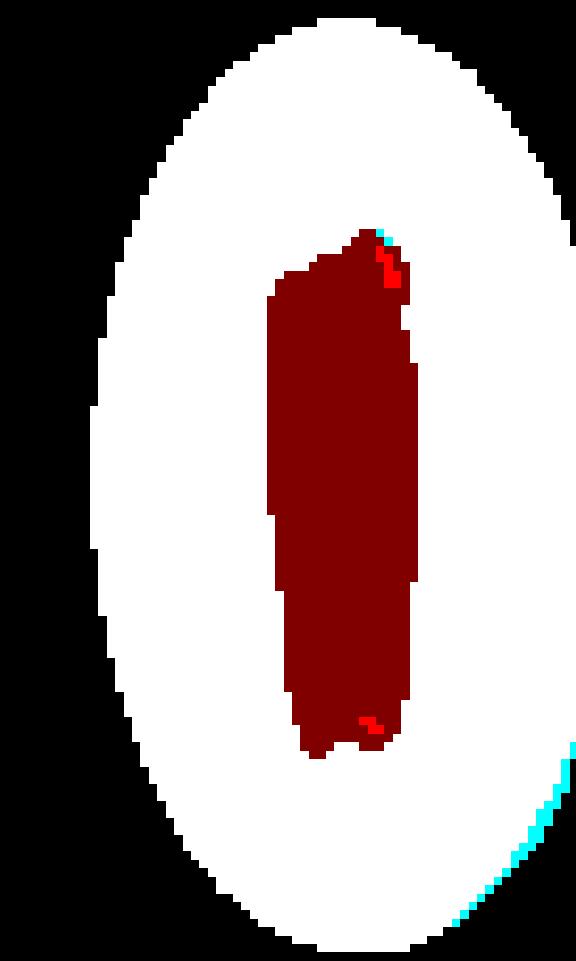
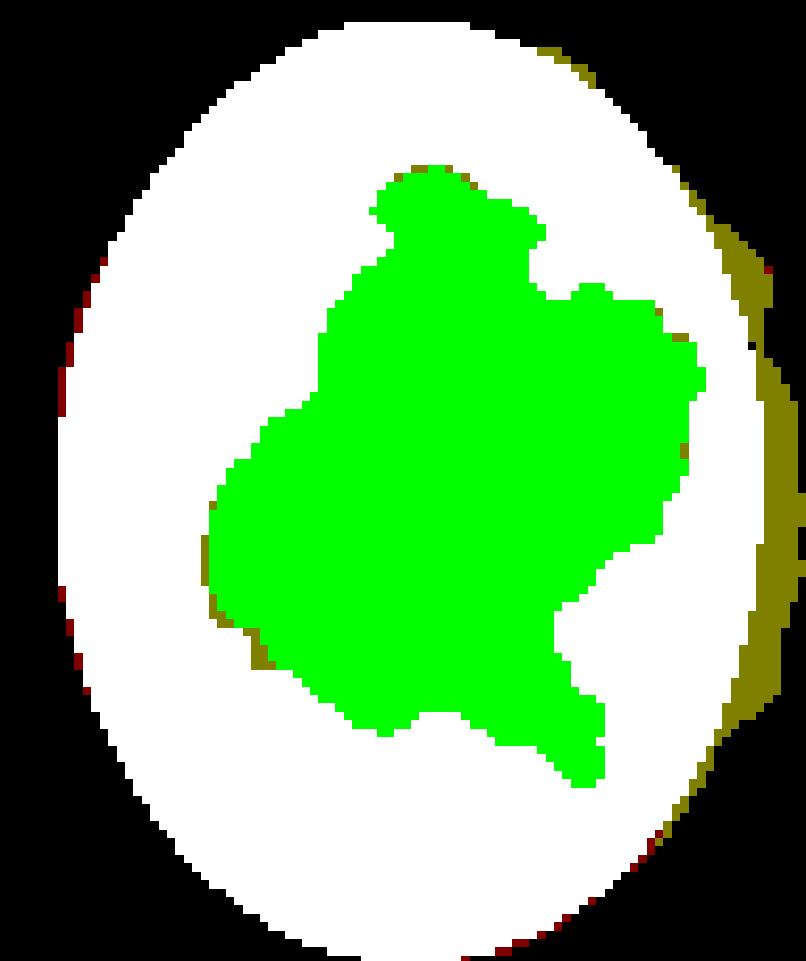
RESULTADOS

Todo esse trabalho gerou algum resultado bom? O que poderia ser melhorado?

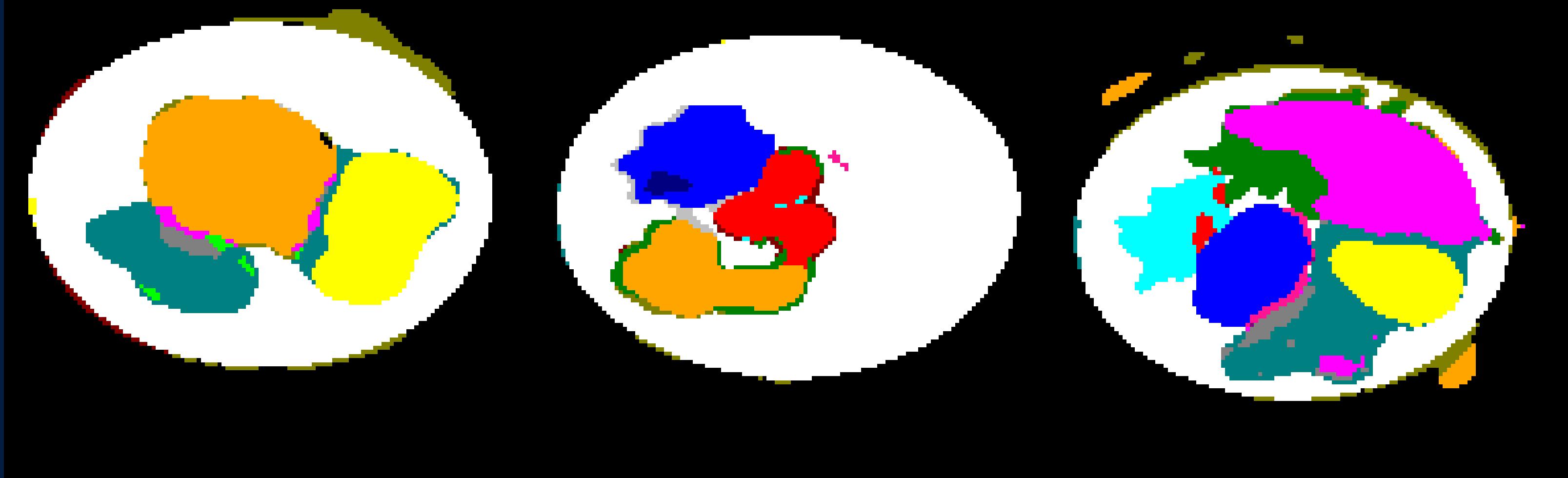
RESULTADO TREINAMENTO 1



RESULTADO TREINAMENTO 2

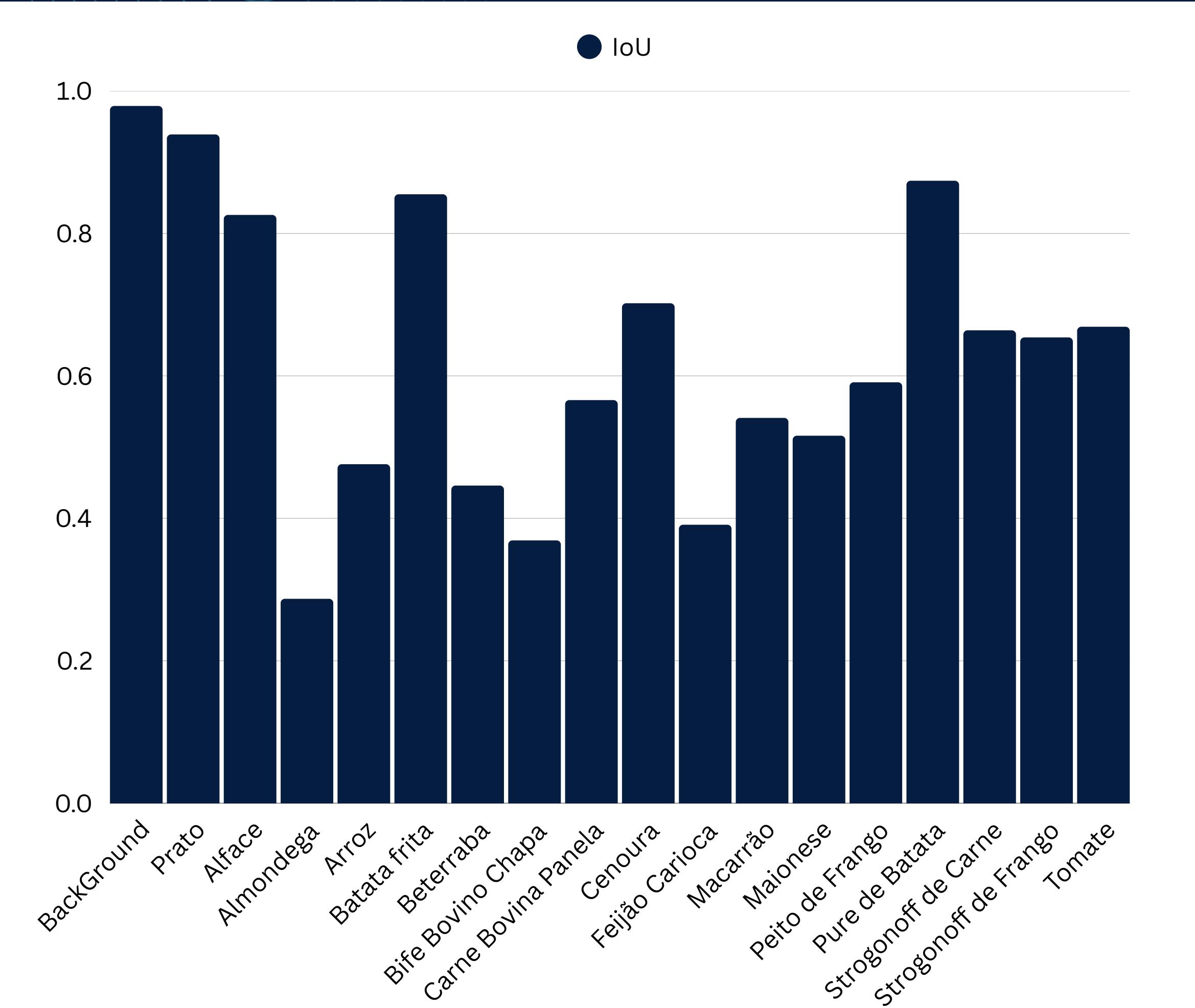


RESULTADO TREINAMENTO 2



Análise do IoU

- Em média, cerca de 50% de precisão;
- Piores classes:
 - Almondegas;
 - Bife Bovino Chapa;
 - Feijão Carioca;
- Melhores Classes:
 - Alface;
 - Batata Frita;
 - Purê de Batata



Obrigado pela atenção



42 99808-7046



acyrmarconato@alunos.utfpt.edu.br

