Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba
DAELN-Departamento de Eletrônica

Visão computacional - Trabalho final

Acyr Eduardo Marconato*

1 DATASET

O conjunto de dados utilizado neste trabalho consiste em imagens RGB de alimentos dispostos sobre pratos, com o objetivo de segmentar semanticamente cada classe de alimento presente em cada imagem. As imagens possuem resolução de 256×256 pixels e foram organizadas em dois subconjuntos principais: **Training** e **Validation**, cada um contendo suas respectivas subpastas de *Images* e *Masks*.

O dataset apresenta um alto grau de desbalanceamento entre classes, sendo que algumas classes de alimentos aparecem em apenas uma fração reduzida do total de imagens. Além disso, classes como o fundo (*background*) e o prato (*plate*) estão presentes em praticamente todas as amostras, representando um desafio adicional no treinamento, uma vez que o modelo pode tender a priorizar essas classes mais frequentes.

Ao todo, o conjunto de dados contempla 18 classes, codificadas por inteiros de 0 a 17, onde cada valor representa uma categoria distinta de alimento ou elemento da imagem (como fundo ou prato). Essas classes foram posteriormente associadas a cores específicas para fins de visualização.

Para garantir maior variabilidade durante o treinamento, foram aplicadas técnicas de aumento de dados (*data augmentation*) como rotações, inversões horizontais e modificações de brilho e contraste. Porém está técnica foi posteriormente abandonada por remover muitas features essenciais para o reconhecimento das classes.

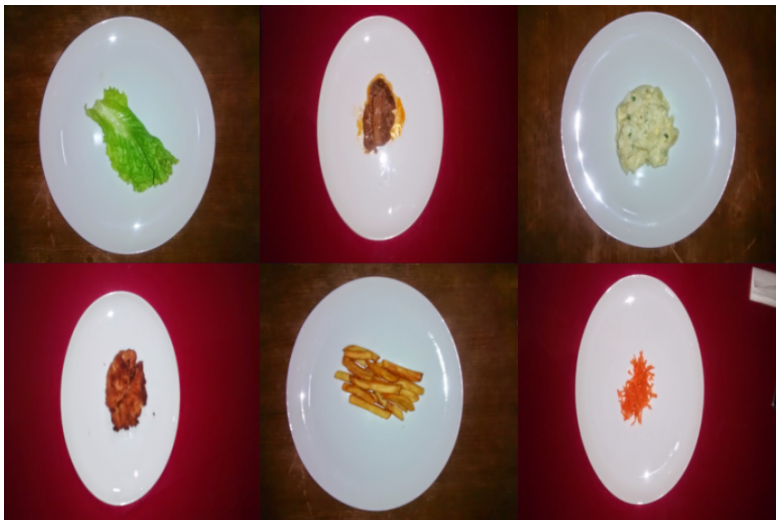

Figura 1 – exemplo de imagens no dataset.

* Universidade Tecnológica Federal do Paraná, Cidade, Paraná, Brasil. acyrmarconato@gmail.com.

2 CRIAÇÃO DE MÁSCARAS

A criação das máscaras foi realizada de forma semiautomática, a partir de imagens rotuladas com base no nome do alimento presente no arquivo. Cada imagem do dataset foi associada a uma classe única de alimento ou a múltiplos alimentos, dependendo da sua categoria.

Inicialmente, máscaras foram geradas utilizando limiares em espaços de cor HSV, com base em faixas características de cor para cada classe de alimento. Para a classe correspondente ao prato, foi utilizada uma máscara complementar que identificava regiões predominantemente brancas ou claras, comuns em superfícies de pratos.

Pequenos buracos ou falhas dentro das máscaras de alimento foram preenchidos por operações morfológicas, como fechamento e preenchimento de regiões internas, desde que não alterassem significativamente a forma geral do objeto.

Para aumentar a robustez do modelo, cada imagem original teve duas cópias sintéticas geradas via técnicas de *data augmentation*, incluindo rotações, ruído, alterações de brilho e contraste, e inversões horizontais. Isso triplicou o tamanho do dataset inicial e introduziu imperfeições visuais que simulam variações do mundo real, como iluminação, ângulo de câmera e oclusão parcial.

Além disso, foi utilizado um processo de correção para evitar que regiões pequenas e isoladas de uma classe fossem confundidas com ruído. Isso foi feito aplicando uma filtragem baseada em área mínima de componentes conectados, removendo ou unificando manchas pequenas com a classe predominante na região.

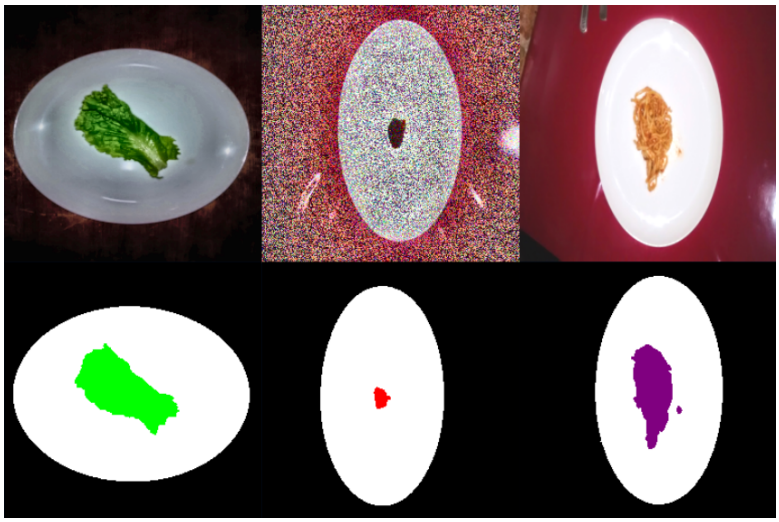

Figura 2 – Resultado da geração de máscaras.

3 ARQUITETURA

A arquitetura utilizada para a tarefa de segmentação semântica foi baseada no modelo *U-Net*, amplamente adotado em tarefas de segmentação por sua estrutura simétrica e capacidade de preservar informações espaciais durante o processo de codificação e decodificação.

O encoder do modelo foi construído a partir da **EfficientNetB0**, uma rede convolucional pré-treinada no ImageNet, utilizada como extratora de características. Camadas intermediárias da EfficientNet foram utilizadas como conexões de atalho (*skip connections*) para transferir informações de baixa e média resolução para o decodificador.

O decodificador consistiu em uma série de blocos de upsampling seguidos por convoluções e operações de normalização e ativação. Cada bloco de upsampling foi conectado a sua respectiva *skip connection*, e antes da concatenação foi aplicada uma **camada de atenção** baseada em SE (Squeeze-and-Excitation) ou CBAM (Convolutional Block Attention Module), com o objetivo de destacar os canais mais relevantes das features vindas do encoder.

A saída do modelo foi uma camada convolucional com 1×1 filtro, gerando um mapa com 18 canais, um para cada classe. A ativação final utilizada foi a **softmax**, adequada para problemas de segmentação com múltiplas classes mutuamente exclusivas.

Além da arquitetura base, foram aplicadas camadas de **SpatialDropout2D** nos blocos do decodificador para melhorar a generalização e evitar o overfitting em regiões específicas.

Por fim, o modelo foi compilado com funções de perda adaptadas ao cenário de classes desbalanceadas, como a **Focal Loss** combinada com a **Dice Loss** e uma versão ponderada da entropia cruzada esparsa, como detalhado na seção de treinamento.

Tabela 1 – Resumo dos principais blocos da arquitetura

Bloco	Descrição	Tamanho da saída
Input	Imagem RGB de entrada	$256 \times 256 \times 3$
Encoder 1	`conv1_relu` (ResNet50)	$128 \times 128 \times 64$
Encoder 2	`conv2_block3_out` (ResNet50)	$64 \times 64 \times 256$
Encoder 3	`conv3_block4_out` (ResNet50)	$32 \times 32 \times 512$
Encoder 4	`conv4_block6_out` (ResNet50)	$16 \times 16 \times 1024$
Bottleneck	`conv5_block3_out` (ResNet50)	$8 \times 8 \times 2048$
Decoder 1	Upsample + Attention + Conv + CBAM	$16 \times 16 \times 256$
Decoder 2	Upsample + Attention + Conv + CBAM	$32 \times 32 \times 128$
Decoder 3	Upsample + Attention + Conv + CBAM	$64 \times 64 \times 64$
Decoder 4	Upsample + Attention + Conv + CBAM	$128 \times 128 \times 32$
Output	Upsample + Conv 1×1 + Softmax	$256 \times 256 \times 18$

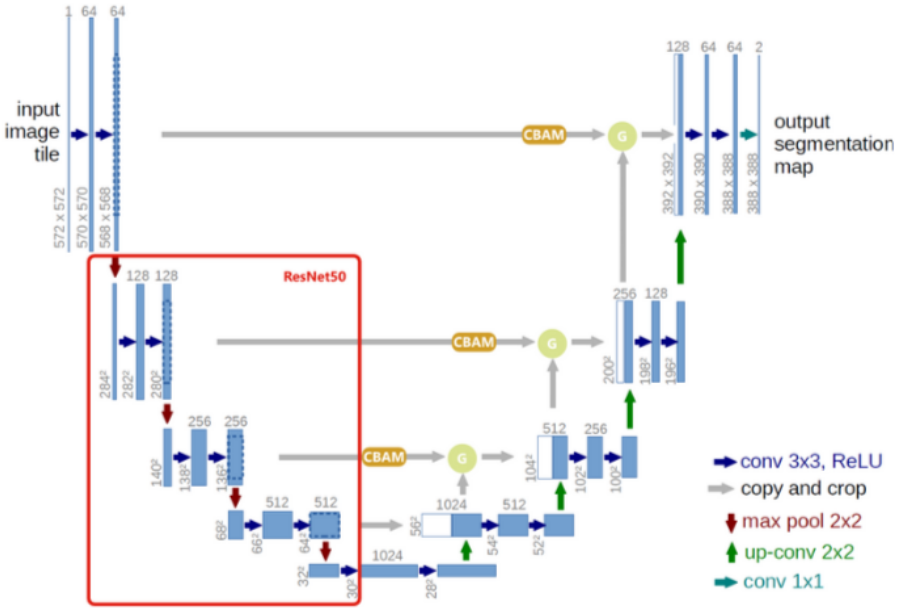

Figura 3 – Imagem de representação da arquitetura utilizada.

4 TREINAMENTO

O treinamento do modelo foi conduzido em duas etapas principais: uma fase inicial com o encoder congelado e uma fase posterior de *fine-tuning*, com todos os pesos do modelo liberados para ajuste.

Na primeira fase, apenas o decodificador foi treinado, mantendo os pesos da EfficientNetB0 congelados. Essa abordagem permite que o modelo aprenda a interpretar os recursos extraídos pelas camadas pré-treinadas, sem perturbar seu conhecimento prévio oriundo do ImageNet. Após algumas épocas, foi realizada a descongelamento total das camadas, permitindo o refinamento completo da arquitetura.

Dado o forte desbalanceamento entre as classes do dataset, especialmente com várias classes de alimentos aparecendo em poucas imagens, foi adotada uma função de perda composta, definida como:

$$\mathcal{L} = 0.7 \cdot \text{CrossEntropyPonderada} + 0.15 \cdot \text{DiceLoss} + 0.15 \cdot \text{FocalLoss} \tag{1}$$

A entropia cruzada ponderada foi utilizada com pesos elevados para as classes que possuíam performance fraca em treinamentos anteriores, alcançando até 15 vezes mais importância relativa em relação às classes comuns, como o fundo ou o prato. A Focal Loss contribuiu para reforçar o aprendizado em pixels mal classificados, enquanto a Dice Loss favoreceu o aprendizado de contornos e áreas pequenas.

Durante o treinamento, foi utilizado o otimizador Adam, com estratégia de redução de taxa de aprendizado (*ReduceLROnPlateau*) baseada na estagnação da validação. A taxa de aprendizado inicial foi de 2×10^{-3}, sendo gradualmente reduzida até um mínimo de 3.9×10^{-6}, conforme necessário.

O modelo foi treinado por 80 épocas, com tamanho de lote (batch size) de 2, devido à limitação de memória da GPU. Foram aplicadas técnicas de *early stopping* manual com base na métrica de *mean IoU* e na análise qualitativa das predições durante o processo.

5 RESULTADOS

Ao final do treinamento, o modelo alcançou uma métrica de *mean Intersection over Union* (mean IoU) de aproximadamente **0,60** no conjunto de validação, com uma perda (*loss*) estabilizada em torno de **0,39**. Esses resultados indicam uma segmentação robusta, especialmente considerando o forte desbalanceamento entre as classes.

A Tabela 2 apresenta o valor de IoU obtido individualmente para cada uma das 18 classes. É possível observar que as classes majoritárias, como fundo (classe 0), prato (classe 1) e alimentos mais frequentes, apresentaram IoUs elevados. No entanto, mesmo classes originalmente pouco representadas, como as classes 3, 5, 7 a 11, também obtiveram resultados satisfatórios após a introdução de pesos específicos na função de perda.

Tabela 2 – IoU por classe no conjunto de validação

Classe	IoU	Classe	IoU
0	0,969	9	0,689
1	0,932	10	0,522
2	0,847	11	0,471
3	0,351	12	0,219
4	0,000	13	0,564
5	0,810	14	0,845
6	0,219	15	0,724
7	0,445	16	0,783
8	0,786	17	0,728

Além dos resultados quantitativos, as predições geradas pelo modelo foram visualizadas com a aplicação de um mapa de cores específico para cada classe, o que facilitou a análise qualitativa da segmentação. Visualmente, o modelo demonstrou bom desempenho na separação de regiões de alimento e prato, mesmo em casos com múltiplas classes ou sobreposição parcial.

Por fim, foram aplicadas técnicas de *test-time augmentation* (TTA), como inversão horizontal, para gerar predições mais estáveis por meio da média de múltiplas inferências. Essa abordagem resultou em pequenas melhorias na qualidade das segmentações finais.

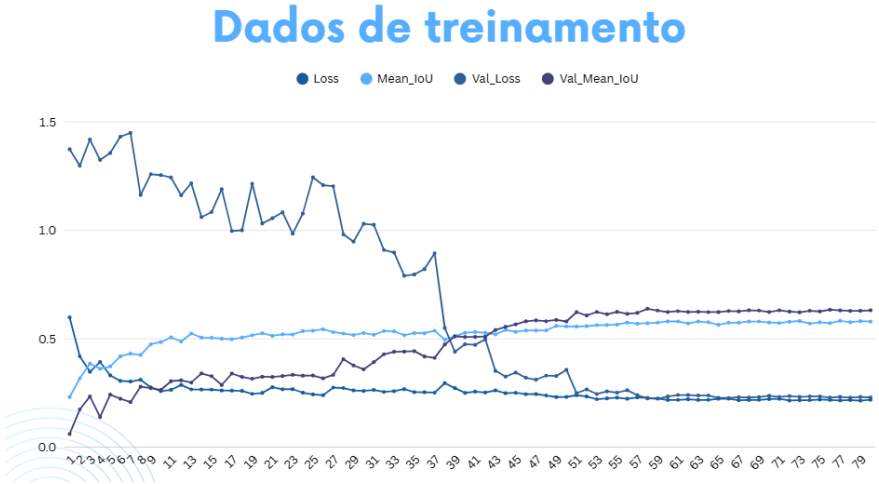

Figura 4 – Gráfico de treinamento.

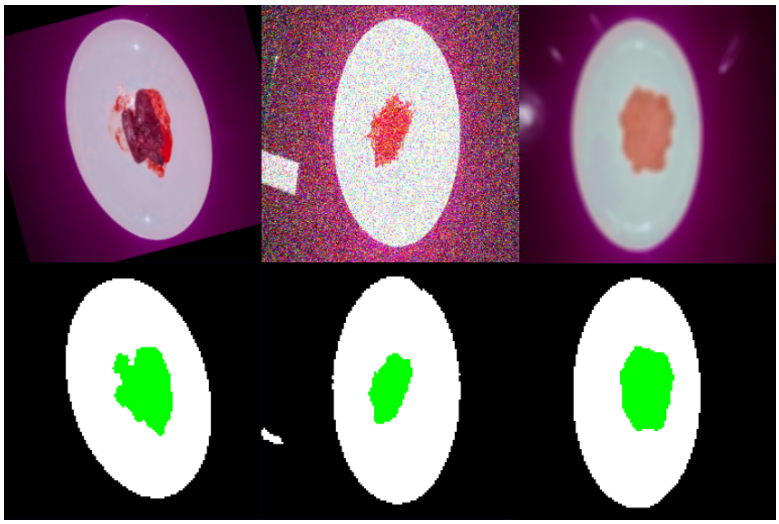

Figura 5 – Resultados de treinamento em 3 classes.

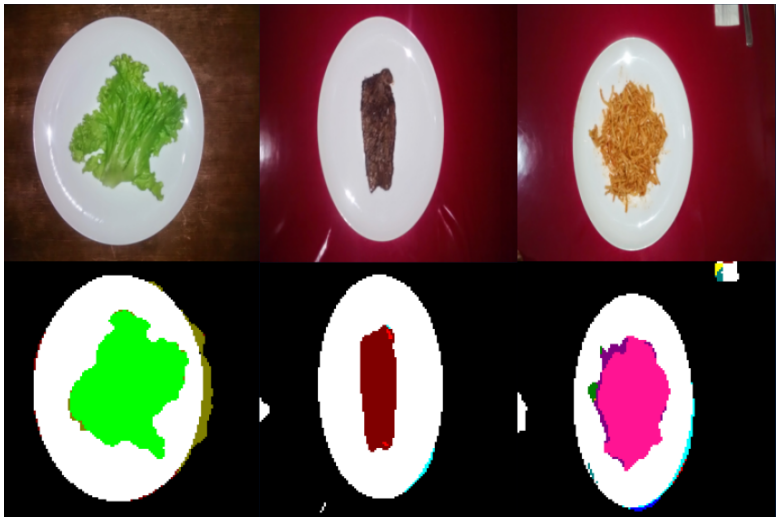

Figura 6 – Resultados de treinamento em 18 classes, aplicado a imagens com um alimento por foto.

Figura 7 – Resultados de treinamento em 18 classes, aplicado a imagens com múltiplos alimentos por foto.

6 CONCLUSÃO

Este trabalho propôs um modelo baseado em U-Net com backbone ResNet50 e módulos de atenção (CBAM e Attention Gates) para segmentação semântica de alimentos em imagens RGB. A abordagem mostrou-se eficaz para lidar com um cenário altamente desbalanceado, contendo 18 classes com frequências significativamente distintas.

Os resultados obtidos demonstram que, com estratégias adequadas de ponderação de classes, combinação de múltiplas funções de perda e uso de transfer learning, é possível alcançar boa segmentação mesmo para classes raras. Ao final do treinamento, o modelo atingiu um *mean IoU* de aproximadamente **0,60** no conjunto de validação, com valores superiores a **0,7** para diversas classes de alimentos.

A introdução de módulos de atenção foi fundamental para melhorar a qualidade das segmentações em regiões pequenas ou com ruído visual, enquanto o uso de pré-treinamento e congelamento progressivo das camadas contribuiu para uma convergência mais estável.

Apesar dos bons resultados, algumas classes extremamente raras ainda apresentaram desempenho limitado, evidenciando a necessidade de estratégias de balanceamento adicionais, como *data augmentation* específico por classe ou amostragem direcionada.

Em síntese, os experimentos confirmam a viabilidade de aplicar redes convolucionais com mecanismos de atenção para segmentação de alimentos, oferecendo resultados promissores para aplicações futuras em reconhecimento nutricional, análise de refeições e monitoramento alimentar automático.