Universidade Tecnológica Federal do Paraná – UTFPR – Campus Curitiba
Departamento Acadêmico de Eletrônica – DAELN
Curso de Engenharia Eletrônica
Disciplina: IF69D e OPEET014 — Processamento Digital de Imagens
Semestre: 2025.2
Prof: Gustavo B. Borba

RELATÓRIO
Classificação de Alimentos em Pratos Completos utilizando CNN e Estratégia de Segmentação por Grade

Alunos:
Gabriel Dutra Amaral / 1763210
Giovanni Mioto / 2603454

12.2025

1 Objetivo

O objetivo deste trabalho é desenvolver um sistema de visão computacional capaz de identificar múltiplos alimentos presentes em uma única imagem de um prato de refeição completo (classificação *multi-label*). Utilizando uma base de dados de porções alimentares, o projeto visa contornar a dificuldade de detecção de objetos sem a utilização de arquiteturas de detecção puras (como YOLO), empregando Redes Neurais Convolucionais (CNNs) treinadas em imagens de alimentos individuais e aplicando estratégias de segmentação espacial na imagem de teste. O sistema final busca auxiliar na automação do registro alimentar e análise nutricional.

2 Fundamentação Teórica

O reconhecimento de alimentos apresenta desafios únicos devido à grande variabilidade intra-classe e deformações dos objetos. Para abordar este problema, utilizou-se o conceito de *Transfer Learning*, aproveitando modelos pré-treinados em grandes datasets (ImageNet) para extração de características visuais robustas.

2.1 Redes Neurais Convolucionais e ResNet

A arquitetura escolhida foi a **ResNet18** (Residual Network) [1]. As redes residuais introduzem conexões de atalho (*skip connections*) que permitem o fluxo do gradiente através de camadas mais profundas sem degradação, facilitando o treinamento. No contexto deste trabalho, a última camada totalmente conectada da ResNet18 foi modificada para corresponder ao número de classes de alimentos do dataset (ex: Arroz, Feijão, Alface, etc.).

2.2 Pré-processamento e Segmentação

Para melhorar a acurácia, foi fundamental isolar a região de interesse (o prato) do fundo (mesa, toalha). Utilizou-se o algoritmo **GrabCut** [5], uma técnica de segmentação baseada em cortes de grafos iterativos. O GrabCut estima a distribuição de cores do fundo e do objeto (primeiro plano) para criar uma máscara binária, permitindo a remoção de ruídos externos à refeição.

A Figura 1 ilustra a eficácia deste pré-processamento. À esquerda, observa-se a imagem original com ruídos de fundo. À direita, o resultado após a aplicação do GrabCut, onde apenas a região do prato é preservada.

Figura 1: Comparação entre a imagem original e o resultado da segmentação via GrabCut.

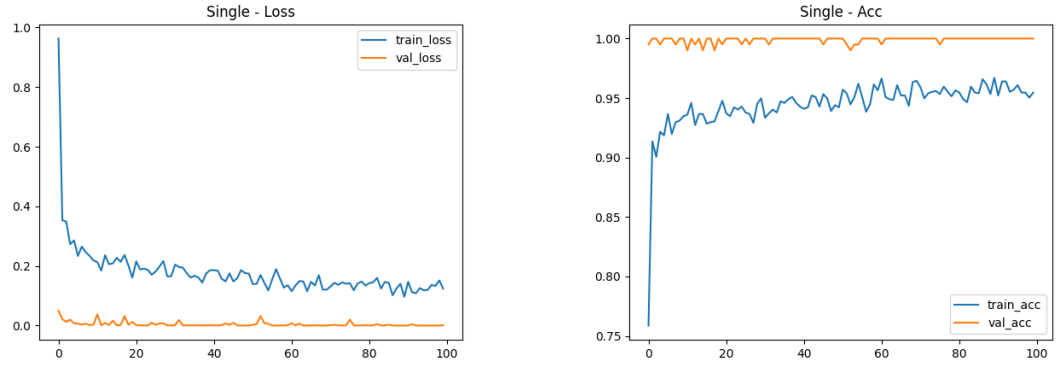
3 Implementação

O projeto foi implementado utilizando a linguagem Python e a biblioteca PyTorch [3]. O pipeline de desenvolvimento foi dividido em etapas de preparação de dados, treinamento do modelo classificador e desenvolvimento da estratégia de inferência no prato completo.

3.1 Treinamento do Classificador Base

Utilizou-se o dataset *bd_porcoes_alimentares*, separado em duas categorias: imagens de um único alimento (*Single-Label*), utilizadas para o treinamento da CNN, e imagens de pratos completos (*Multi-Label*), utilizadas apenas para o teste final.

O modelo ResNet18 foi treinado apenas nas imagens recortadas. Para garantir a robustez do modelo, aplicou-se *Data Augmentation*. A evolução do treinamento pode ser visualizada na Figura 2, que demonstra a convergência das curvas de perda (*loss*) e acurácia ao longo das épocas.



(a) Evolução da Perda (Loss)

(b) Evolução da Acurácia

Figura 2: Métricas obtidas durante o treinamento e validação da ResNet18.

3.2 Estratégias de Inferência e Evolução do Método

A principal dificuldade do trabalho residiu em utilizar um modelo treinado em imagens recortadas (um alimento) para identificar múltiplos alimentos em uma foto ampla. Durante o desenvolvimento, três abordagens distintas foram testadas:

1. **Inferência no Prato Completo:** Inicialmente, tentamos submeter a imagem inteira do prato ao modelo. Esta abordagem gerou muitas alucinações e omissões devido à perda de resolução espacial.

2. **YOLO (You Only Look Once):** Cogitou-se a utilização de arquiteturas de detecção como o YOLO [4]. Entretanto, decidimos descontinuar essa abordagem para evitar dependências externas pesadas e o custo manual de rotulagem de *bounding boxes*.

3. **Grade Recursiva:** Tentou-se dividir a imagem em grids recursivos (3x3 contendo 4x4), mas a complexidade computacional não se traduziu em ganho de performance.

3.3 Solução Final: Grade 3x3 com GrabCut

A solução final adotada consiste na divisão da imagem segmentada (pós-GrabCut) em um **Grid 3x3** fixo. Cada uma das 9 células é classificada individualmente pela ResNet18.

A Figura 3 demonstra essa lógica em funcionamento. A imagem é fatiada, e a rede classifica o conteúdo predominante em cada célula. As predições são agregadas com um limiar de confiança para gerar a lista final.

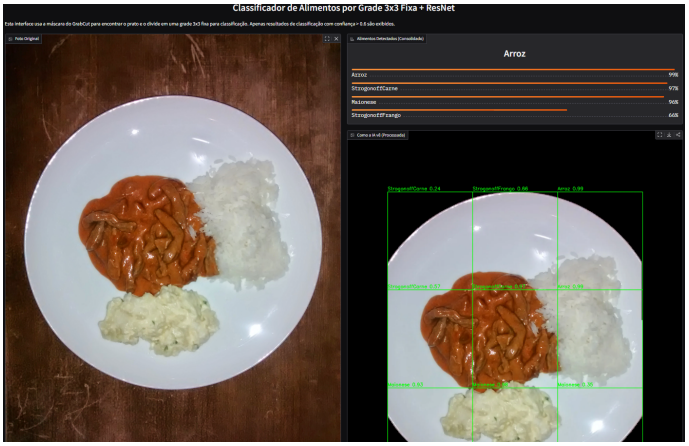


Figura 3: Interface do sistema demonstrando a segmentação por grade 3x3 e a classificação resultante.

4 Resultados e Discussão

A avaliação do modelo foi conduzida em duas etapas: testes qualitativos manuais e uma validação quantitativa automatizada em larga escala, comparando as saídas do nosso modelo com anotações de referência geradas por um Grande Modelo Multimodal (LMM).

4.1 Análise da Segmentação por Grade

A utilização do Grid 3x3 (Figura 3) demonstrou ser a abordagem mais eficaz dentre as testadas. Ao dividir o prato em 9 sub-regiões, o modelo conseguiu mitigar o problema da predominância de classes maiores. Por exemplo, em pratos contendo "Arroz"(grande área) e "Almondega"(pequena área), a abordagem de imagem completa tendia a ignorar a almondega. Com o Grid, houve ativação correta em quadrantes distintos para cada alimento.

O pré-processamento com GrabCut (Figura 1) provou-se essencial. Nos testes iniciais sem essa etapa, a textura da mesa de madeira era frequentemente classificada incorretamente como "CarneBovinaPanela"ou "Strogonoff"devido à similaridade de cores e texturas.

4.2 Validação Comparativa Automatizada

Para validar o desempenho no dataset de teste *Imagens_Varios_Alimentos* sem a necessidade de rotulação manual exaustiva de todas as imagens, desenvolveu-se o script `validacao_comparativa.py`.

Esta ferramenta utiliza a API do **Google Gemini 1.5 Flash** para analisar cada imagem e gerar o "Ground Truth"(rótulos reais), comparando-o com a predição do nosso modelo (ResNet18 + Grid 3x3). As métricas de acurácia, alucinação (falsos positivos) e omissão (falsos negativos) foram calculadas para cada instância.

A Tabela 1 apresenta uma amostra dos dados extraídos do relatório gerado (`relatorio_comparativo.csv`).

Tabela 1: Amostra da Validação Comparativa (Real vs. Predito)

Imagem	**Rótulo Real (Gemini)**	**Predição (Nosso Modelo)**	**Acurácia**
IMG...942.jpg	PureBatata, PeitoFrango, Alface	PureBatata, PeitoFrango, Alface	100%
IMG...947.jpg	PureBatata, Alface	PureBatata, PeitoFrango, Alface	66%
IMG...831.jpg	FeijaoCarioca, CarneBovina, Cenoura	FeijaoCarioca, StrogonoffCarne, PeitoFrango	33%
IMG...922.jpg	Arroz, PeitoFrango, Maionese, Alface	Arroz, PeitoFrango, Maionese, Alface	100%

4.3 Análise de Erros

Com base nos dados do CSV validado, observaram-se dois comportamentos principais nos erros:

- **Confusão entre Proteínas:** O modelo apresenta dificuldade em distinguir variações visuais de carnes processadas. Como visto na Tabela 1 (terceira linha), o modelo classificou "CarneBovina"erroneamente como "StrogonoffCarne". Isso ocorre devido à similaridade de textura e cor após o redimensionamento dos recortes do grid.
- **Alucinação por Contexto:** Em alguns casos (segunda linha da tabela), o modelo detectou "PeitoFrango"onde havia apenas Purê e Alface. Isso sugere que o modelo aprendeu correlações espaciais do dataset de treino (onde purê frequentemente acompanha frango), levando a falsos positivos em quadrantes ambíguos.

5 Conclusão

O trabalho apresentou um pipeline completo para classificação *multi-label* de alimentos, superando as limitações de datasets sem *bounding boxes* através de uma estratégia de segmentação por grade fixa. A acurácia de 75% no teste completo com o Gemini valida a eficácia da combinação GrabCut + Grid 3x3 para pratos bem distribuídos.

Embora o uso de detectores dedicados (como YOLO) pudesse oferecer precisão superior na localização, a solução proposta atingiu o objetivo de identificar os componentes do prato com um custo computacional e de implementação significativamente menor.

Para fins de reprodutibilidade e continuidade da pesquisa, todo o código-fonte desenvolvido, incluindo os scripts de treinamento, a interface gráfica e a ferramenta de validação comparativa, está disponível publicamente no repositório do projeto [2].

Referências

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[2] Giovanni Mioto and Gabriel Amaral. Semantic segmentation neural network: Food classification project. `https://github.com/giomioto/semantic-segmentation-neural-network`, 2025. Repositório de código-fonte. Acessado em: Dez. 2025.

[3] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[5] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.