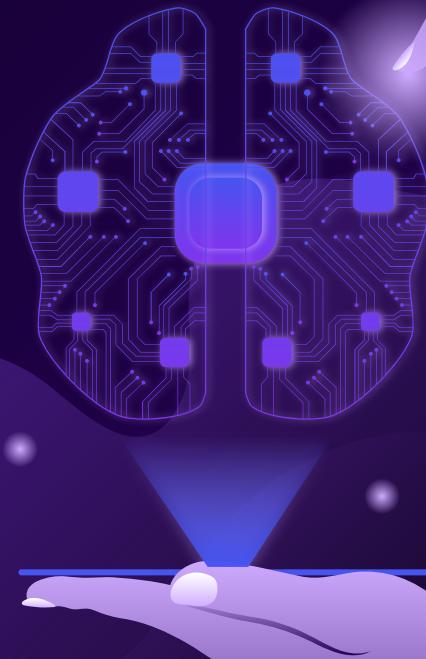


Classificação de Alimentos em Pratos Completos utilizando CNN

**Aplicação de CNN (ResNet18)
com Estratégia de Segmentação
por Grade**

Autores: Gabriel Dutra Amaral e Giovanni Mioto



O Dataset

- Imagens vindas do banco de dados bd_porcoes_alimentares, disponível no [GitHub](#);
- Duas pastas principais:
 - **Imagens_um_Alimento:**
 - 16 pastas com 2011 imagens individuais de diversos alimentos.
 - **Imagens_Varios_Alimentos:**
 - 289 imagens contendo diversos alimentos em um prato (até mesmo alguns que não estão em Imagens_um_Alimento).

A Tarefa escolhida

- Identificar quais alimentos estão presentes em um prato, de acordo com os alimentos previstos na pasta Imagens_um_Alimento;
- Uma Rede Neural Convolucional (CNN) foi treinada para permitir a execução desta Task.

Pipeline de Treinamento

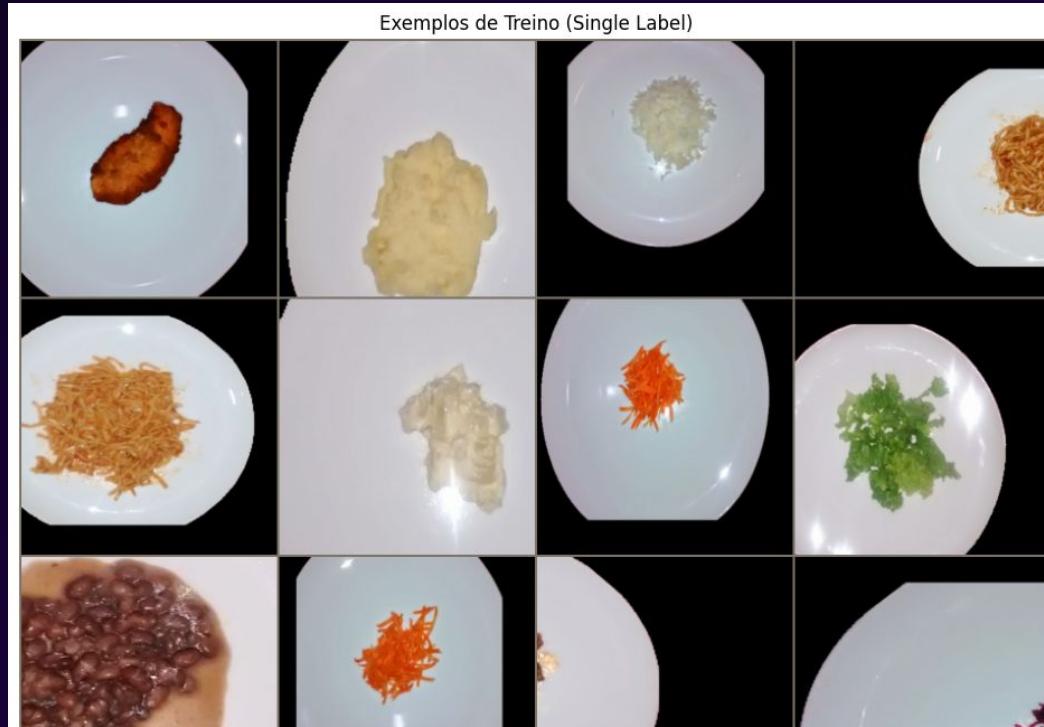
- **Arquitetura:** Dentre as disponíveis, a que nos deu o melhor resultado foi a **ResNet18** (Redes Neurais Residuais com 18 camadas de aprendizado e cerca de 11 milhões de parâmetros)
 - **Estratégia:** Utilização de **Transfer Learning** (pesos da ImageNet).
- **Pré-processamento:** GrabCut para remover os pixels de fundo (mesa) e focar apenas no prato (alimento em si);

Pipeline de Treinamento

- **Data Augmentation:** Aplicação de **RandomResizedCrop** e **RandomHorizontalFlip** para aumentar a variabilidade e evitar que a rede memorize posições fixas.
- **Estratégia de Aprendizado: Treinamento Single-Label**
 - O modelo foi treinado utilizando apenas imagens de alimentos isolados para garantir a robustez na identificação de texturas individuais.

Resultados Obtidos (Treinamento)

- Amostra com pré-processamento e Data Augmentation:



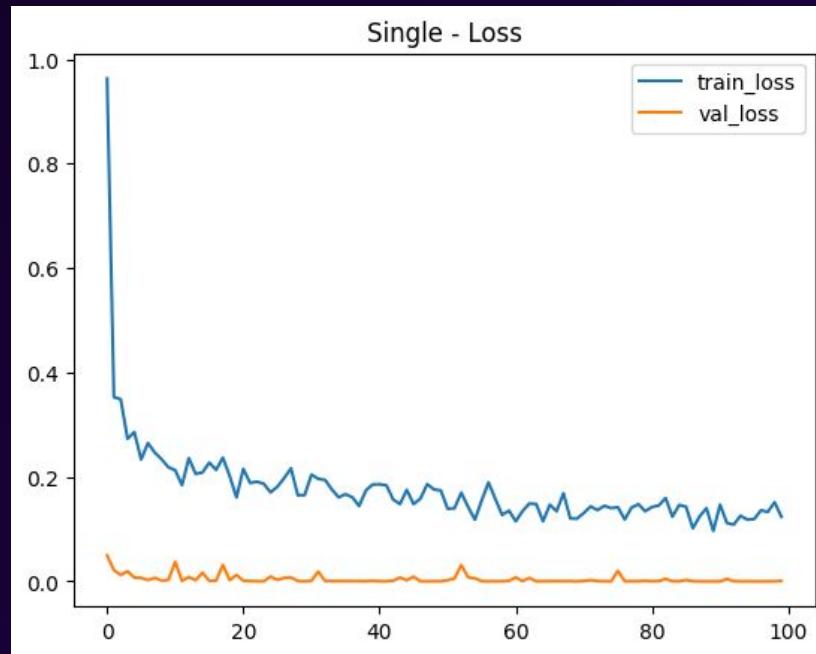
Resultados Obtidos (Treinamento)

- O modelo foi treinado com 100 épocas para garantir um resultado melhor:

```
Epoch 91 train_loss 0.1469 train_acc 0.9521 | val_loss 0.0000 val_acc 1.0000
Train single epoch 92/100: 100%|██████████| 101/101 [01:00<00:00, 1.67it/s]
Epoch 92 train_loss 0.1118 train_acc 0.9639 | val_loss 0.0046 val_acc 1.0000
Train single epoch 93/100: 100%|██████████| 101/101 [01:09<00:00, 1.46it/s]
Epoch 93 train_loss 0.1088 train_acc 0.9639 | val_loss 0.0004 val_acc 1.0000
Train single epoch 94/100: 100%|██████████| 101/101 [01:04<00:00, 1.57it/s]
Epoch 94 train_loss 0.1256 train_acc 0.9552 | val_loss 0.0002 val_acc 1.0000
Train single epoch 95/100: 100%|██████████| 101/101 [01:02<00:00, 1.62it/s]
Epoch 95 train_loss 0.1182 train_acc 0.9571 | val_loss 0.0001 val_acc 1.0000
Train single epoch 96/100: 100%|██████████| 101/101 [01:00<00:00, 1.67it/s]
Epoch 96 train_loss 0.1194 train_acc 0.9608 | val_loss 0.0001 val_acc 1.0000
Train single epoch 97/100: 100%|██████████| 101/101 [01:00<00:00, 1.66it/s]
Epoch 97 train_loss 0.1362 train_acc 0.9546 | val_loss 0.0001 val_acc 1.0000
Train single epoch 98/100: 100%|██████████| 101/101 [01:01<00:00, 1.65it/s]
Epoch 98 train_loss 0.1331 train_acc 0.9546 | val_loss 0.0001 val_acc 1.0000
Train single epoch 99/100: 100%|██████████| 101/101 [01:01<00:00, 1.65it/s]
Epoch 99 train_loss 0.1515 train_acc 0.9502 | val_loss 0.0003 val_acc 1.0000
```

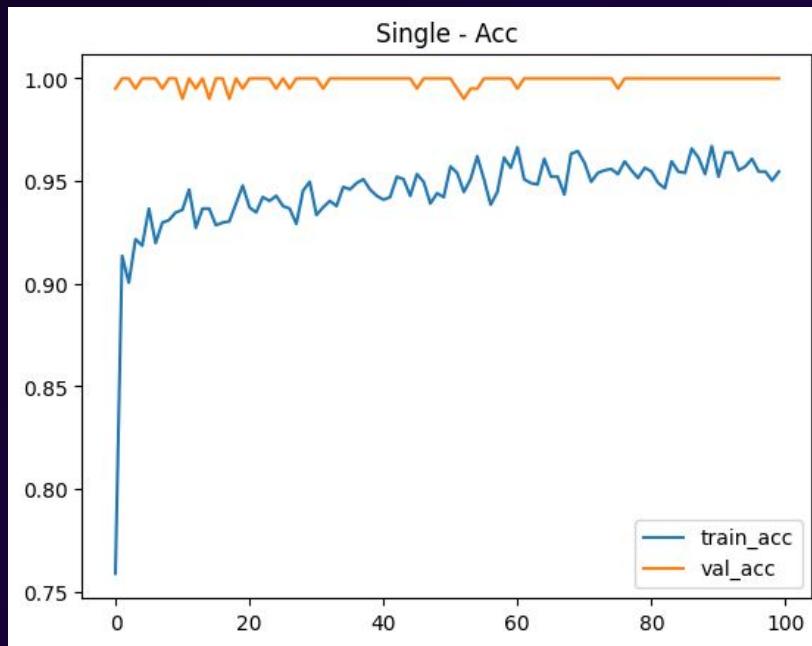
Resultados Obtidos (Treinamento)

- Este treinamento resultou no seguinte gráfico de perda:



Resultados Obtidos (Treinamento)

- E no seguinte gráfico de acurácia:



Estratégia de Inferência

- Como o modelo aprendeu a visualizar um item por vez, não seria possível classificar o prato todo de uma vez só;
- **Solução (Algoritmo):**
 - **1.** Detecção da área do prato via máscara (GrabCut);
 - **2.** Divisão da imagem em um Grid (Grade 3x3);
 - **3.** Classificação independente de cada célula da grade;
 - **4.** Threshold: apenas alimentos acima de 60% de certeza.

Estratégia de Inferência

Classificador de Alimentos por Grade 3x3 Fixa + ResNet

Esta interface usa a máscara do GrabCut para encontrar o prato e o divide em uma grade 3x3 fixa para classificação. Apenas resultados de classificação com confiança > 0.6 são exibidos.

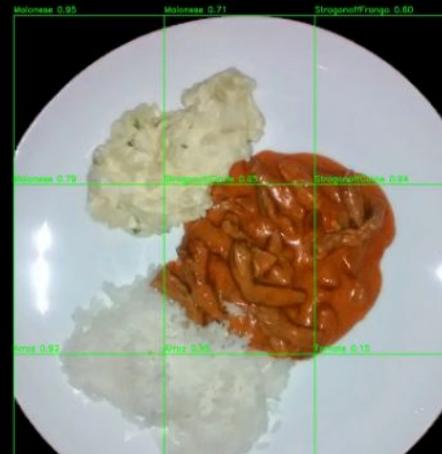
Foto Original



Alimentos Detectados (Consolidado)

Alimento	Confiança (%)
StrogonoffCarne	95%
Azeite	94%
Maionese	82%

Como a Iável (Processado)



Resultados Quantitativos

- **Metodologia de Teste:**
 - Um Script de validação foi criado, rodando sobre o dataset presente na pasta Imagens_Varios_Alimentos;
 - As imagens são pré-processadas e comparadas com o gabarito feito pelo usuário;
 - O resultado é retornado em CSV e no terminal.

Resultados Quantitativos

- **Métricas obtidas:**

- **Precision:** das previsões positivas, quantas estavam corretas;
- **Recall:** dos casos positivos reais, quantos o modelo identificou;
- **F1-Score:** média harmônica entre Precisão e Recall;
- **Acurácia:** porcentagem de previsões corretas sobre o total de amostras.

RESULTADO FINAL				
	precision	recall	f1-score	support
Alface	1.00	1.00	1.00	3
Almondega	0.00	0.00	0.00	2
Arroz	1.00	1.00	1.00	4
BatataFrita	1.00	1.00	1.00	4
Beterraba	1.00	1.00	1.00	3
BifeBovinoChapa	0.00	0.00	0.00	0
CarneBovinaPanela	0.00	0.00	0.00	0
Cenoura	1.00	0.67	0.80	3
FeijaoCarioca	1.00	1.00	1.00	1
Macarrao	0.00	0.00	0.00	0
Maionese	1.00	1.00	1.00	2
PeitoFrango	1.00	1.00	1.00	2
PureBatata	1.00	1.00	1.00	4
StrogonoffCarne	1.00	1.00	1.00	4
StrogonoffFrango	0.50	1.00	0.67	2
Tomate	1.00	1.00	1.00	3
micro avg	0.94	0.92	0.93	37
macro avg	0.72	0.73	0.72	37
weighted avg	0.92	0.92	0.91	37
samples avg	0.93	0.92	0.92	37

Acurácia Exata (Acertou o prato 100%): 81.82%

Resultados Quantitativos

- **Comparação direta com Gemini sendo a base das respostas.**
 - **265 imagens testadas**
 - **Métricas obtidas:**
 - Recall;
 - Acurácia.

Acurácia Média	72,38
Recall Global	70,52

Conclusão

- **Grid Crop (Grade)** permitiu a realização de uma tarefa Multi-Label utilizando um classificador Single-Label;
- **Eficiência da Arquitetura:** A ResNet18 demonstrou ser o equilíbrio ideal entre desempenho e custo computacional;
- **Foco na Textura:** O modelo foi capaz de generalizar bem as características de textura dos alimentos.

Referências

- [Geeks for Geeks] ResNet18 from scratch using pytorch.
<https://www.geeksforgeeks.org/deep-learning/resnet18-from-scratch-using-pytorch/>;
- [Dataset] SHIGA, Yuri. Banco de Dados de Porções Alimentares.
Disponível em: github.com/yurishiga/bd_porcoes_alimentares.