



Inspeção de pacotes com Deep Learning

Por:

Davi Brilhante

Guilherme Máximo

Matheus Stepple

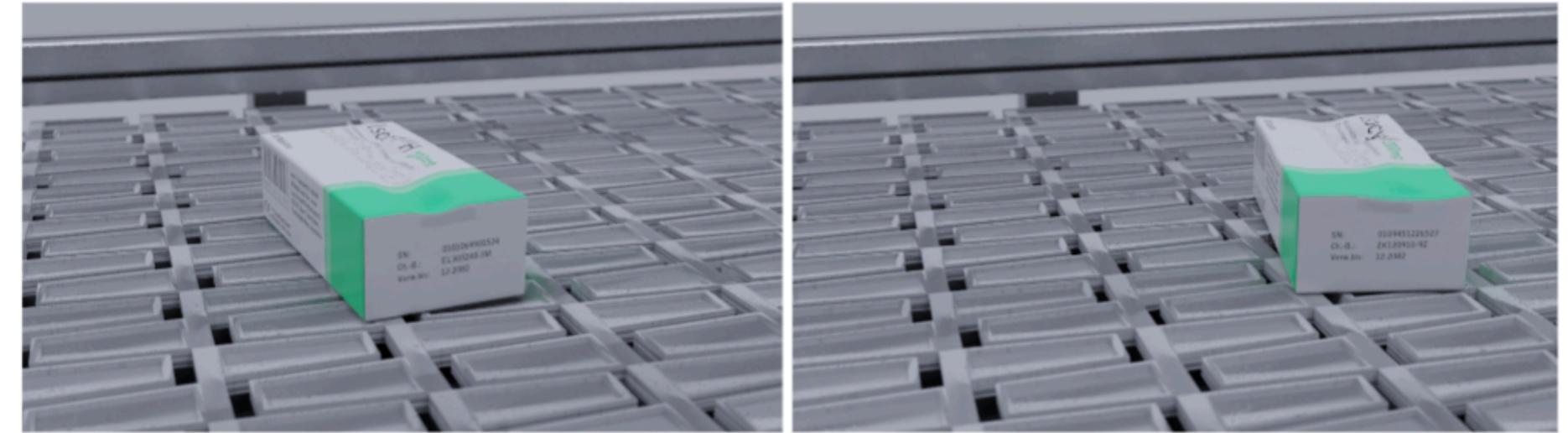
Vinícius Sá de Melo

O desafio

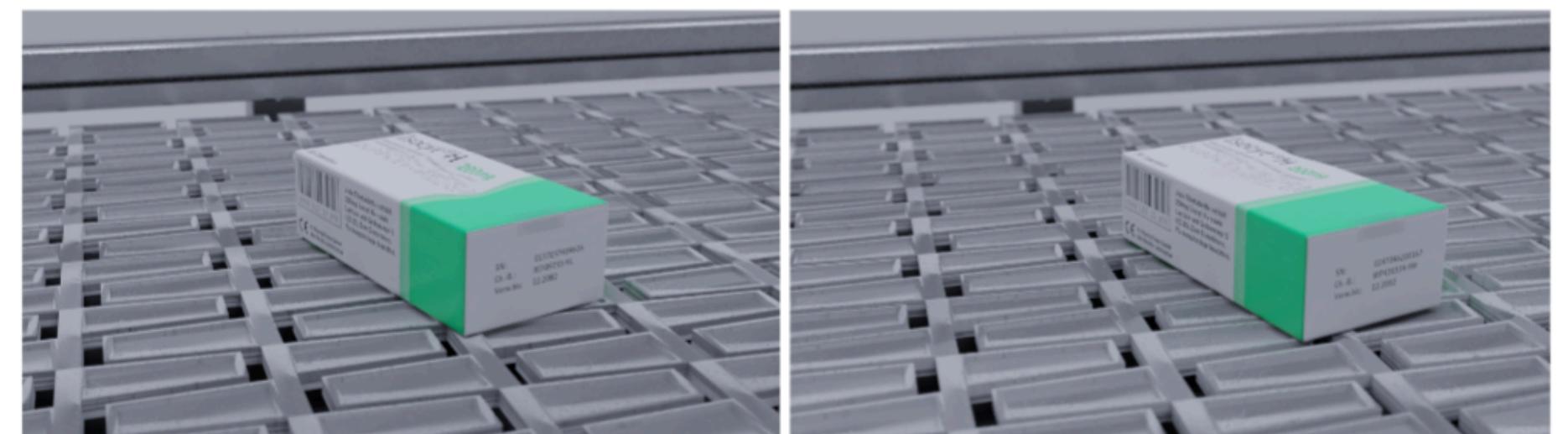
Estrutura do Dataset:
archive/

```
├── damage/  
│   └── side/  
│       └── top/  
└── intact/  
    ├── side/  
    └── top/  
└── interpretabilidade/  
    ├── side/  
    └── top/
```

Exemplos de pacotes danificados:

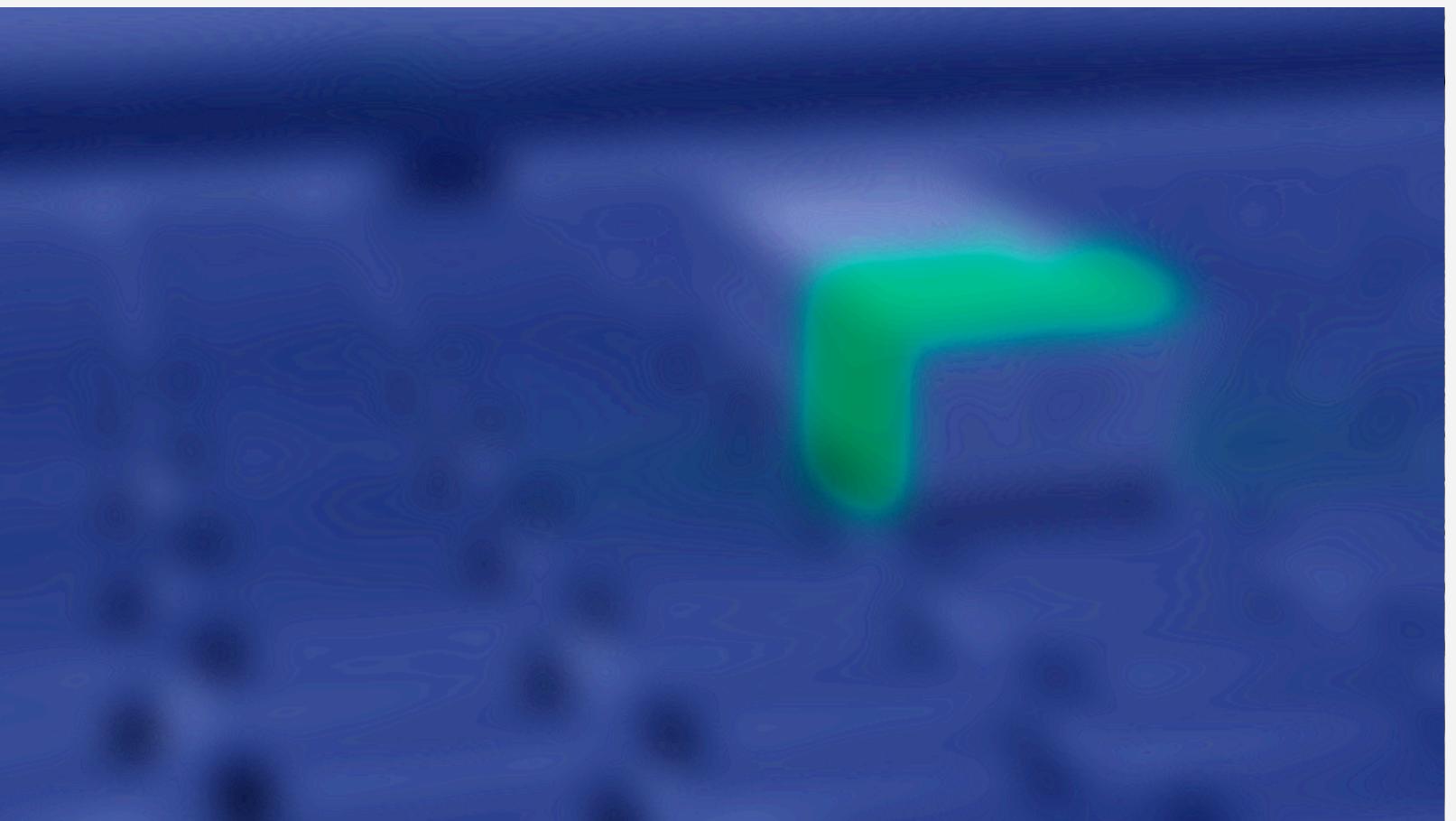


Exemplos de pacotes intactos:



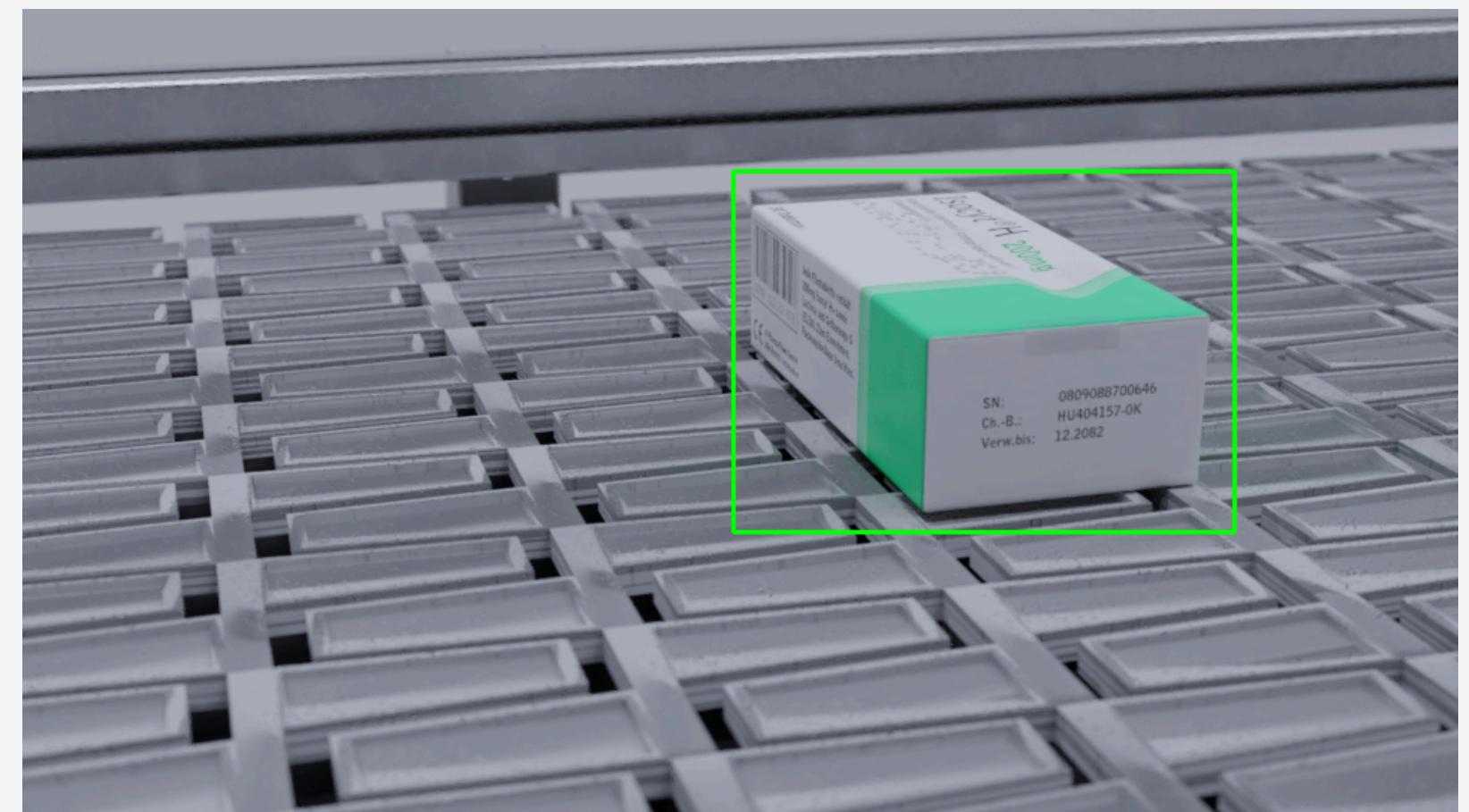
Pré-processamento

- Encontrar a caixa e cortar a imagem (crop)
 - Processamento de imagens
 - Visão computacional clássica



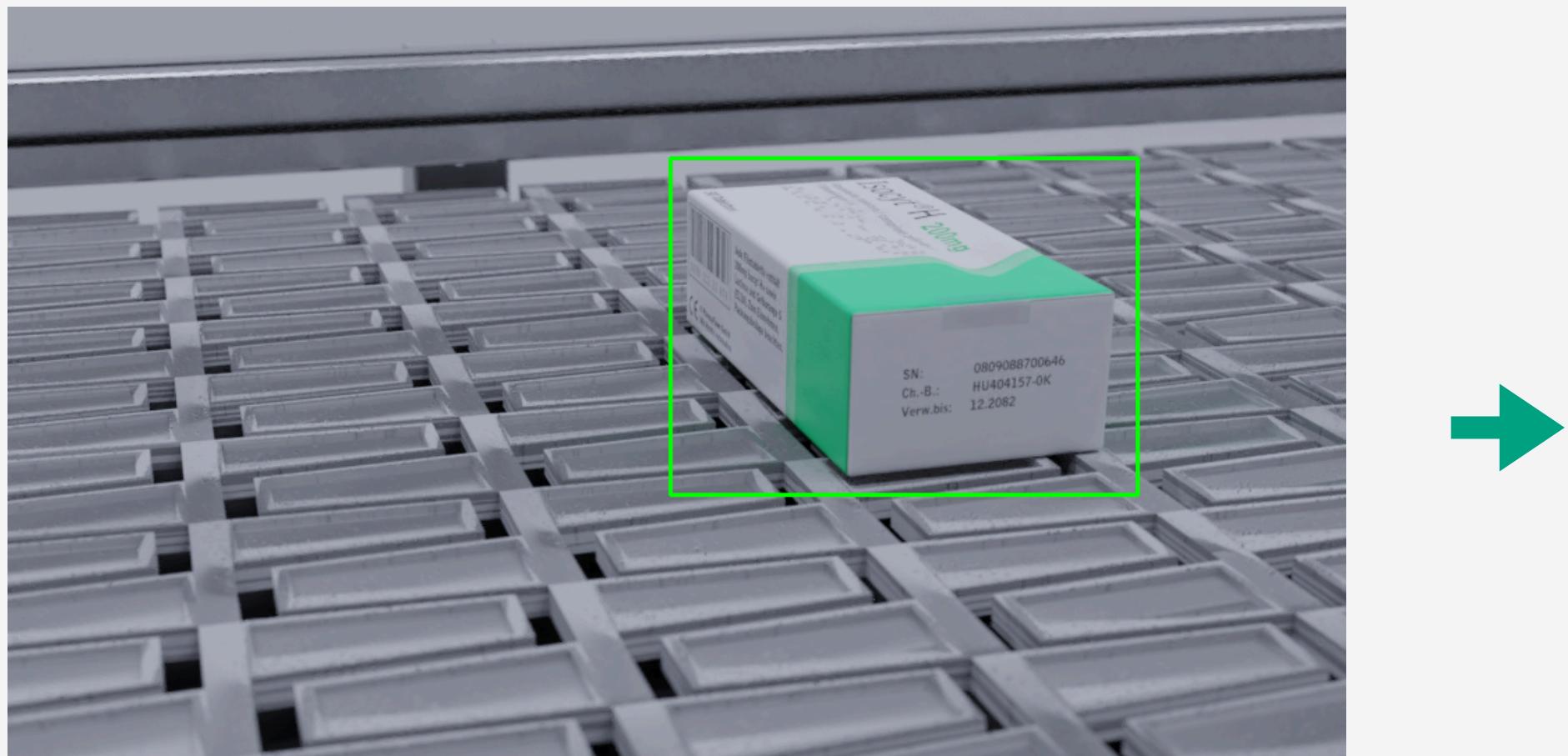
Pré-processamento

- Encontrar a caixa e cortar a imagem (crop)
 - Processamento de imagens
 - Visão computacional clássica



Pré-processamento

- Encontrar a caixa e cortar a imagem (crop)
 - Processamento de imagens
 - Visão computacional clássica



Pré-processamento

- Filtros: Sharpening



Imagen original



Filtro de sharpening

Pré-processamento

- Filtros: Clahe (Equalização de Histograma Adaptativa com Limite de Contraste)



Imagen original



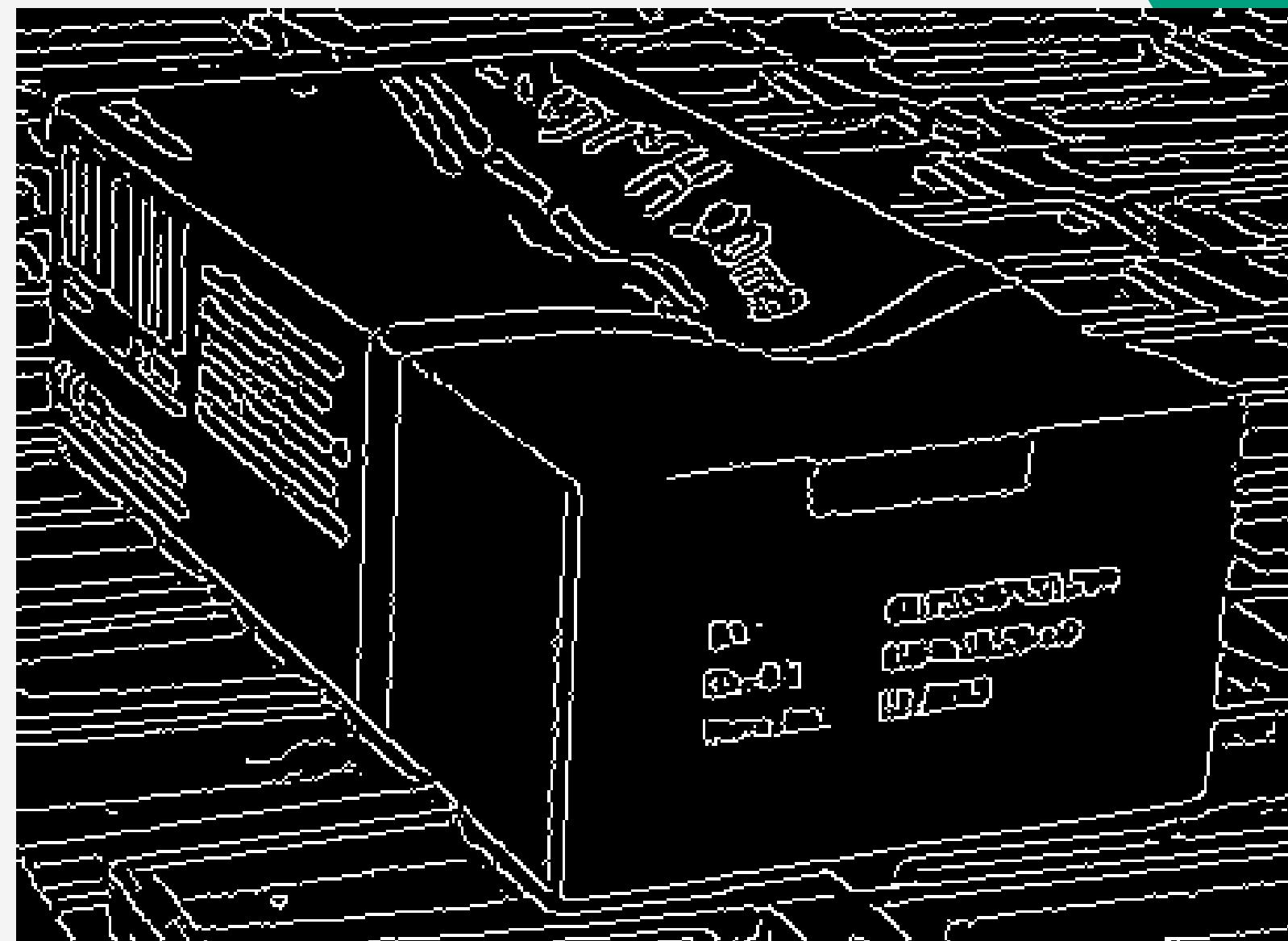
Filtro de Clahe

Pré-processamento

- Filtros: Canny



Imagen original



Filtro de Canny

Pré-processamento

- Filtros: scahrr



Imagen original



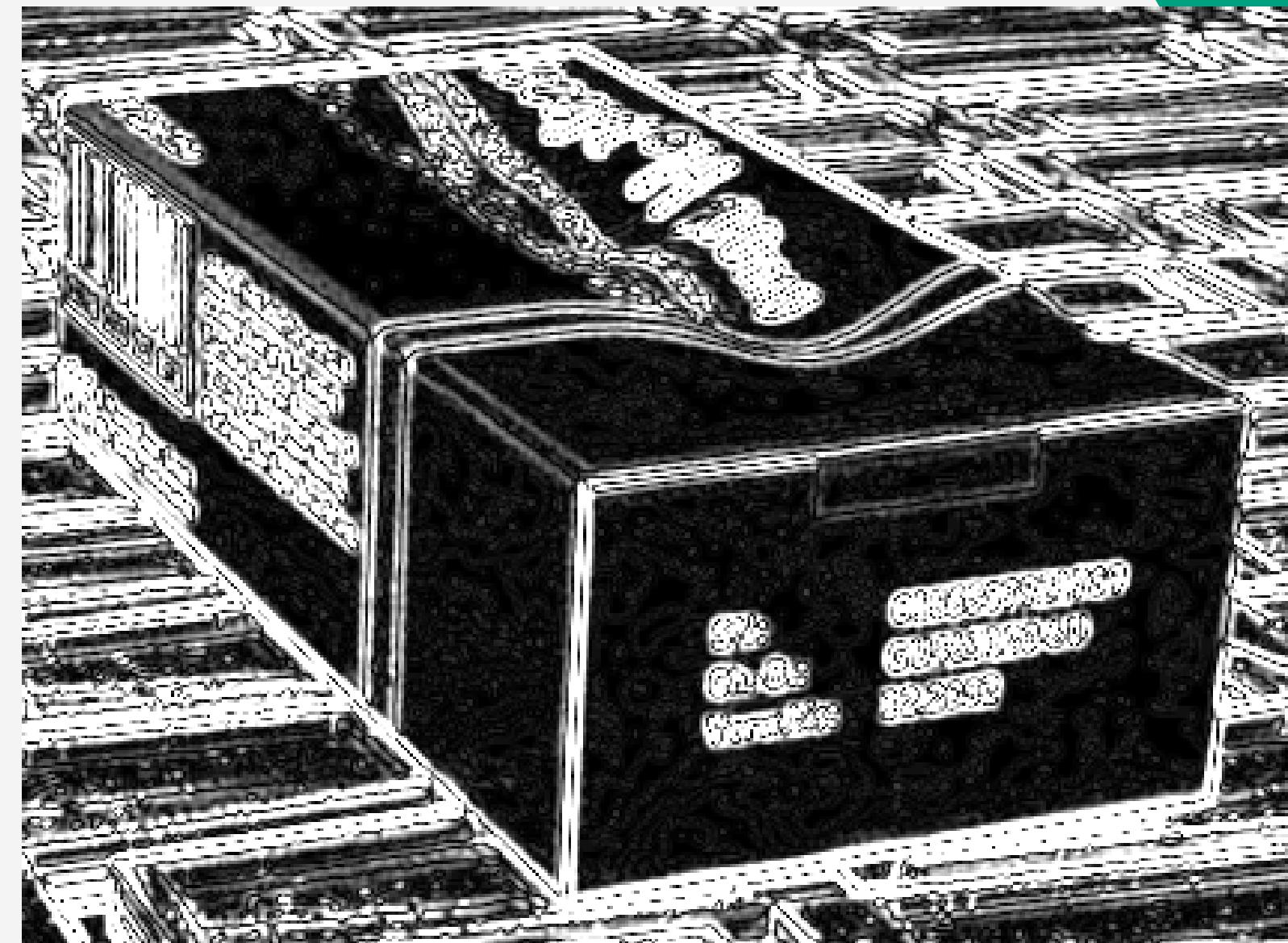
Filtro de scahrr

Pré-processamento

- Filtros: Laplace



Imagen original



Filtro de Laplace

Pré-processamento

- Fundo removido



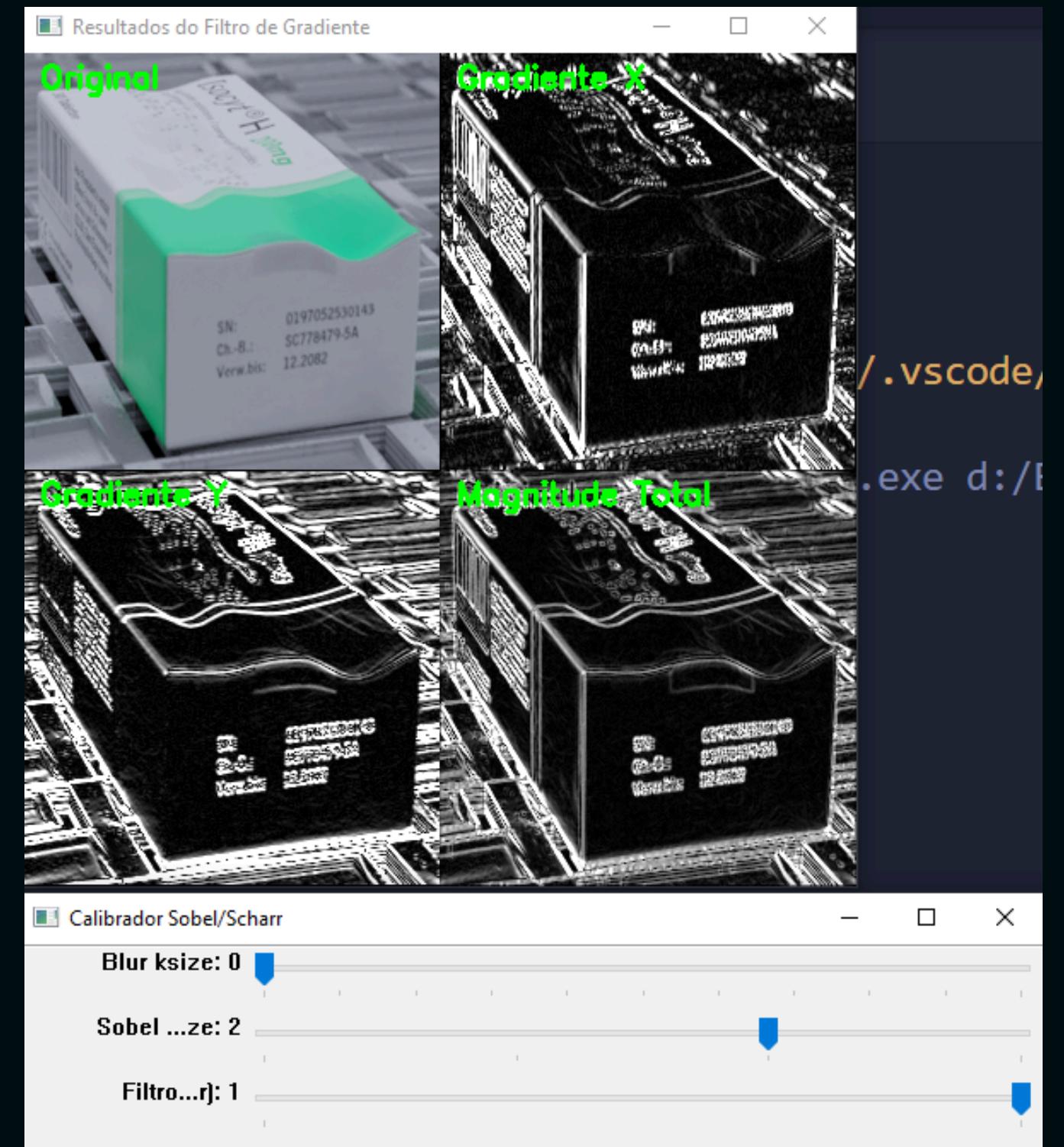
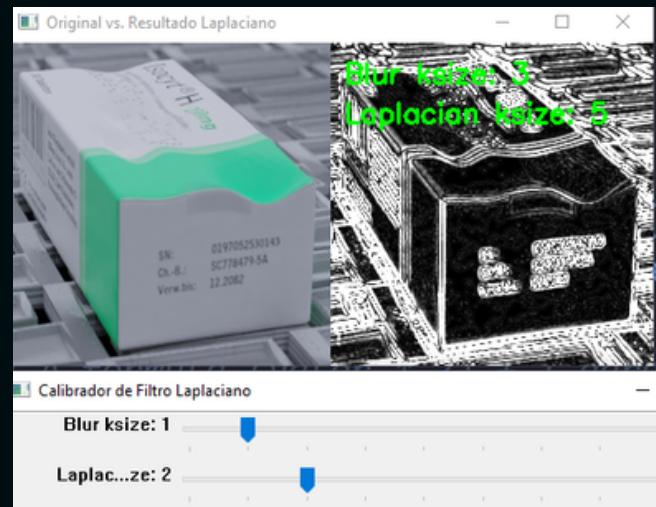
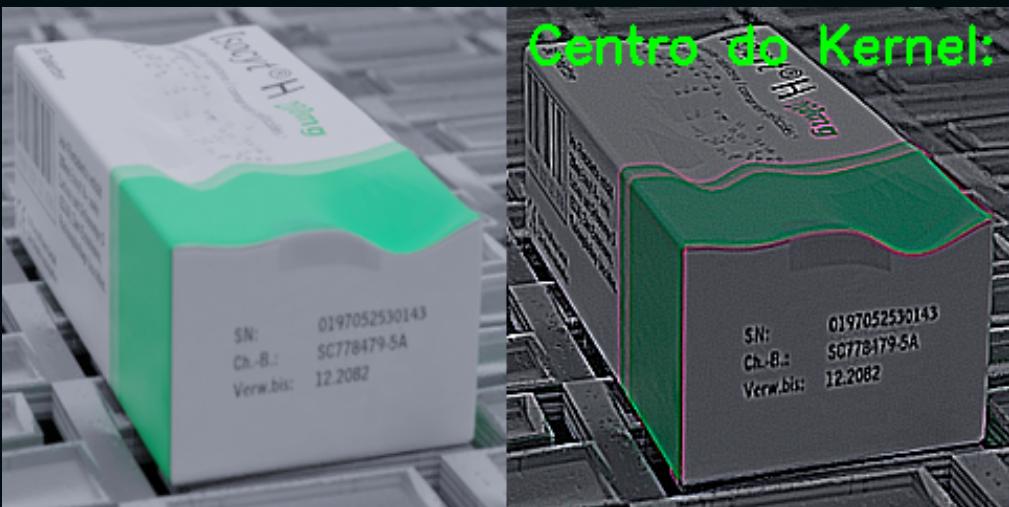
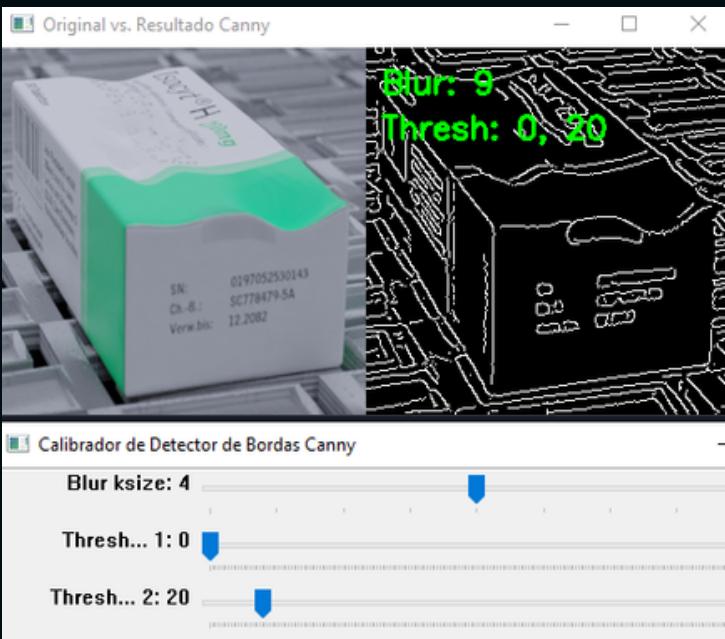
Imagen original



Fundo removido

Pré-processamento

- Metodologia de calibração dos filtros



CNN - Arquitetura Própria

Arquitetura formada por:

- Rede Sequential com 3 blocos Conv2D + MaxPooling.
- Camadas Conv2D com filtros 32, 64 e 128 e ReLU.
- Flatten + Dense (512) com Dropout de 50%.
- Saída com sigmoid para classificação binária.

Pré-processamento e Dados

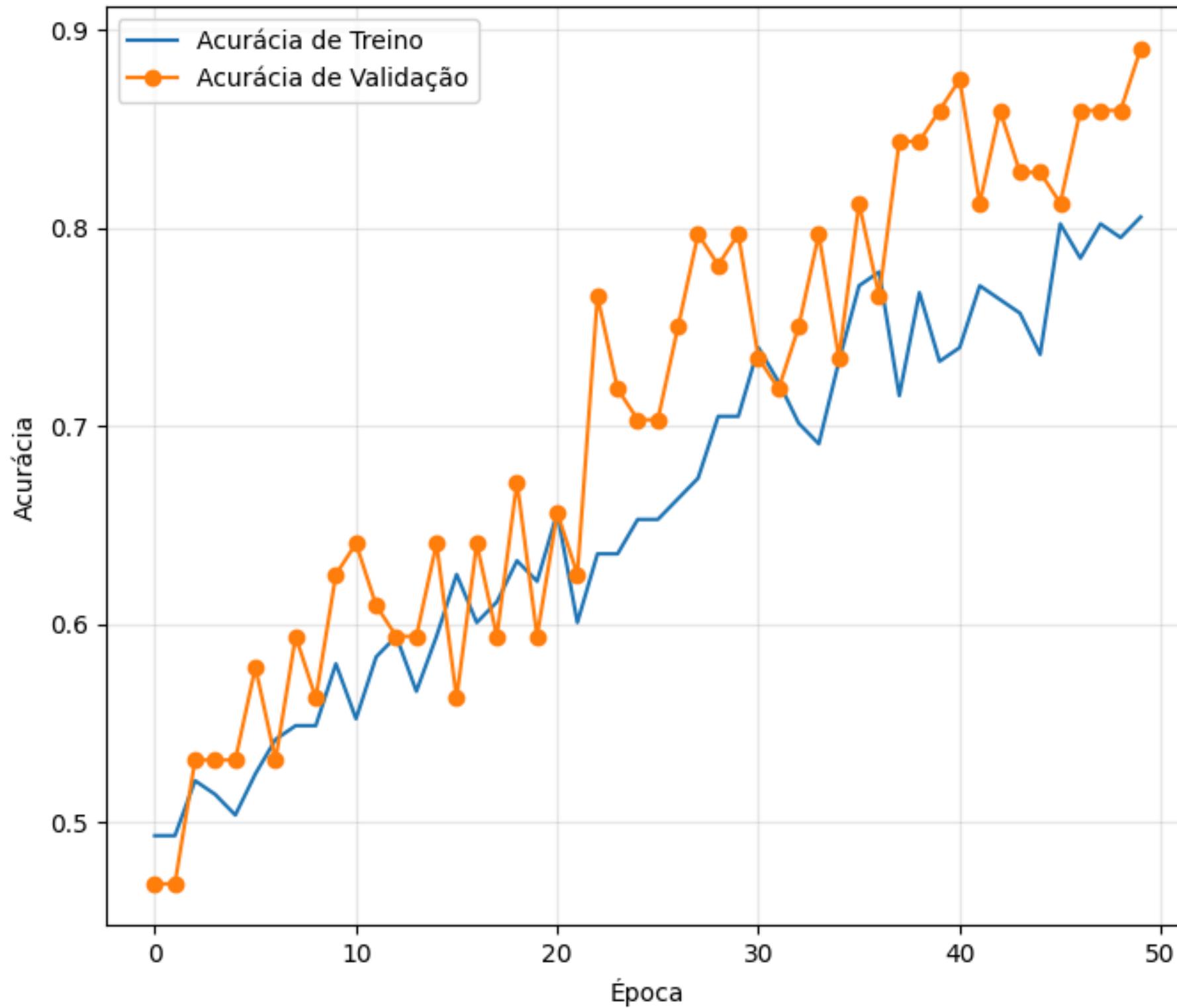
- Imagens redimensionadas para 224×224.
- Data augmentation leve (rotação, zoom, translação).
- Normalização aplicada manualmente no carregamento (/255).
- Gerador personalizado para misturar top e side.

Treinamento e Otimização

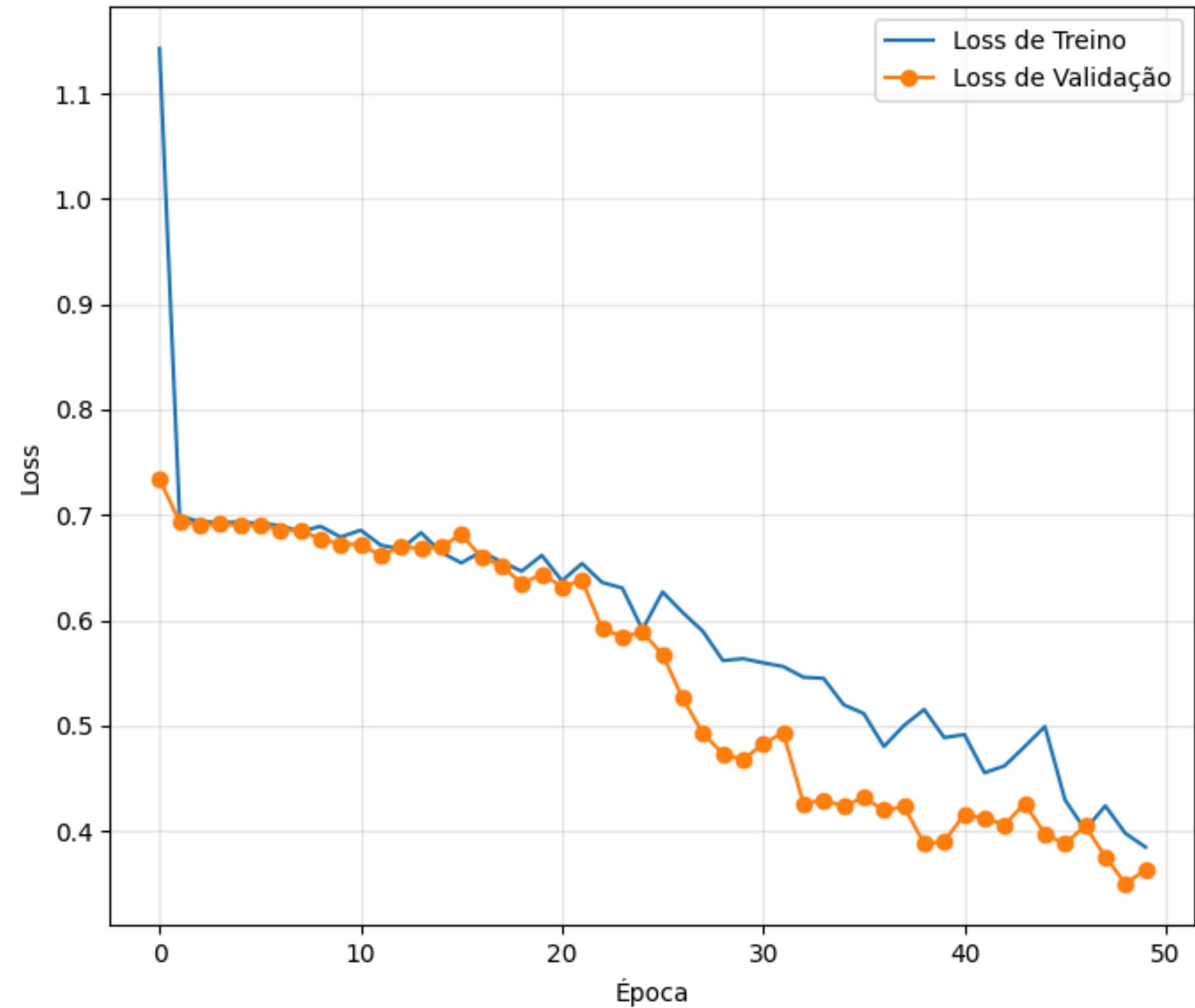
- Otimizador Adam com LR = 0.0005.
- Função de perda: binary_crossentropy.
- Callbacks: ModelCheckpoint, EarlyStopping, ReduceLROnPlateau.
- Divisão 80/20 entre treino e validação com embaralhamento.

Resultado Final - Arquitetura Clássica

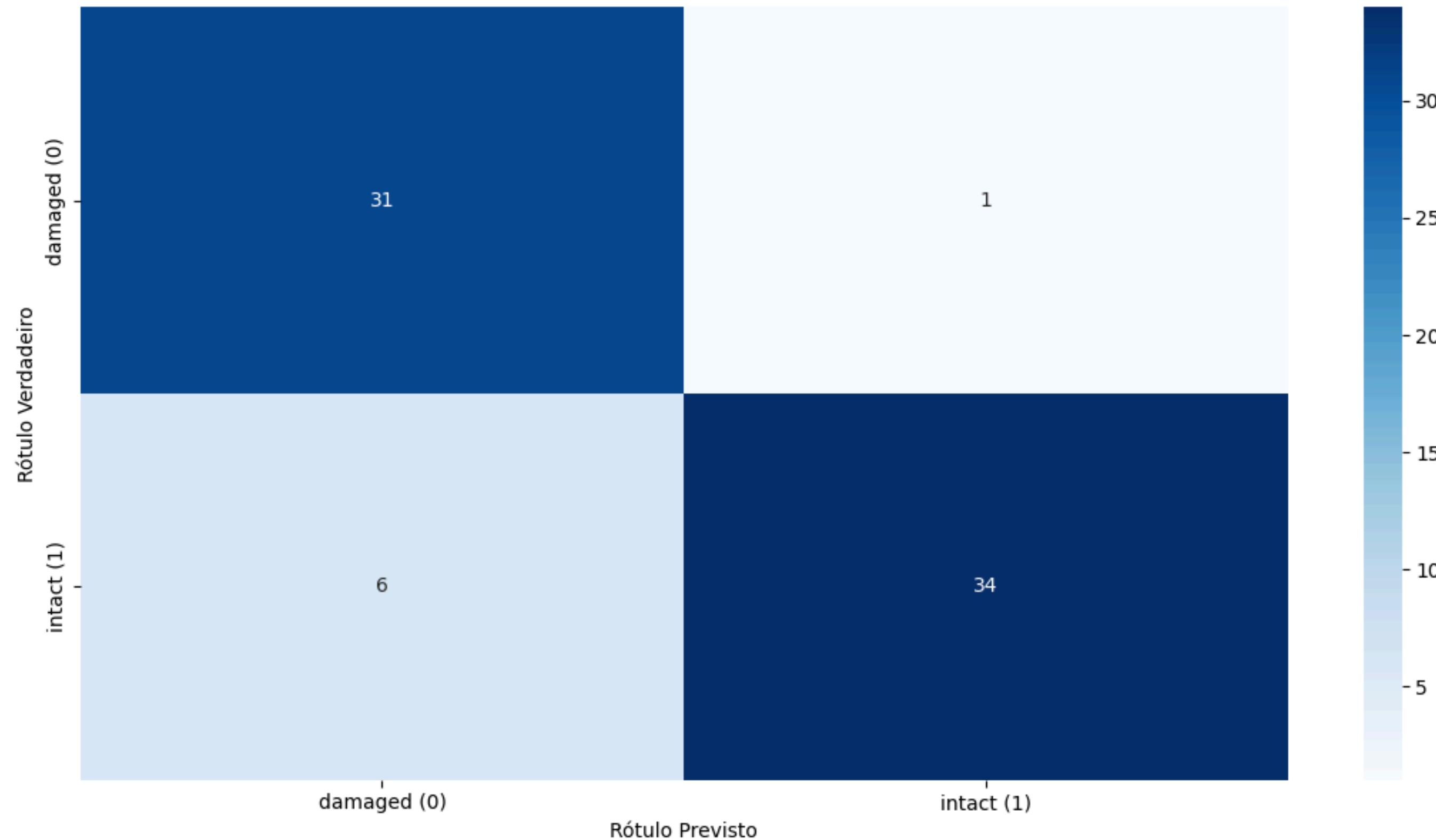
Evolução da Acurácia



Evolução da Loss



Matriz de Confusão



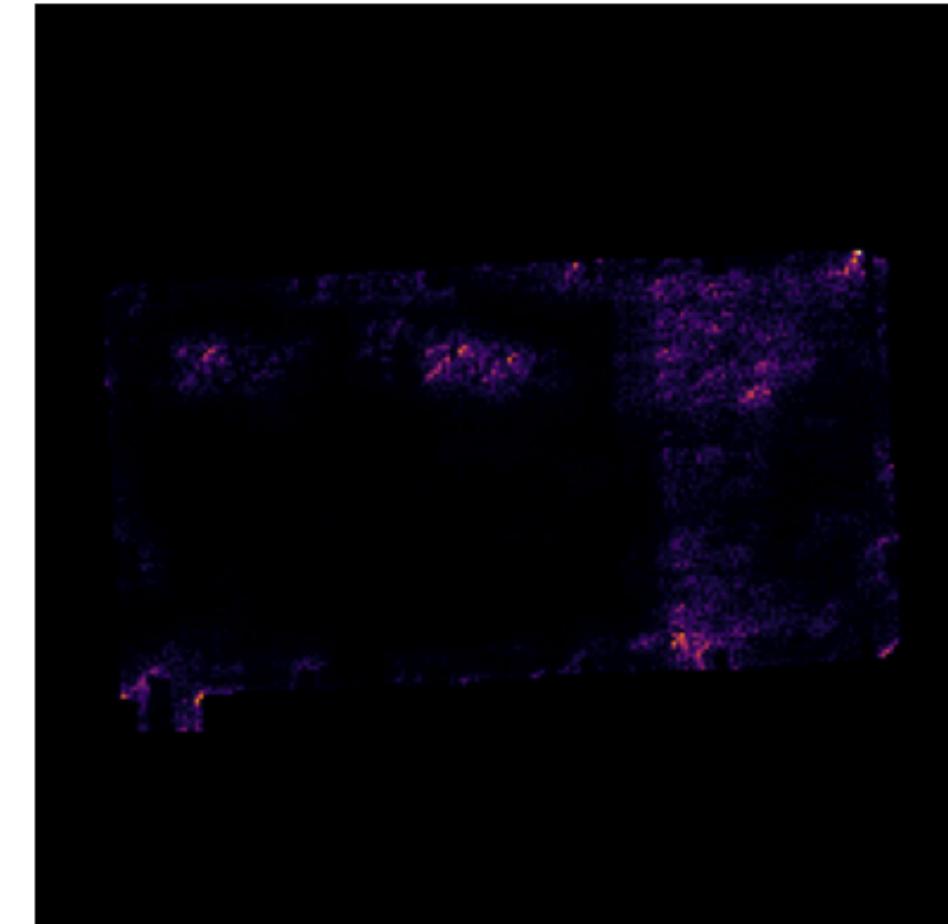
CNN - Arquitetura Própria Interpretadores

Análise de Interpretabilidade - Removed BG

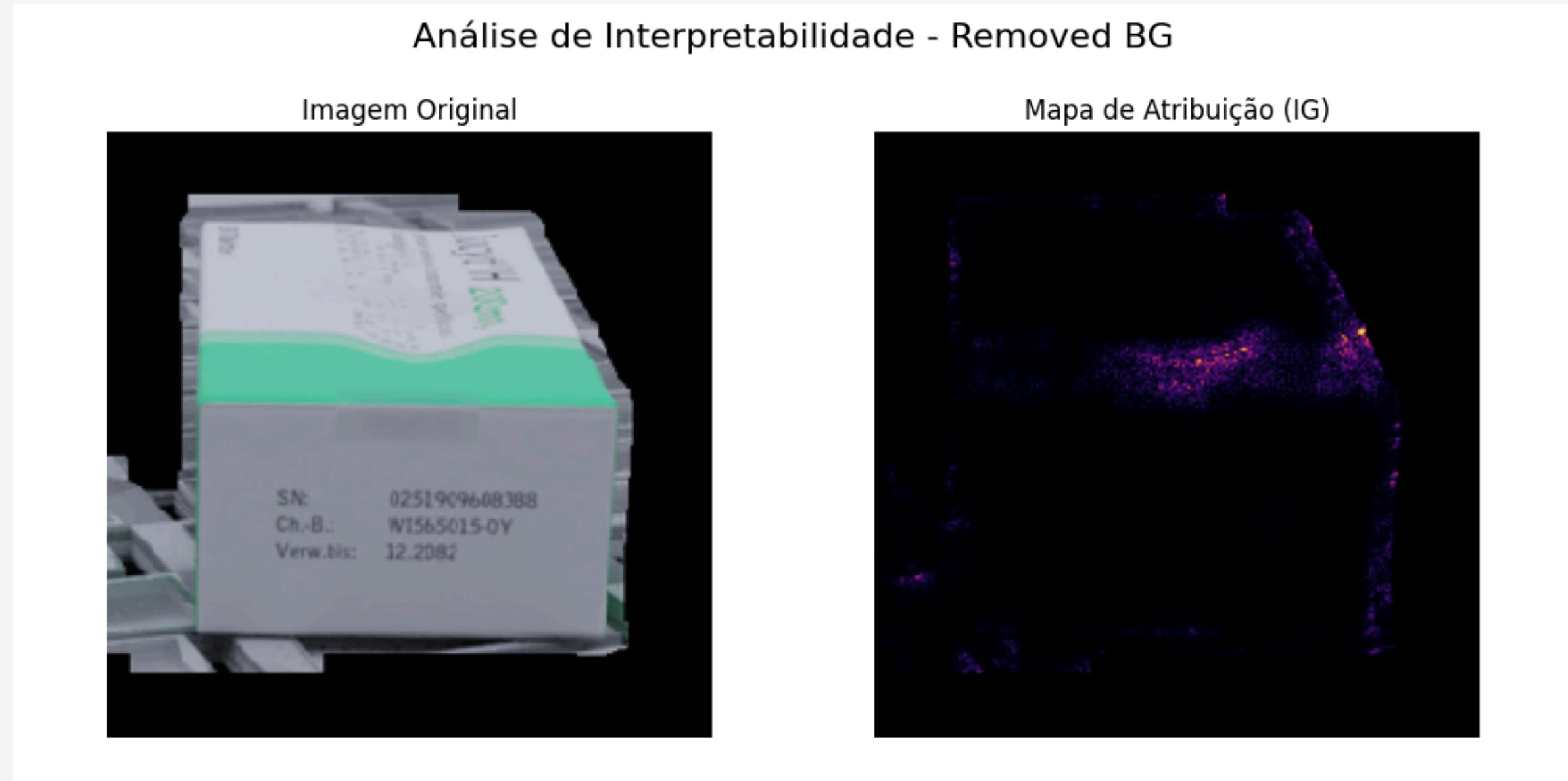
Imagen Original



Mapa de Atribuição (IG)



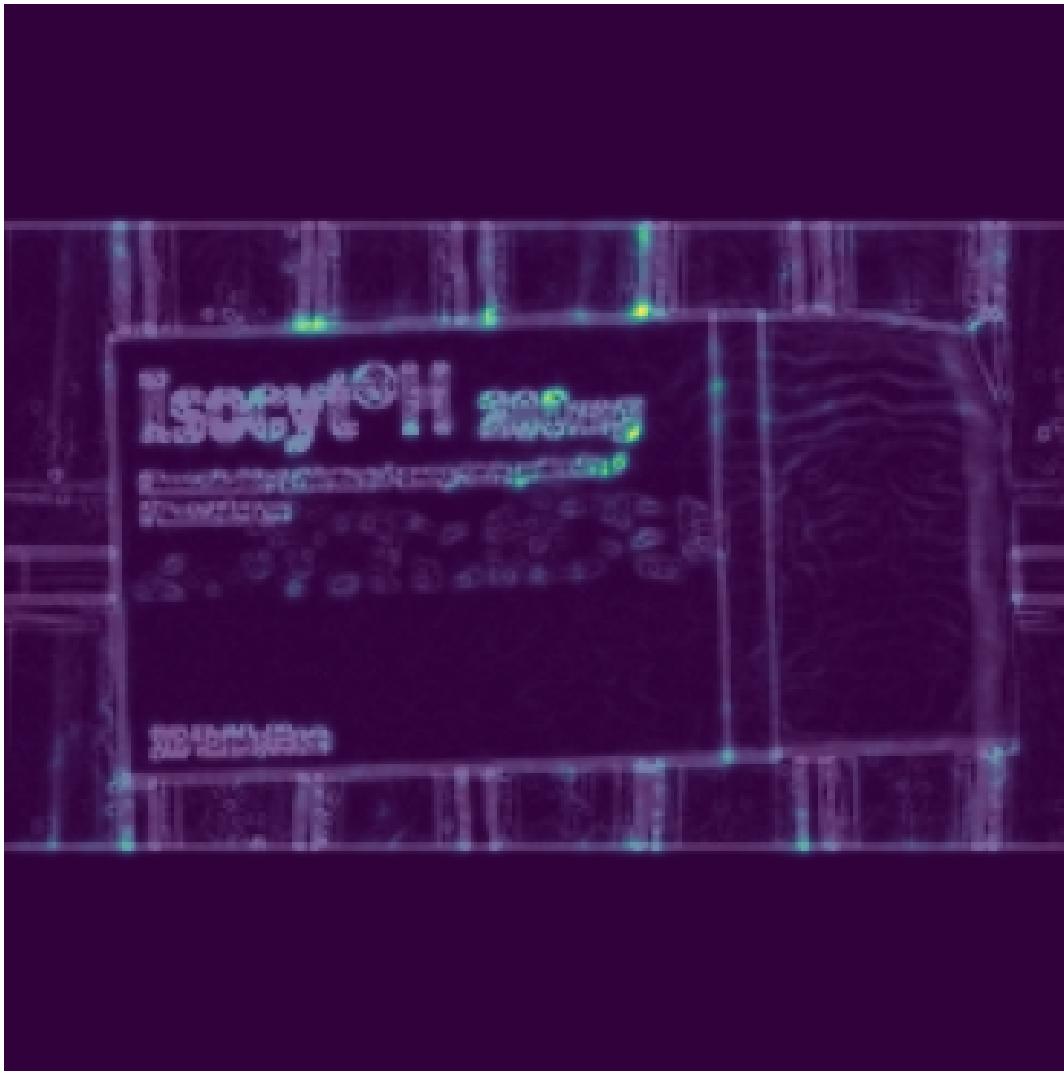
CNN - Arquitetura Própria Interpretadores



CNN - Arquitetura Própria Interpretadores



CNN - Arquitetura Própria Interpretadores



Transfer Learning

Arquitetura:

- Para o modelo top, foi utilizado como base o modelo VGG16
- Para o modelo side, foi utilizado como base o modelo EfficientNetB0
- Modelo congelado para evitar perder os pesos
- Adicionado camada de entrada
- Camada GlobalAveragePooling2D
- Saída com sigmoid para classificação binária.

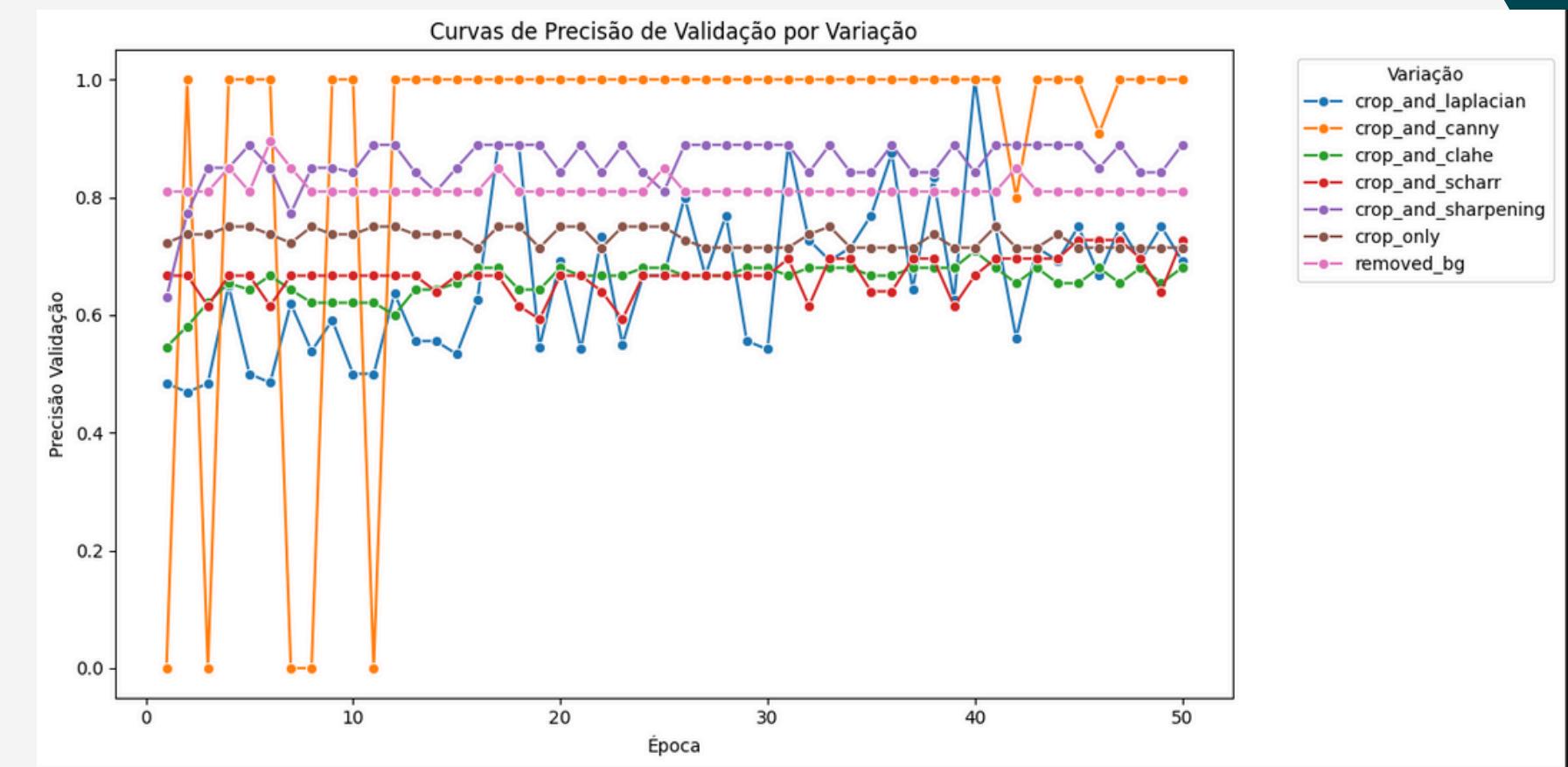
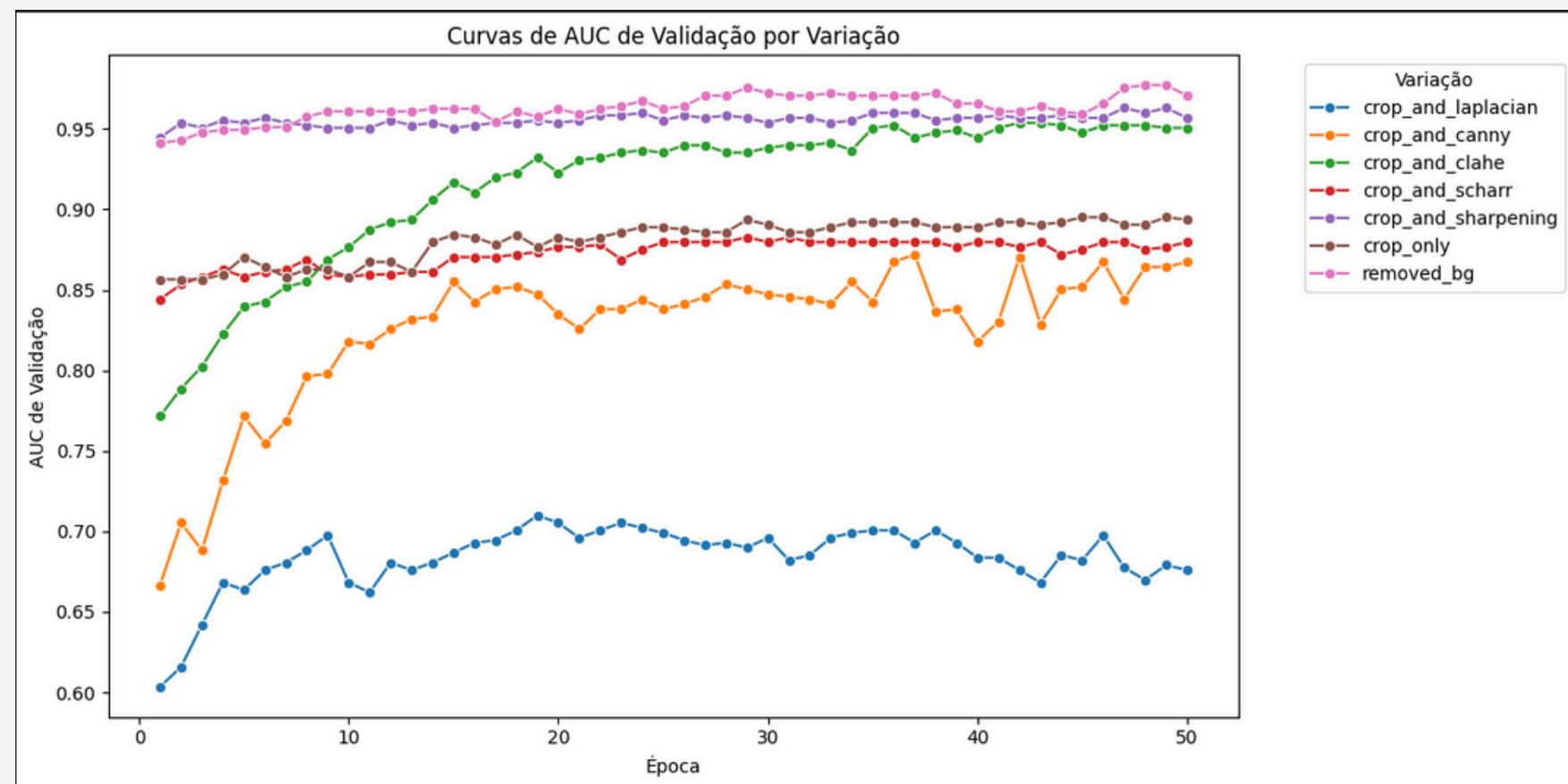
Pré-processamento e Dados

- Imagens redimensionadas para 224×224.
- Data augmentation leve (rotação, zoom, translação).
- Normalização por Sample_wise center .

Transfer Learning

- Side

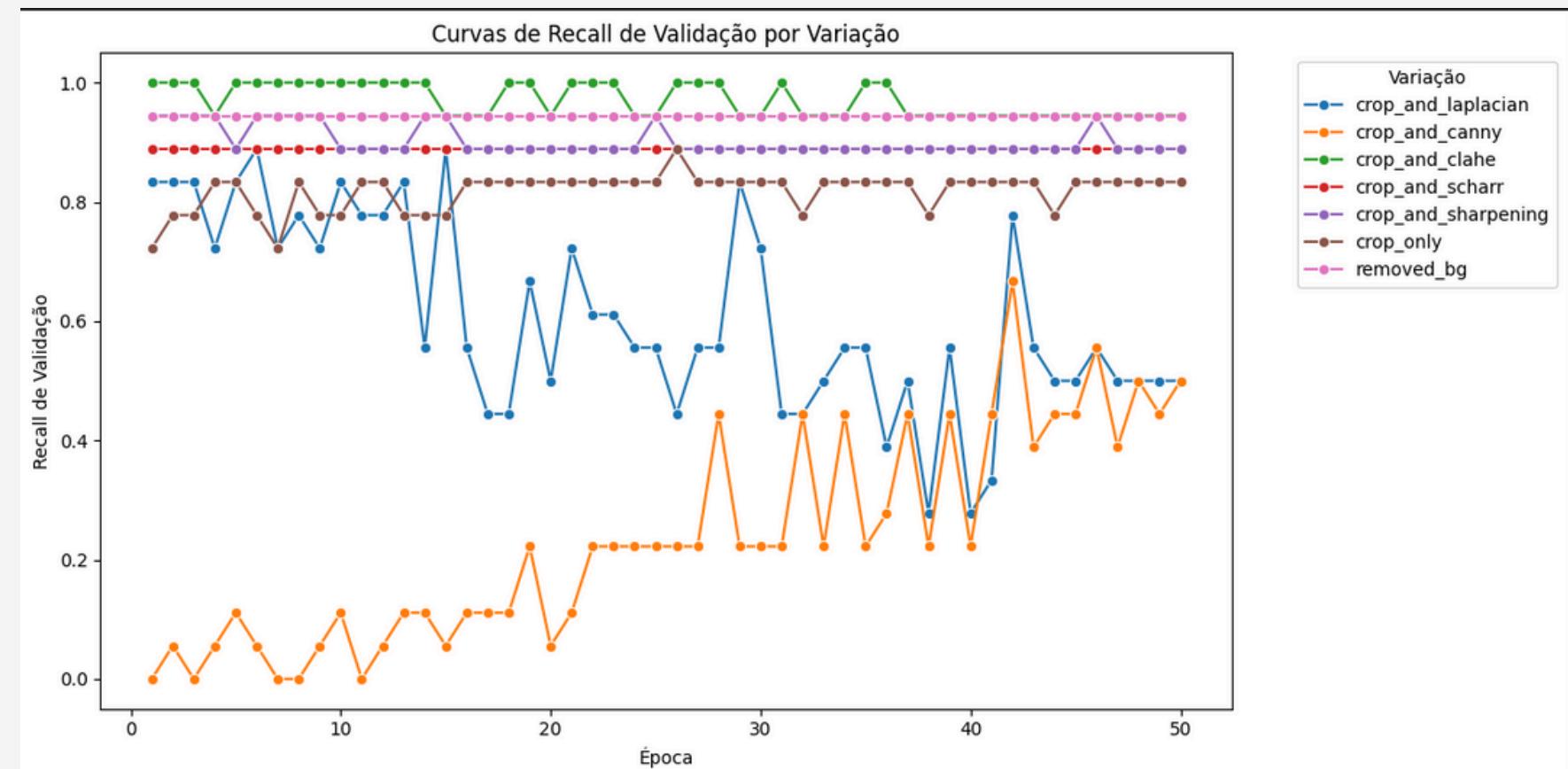
Comparando cada tipo de filtro:



Transfer Learning

- Side

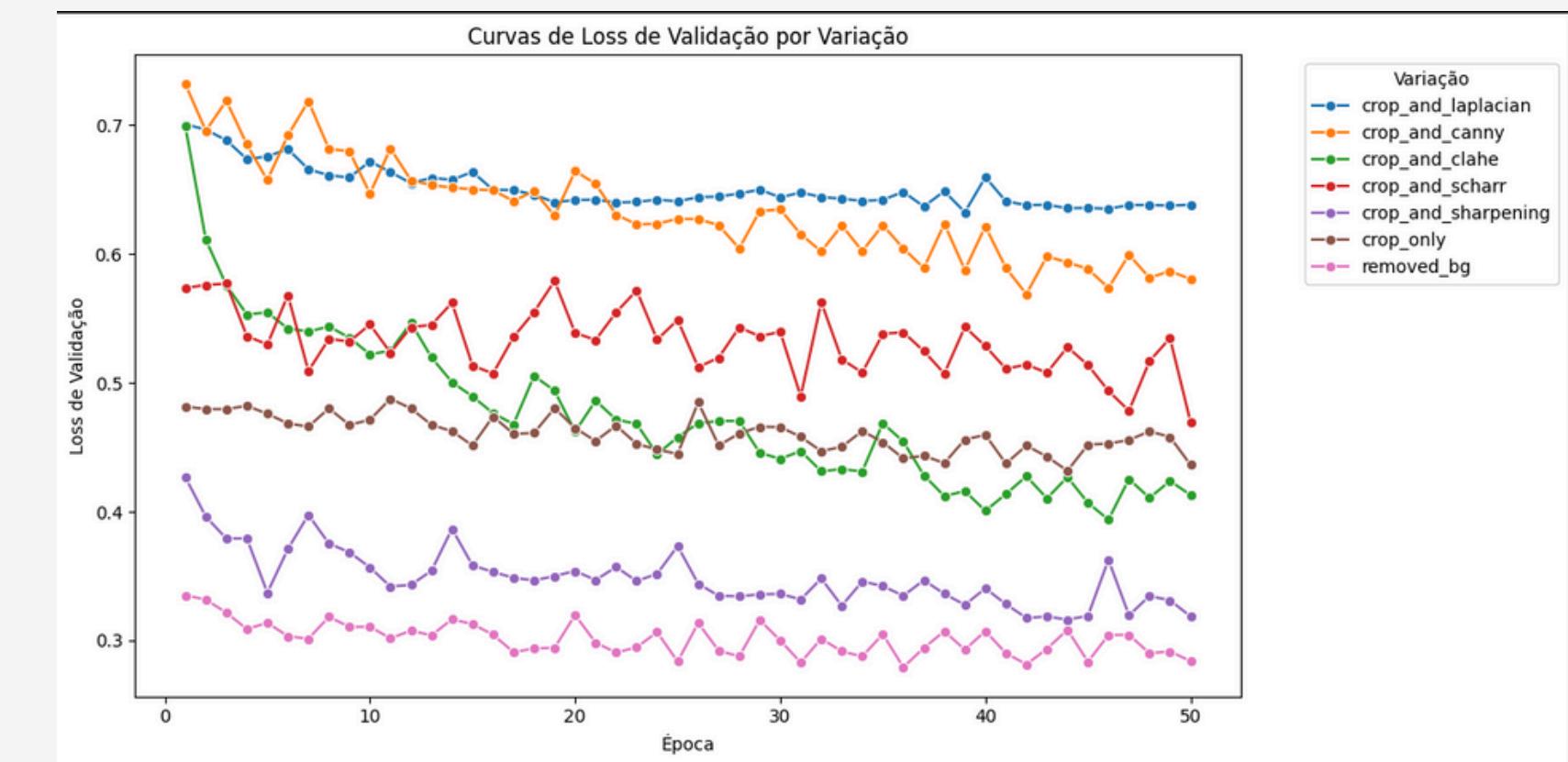
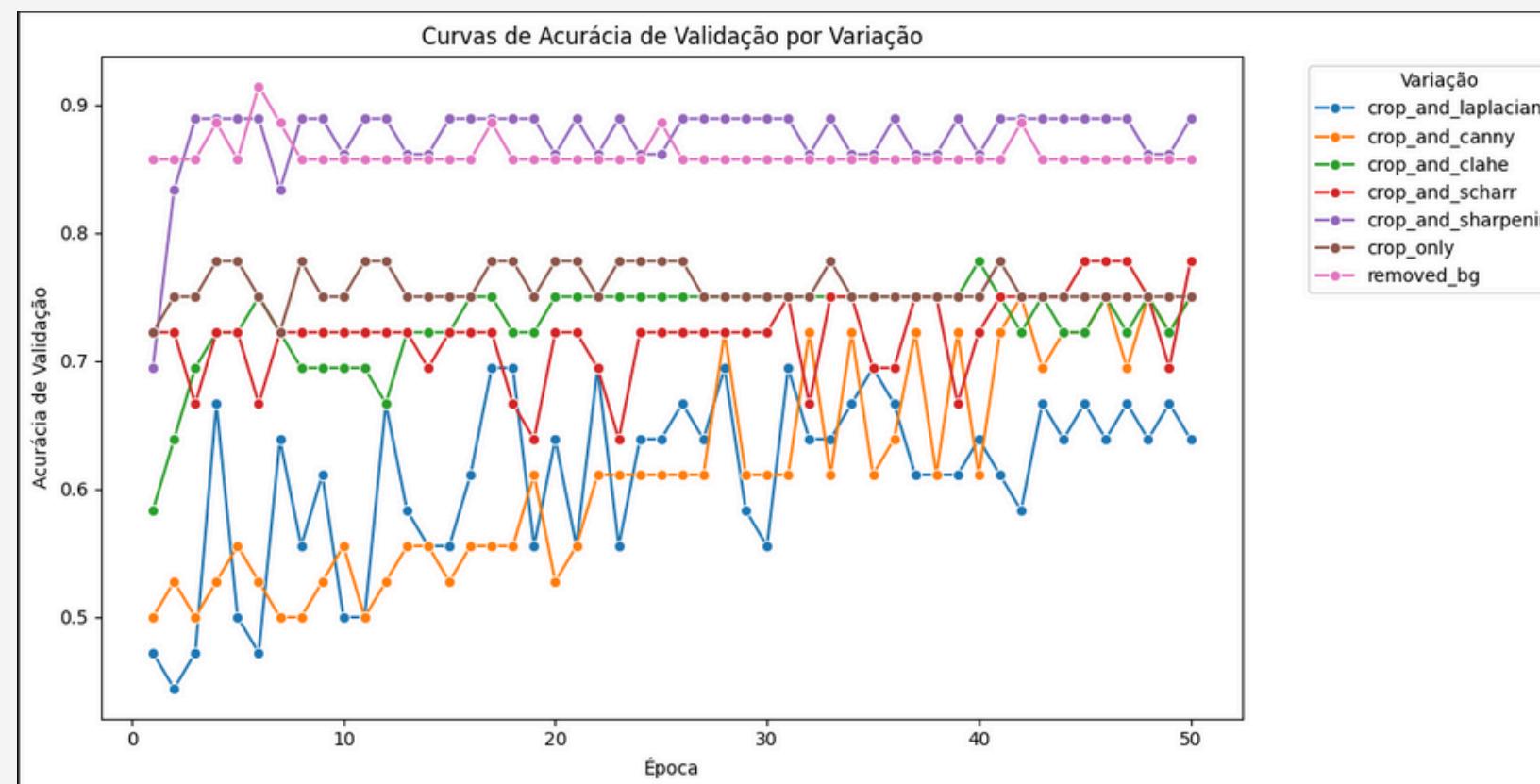
Comparando cada tipo de filtro:



Transfer Learning

- Side

Comparando cada tipo de filtro:



Transfer Learning

- Resultado

```
2023-07-11 07:01:50.525910: I tensorflow/core/tensorflow/service/gpu/dataloader/conv_algorithm_picker.cc:515] omitted potential  
1/1 ━━━━━━━━━━ 19s 19s/step - AUC: 0.9900 - Precision: 0.8333 - Recall: 1.0000 - accuracy: 0.9000 - loss: 0.3004
```

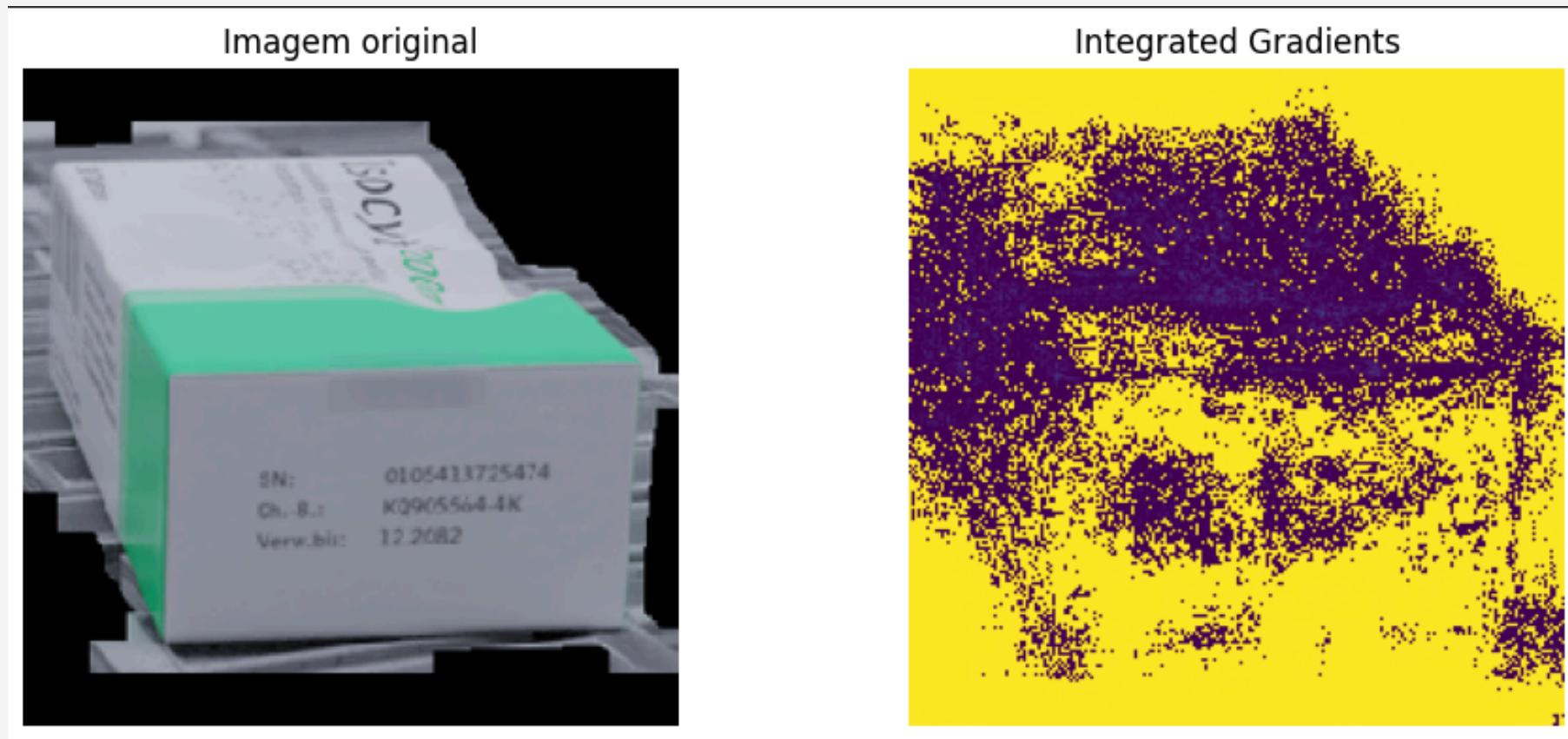
Como o Recall foi 100%, logo foram 2 falsos positivos

Transfer Learning

- Side

Analisando o que o modelo vê

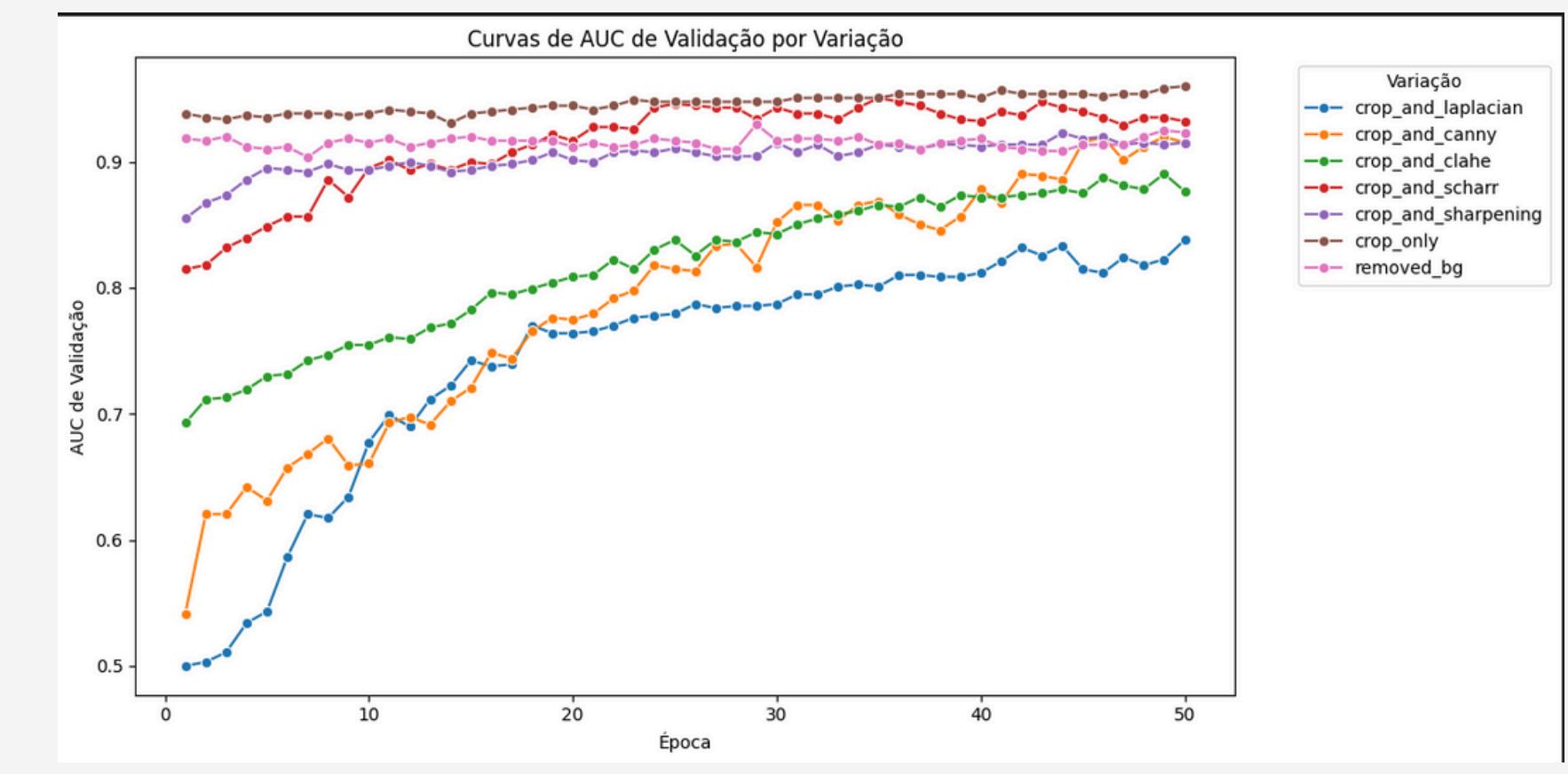
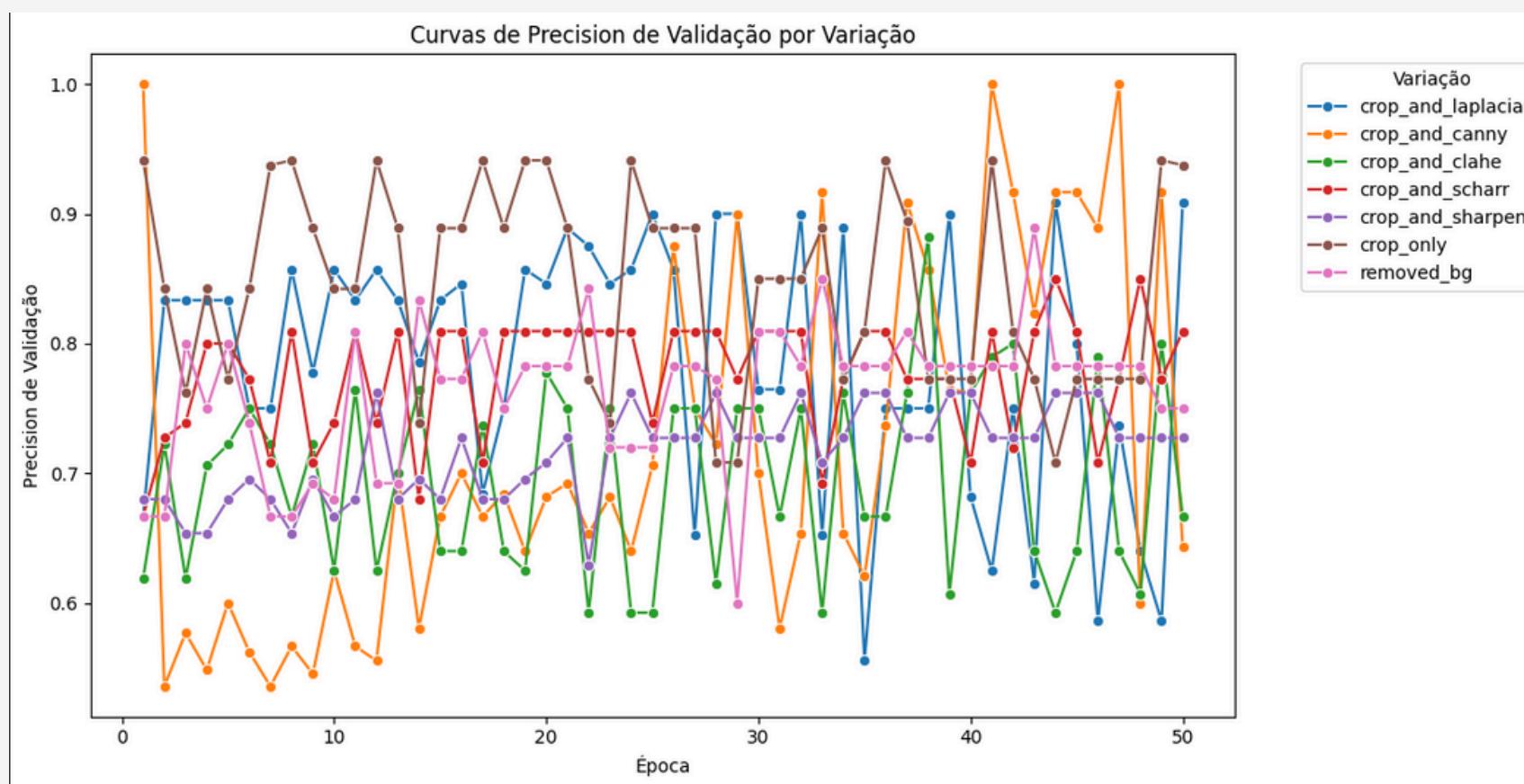
- Removed_bg



Transfer Learning

- Top

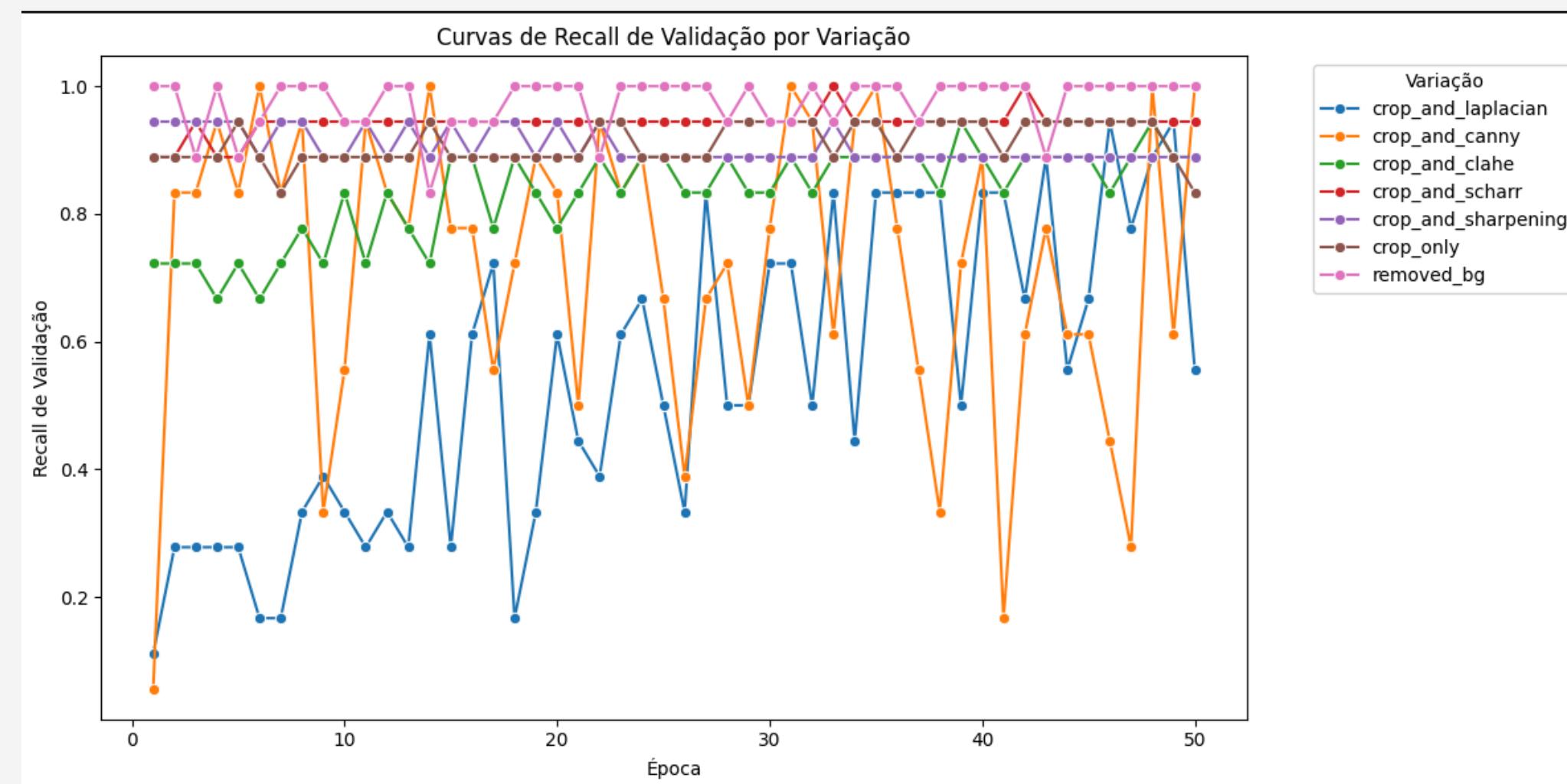
Comparando cada tipo de filtro:



Transfer Learning

- Top

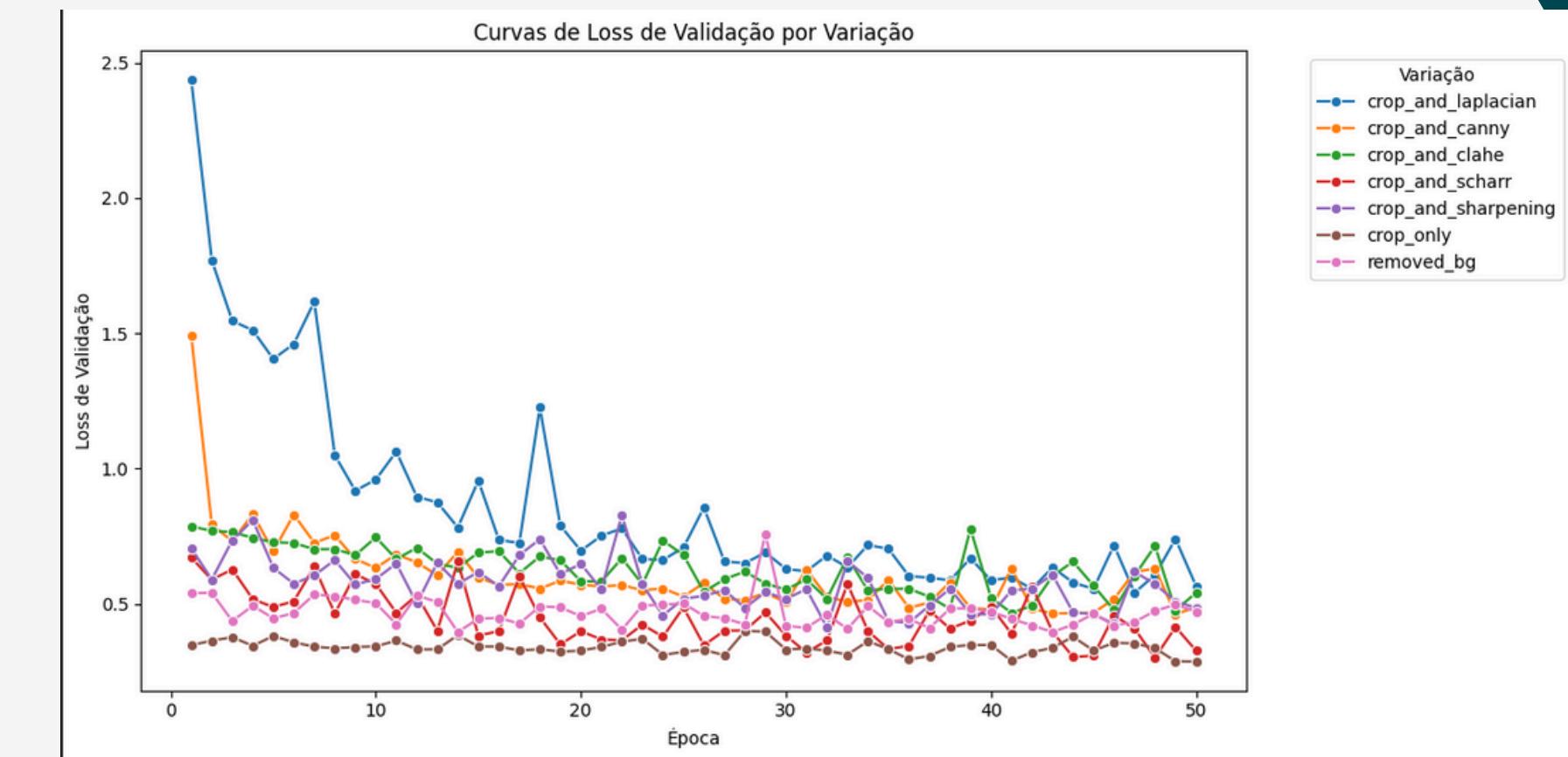
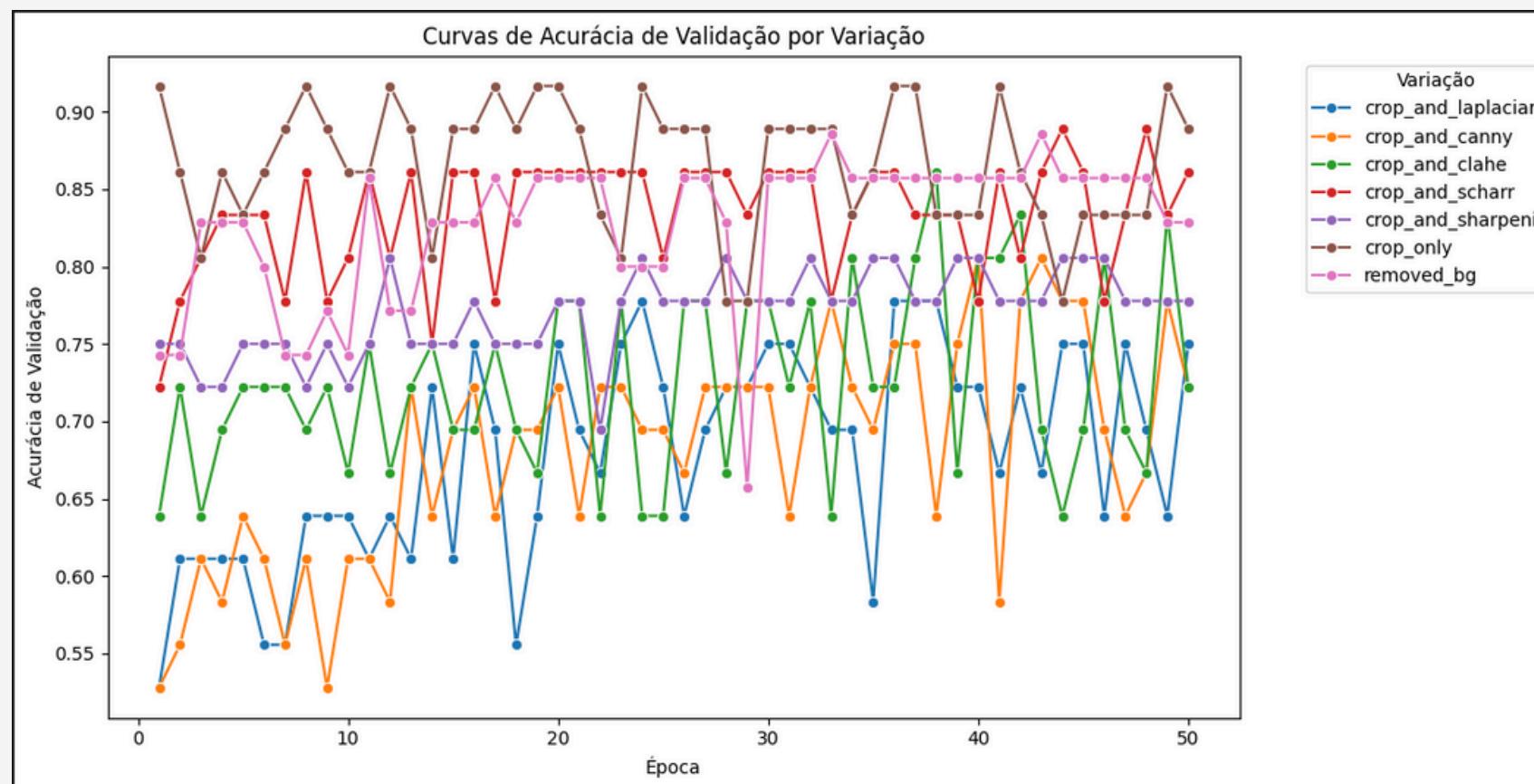
Comparando cada tipo de filtro:



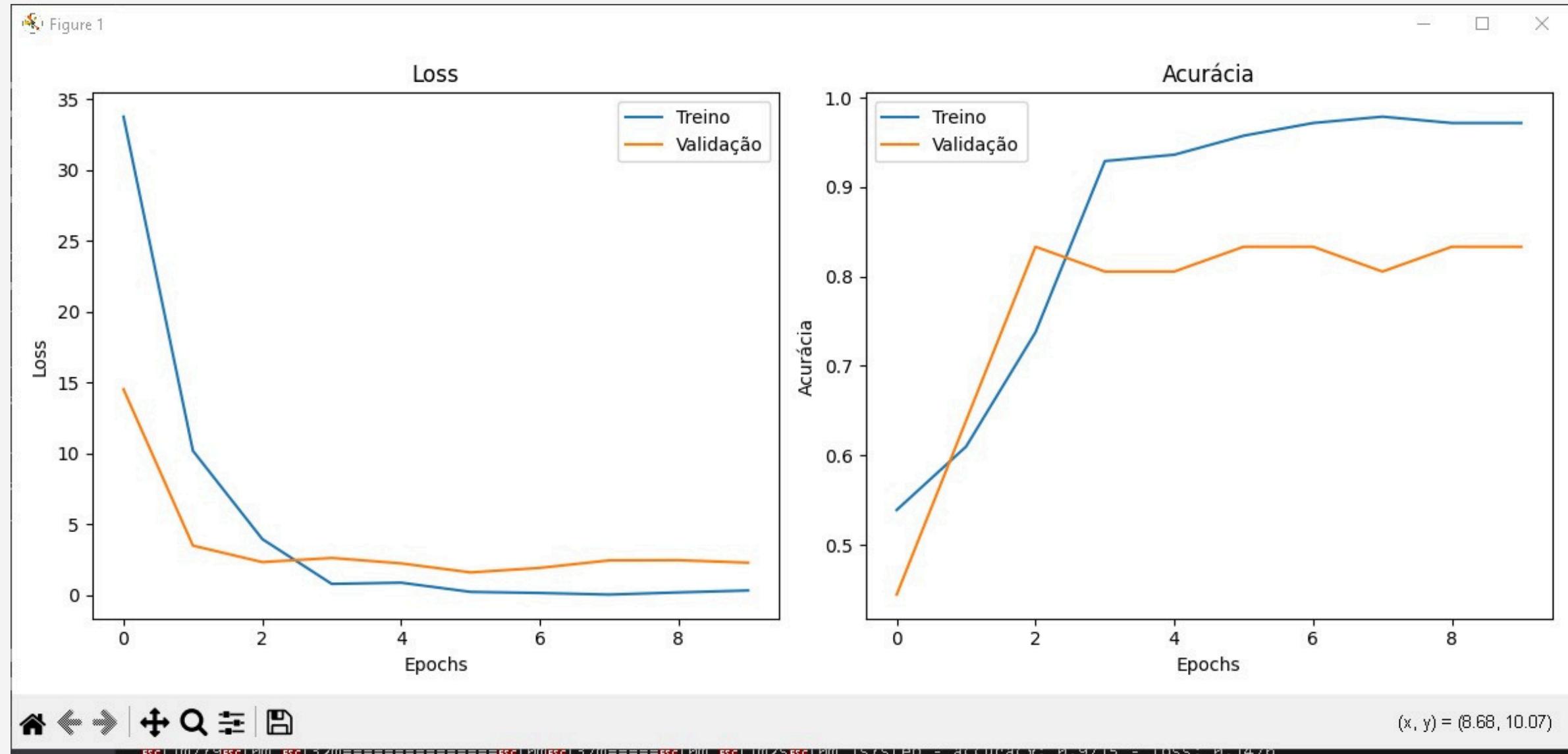
Transfer Learning

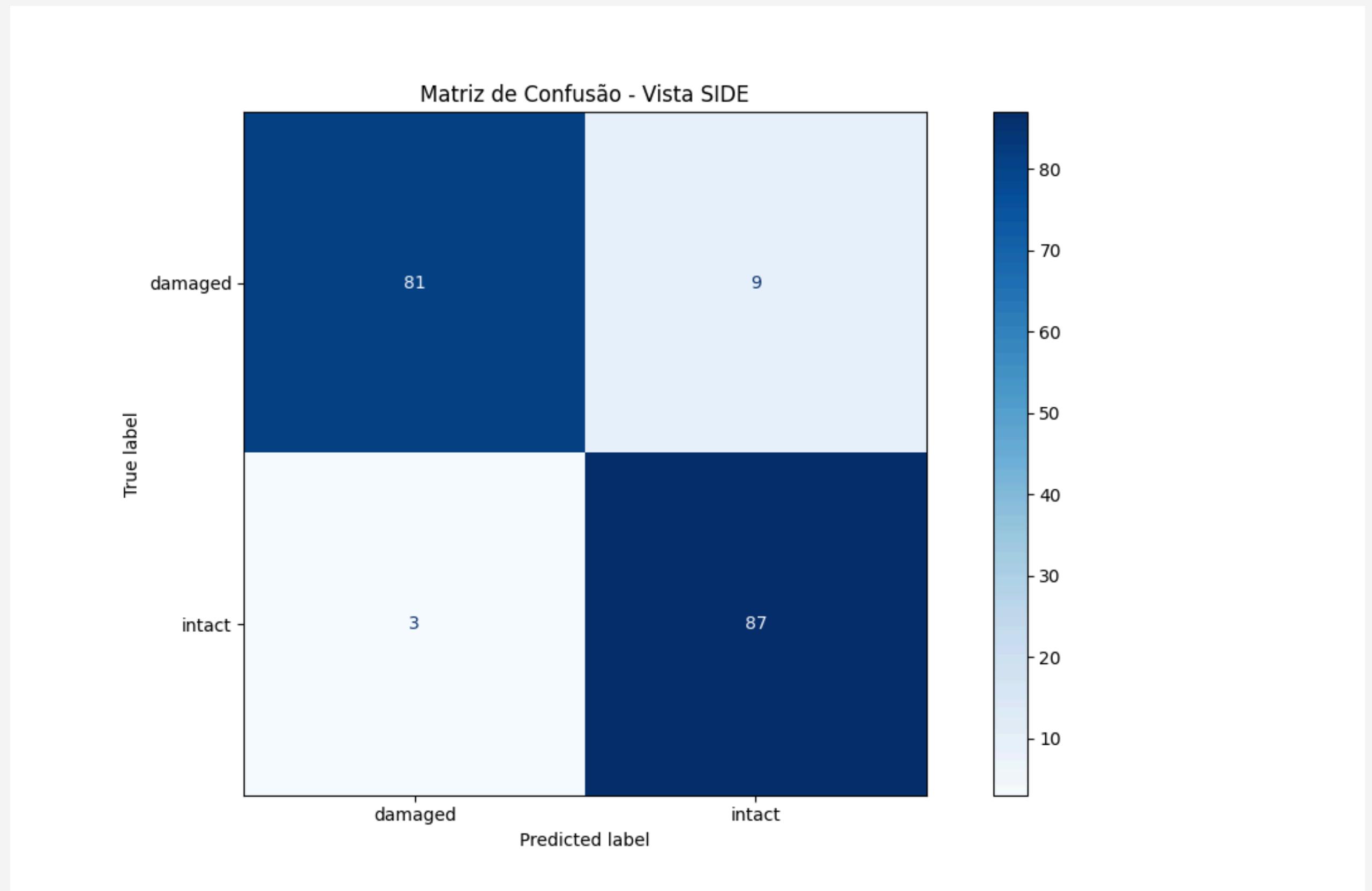
- Top

Comparando cada tipo de filtro:



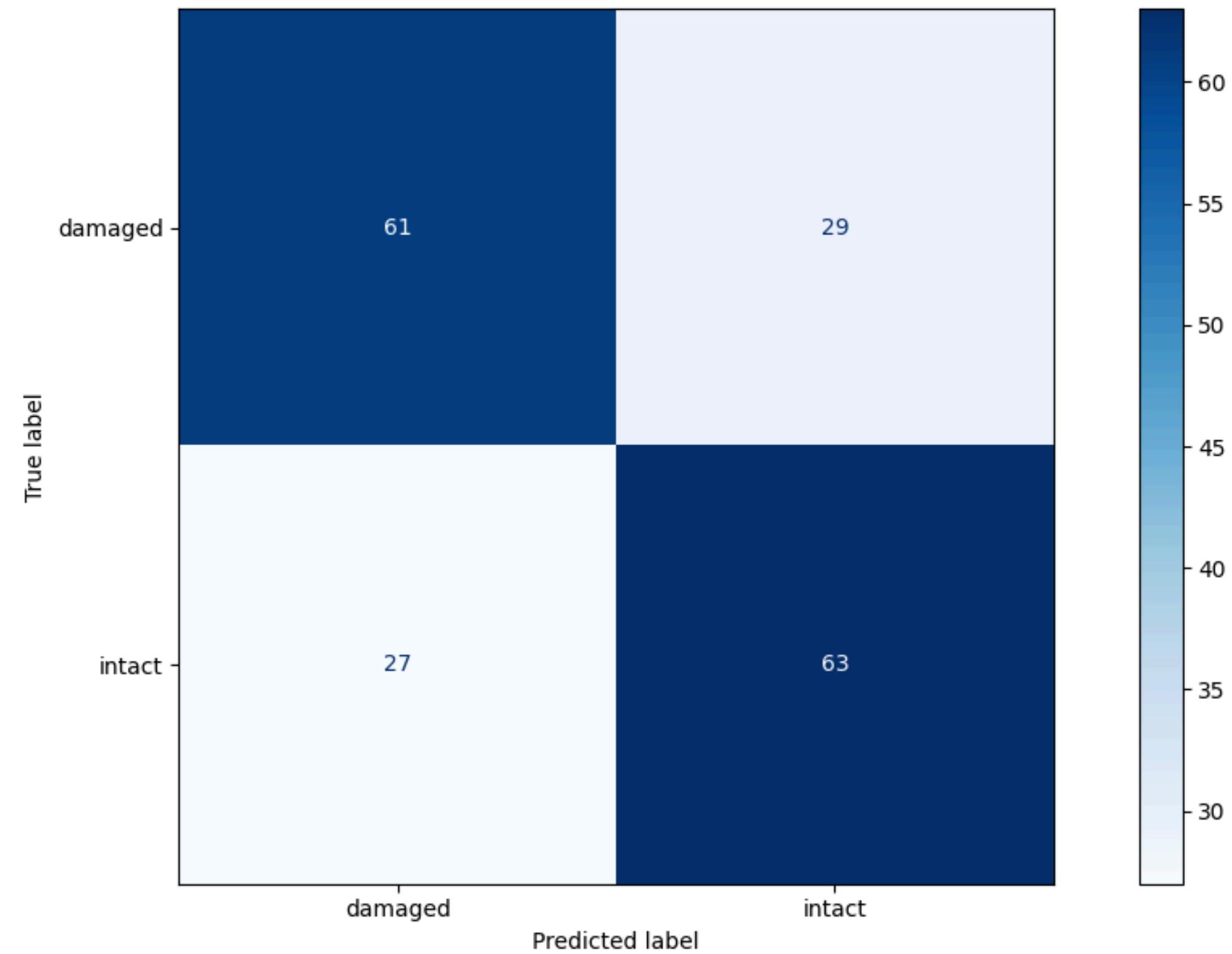
Transfer Learning







Matriz de Confusão - Vista TOP



Conclusão

- **Resultado**
 - O modelo 1 foi eficaz para encontrar bordas, e áreas de contraste
 - O modelo 2 funcionou melhor com fundo removido e focou bem nos detalhes superiores da caixa
 - O modelo de top teve uma acurácia baixa, possivelmente devido a similaridade nos dados