Classificação de Raio-X

Problema multi classe

Grupo

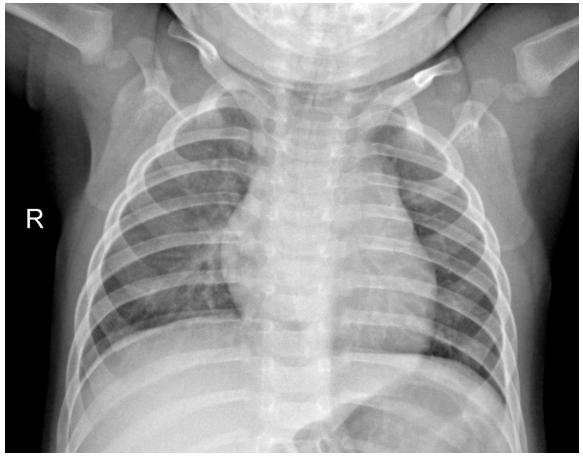
- Vitor Beneti Martins - nºUSP 11877635
- Gabriel Lima Alves - nºUSP 12558547
- João Pedro Rodrigues Freitas - nºUSP 11316552

Objetivo

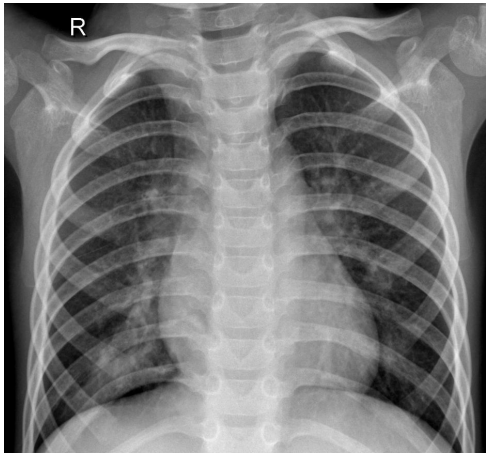
- Criar um modelo para classificar imagens de raio-x
  - Imagens normais
  - Imagens virais
  - Imagens covid
- 3 abordagens
  - CNN simples
  - ResNet18
  - ViT

Exemplos de imagem

Normal



Vírus



Covid

Pré-processamento

- Padronização dos tamanhos das imagens: 224x224
- Imagens com 3 canais
  - algumas contém tons de azul
- Dataset pequeno: 1196 imagens
  - Geração de imagens por meio do espelhamento
- Total de 2392 imagens
- Classes ligeiramente desbalanceadas
  - Covid (classe 0): 590 imagens
  - Normal (classe 1): 936 imagens
  - Vírus (classe 2): 866 imagens

Pré-processamento

- Divisão
  - Treino 70%
    - Covid (Classe 0): 416 imagens
    - Normal (Classe 1): 647 imagens
    - Vírus (Classe 2): 611 imagens
  - Validação 15%
    - Covid: 90 imagens
    - Normal: 136 imagens
    - Vírus: 132 imagens
  - Teste 15%
    - Covid: 84 imagens
    - Normal: 153 imagens
    - Vírus: 123 imagens

Treinamento Geral

- Otimizador: Adam
  - lr = 0.001
  - Taxa de aprendizado adaptativa
- Loss Function: Cross Entropy Loss
  - Adequado para classificação multi-classe

Baseline

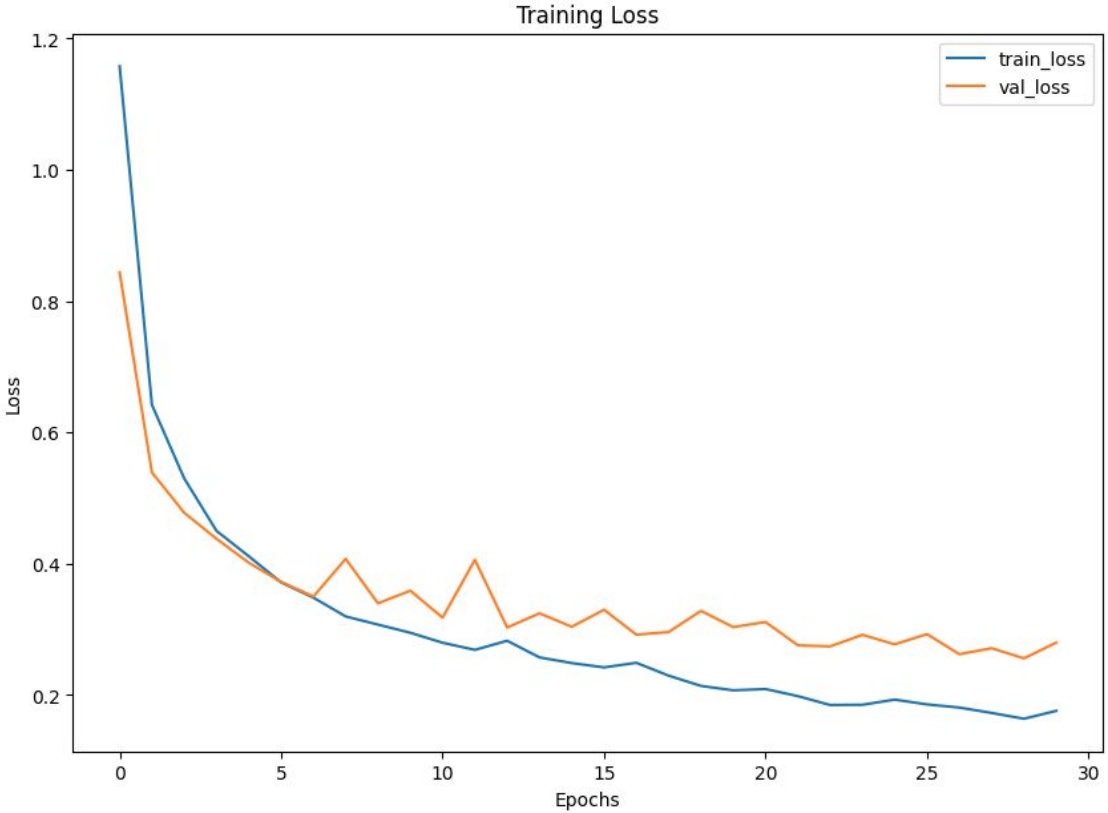
Baseline - Modelo - Features

- Entrada 3x224x224
-
- nn.**Conv2d**(3, 8, kernel_size=3, stride=1, padding=1),
- nn.**BatchNorm2d**(8),
- nn.**ReLU**(),
- nn.**MaxPool2d**(kernel_size=2, stride=2),
-
- nn.**Conv2d**(8, 16, kernel_size=3, stride=1, padding=1),
- nn.**BatchNorm2d**(16),
- nn.**ReLU**(),
- nn.**MaxPool2d**(kernel_size=2, stride=2),

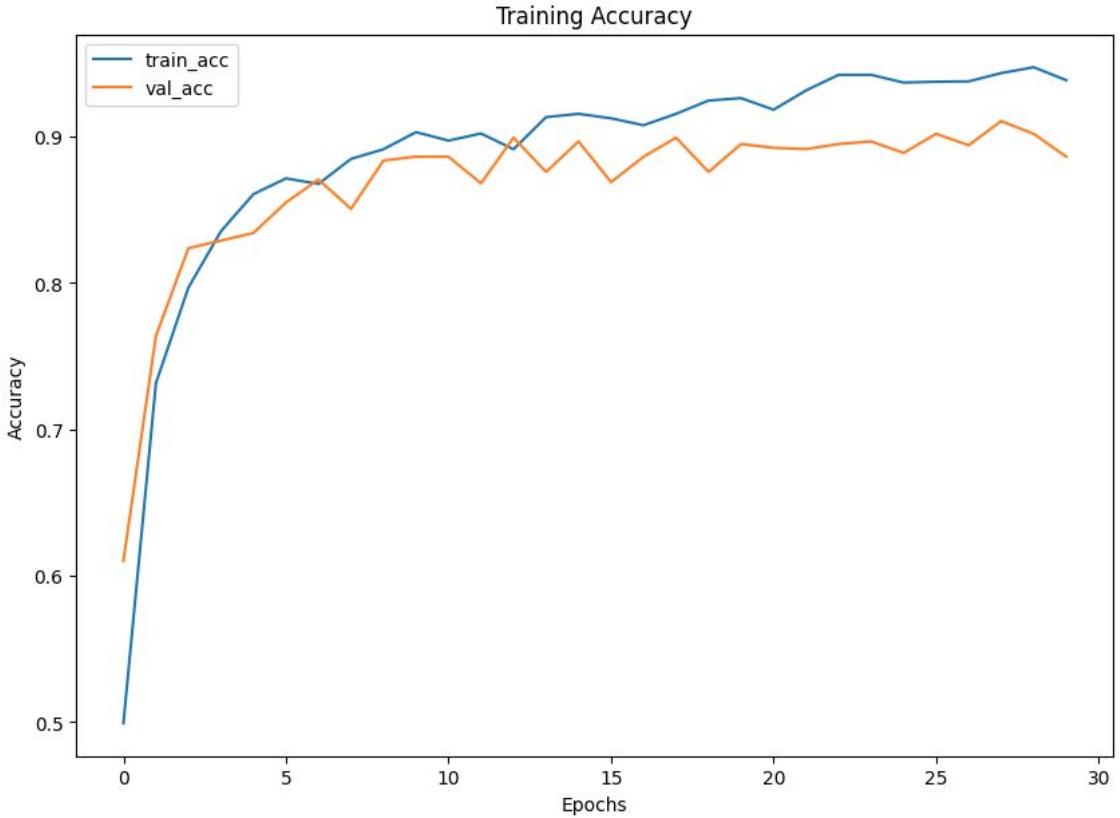
Baseline - Modelo - Classificação

- Entrada 16x56x56
-
- nn.**AdaptiveMaxPool2d**((1, 1)), # Global Max Pooling
- nn.**Flatten**(),
- nn.**Linear**(16, num_diseases)
-
- Saída: 3 classes possíveis

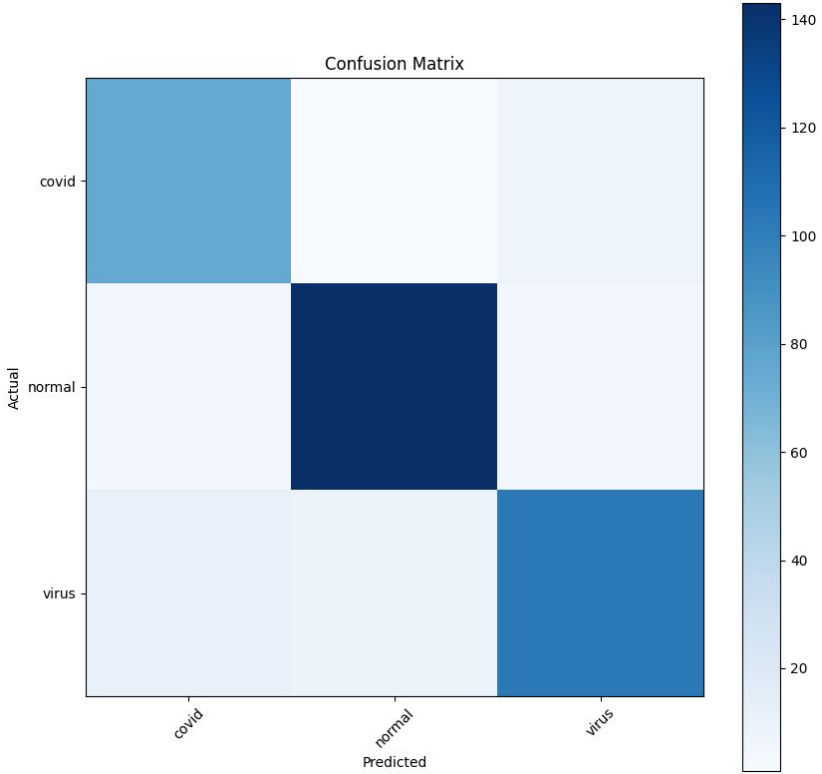
Resultados - Loss



Resultados - Acurácia



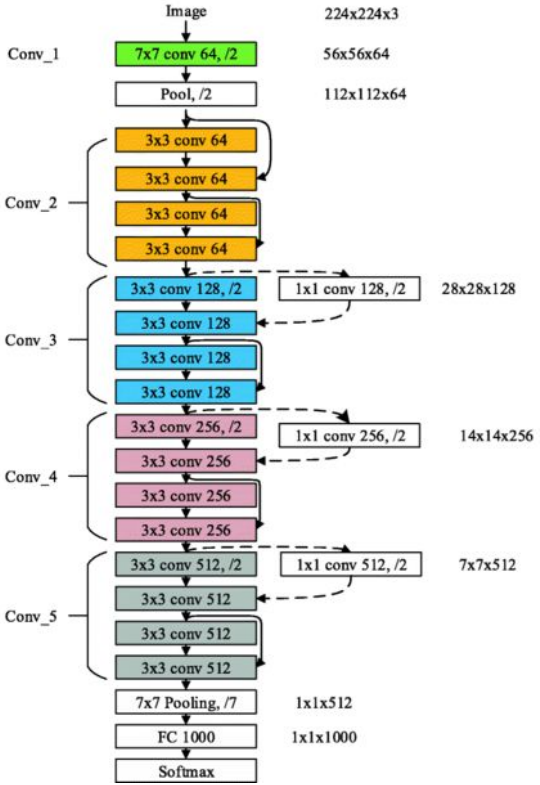
Resultados - Matriz de Confusão e Métricas

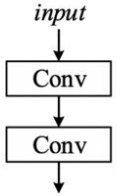


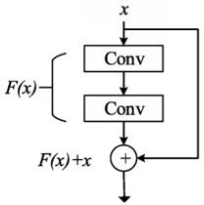
	precision	recall	f1-score	support
covid	0.82	0.89	0.86	84
normal	0.93	0.93	0.93	153
virus	0.89	0.84	0.86	123
accuracy			0.89	360
macro avg	0.88	0.89	0.88	360
weighted avg	0.89	0.89	0.89	360

Resnet18

ResNet - overview







Resnet - Modelo - Features

- **Camada de entrada**
  - **conv1:** Convolução com 64 filtros de tamanho 7x7, stride 2, padding 3. Reduz a resolução da imagem de entrada de 224x224 para 112x112.
  - **bn1:** Normalização em Batch (Batch Normalization) para a saída da convolução inicial.
  - **relu:** Função de ativação ReLU.
  - **maxpool:** MaxPooling com kernel de 3x3 e stride 2. Reduz a resolução da imagem para 56x56.

Resnet - Modelo - Features

- **Blocos Residuais**
  - **Convolução 1 (conv1):** Convolução com filtros 3x3.
  - **BatchNorm 1 (bn1):** Normalização em Batch após a primeira convolução.
  - **ReLU (relu):** Função de ativação ReLU.
  - **Convolução 2 (conv2):** Convolução com filtros 3x3.
  - **BatchNorm 2 (bn2):** Normalização em Batch após a segunda convolução.
  - **Skip Connection:** Adiciona a entrada original ao resultado das duas convoluções anteriores

Resnet - Modelo - Features

- ## Camadas Residuais
  - **A resnet18 possui 4 camadas residuais sendo cada uma delas contendo 2 blocos residuais**
  - **Camada 1:** Ambos os blocos têm 64 filtros e mantêm a resolução de 56x56.
  - **Camada 2:** O primeiro bloco reduz a resolução para 28x28 (stride 2) e aumenta os filtros para 128. O segundo bloco mantém a resolução de 28x28.
  - **Camada 3:** O primeiro bloco reduz a resolução para 14x14 (stride 2) e aumenta os filtros para 256. O segundo bloco mantém a resolução de 14x14.
  - **Camada 4:** O primeiro bloco reduz a resolução para 7x7 (stride 2) e aumenta os filtros para 512. O segundo bloco mantém a resolução de 7x7.

Resnet - Modelo - Features

- **Camada de Classificação:**
  - `avgpool`**:** Pooling adaptativo para 1x1, reduzindo a resolução da imagem para 1x1, essencialmente extraindo a média de cada filtro.
  - `fc`**:** Camada totalmente conectada (fully connected) que mapeia os 512 filtros finais para as 3 classes de saída.

Resnet - Características

Vantagens:

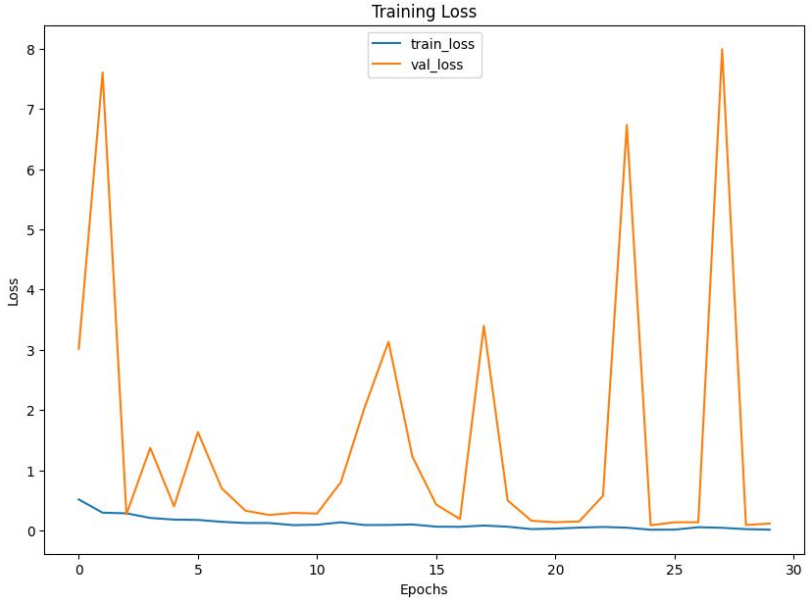
- Mitiga o problema de degradação em redes profundas (gradiente desaparecendo), permite treinamento de redes mais profundas
- Preservação de informações cruciais durante o treinamento, resultando em um desempenho aprimorado e maior capacidade de aprendizado

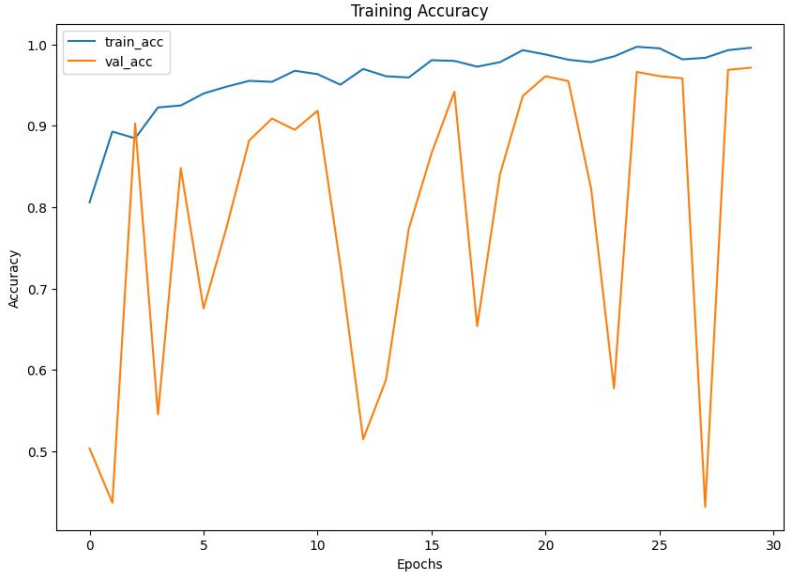
Desvantagens:

- Potencial para overfitting maior em banco de dados menores
- Aumento do consumo de memória e recursos computacionais devido à profundidade da rede

Resnet - Resultados

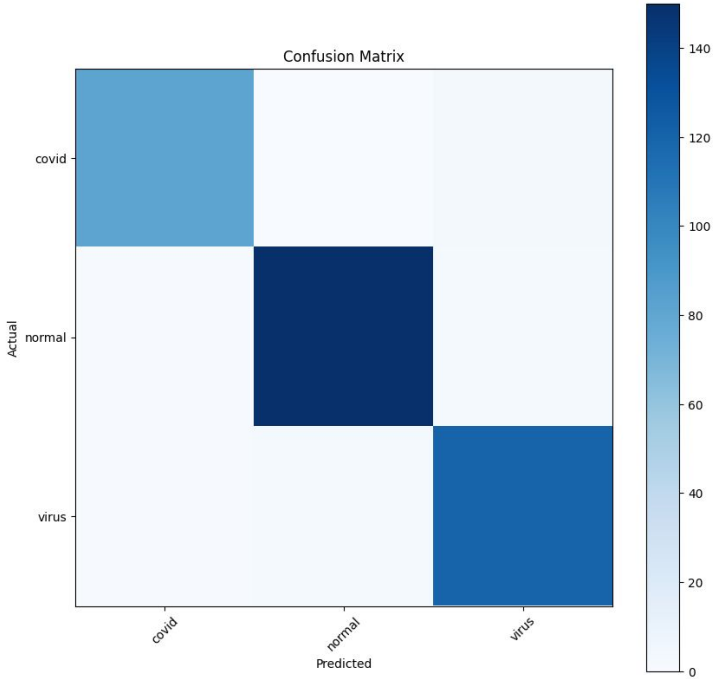
* Treinado por 30 épocas





Resnet - Resultados

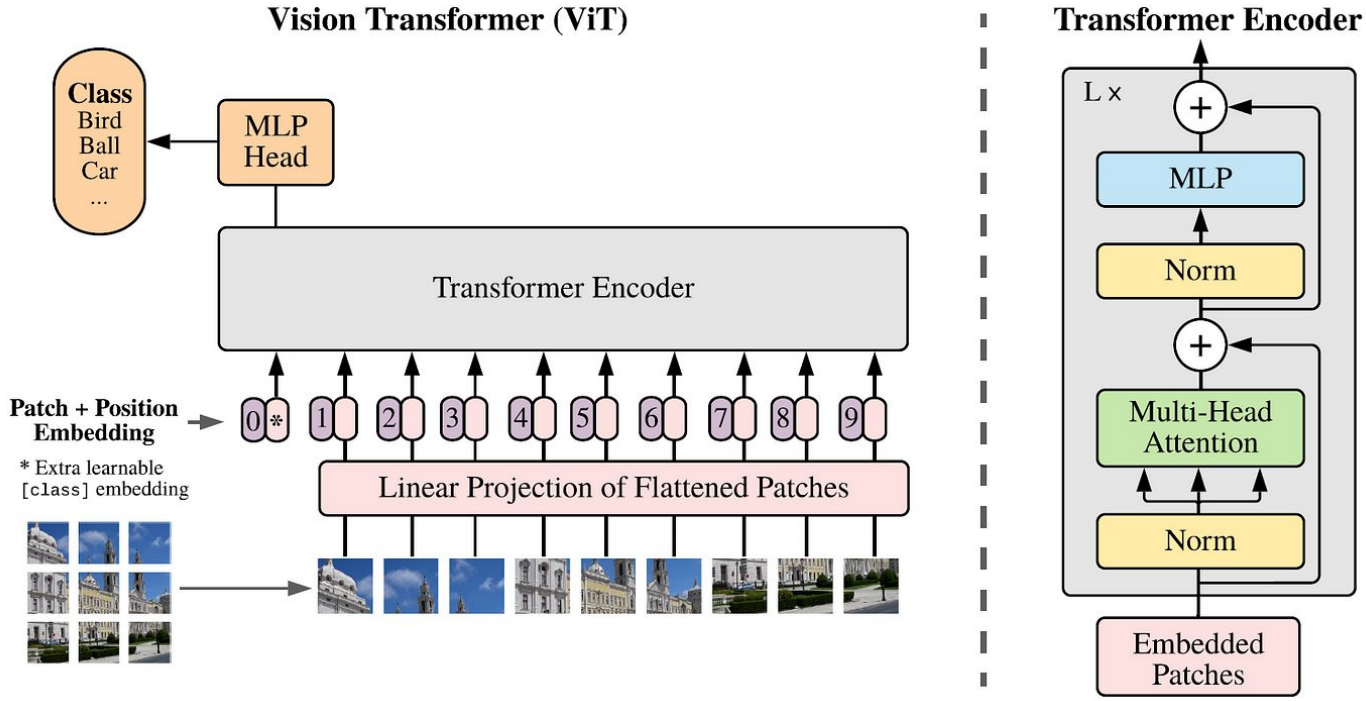
* Treinado por 30 épocas



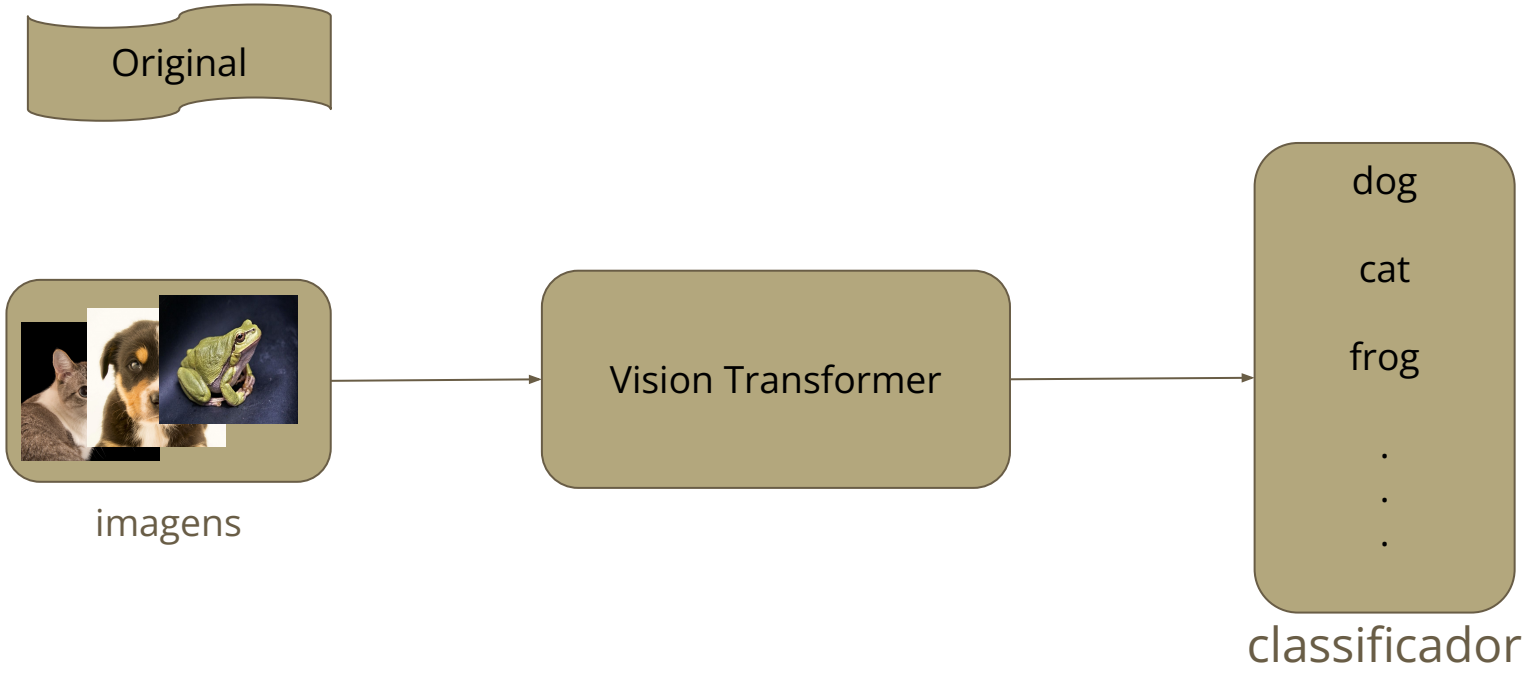
	precision	recall	f1-score	support
covid	0.98	0.96	0.97	84
normal	0.99	0.98	0.98	153
virus	0.96	0.98	0.97	123
accuracy			0.97	360
macro avg	0.97	0.97	0.97	360
weighted avg	0.98	0.97	0.98	360

Vision Transformer

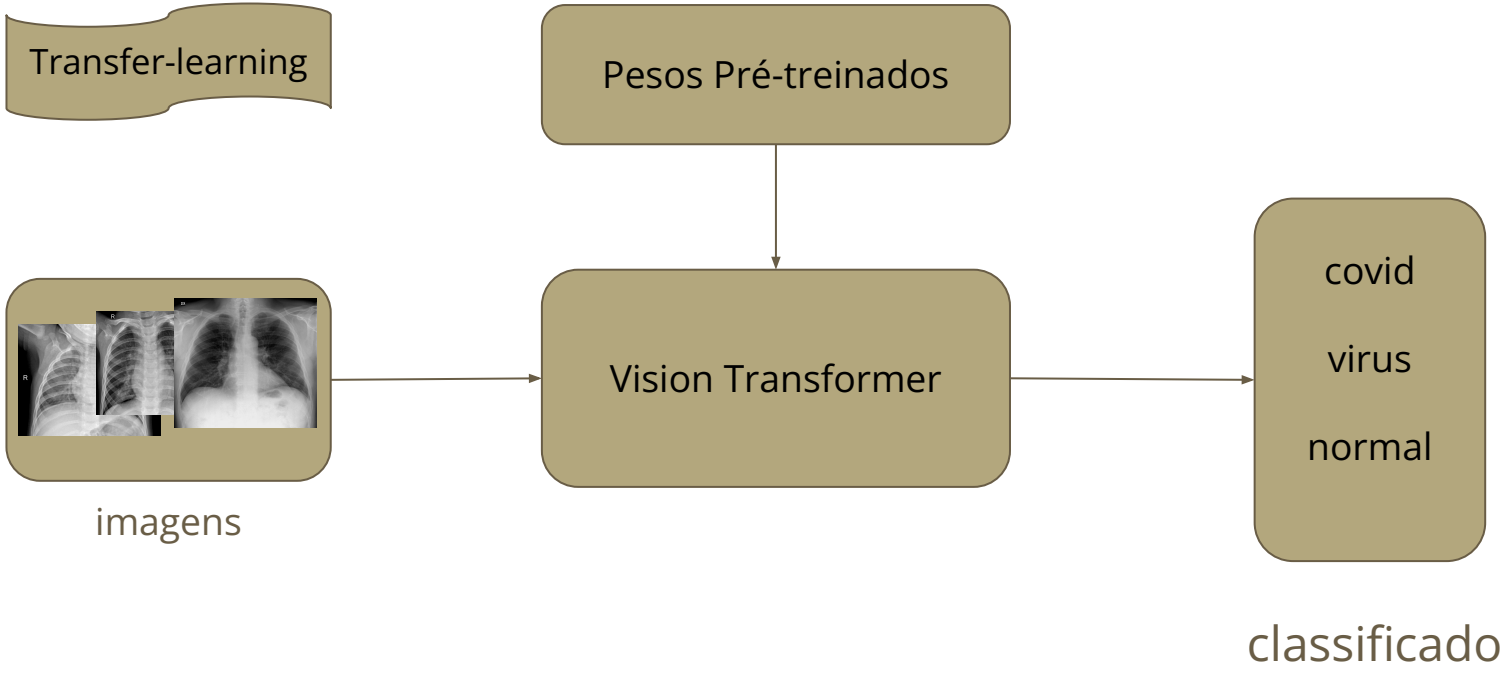
ViT - overview



ViT - transfer learning



ViT - transfer learning

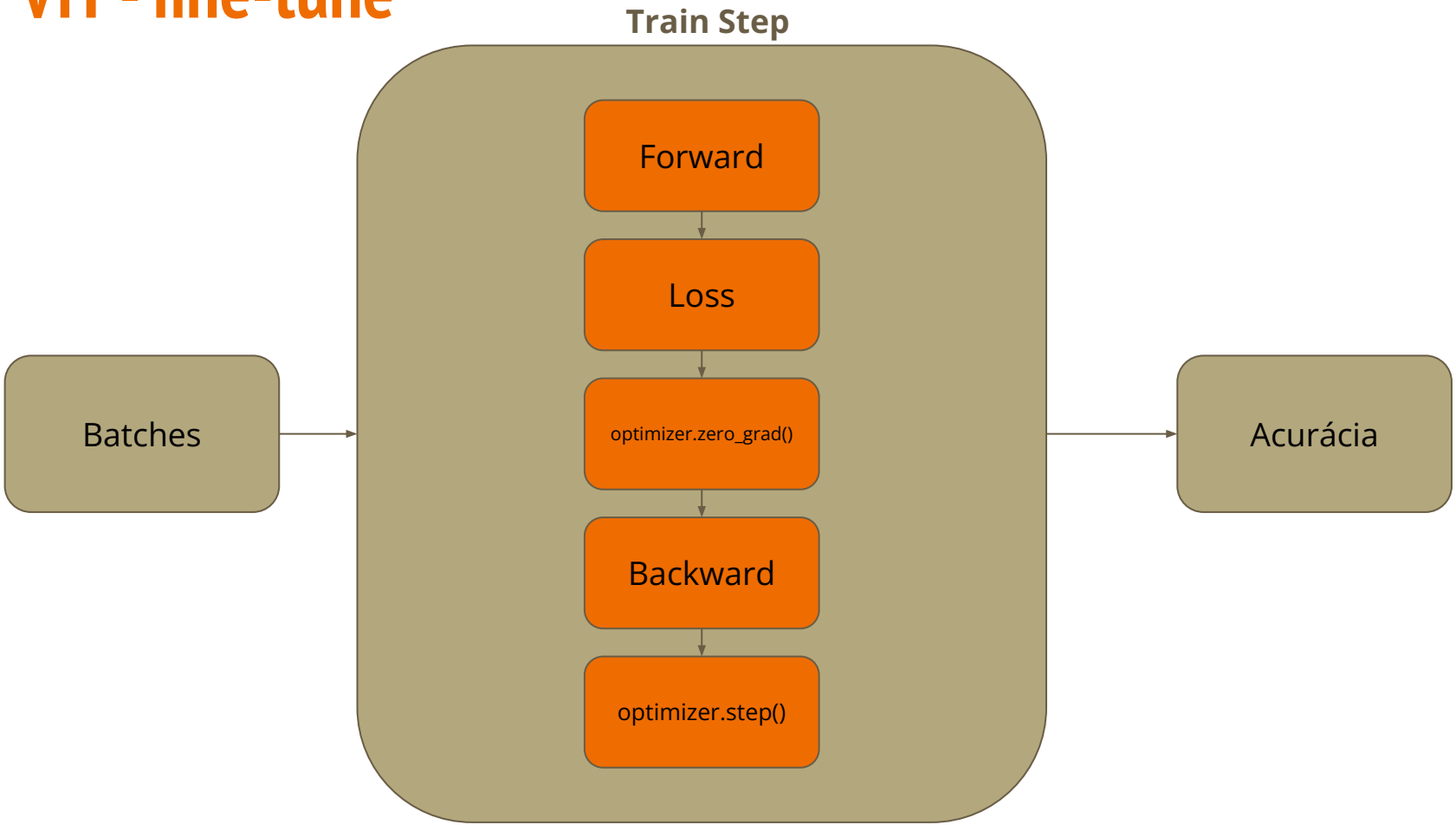


ViT - transfer learning

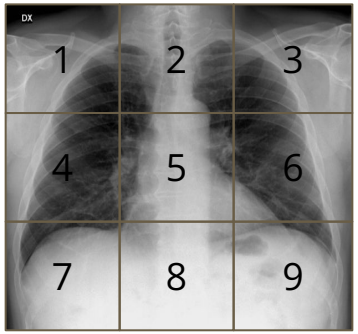
Vantagens:

- Economia de tempo e recursos
- Performance melhorada
- Facilidade de implementação
- Menor risco de overfitting

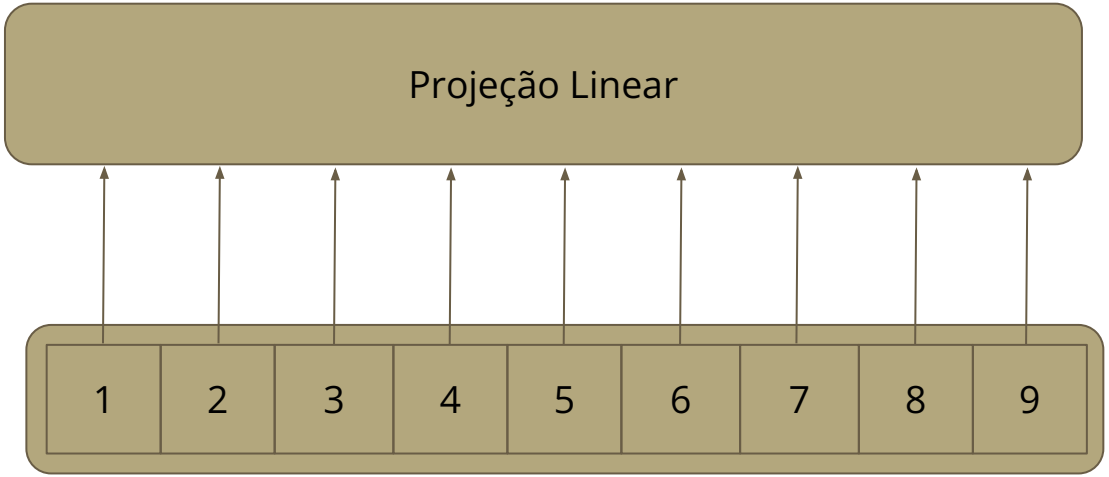
ViT - fine-tune



ViT - fine-tune

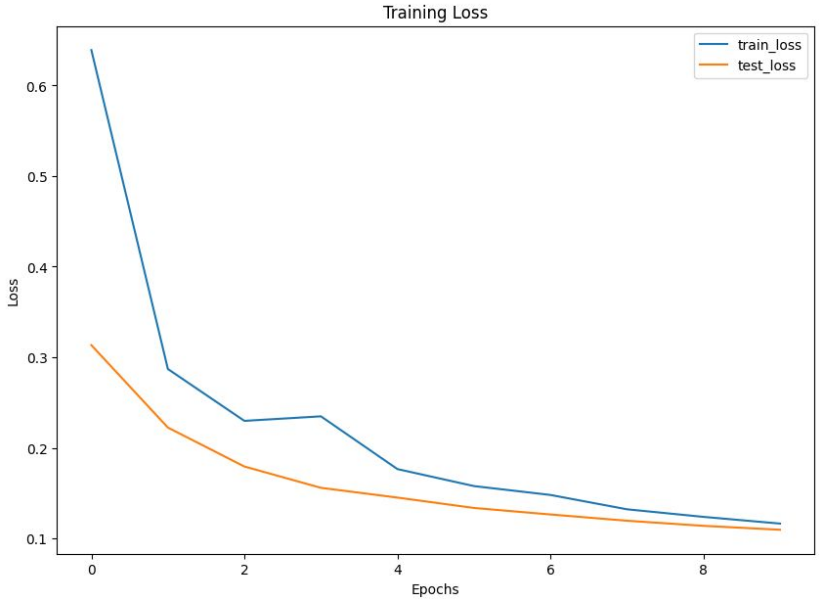


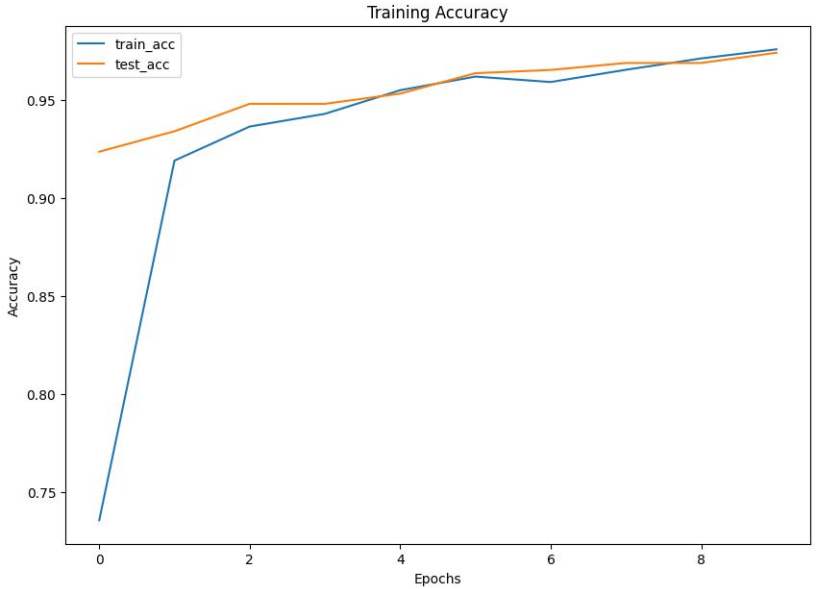
Dividir em patches



ViT - Resultados

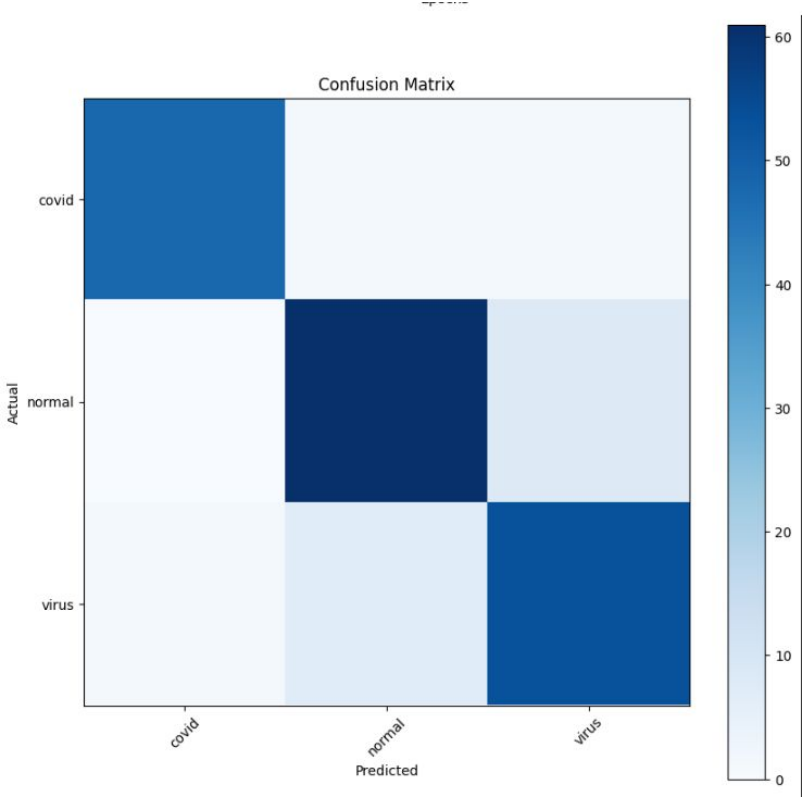
* Treinado por 10 épocas





ViT - Resultados

* Treinado por 10 épocas



	precision	recall	f1-score	support
covid	0.98	0.96	0.97	50
normal	0.88	0.88	0.88	69
virus	0.85	0.87	0.86	61
accuracy			0.90	180
macro avg	0.91	0.90	0.91	180
weighted avg	0.90	0.90	0.90	180