Equipe 4 - Classificação Covid-19

Alunos:

Paulo Douglas Melo da Silva
Paulo Cesar Ribeiro Paiva Filho

Sumário

- Contextualização do problema
- Apresentação do código analisado pela equipe
- Análise das principais problemáticas encontradas
- Propostas de melhoria
- Resultados alcançados

Contextualização do problema

- Desafios da Covid-19
- Identificação de danos aos pulmões
- Classificação de pulmões
  - Covid-19
  - Pneumonia
  - Saudável

Base de dados: https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset

- 137 - Covid-19
- 90 - Pneumonia
- 90 - Saudável

Normal

Viral Pneumonia

Covid

Apresentação do código analisado pela equipe

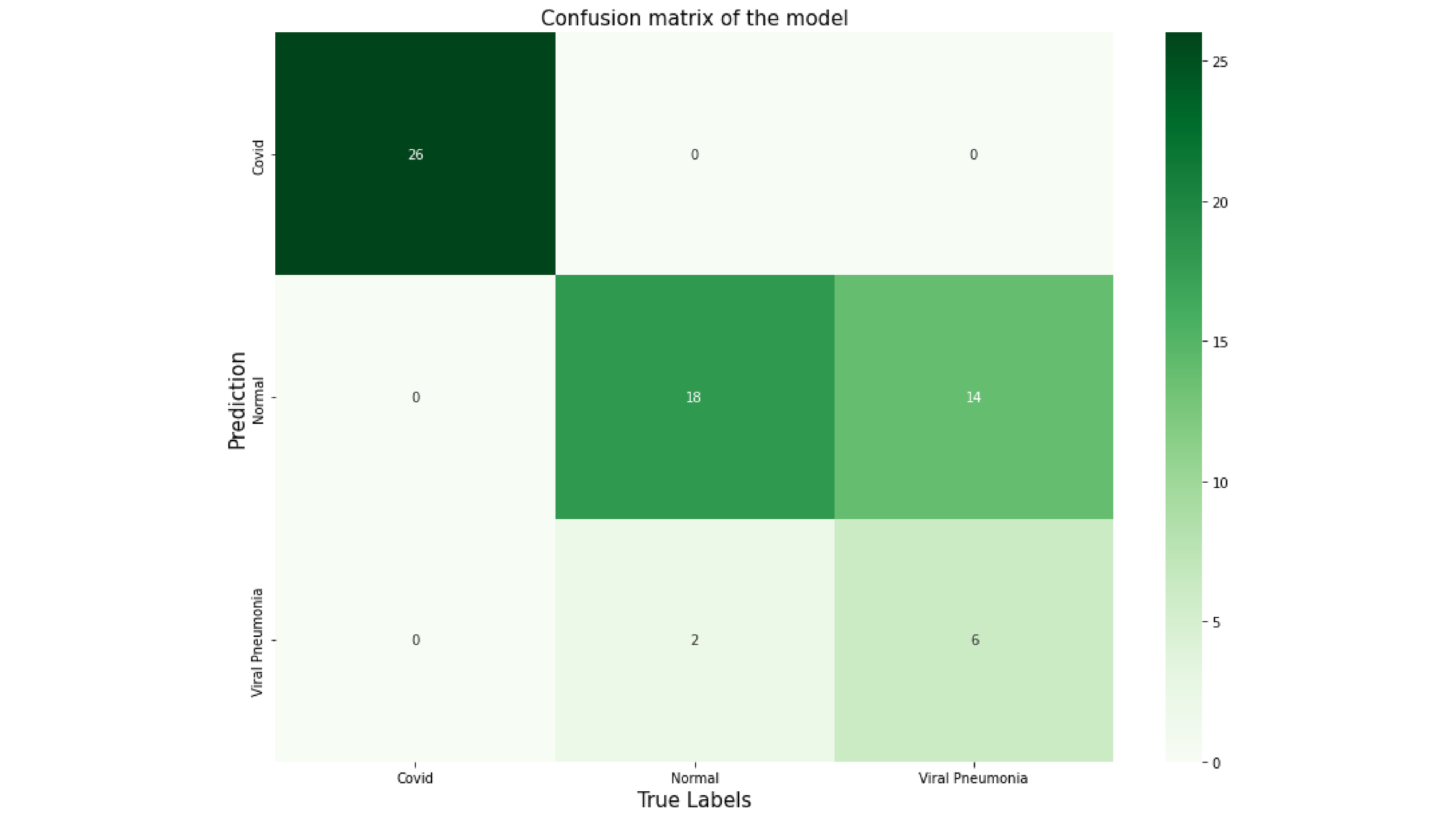
```
Model                                    [32, 3]                   --
├─ResNet: 1-1                           [32, 3]                   --
│    └─Conv2d: 2-1                      [32, 64, 128, 128]        9,408
│    └─BatchNorm2d: 2-2                 [32, 64, 128, 128]        128
│    └─ReLU: 2-3                        [32, 64, 128, 128]        --
│    └─MaxPool2d: 2-4                   [32, 64, 64, 64]          --
│    └─Sequential: 2-5                  [32, 64, 64, 64]          --
│    │    └─BasicBlock: 3-1             [32, 64, 64, 64]          73,984
│    │    └─BasicBlock: 3-2             [32, 64, 64, 64]          73,984
│    └─Sequential: 2-6                  [32, 128, 32, 32]         --
│    │    └─BasicBlock: 3-3             [32, 128, 32, 32]         230,144
│    │    └─BasicBlock: 3-4             [32, 128, 32, 32]         295,424
│    └─Sequential: 2-7                  [32, 256, 16, 16]         --
│    │    └─BasicBlock: 3-5             [32, 256, 16, 16]         919,040
│    │    └─BasicBlock: 3-6             [32, 256, 16, 16]         1,180,672
│    └─Sequential: 2-8                  [32, 512, 8, 8]           --
│    │    └─BasicBlock: 3-7             [32, 512, 8, 8]           3,673,088
│    │    └─BasicBlock: 3-8             [32, 512, 8, 8]           4,720,640
│    └─AdaptiveAvgPool2d: 2-9           [32, 512, 1, 1]           --
│    └─Sequential: 2-10                 [32, 3]                   --
│    │    └─Linear: 3-9                 [32, 512]                 262,656
│    │    └─ReLU: 3-10                  [32, 512]                 --
│    │    └─Dropout: 3-11               [32, 512]                 --
│    │    └─Linear: 3-12                [32, 512]                 262,656
│    │    └─ReLU: 3-13                  [32, 512]                 --
│    │    └─Dropout: 3-14               [32, 512]                 --
│    │    └─Linear: 3-15                [32, 3]                   1,539
==========================================================================================
```

Apresentação do código analisado pela equipe

```
# get image means and stds
stats = ([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
```

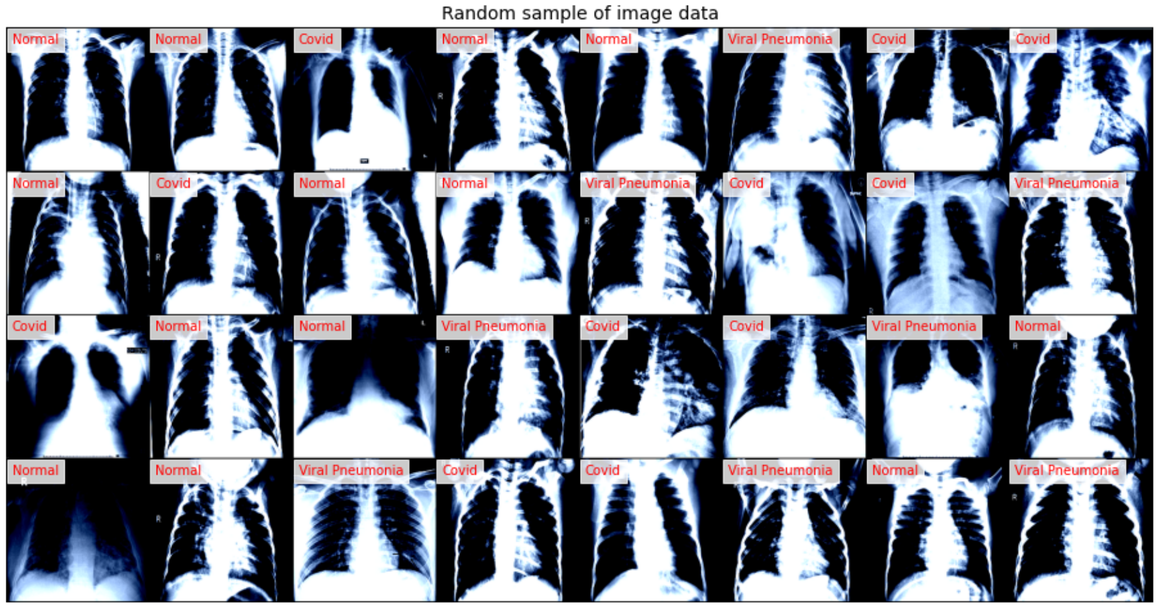
```
# Add custom transforms on top of needed transforms for the pretrained model
base_transforms = [
    tt.Resize((256,256), interpolation = tt.InterpolationMode.BICUBIC),
]

transforms_custom = tt.Compose([
    *base_transforms,
    tt.RandomCrop(224),
    tt.RandomHorizontalFlip(),
    tt.RandomRotation(30),
    tt.RandomVerticalFlip(),
    tt.ToTensor(),
    tt.Normalize(*stats)
])

test_transforms = tt.Compose([
    *base_transforms,
    tt.ToTensor(),
    tt.Normalize(*stats)
])
```

```
model = torchvision.models.resnet18(pretrained=True)

if self.train_backbone_params:
  for param in model.parameters():
    param.requires_grad = True
else:
  for param in model.parameters():
    param.requires_grad = False

classifier = nn.Sequential(
     nn.Linear(in_features=512, out_features=512),
     nn.ReLU(),
     nn.Dropout(0.2),
     nn.Linear(512,512),
     nn.ReLU(),
     nn.Dropout(0.2),
     nn.Linear(512,3)
)

model.fc = classifier
```


Confusion matrix of the model
Prediction
Covid
Normal
Viral Pneumonia
26
0
0
0
18
14
0
2
6
Covid
Normal
Viral Pneumonia
True Labels
25
20
15
10
5
0

Análise das principais problemáticas encontradas

- Transformações arbitrárias nas imagens

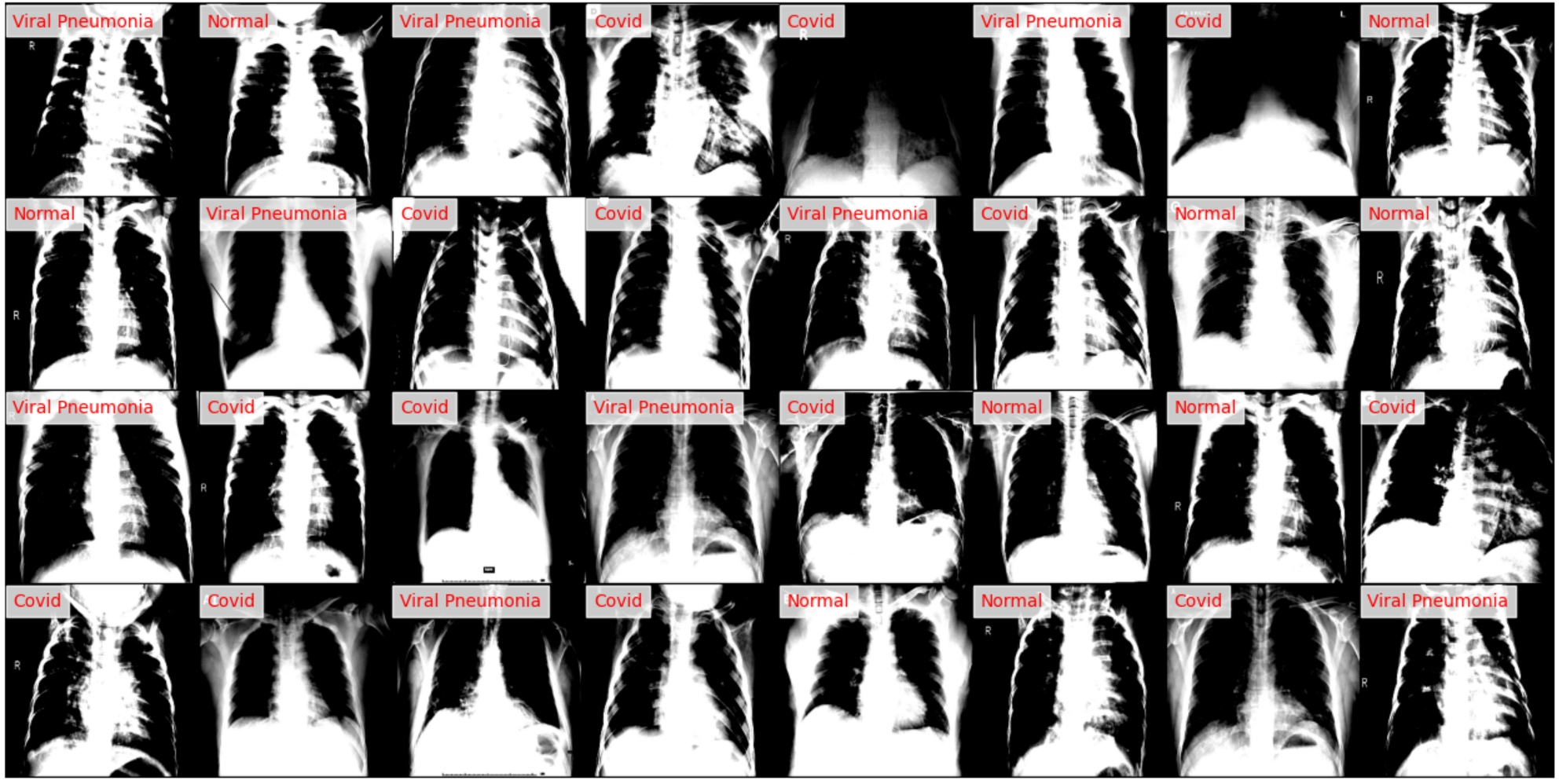


- Overfiting durante o treinamento

Propostas de melhoria

```
stats = ([0.48, 0.48, 0.48], [0.22, 0.22, 0.22])

# Add custom transforms on top of needed transforms for the pretrained model
base_transforms = [
    tt.Resize((256,256), interpolation = tt.InterpolationMode.BICUBIC),
    tt.Grayscale(3)
]

transforms_custom = tt.Compose([
    *base_transforms,
    tt.RandomCrop(240),
    tt.RandomHorizontalFlip(),
    tt.RandomRotation(30),
    tt.RandomVerticalFlip(),
    tt.ToTensor(),
    tt.Normalize(*stats)
])

test_transforms = tt.Compose([
    *base_transforms,
    tt.ToTensor(),
    tt.Normalize(*stats)
])
```

Random sample of image data



```
model = torchvision.models.resnet18(pretrained=True)

if self.train_backbone_params:
  for param in model.parameters():
    param.requires_grad = True
else:
  for param in model.parameters():
    param.requires_grad = False

classifier = nn.Sequential(
     nn.Linear(in_features=512, out_features=512),
     nn.ReLU(),
     nn.Dropout(0.3),
     nn.Linear(512,512),
     nn.ReLU(),
     nn.Dropout(0.3),
     nn.Linear(512,3)
)

model.fc = classifier
```

“Improving neural networks by preventing co-adaptation of feature detectors”, publicado em 2012 por Geoffrey Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever e Ruslan Salakhutdinov

Resultados alcançados