



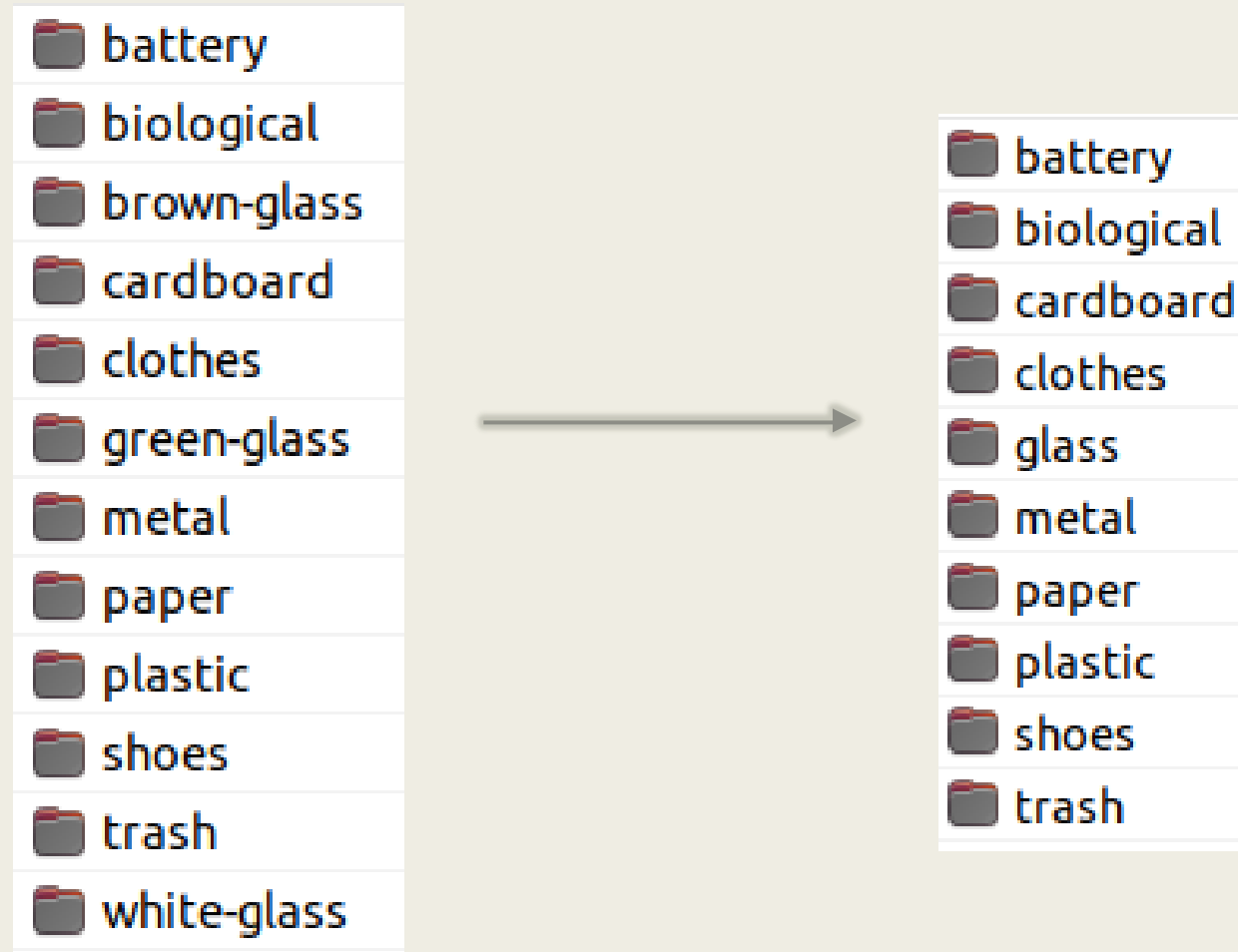
INTELIGÊNCIA COMPUTACIONAL

Projeto Fase - 1

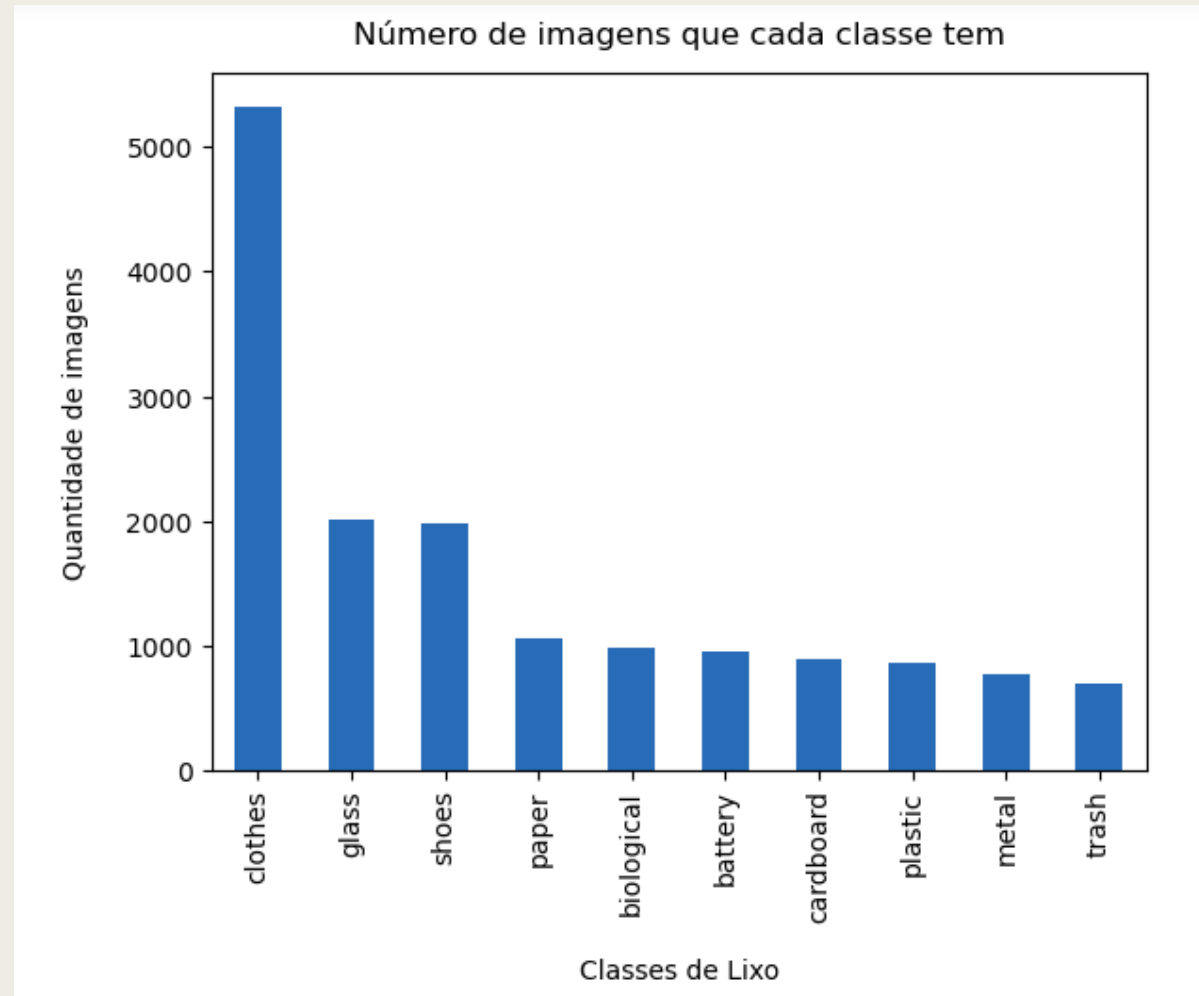
Tratamento de Lixo - Dataset

- Total de 15515 exemplos;
- 12 classes;
- Problema de classificação;
- Não é balanceado.

Ajustes no Dataset



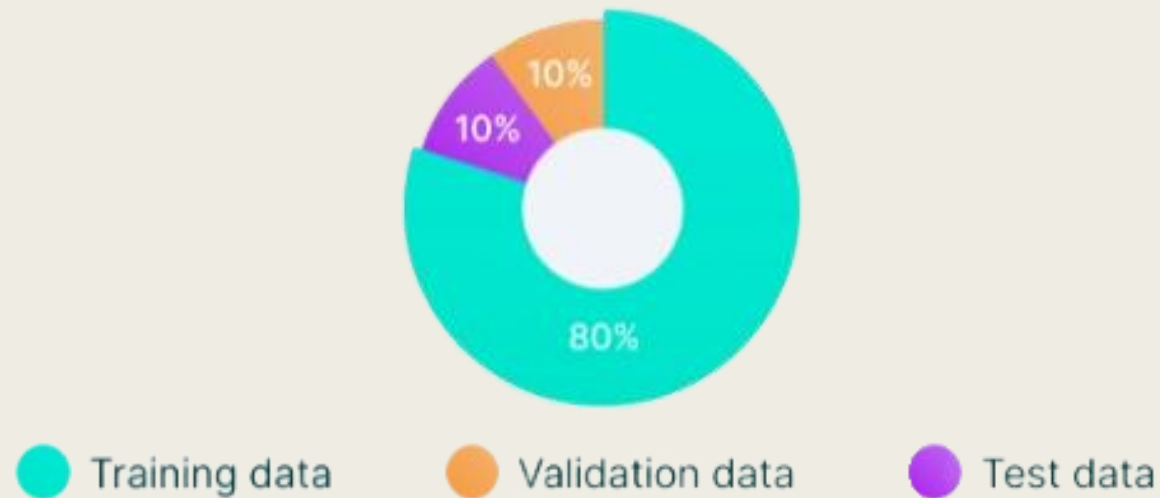
Distribuição das imagens pelas classes



Construção da Rede Neuronal

```
# mobilenetv2 é um modelo (deep learning) do tipo CNN que é pré-treinado, ou seja,  
# já foi treinada para classificar imagens, sendo esta capaz de classificar  
# 1000 categorias das mesmas  
mobilenetv2_layer = mobilenetv2.MobileNetV2(include_top = False, input_shape = (IMAGE_WIDTH, IMAGE_HEIGHT, IMAGE_CHANNELS),  
                                             weights = 'imagenet')  
  
# O modelo mobilenetv2 já está pré-treinado, logo, não queremos que estas camadas que ele traz  
# sejam treinadas novamente pelo tensorflow  
mobilenetv2_layer.trainable = False  
  
model = Sequential()  
  
model.add(keras.Input(shape=(IMAGE_WIDTH, IMAGE_HEIGHT, IMAGE_CHANNELS)))  
  
# Cria uma camada para aplicar o pré-processamento na imagem, para ter as características pretendidas  
def mobilenetv2_preprocessing(img):  
    return mobilenetv2.preprocess_input(img)  
  
model.add(Lambda(mobilenetv2_preprocessing))  
  
#model.add(tf.keras.layers.BatchNormalization()) | Demora mt tempo a treinar com esta camada +/- 15 por epoch  
  
# Camadas pre treinadas  
model.add(mobilenetv2_layer)  
  
model.add(tf.keras.layers.GlobalAveragePooling2D())  
model.add(Flatten(name="featuresCamadaFlatten"))  
model.add(tf.keras.layers.Dropout(0.3))  
model.add(Dense(len(categories), activation='softmax'))  
  
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['categorical_accuracy'])  
model.summary()  
  
feature_extractor = keras.Model(  
    inputs=model.inputs,  
    outputs=model.get_layer(name="featuresCamadaFlatten").output,  
)  
  
x = tf.ones((1, 224, 224, 3))  
features = feature_extractor(x)  
print("Número de Features da Camada featuresCamadaFlatten:")  
print(features)
```

Divisão em Treino, Validação e Teste



Num Imagens de Treino = 12412 Num Imagens de Validação = 1551 Num Imagens de Teste = 1552

Softmax

● Data Augmentation - Teste

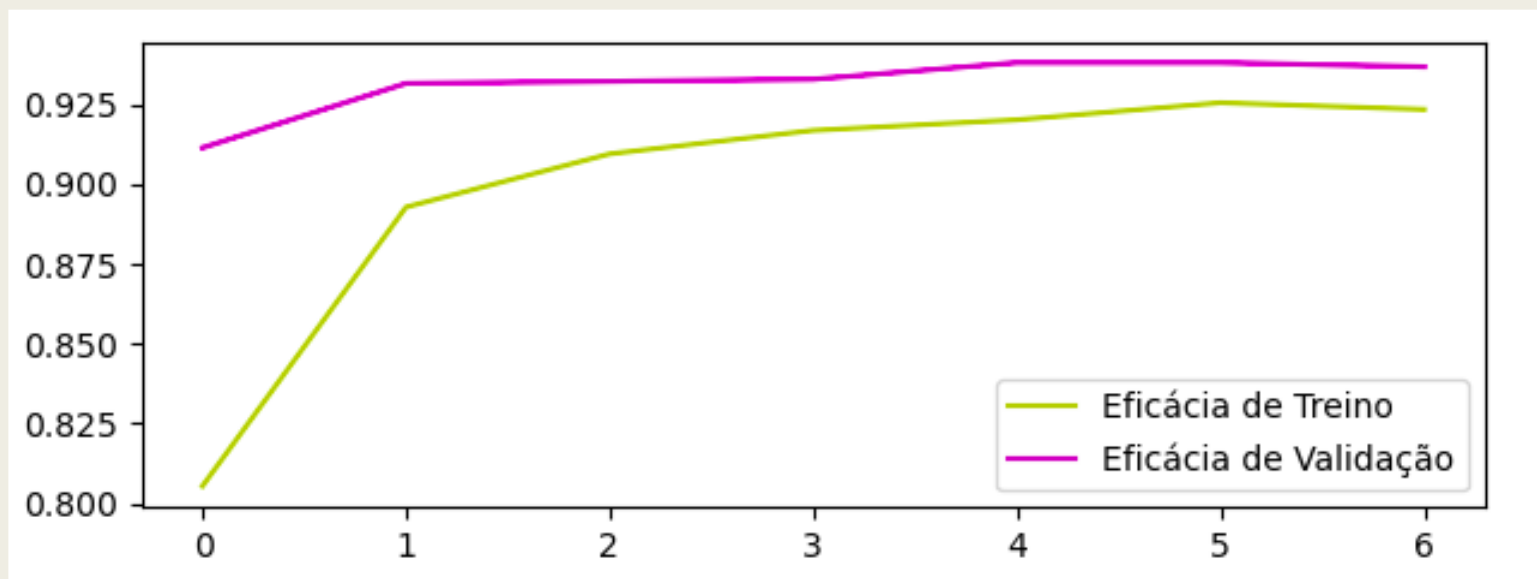
GlobalAveragePooling2D

```
Epoch 1/20
193/193 [=====] - 331s 2s/step - loss: 0.6619 - categorical_accuracy: 0.7992 - val_loss:
0.3179 - val_categorical_accuracy: 0.9115
Epoch 2/20
193/193 [=====] - 341s 2s/step - loss: 0.3374 - categorical_accuracy: 0.8907 - val_loss:
0.2730 - val_categorical_accuracy: 0.9219
Epoch 3/20
193/193 [=====] - 348s 2s/step - loss: 0.2827 - categorical_accuracy: 0.9060 - val_loss:
0.2535 - val_categorical_accuracy: 0.9219
Epoch 4/20
193/193 [=====] - 350s 2s/step - loss: 0.2559 - categorical_accuracy: 0.9149 - val_loss:
0.2452 - val_categorical_accuracy: 0.9277
Epoch 5/20
193/193 [=====] - 342s 2s/step - loss: 0.2430 - categorical_accuracy: 0.9176 - val_loss:
0.2477 - val_categorical_accuracy: 0.9290
Epoch 6/20
193/193 [=====] - 345s 2s/step - loss: 0.2278 - categorical_accuracy: 0.9266 - val_loss:
0.2529 - val_categorical_accuracy: 0.9277
Epoch 7/20
193/193 [=====] - 347s 2s/step - loss: 0.2223 - categorical_accuracy: 0.9263 - val_loss:
0.2194 - val_categorical_accuracy: 0.9362
Epoch 8/20
193/193 [=====] - 348s 2s/step - loss: 0.2137 - categorical_accuracy: 0.9289 - val_loss:
0.2238 - val_categorical_accuracy: 0.9336
Epoch 9/20
193/193 [=====] - ETA: 0s - loss: 0.2187 - categorical_accuracy: 0.9240Restoring model we
ights from the end of the best epoch: 7.
193/193 [=====] - 343s 2s/step - loss: 0.2187 - categorical_accuracy: 0.9240 - val_loss:
0.2227 - val_categorical_accuracy: 0.9323
Epoch 9: early stopping
```

Softmax

GlobalAveragePooling2D

● Data Augmentation - Teste



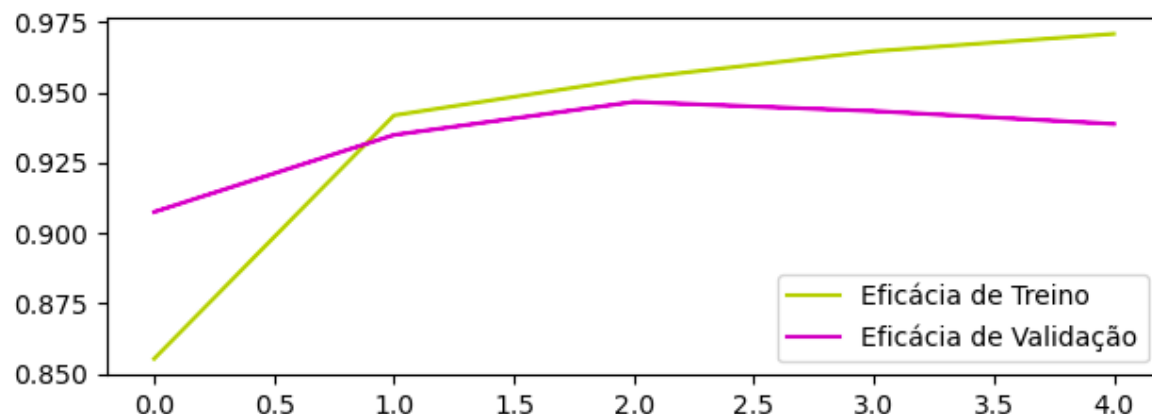
Eficácia de teste = 93.69 %

Softmax

● Data Augmentation - Teste

GlobalAveragePooling2D

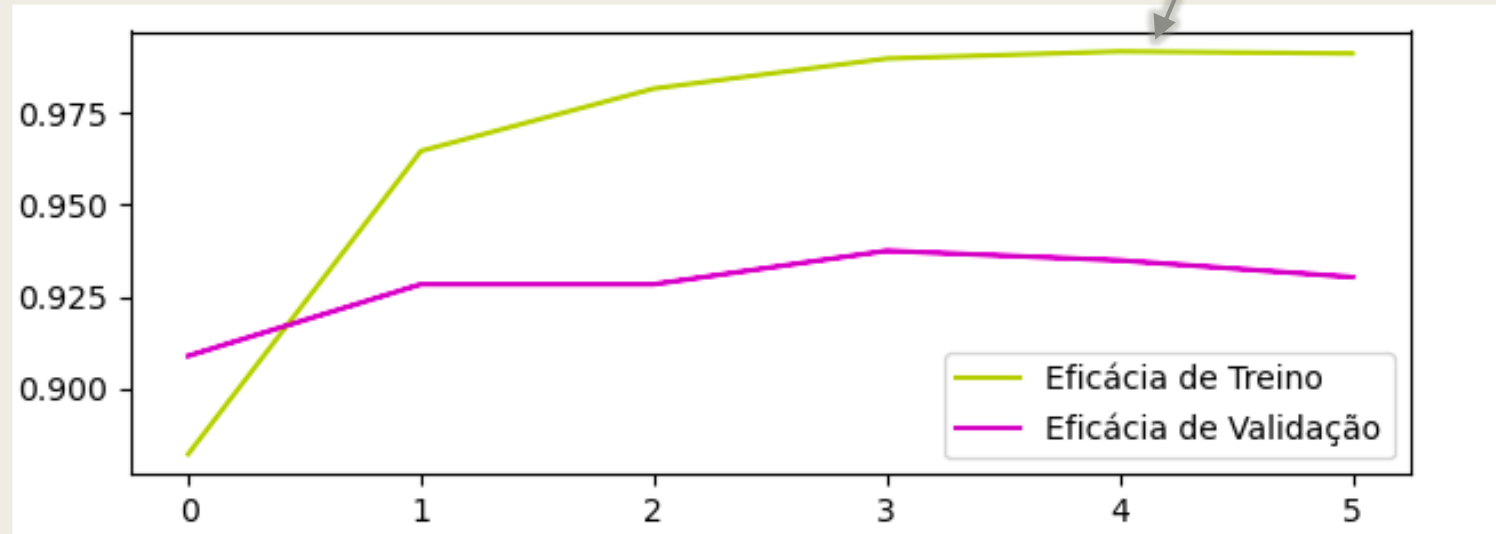
```
Epoch 1/20
193/193 [=====] - 284s 1s/step - loss: 0.4882 - categorical_accuracy: 0.8554 - val_loss:
0.2689 - val_categorical_accuracy: 0.9076
Epoch 2/20
193/193 [=====] - 282s 1s/step - loss: 0.1962 - categorical_accuracy: 0.9419 - val_loss:
0.2049 - val_categorical_accuracy: 0.9349
Epoch 3/20
193/193 [=====] - 283s 1s/step - loss: 0.1498 - categorical_accuracy: 0.9550 - val_loss:
0.1789 - val_categorical_accuracy: 0.9466
Epoch 4/20
193/193 [=====] - 279s 1s/step - loss: 0.1230 - categorical_accuracy: 0.9646 - val_loss:
0.1821 - val_categorical_accuracy: 0.9434
Epoch 5/20
193/193 [=====] - ETA: 0s - loss: 0.1036 - categorical_accuracy: 0.9708Restoring model we
ights from the end of the best epoch: 3.
193/193 [=====] - 279s 1s/step - loss: 0.1036 - categorical_accuracy: 0.9708 - val_loss:
0.1680 - val_categorical_accuracy: 0.9388
Epoch 5: early stopping
```



Eficácia de teste = 94.33 %

Flatten - Teste

```
model.add(Flatten(name="featuresCamadaFlatten"))  
#model.add(tf.keras.layers.GlobalAveragePooling2D(name="featuresCamadaGlobal"))  
model.add(Dense(len(categories), activation='softmax'))
```



Eficácia de teste = 93.69 %

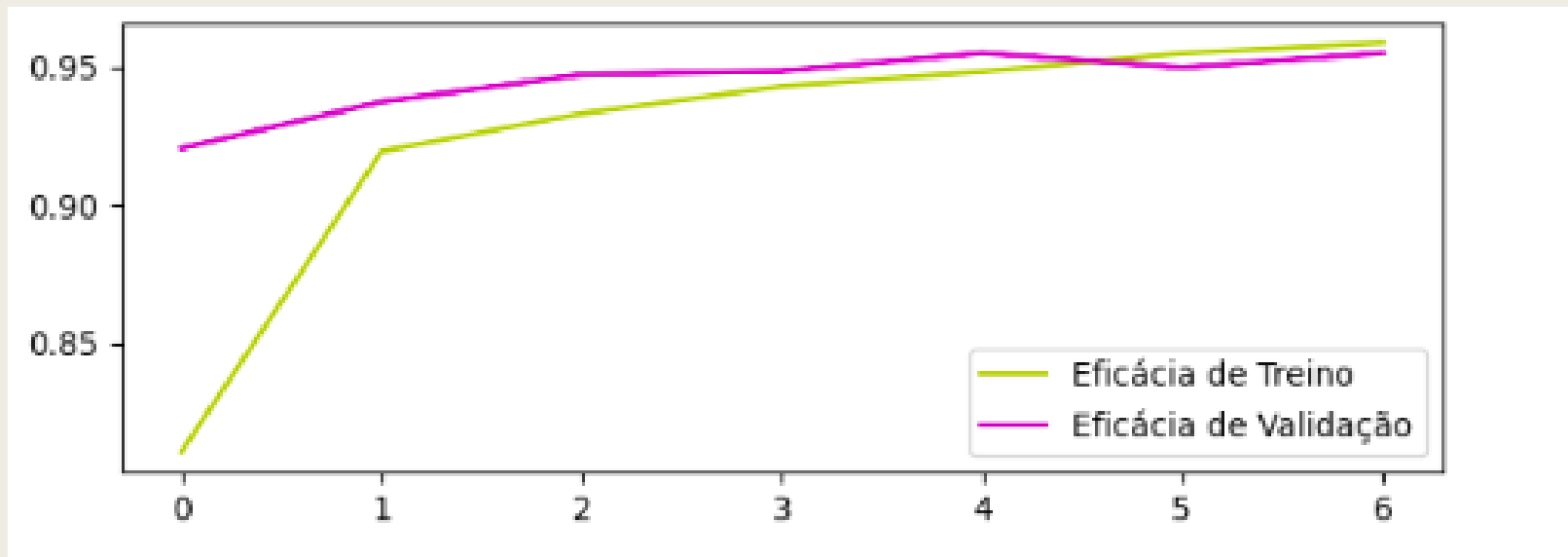
GlobalAveragePooling2D + Relu Flatten + Relu

```
Epoch 1/20
193/193 [=====] - 280s 1s/step - loss: nan - categorical_accuracy: 0.2017 - val_loss: nan
- val_categorical_accuracy: 0.0618
Epoch 2/20
193/193 [=====] - 280s 1s/step - loss: nan - categorical_accuracy: 0.0611 - val_loss: nan
- val_categorical_accuracy: 0.0625
Epoch 3/20
193/193 [=====] - ETA: 0s - loss: nan - categorical_accuracy: 0.0613Restoring model weights
from the end of the best epoch: 1.
193/193 [=====] - 281s 1s/step - loss: nan - categorical_accuracy: 0.0613 - val_loss: nan
- val_categorical_accuracy: 0.0625
Epoch 3: early stopping
```

Eficácia de teste = 5.8 %

Eficácia de teste = 6.51 %

GlobalAveragePooling2D + Flatten + Dropout + Softmax



Eficácia de teste = 94.85 %

Previsões

Número de vezes que a classe bateria apareceu no conjunto de dados.

	support
battery	85
biological	104
cardboard	87
clothes	539
glass	200
metal	69
paper	114
plastic	99
shoes	191
trash	64
accuracy	0.95 1552

Conclusão

- Se cada ecoponto tivesse uma camara, este modelo seria capaz de reconhecer qual o objeto que está a ser despejado e se este está a ser colocado ou não no sítio certo.