

Relatório de experimentos com classificação - Projeto Final IA

1. Especificações

ERICK ECKERMANN CARDOSO
GRR20186075

Trabalho da disciplina de Inteligência Artificial. O objetivo consiste em classificar as imagens de uma base em 6 classes, consistindo nos personagens dos Simpsons. Para realizar tal tarefa, deveria-se extrair um vetor de características (representação) de cada imagem. As bases de treino e validação são compostas de 253 e 106 imagens.

Trabalho implementado em Python 3, utilizando o módulo Tensor Flow e a API Keras para criação, treinamento do modelo pela base, extração do vetor de características e classificação. Para a classificação, utilizou-se o script de classificação com KNN com $k = 1$.

Os imports utilizados foram:

- import numpy as np
- import tensorflow as tf
- import keras
- from keras.layers import GlobalAveragePooling2D, Dense, Dropout, Flatten
- from keras.models import Model
- from keras.callbacks import EarlyStopping
- from keras.applications import VGG16, ResNet50, Xception, ResNet50V2, EfficientNetB4
- from keras.applications.resnet import preprocess_input
- from matplotlib import pyplot as plt
- import pandas as pd

Os testes foram feitos com um total de 4 modelos disponibilizados pela API Keras. Estes são: VGG16, ResNet50, ResNet50V2, EfficientNetB4. O único onde foi realizado o treinamento pela camada densa foi ResNet50, onde foram atingidos os melhores resultados. Nestes mesmos modelos, foram feitos testes também com as imagens da base cortadas, onde a acurácia obteve aumento significativo.

Diversos parâmetros foram modificados para procurar a melhor acurácia na validação. Será apresentado neste relatório, em maior parte, os parâmetros que receberam maior desempenho em cada base.

2. Resultado dos experimentos

2.1 VGG16

Não foi realizado o treinamento do modelo nos testes. O melhor resultado com a extração simples, utilizando a base com as imagens **não** cortadas foi a seguinte:

Número de Bach: 32

Paciência: 15

Accuracy: 0.37735849056603776

Matriz de confusão:

```
[[16 2 9 6 1 1]
 [ 3 3 3 1 0 1]
 [10 4 9 0 1 1]
 [ 3 2 4 4 0 0]
 [ 3 2 2 0 3 2]
 [ 0 2 2 0 1 5]]
```

Resultados por classe:

	precision	recall	f1-score	support
0	0.46	0.46	0.46	35
1	0.20	0.27	0.23	11
2	0.31	0.36	0.33	25
3	0.36	0.31	0.33	13
4	0.50	0.25	0.33	12
5	0.50	0.50	0.50	10
accuracy			0.38	106
macro avg	0.39	0.36	0.36	106
weighted avg	0.39	0.38	0.38	106

O melhor resultado com a extração simples, utilizando a base com as imagens cortadas foi a seguinte:

Número de Bach: 32

Paciência: 10

Accuracy: 0.6037735849056604

Matriz de confusão:

```
[[25 2 2 2 2 2]
 [ 1 9 1 0 0 0]
 [ 5 1 11 4 1 3]
 [ 2 0 2 7 1 1]
 [ 2 0 1 4 5 0]
 [ 0 0 1 1 1 7]]
```

Resultados por classe:

	precision	recall	f1-score	support
0	0.71	0.71	0.71	35
1	0.75	0.82	0.78	11
2	0.61	0.44	0.51	25
3	0.39	0.54	0.45	13
4	0.50	0.42	0.45	12
5	0.54	0.70	0.61	10
accuracy			0.60	106
macro avg	0.58	0.60	0.59	106
weighted avg	0.61	0.60	0.60	106

2.2 ResNet50

Os testes para esse modelo foram realizados com e sem treinamento, utilizando a base com as imagens cortadas e não.

2.2.1 Sem treinamento

- Para a base com as imagens não cortadas obteve-se os seguintes resultados com os parâmetros:

Número de Bach: 32

Paciência: 15

Accuracy: 0.5

Matriz de confusão:

```
[[17 1 8 7 1 1]
 [ 1 7 2 0 0 1]
 [ 9 1 11 4 0 0]
 [ 2 2 0 9 0 0]
 [ 5 1 3 1 2 0]
 [ 1 1 1 0 0 7]]
```

Resultados por classe:

	precision	recall	f1-score	support
0	0.49	0.49	0.49	35
1	0.54	0.64	0.58	11
2	0.44	0.44	0.44	25
3	0.43	0.69	0.53	13
4	0.67	0.17	0.27	12
5	0.78	0.70	0.74	10
accuracy			0.50	106
macro avg	0.56	0.52	0.51	106
weighted avg	0.52	0.50	0.49	106

- Imagens cortadas:

Número de Bach: 32

Paciência: 15

Accuracy: 0.6226415094339622

Matriz de confusão:

```
[[29 1 3 1 0 1]
 [ 2 7 2 0 0 0]
 [ 7 0 15 1 1 1]
 [ 5 1 0 5 1 1]
 [ 2 0 3 3 4 0]
 [ 2 1 1 0 0 6]]
```

Resultados por classe:

	precision	recall	f1-score	support
0	0.62	0.83	0.71	35
1	0.70	0.64	0.67	11
2	0.62	0.60	0.61	25
3	0.50	0.38	0.43	13
4	0.67	0.33	0.44	12
5	0.67	0.60	0.63	10

accuracy			0.62	106
macro avg	0.63	0.56	0.58	106
weighted avg	0.62	0.62	0.61	106

Percebe-se que houve um significativo aumento na acurácia para os testes com as imagens cortadas. Isso se deve provavelmente pelo menor número de características que atrapalham a classificação.

2.2.2 Com treinamento

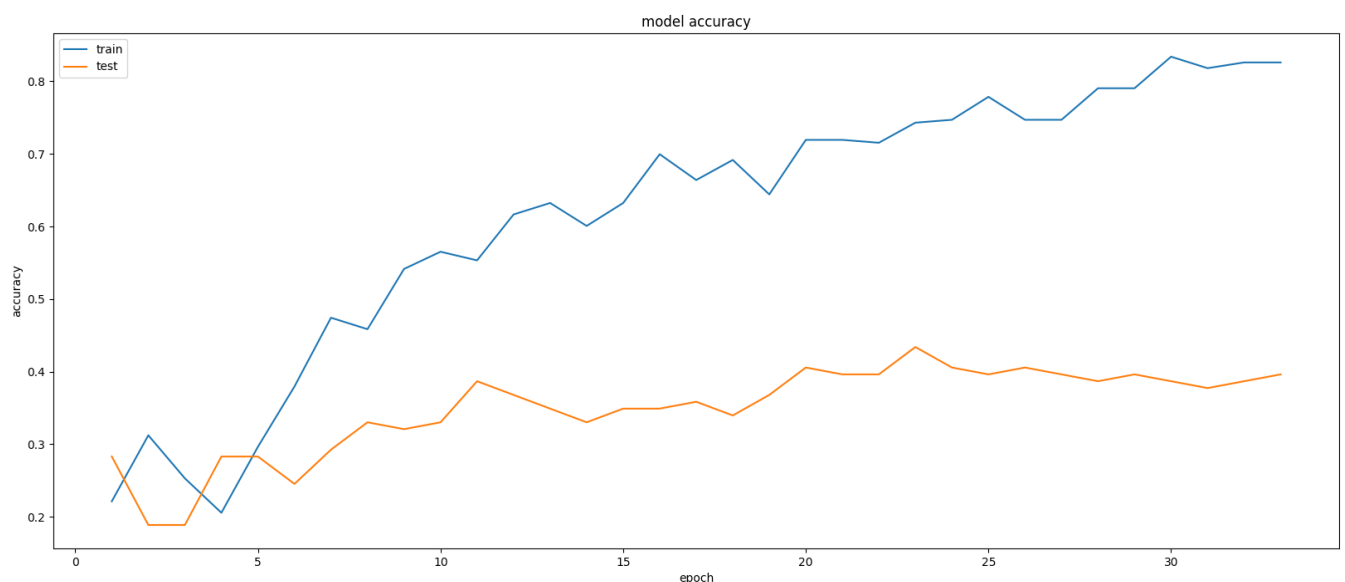
O treinamento foi realizado em **camada densa**, alterando os parâmetros para se buscar os melhores resultados.

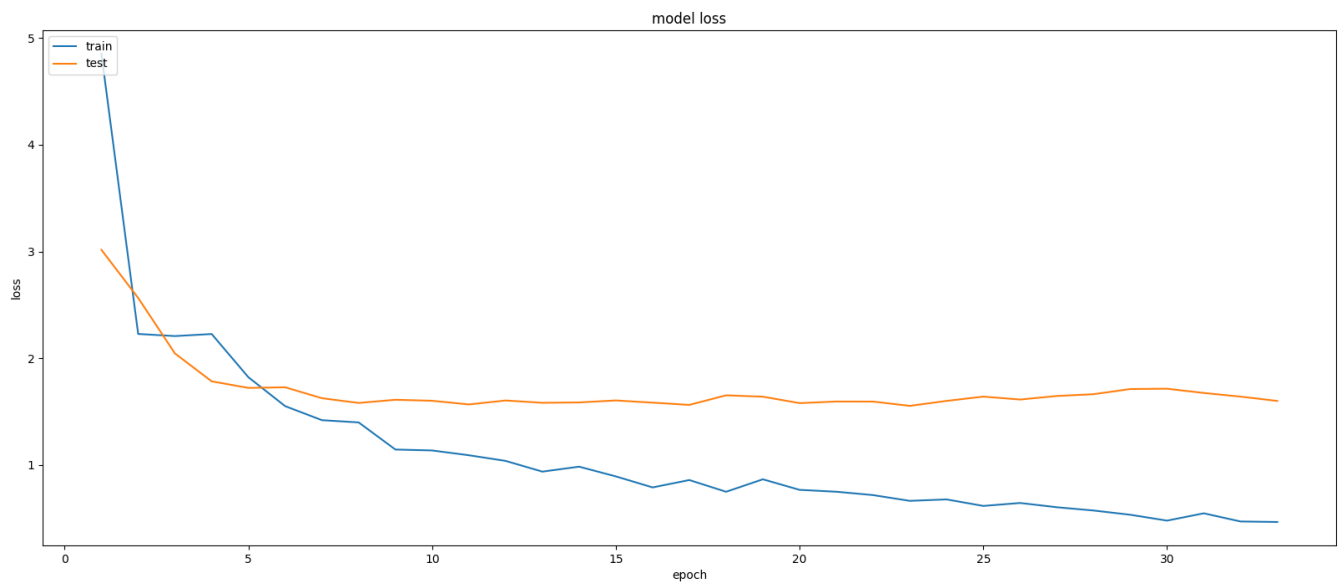
Para o primeiro treinamento as camadas convolucionais não são treinadas.

A camada densa não teve ativação, e a loss foi definida como `from_logits=True`.

O otimizador utilizado foi o Adam com a taxa de aprendizado padrão.

Imagens não cortadas





Número de batch: 32

Paciência: 10

Accuracy: 0.4339622641509434

Matriz de confusão:

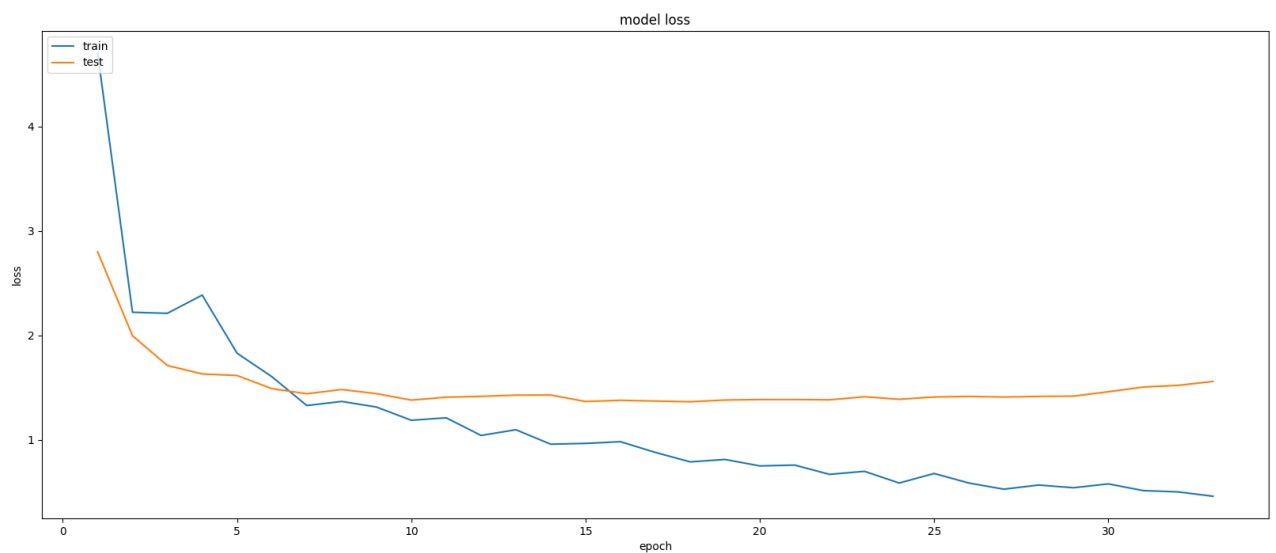
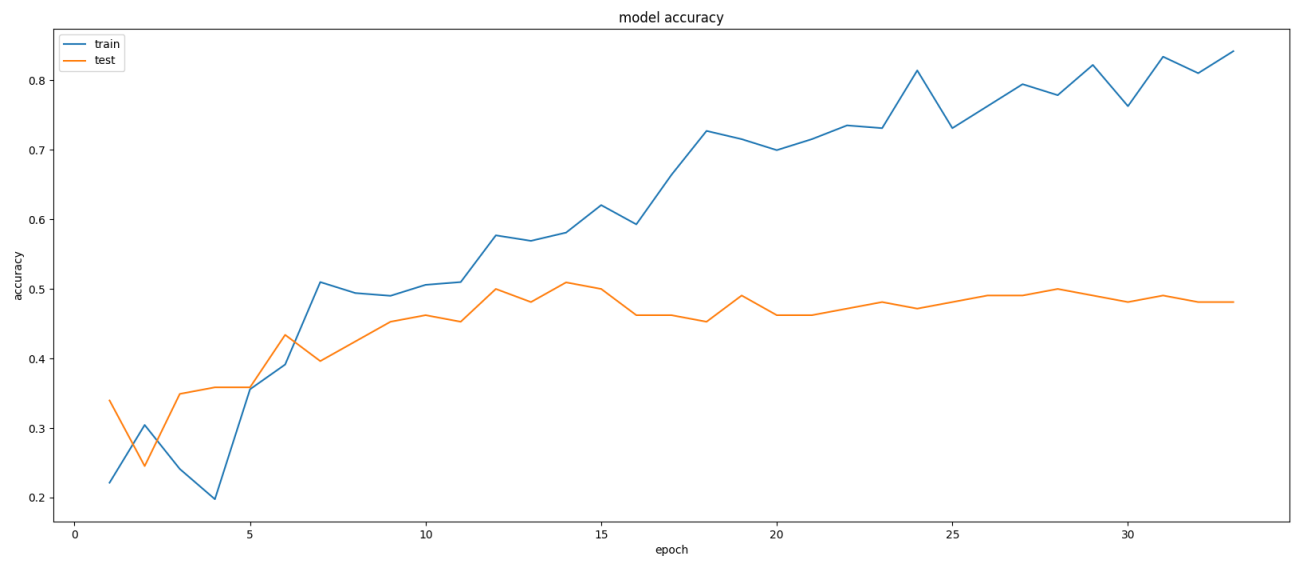
```
[[22 0 10 0 2 1]
 [ 2 4 4 0 1 0]
 [11 0 11 1 1 1]
 [ 5 3 2 3 0 0]
 [ 6 0 5 0 1 0]
 [ 2 0 3 0 0 5]]
```

Resultados para as classes:

	precision	recall	f1-score	support
0	0.46	0.63	0.53	35
1	0.57	0.36	0.44	11
2	0.31	0.44	0.37	25
3	0.75	0.23	0.35	13
4	0.20	0.08	0.12	12
5	0.71	0.50	0.59	10

accuracy	0.43		106	
macro avg	0.50	0.37	0.40	106
weighted avg	0.47	0.43	0.42	106

Resultado aumentando a paciência da rede



Paciência: 15

Número de Bach: 32

Accuracy: 0.5377358490566038

Matriz de confusão:

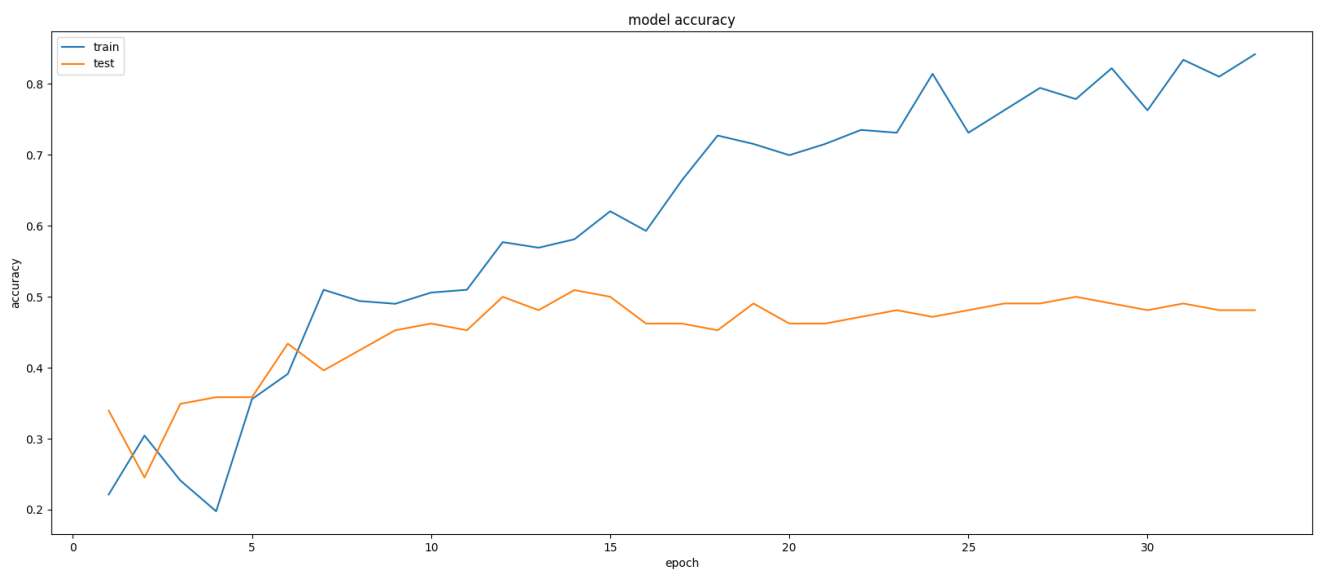
```
[[25 0 8 1 0 1]
 [ 1 9 1 0 0 0]
 [ 5 3 14 0 2 1]
 [ 4 2 5 2 0 0]
 [ 3 0 4 0 3 2]
 [ 1 0 5 0 0 4]]
```

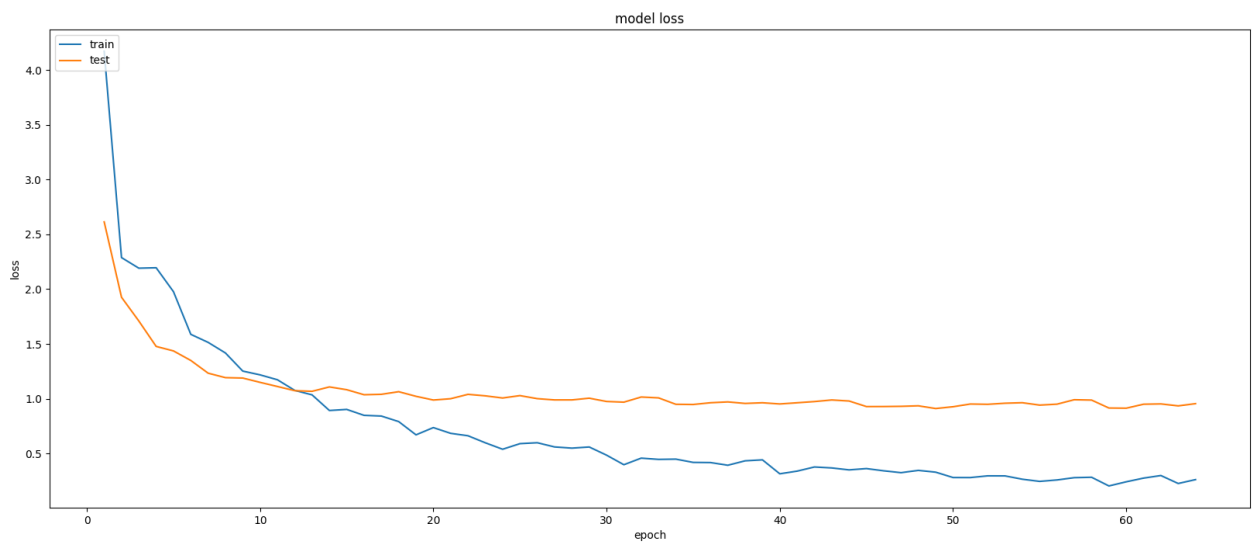
Resultados para as classes

	precision	recall	f1-score	support
0	0.64	0.71	0.68	35
1	0.64	0.82	0.72	11
2	0.38	0.56	0.45	25
3	0.67	0.15	0.25	13
4	0.60	0.25	0.35	12
5	0.50	0.40	0.44	10
accuracy		0.54		106
macro avg	0.57	0.48	0.48	106
weighted avg	0.56	0.54	0.52	106

Percebe-se um pequeno aumento na acurácia devido ao aumento da paciência no treinamento.

Imagens da base cortadas





Paciência: 15

Número de Bach: 32

Accuracy: 0.7264150943396226

Matriz de confusão:

```
[[31 0 2 0 1 1]
 [0 7 3 1 0 0]
 [4 0 20 0 1 0]
 [6 2 0 5 0 0]
 [3 0 0 4 5 0]
 [0 0 1 0 0 9]]
```

Resultados para as classes:

	precision	recall	f1-score	support
0	0.70	0.89	0.78	35
1	0.78	0.64	0.70	11
2	0.77	0.80	0.78	25
3	0.50	0.38	0.43	13
4	0.71	0.42	0.53	12
5	0.90	0.90	0.90	10
accuracy	0.73			106
macro avg	0.73	0.67	0.69	106
weighted avg	0.72	0.73	0.71	106

Houve um aumento considerável na acurácia geral.

2.3 ResNet50V2

- Imagens não cortadas:

Número de Bach: 32

Paciência: 15

Accuracy: 0.44339622641509435

Matriz de confusão:

```
[[15 0 15 5 0 0]
 [ 2 0 5 1 0 3]
 [ 7 0 16 0 1 1]
 [ 6 0 2 4 0 1]
 [ 2 0 6 0 4 0]
 [ 1 0 1 0 0 8]]
```

Resultados por classe:

	precision	recall	f1-score	support
0	0.45	0.43	0.44	35
1	0.00	0.00	0.00	11
2	0.36	0.64	0.46	25
3	0.40	0.31	0.35	13
4	0.80	0.33	0.47	12
5	0.62	0.80	0.70	10

accuracy			0.44	106
macro avg	0.44	0.42	0.40	106
weighted avg	0.43	0.44	0.42	106

- Imagens cortadas

Número de Bach: 32

Paciência: 15

Accuracy: 0.4528301886792453

Matriz de confusão:

```
[[25 0 5 3 2 0]
 [ 4 4 3 0 0 0]
 [12 0 9 2 0 2]
 [ 8 0 1 3 1 0]
 [ 4 0 4 2 2 0]
 [ 3 1 1 0 0 5]]
```

Resultados por classe:

	precision	recall	f1-score	support
0	0.45	0.71	0.55	35
1	0.80	0.36	0.50	11
2	0.39	0.36	0.37	25
3	0.30	0.23	0.26	13
4	0.40	0.17	0.24	12
5	0.71	0.50	0.59	10

accuracy			0.45	106
macro avg	0.51	0.39	0.42	106

weighted avg 0.47 0.45 0.44 106

Perceba que não houve aumento significativo para as duas bases nesse caso.

2.4 EfficientNetB4

- Imagens das bases cortadas

Número de Bach: 32

Paciência: 15

Accuracy: 0.5566037735849056

Matriz de confusão:

```
[[27 0 4 3 1 0]
 [ 1 8 1 1 0 0]
 [12 1 7 2 1 2]
 [ 3 0 4 6 0 0]
 [ 2 2 1 0 7 0]
 [ 1 1 2 0 2 4]]
```

Resultados por classe:

	precision	recall	f1-score	support
0	0.59	0.77	0.67	35
1	0.67	0.73	0.70	11
2	0.37	0.28	0.32	25
3	0.50	0.46	0.48	13
4	0.64	0.58	0.61	12
5	0.67	0.40	0.50	10

accuracy			0.56	106
macro avg	0.57	0.54	0.54	106
weighted avg	0.55	0.56	0.54	106

- Imagens cortadas

Número de Bach: 32

Paciência: 15

Accuracy: 0.5566037735849056

Matriz de confusão:

```
[[29 2 2 1 0 1]
 [ 3 4 1 1 2 0]
 [12 5 6 0 2 0]
 [ 2 2 1 7 0 1]
 [ 1 1 0 2 8 0]
 [ 4 0 1 0 0 5]]
```

Resultados por classe:

	precision	recall	f1-score	support
0	0.57	0.83	0.67	35
1	0.29	0.36	0.32	11
2	0.55	0.24	0.33	25

3	0.64	0.54	0.58	13
4	0.67	0.67	0.67	12
5	0.71	0.50	0.59	10
accuracy			0.56	106
macro avg	0.57	0.52	0.53	106
weighted avg	0.57	0.56	0.54	106

Perceba que não houve diferença na acurácia para as bases nesse modelo.

3. Conclusão

Em geral, os resultados com as bases onde as imagens tiveram seus tamanhos cortados em 20%, os resultados foram melhores e mantendo na média dos 60% de acurácia.

Com a rede treinada utilizando a camada Densa, o maior resultado atingido foi em média 72%, o que é significativamente melhor do que os outros resultados. Percebe-se também que, no geral, aumentar a paciência interfere no resultado positivamente, porém o custo computacional também aumenta dado ao aumento de iterações. Modificar o `batch_size` em geral não trouxe mudanças significativas.

Para uma rede apresentar resultados melhores em geral, deve-se haver uma base de treino maior e de maior qualidade, que permita extrair a maior quantidade possível de informações de representação. Assim, pode-se concluir que os resultados com as imagens cortadas tiveram melhor eficiência pois cortar a imagem permitia uma melhor representação, tornando a base mais bem preparada.