

# Relatório de experimentos com classificação - Projeto Final IA

Professor: Aurora Trinidad Ramirez Pozo

Aluno: Erick Eckermann Cardoso

GRR20186075

## 1. Especificações

Projeto final da disciplina de Inteligência Artificial. O tema escolhido foi o de classificação de imagens utilizando redes neurais convolucionais (CNN), que é uma classe de rede neural utilizada para processamento e análise de imagens, onde foram utilizadas uma base de imagens para treinamento e outra para a validação da rede treinada. As imagens foram divididas em 6 classes, baseadas nos personagens da família de Os Simpsons, onde cada classe representa um personagem da família e a última classe seria a família toda completa.

A implementação e os conceitos foram baseados no artigo de Marcos Tanaka, "Classificação de imagens com deep learning e TensorFlow", disponível no endereço web: <https://imasters.com.br/back-end/classificacao-de-imagens-com-deep-learning-e-tensorflow>. O algoritmo foi implementado em **Python 3** e foi utilizado o módulo **Tensor Flow e a API Keras**, para os modelos pré-treinados.

Neste exemplo não foi escrita uma CNN do zero, mas sim treinados alguns modelos prontos utilizando um processo chamado Transfer Learning. Com Transfer Learning, usamos uma CNN já treinada e adicionamos uma camada a mais, treinando apenas esta camada para nosso objetivo. Os testes foram feitos com um total de 3 modelos pré-treinados disponibilizados pela API Keras. Estes são: **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 em 20%, ou seja, 20% de cada imagem foi cortada, onde a acurácia obteve aumento significativo devido ao menor número de características para serem avaliadas..

Diversos parâmetros foram modificados para procurar a melhor acurácia na validação. Para todos os testes, o número de épocas para o treinamento foi fixado em 100. Será apresentado neste relatório, em maior parte, os parâmetros que receberam maior desempenho em cada base.

## 2. Metodologia

Foi utilizado a metodologia de transfer learning para re-treinar os modelos para objetivo específico de classificar a família dos Simpsons. Também foram realizados testes com as redes sem o treinamento, a fim de comparação.

O processo de treinamento e classificação se encontra no código Simpsons.py. Quando é executado, o algoritmo procura a base com o caminho especificado e extrai as bases de treinamento e validação. Então o modelo especificado é treinado e então se inicia a validação. Para a classificação, foi utilizado um script baseado em classificação KNN com o  $k = 1$ . O algoritmo KNN permite com que você faça previsões de dados com base nos K vizinhos mais próximos a esse ponto, indicando que por estar próximo daqueles pontos, ele

pertence à mesma classe. O script que foi baseado se encontra no código knn.py. Os resultados são impressos no terminal, indicando a acurácia para cada Classe.

### 3. Resultado dos experimentos

#### 3.2 ResNet50

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

##### 3.2.1 Sem treinamento

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

Numéro de Bach: 32

Paciência: 15

Accuracy: 0.5

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:

Numéro de Bach: 32

Paciência: 15

Accuracy: 0.6226415094339622

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.

##### 3.2.2 Com treinamento

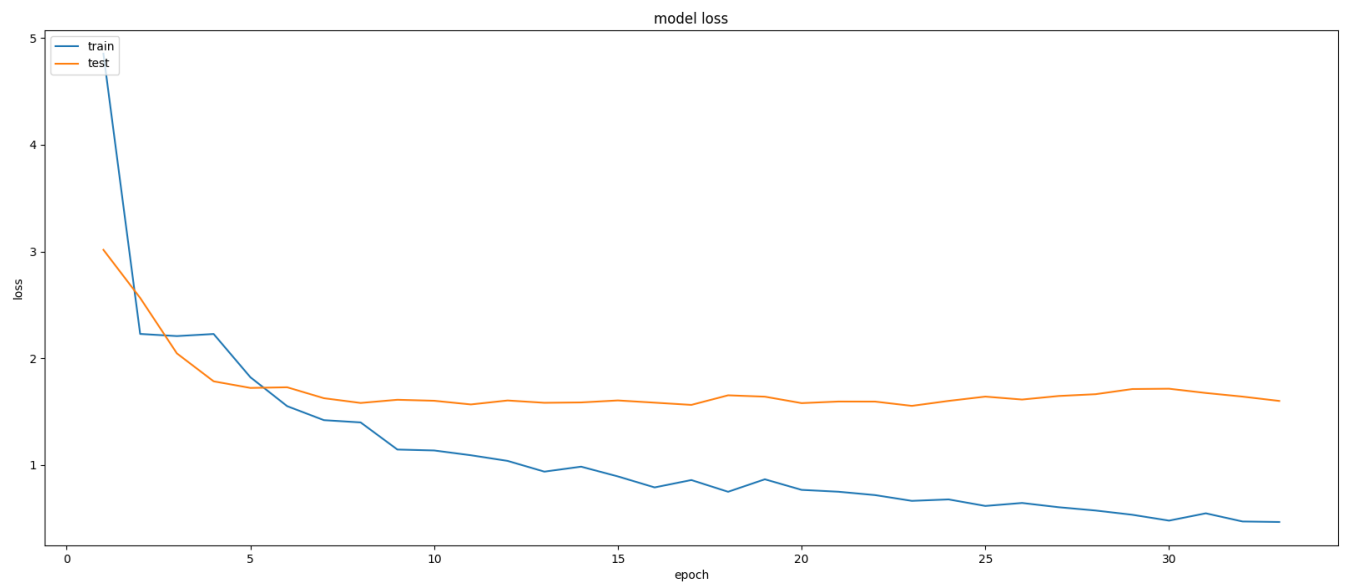
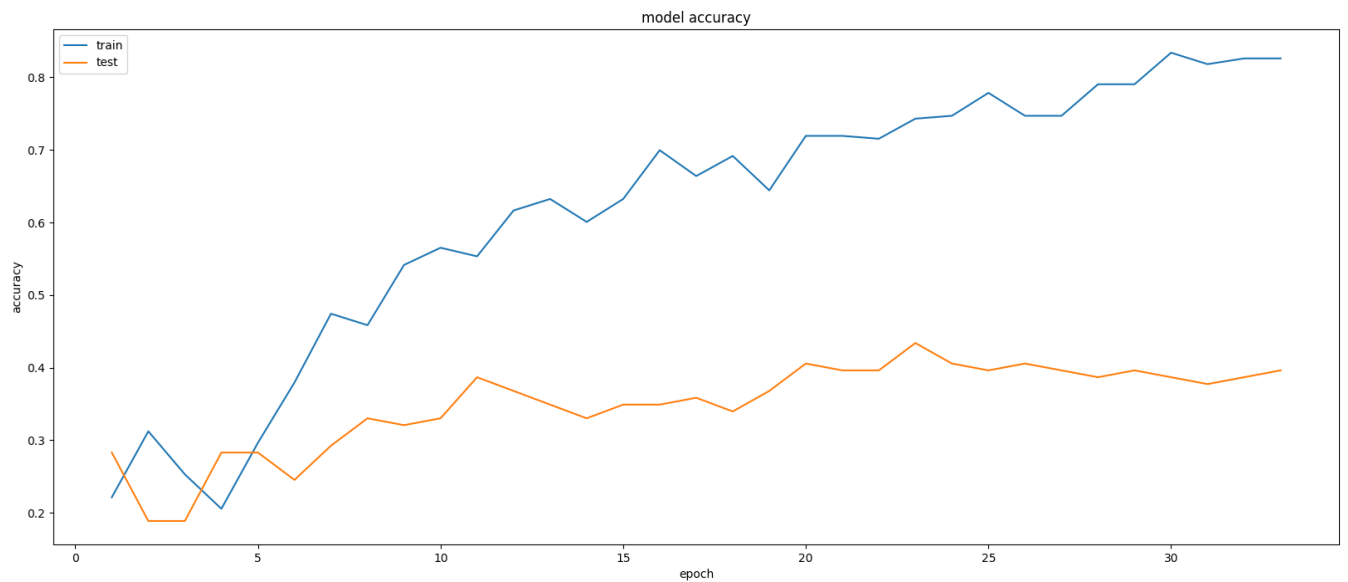
O treinamento foi realizado em **camada densa**, alterando os parâmetros para se buscar os melhores resultados.

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

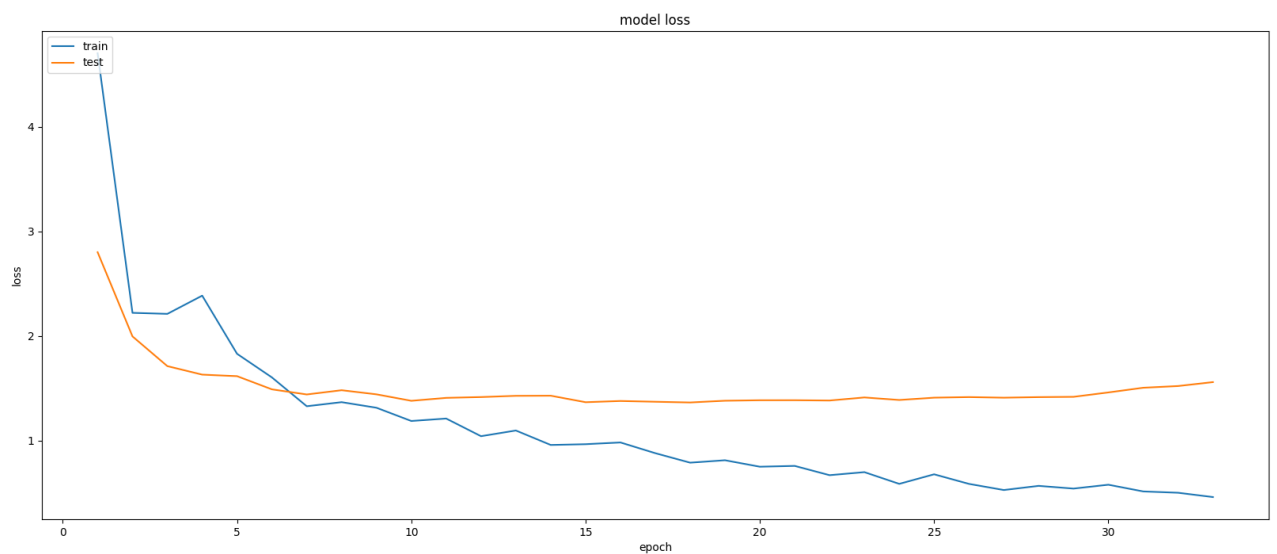
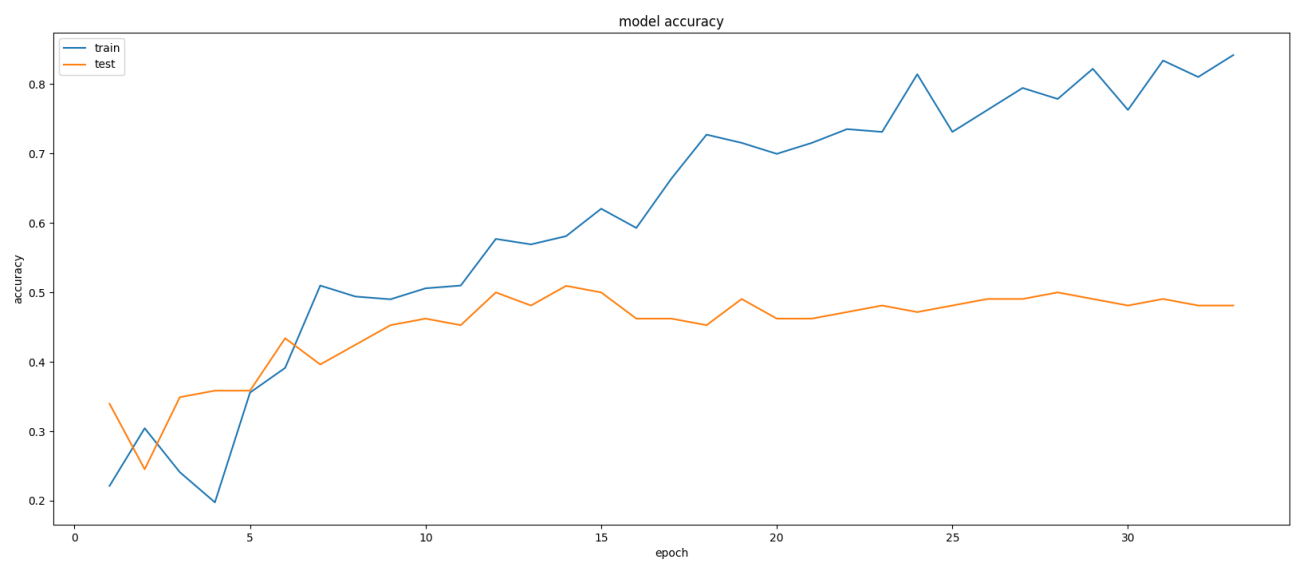
Accuracy: 0.4339622641509434

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

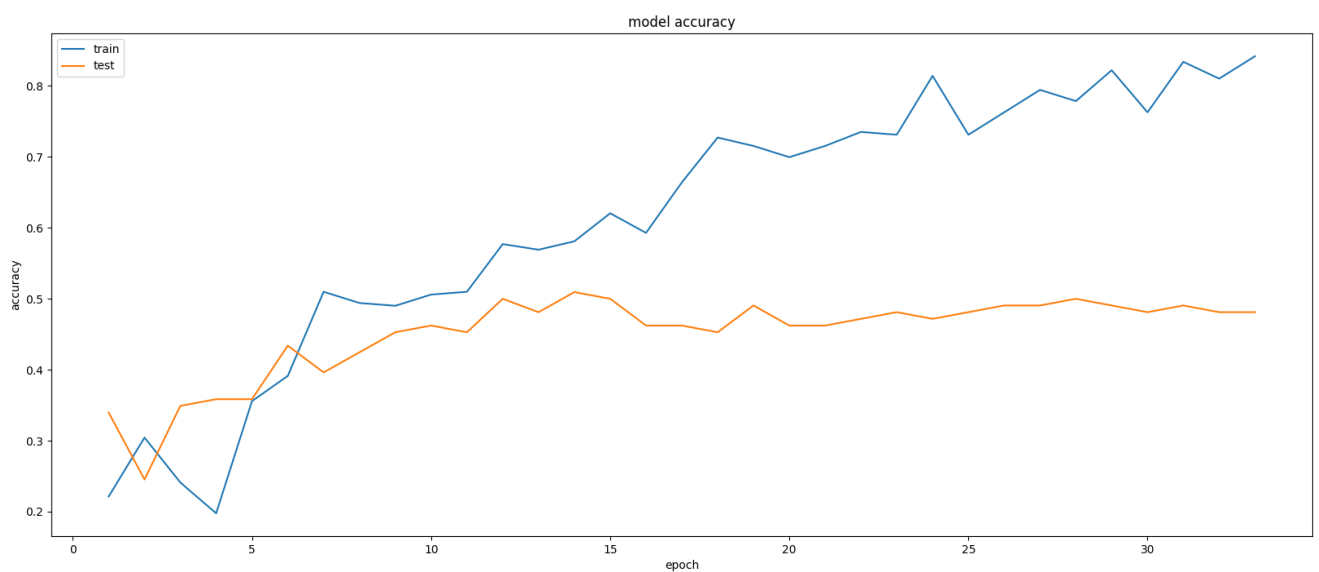
Accuracy: 0.5377358490566038

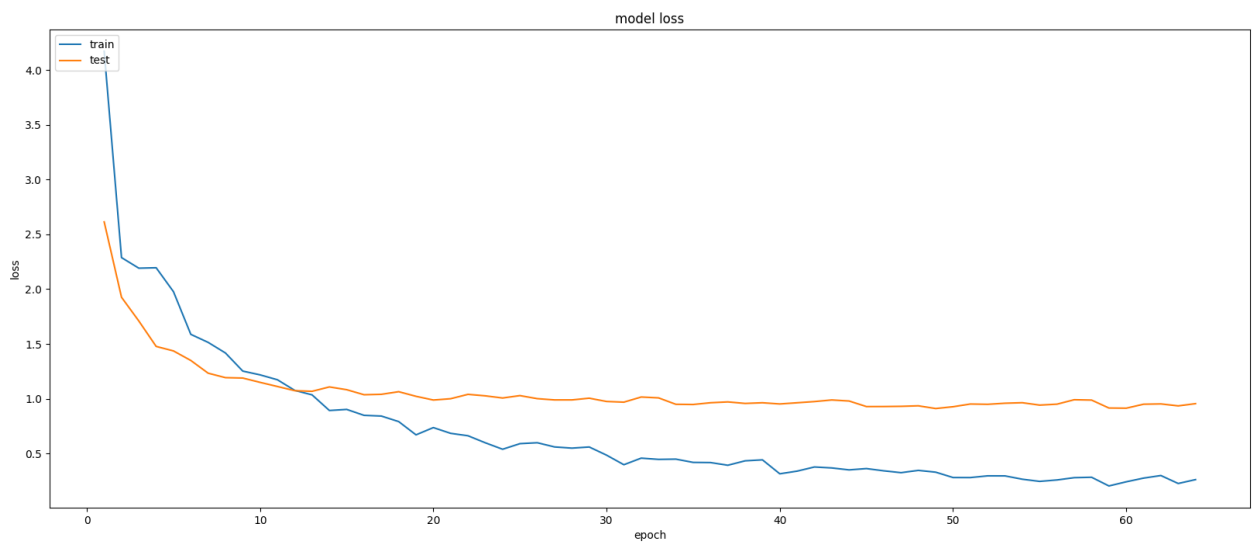
Resultados para as classes:

	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**

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

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

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

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 não cortadas

Número de Bach: 32

Paciência: 15

Accuracy: 0.5566037735849056

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.

## 4. 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.