

Representação e classificação utilizando IA

...

Erick Eckermann Cardoso



Objetivo

- O objetivo consiste treinar uma rede Neural para classificar as imagens de uma base em 6 classes, cada classe 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.

Especificação

- 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 da classificação com KNN com $k = 1$.



Modelos

- 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.
- Com o objetivo de aumentar a acurácia, foram feitos testes com as imagens das bases normais e cortadas.

Resultados sem corte e sem treinamento

- VGG16
 - Numéro de Bach: 32
 - Paciência: 15
 - Accuracy: 0.37735849056603776
- ResNet50V2
 - Numéro de Bach: 32
 - Paciência: 15
 - Accuracy: 0.44339622641509435
-
- ResNet50
 - Numéro de Bach: 32
 - Paciência: 15
 - Accuracy: 0.5
- EfficientNetB4
 - Numéro de Bach: 32
 - Paciência: 15
 - Accuracy: 0.5566037735849056

Resultados com cortes e sem treinamentos

- VGG16
- Número de Bach: 32
- Paciência: 10
- Accuracy: 0.6037735849056604

- ResNet50V2
- Número de Bach: 32
- Paciência: 15
- Accuracy: 0.4528301886792453

- ResNet50
- Número de Bach: 32
- Paciência: 15
- Accuracy: 0.6226415094339622

- EfficientNetB4
- Número de Bach: 32
- Paciência: 15
- Accuracy: 0.5566037735849056

Rede treinada (ResNet50)

- 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.

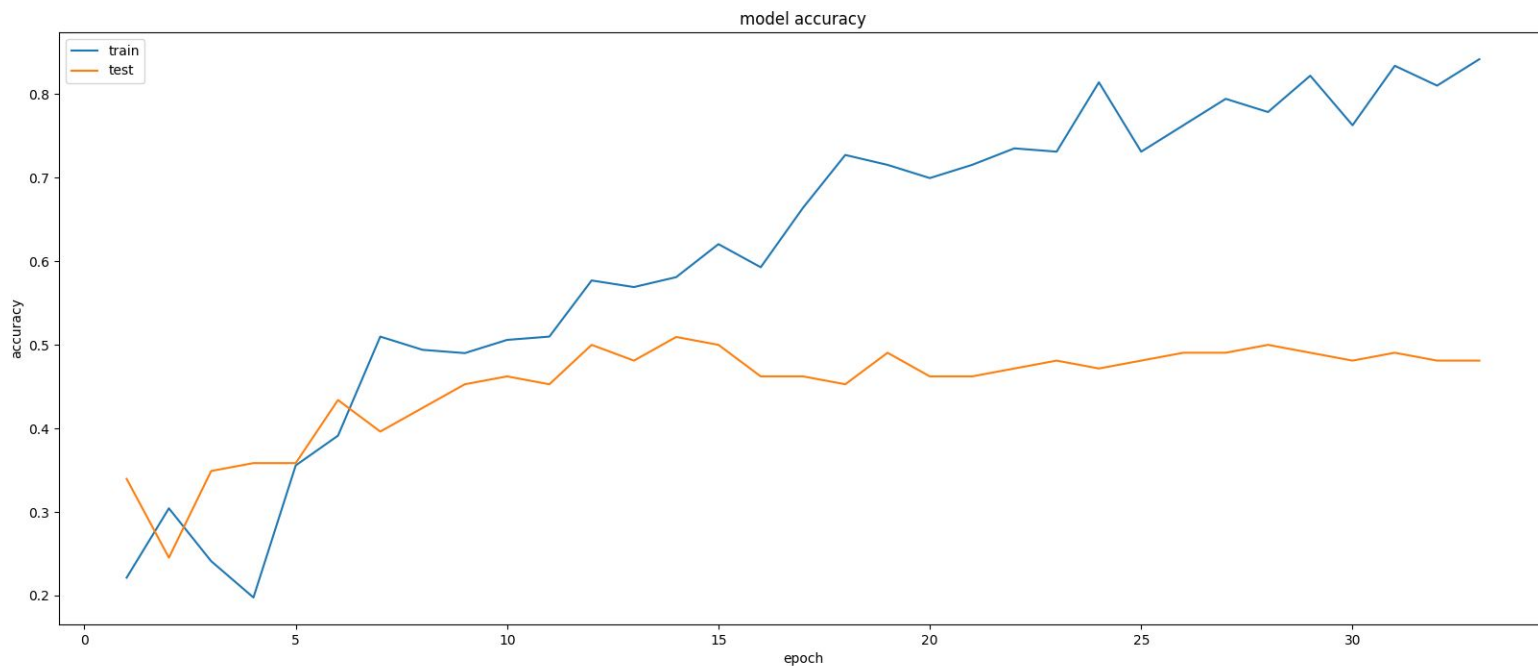


Resultados

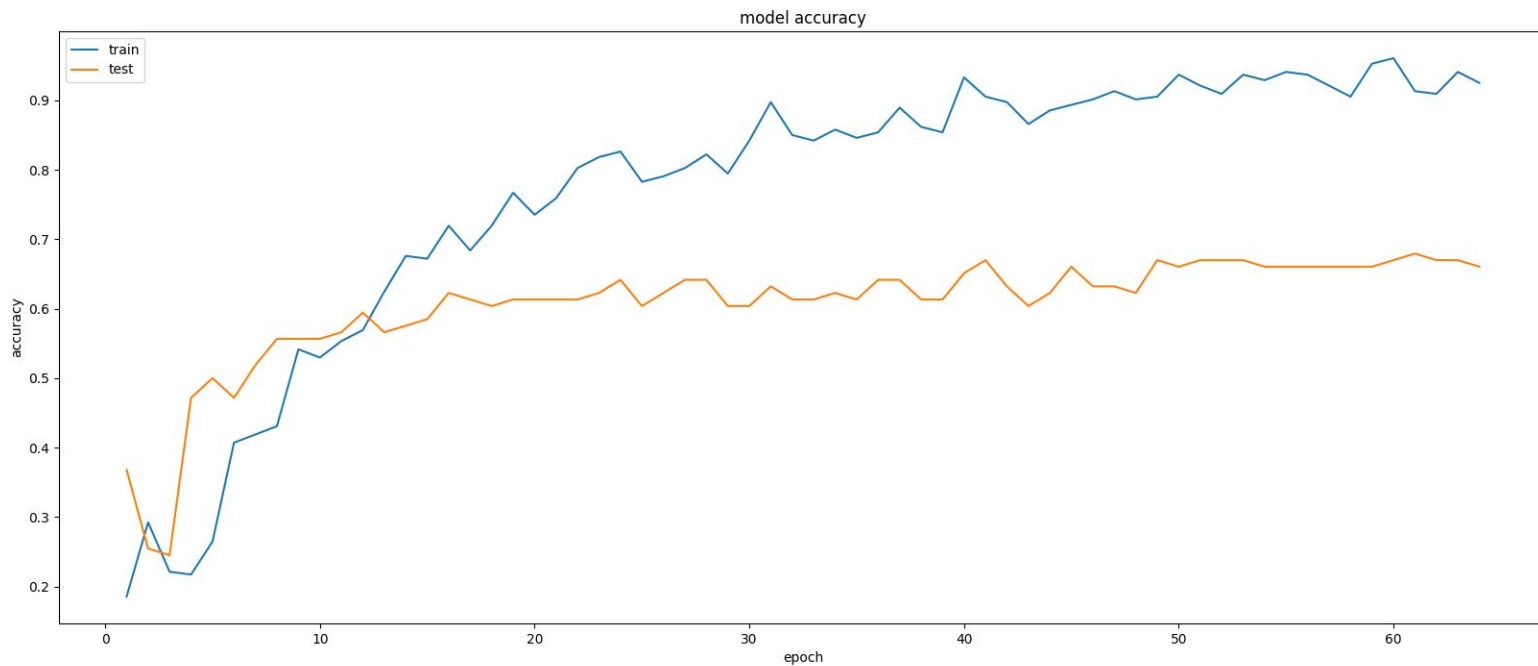
- Base sem imagens cortadas
- Paciência: 15
- Número de Bach: 32
- Accuracy: 0.5377358490566038

- Base com imagens cortadas
- Paciência: 15
- Número de Bach: 32
- Accuracy: 0.7264150943396226

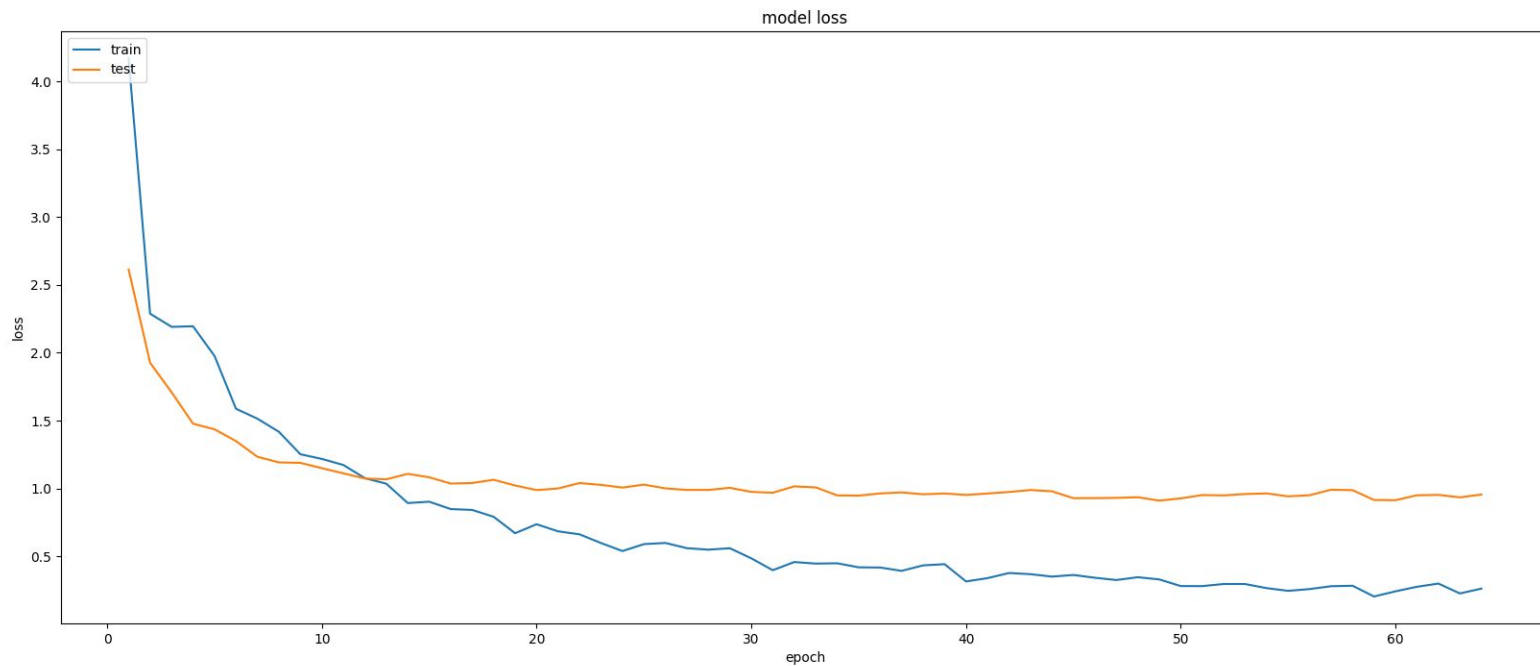
Acurácia para base não modificada



Acurácia base modificada



Loss base modificada





Obrigado