

APLICAÇÃO DE DATA MINING
PARA IDENTIFICAÇÃO DE

RAÇAS DE CACHORROS

ORIENTADOR:

ROBERTO SANTOS

AUTORES:

BRUNO FRANCISCO DE LIMA SOUSA

DIANE SANTOS DOS REIS

ELENINHA ALICE SANTOS

GABRIEL SILVA EZEQUIEL

VITOR FERNANDES PEREIRA













FACULDADE:

FACULDADE IMPACTA DE TECNOLOGIA

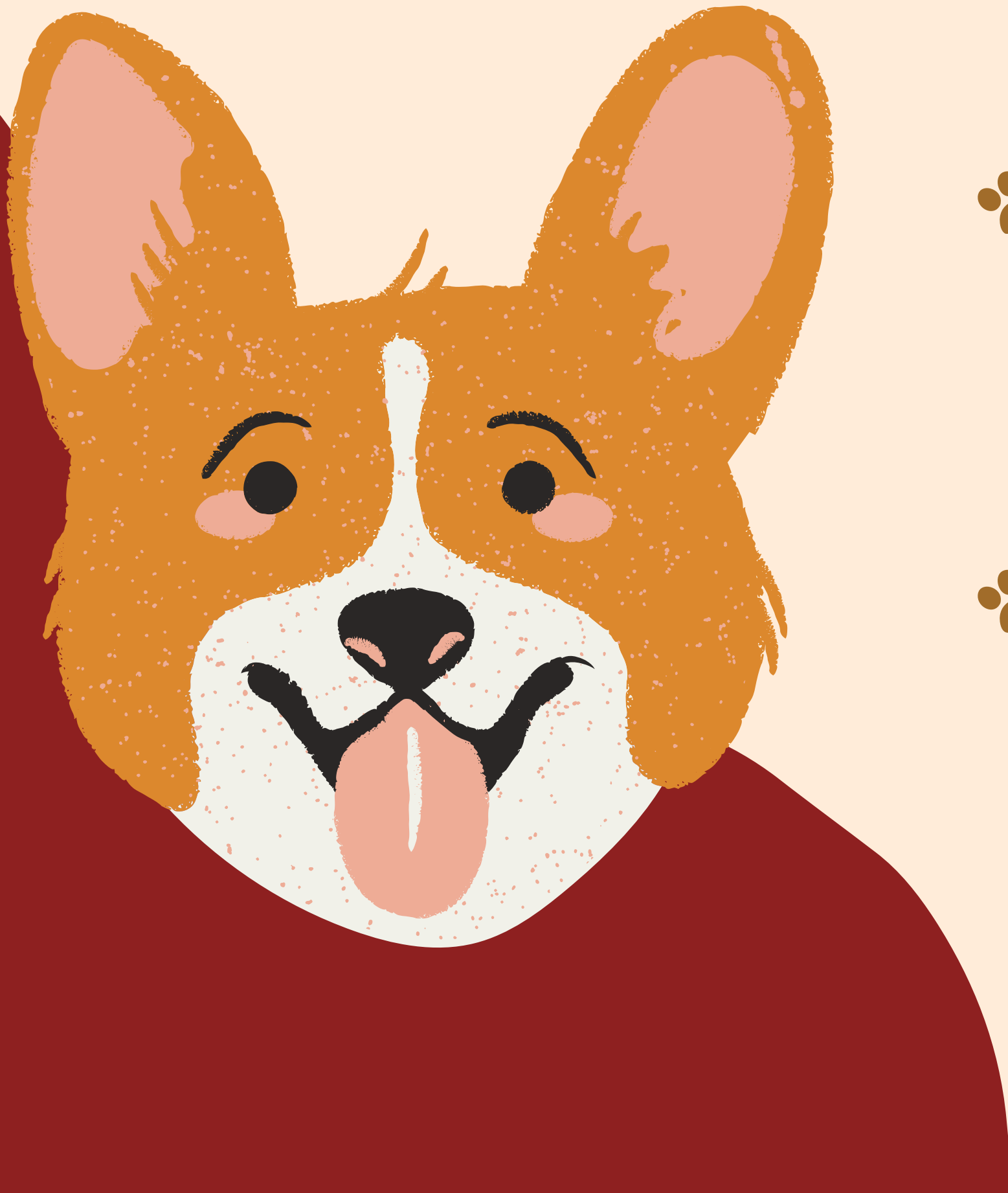


Resumo do Trabalho



-  MOTIVAÇÃO
-  PROPOSTA
-  DESCRIÇÃO DA SOLUÇÃO
-  METODOLOGIA
-  ENTENDIMENTO DOS DADOS
-  SEPARAÇÃO DOS DADOS
-  MODELAGEM
-  METRICAS DE AVALIAÇÃO
-  PREDIÇÃO
-  DESEMPENHO POR RAÇAS
-  IMPLANTAÇÃO DA SOLUÇÃO
-  CONCLUSÃO

Motivação



- 🐾 Nos últimos anos, houve um salto significativo na adoção/compra de cães. Junto com esse crescimento, também se inspirou dificuldade na escolha da raça, levando a escolhas equivocadas na hora da aquisição
- 🐾 Muitos adotantes acabam escolhendo seus animais de estimação pela aparência física, porte ou moda, em vez de considerar também as principais características da raça. Essa dificuldade de reconhecimento e distinção de raças pode levar a frustrações, levando à devolução e até ao abandono dos animais de estimação.

Proposta

Desenvolvimento de uma aplicação de reconhecimento de raças através de imagens, com a intenção de auxiliar potenciais adotantes na seleção responsável de seus animais de estimação.



Descrição da Solução



Desenvolvimento de um modelo para reconhecimento de raças de cachorros através de imagem;



O modelo pode ser disponibilizado por meio de API;



Se integrado à aplicativos de adoção ou redes de PetShop, o adotante conseguirá fazer o upload da foto do cachorro que deseja descobrir a raça.

Entendimento dos dados

Foram coletadas 5.317 imagens, abrangendo 20 raças distintas, com aproximadamente 260 imagens para cada raça. Este conjunto foi construído utilizando imagens amplamente estendidas em termos de ambientes, condições de iluminação, ângulos de captura e contextos.

As imagens foram obtidas a partir do estudo realizado por Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao e Li Fei-Fei, que desenvolveu um conjunto de dados inovador para Categorização de Imagens de Alta Resolução. O estudo foi apresentado no First Workshop on Fine-Grained Visual Categorization (FGVC), parte da IEEE Conference on Computer Vision and Pattern Recognition (CVPR) em 2011.

Fonte: <https://www.kaggle.com/datasets/jessicali9530/stanford-dogs-dataset>



Separação dos dados

As imagens foram separadas em conjuntos de 70% para treino, e 15% para validação e teste, resultando em:

- Conjunto de Treino com 3717 imagens
- Conjunto de Validação com 800 imagens
- Conjunto de Teste com 800 imagens

```
[ ] # Pré-processamento de dados
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
train_set = train_datagen.flow_from_directory('/content/Images', target_size=(256, 256), batch_size=32, class_mode='categorical')

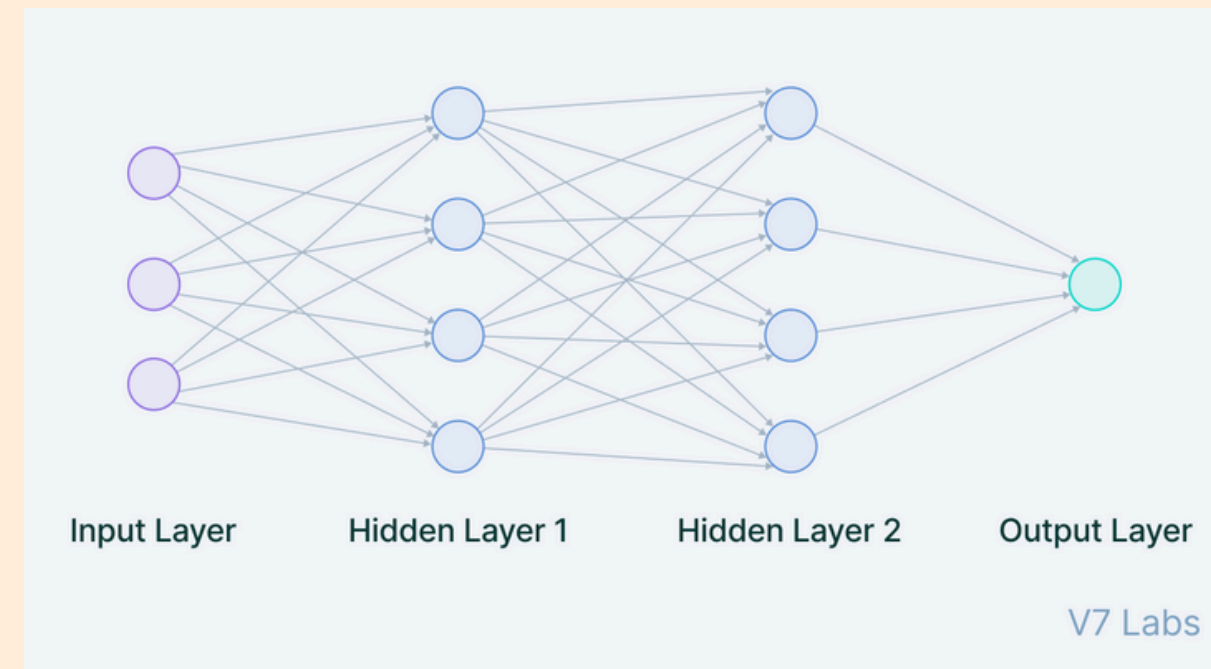
validation_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
validation_set = validation_datagen.flow_from_directory('/content/Images_validation', target_size=(256, 256), batch_size=32, class_mode='categorical')

test_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_set = test_datagen.flow_from_directory('/content/Images_teste', target_size=(256, 256), batch_size=32, class_mode='categorical')
```

Modelagem

Construção do modelo CNN - Rede Neural Convolucional, utilizando Sequential

Sequential é especialmente útil para problemas onde é necessário classificar imagens em categorias específicas, realizar reconhecimento de padrões ou efetuar a categorização de imagens.



```
model = Sequential()

model.add(Conv2D(64, (3, 3), input_shape=(256, 256, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=2))
```


Métricas de Avaliação

Em geral, no conjunto de teste, a acurácia foi considerada a métrica de maior importância para o problema.

```
[ ] #Avaliação do Modelo

# Obter previsões do conjunto de teste
predictions = model.predict(test_set)
predicted_classes = np.argmax(predictions, axis=1)

# Obter rótulos verdadeiros do conjunto de teste
true_classes = test_set.classes

# Calcular as métricas
accuracy = accuracy_score(true_classes, predicted_classes)
precision = precision_score(true_classes, predicted_classes, average='weighted')
recall = recall_score(true_classes, predicted_classes, average='weighted')
f1 = f1_score(true_classes, predicted_classes, average='weighted')

print(f'Acurácia: {round(accuracy, 2)}')
print(f'Precisão: {round(precision, 2)}')
print(f'Recall: {round(recall, 2)}')
print(f'F1-Score: {round(f1, 2)}')
```



```
26/26 [=====] - 11s 431ms/step
Acurácia: 0.78
Precisão: 0.65
Recall: 0.70
F1-Score: 0.73
```

Predição

A função predição prepara uma imagem para a entrada do modelo, faz uma predição usando o modelo treinado, mapeia a predição para a raça correspondente e exibe a imagem original com a raça prevista.

```
def predicao(image_path):  
    img = image.load_img(image_path, target_size=(256, 256))  
    img = image.img_to_array(img)  
    img = np.expand_dims(img, axis=0)  
  
    prediction = model.predict(img)  
  
    class_indices = train_set.class_indices  
  
    predicted_breed = [breed for breed, index in class_indices.items() if index == np.argmax(prediction)][0]  
    img = image.load_img(image_path)  
    img.show()  
  
    print(f'A raça prevista é: {predicted_breed}')
```

Predição

```
✓  
1s [43] predicao(r'/content/teste_final/Galgo_afegao1.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao2.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao3.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao4.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao5.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao6.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao7.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao8.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao9.jpg')  
      predicao(r'/content/teste_final/Galgo_afegao10.jpg')
```

```
1/1 [=====] - 0s 19ms/step  
A raça prevista é: n02088094-Galgo_afegao  
1/1 [=====] - 0s 17ms/step  
A raça prevista é: n02088094-Galgo_afegao  
1/1 [=====] - 0s 18ms/step  
A raça prevista é: n02088094-Galgo_afegao  
1/1 [=====] - 0s 17ms/step  
A raça prevista é: n02090622-borzoi  
1/1 [=====] - 0s 17ms/step  
A raça prevista é: n02088094-Galgo_afegao  
1/1 [=====] - 0s 18ms/step  
A raça prevista é: n02086079-Pekinese  
1/1 [=====] - 0s 18ms/step  
A raça prevista é: n02088094-Galgo_afegao  
1/1 [=====] - 0s 18ms/step  
A raça prevista é: n02088094-Galgo_afegao  
1/1 [=====] - 0s 18ms/step  
A raça prevista é: n02088094-Galgo_afegao  
1/1 [=====] - 0s 19ms/step  
A raça prevista é: n02088094-Galgo_afegao
```





80%



80%



80%



80%



90%



70%



80%



90%



70%



90%



80%



80%



80%



80%



80%



90%



90%



90%



80%



90%

Implantação da solução

- 🐾 Distribuída por API
- 🐾 Fácil integração com APP's e Sites



Conclusão



- Foram realizados diversos testes para as 20 raças de cães. O modelo de parceria, em média, de 5 a 6 raças a cada 10 tentativas quando o conteúdo da imagem era apenas o cão. Em imagens em que os donos estavam bem próximos, foi observado uma dificuldade em prever as raças;
- Não foram encontrados projetos que tenham utilizado o mesmo conjunto de dados deste trabalho. Contudo, quando comparados com projetos pares (reconhecimento de cães e gatos) no site Kaggle, os resultados encontrados foram
- Pode-se adaptar este projeto para que ele seja replicado, também, para outros tipos de animais.

Obrigado!

