

# O USO DE MÁQUINA ENSINÁVEL APLICADA PARA RECONHECIMENTO PESSOAS E OBJETOS EM LUGARES CONTROLADOS

Alexandre de Araújo

1

TP558 - Tópicos Avançados em Aprendizado de Máquina

06/2024

# SUMÁRIO

1. Introdução
2. Objetivos
3. Métodos
4. Solução Proposta Utilizando RNA
5. Resultados
6. Conclusão
7. Trabalhos Futuros

# INTRODUÇÃO

- Atualmente, o reconhecimento de pessoa ou então a identificação de objetos por meio de dados de vídeo, tem se tornado essencial na visão computacional.
- Tais como, estação de metrô, aeroportos, indústria e ambientes que necessita de monitoramento permanente ou então em setores de fluxo de mercadorias.

# OBJETIVOS

- Apresentar ao usuário inexperiente as noções de básica de implementação de maquina ensináveis. Aplicando o modelo para classificar imagens de pessoas e objetos em lugares controlados.

# MÉTODOS

- Ferramenta Teachable Machine Google;
- Método baseado em Rede Neural Convolucional de camadas profundas;

# SOLUÇÃO PROPOSTA UTILIZANDO RNA

## ▪ Coleta de dados

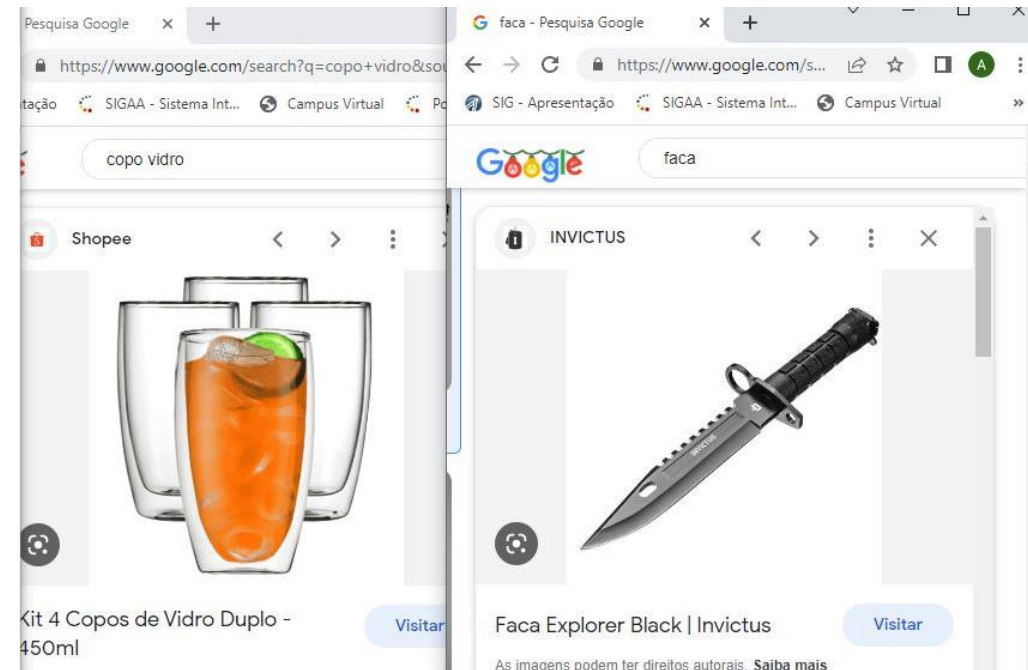
### ➤ Modelo A (Face)

- 450 Imagens de fase com mascaras
- 450 Imagens de fase sem mascaras

### ➤ Modelo B (Objetos)

- 180 Imagens de objetos  
(Arma, Cigarro, Copo de vidro, Droga, Faca e Taco)

Figura 1 - Selecionando imagem para bando de dados



Fonte: Elaborado pelo autor.

# SOLUÇÃO PROPOSTA UTILIZANDO RNA

- Redimensionamento de imagem
  - Tamanho 160 pixels x 160 pixels

Figura 2 - Redimensionamento de imagem



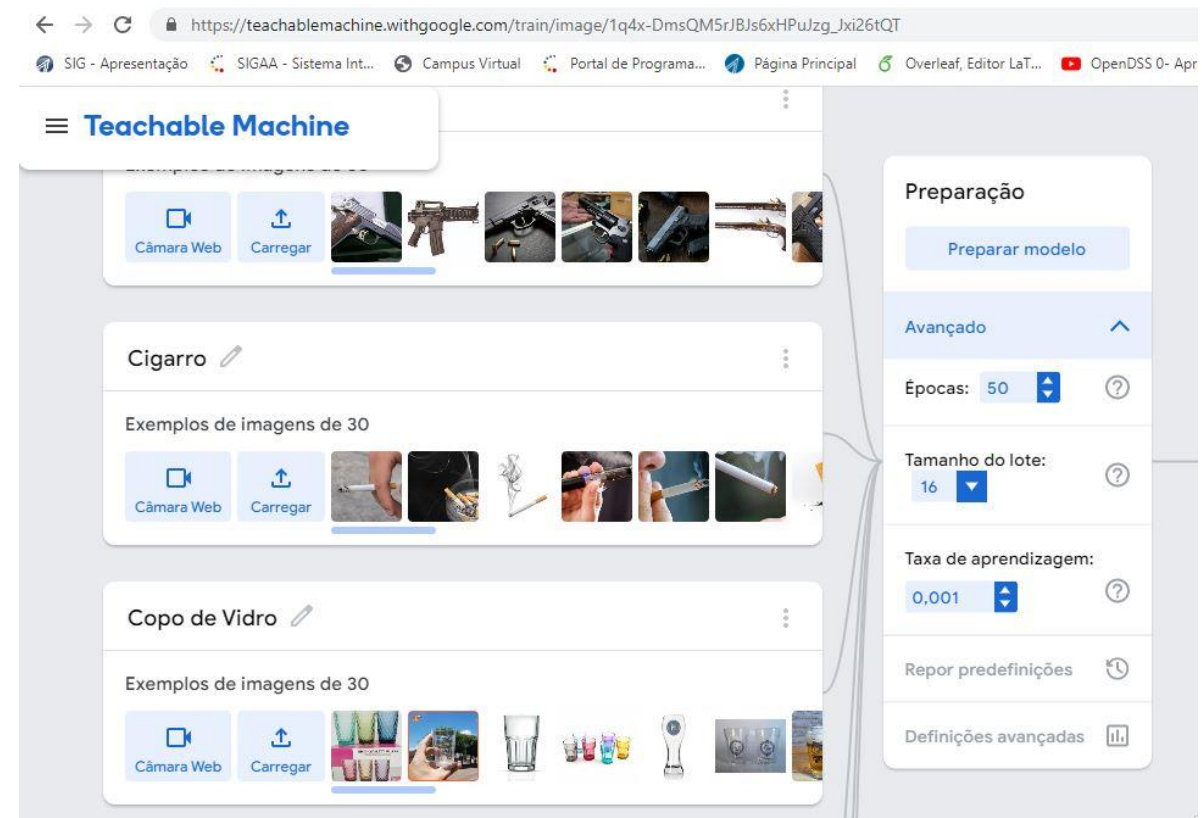
Fonte: Elaborado pelo autor.

# SOLUÇÃO PROPOSTA UTILIZANDO RNA

## ▪ Teachable Machine Google

- Definir classe
- Carregar imagens
  - Épocas
  - Tamanho do lote
  - Taxa de aprendizagem
- Iniciar preparação

Figura 3 - Pagina web Teachable Machine Google



Fonte: Elaborado pelo autor.



# SOLUÇÃO PROPOSTA UTILIZANDO RNA

- Script

```
# Bibliotecas
from keras.models import load_model
from PIL import Image, ImageOps #Install pillow instead of PIL
import numpy as np
from google.colab.patches import cv2_imshow
import cv2

# Carrega o modelo
model = load_model('keras_model.h5', compile=False)

class_names = open('labels.txt', 'r').readlines() # Carregar os rótulos

# Crie a matriz da forma certa para alimentar o modelo keras
# O 'comprimento' ou número de imagens que você pode colocar no array é
# Determinado pela primeira posição na tupla de forma, neste caso 1..
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

image = Image.open('01.jpg').convert('RGB') # Substitua isso pelo caminho para sua imagem

# Redimensionar a imagem para 224x224 com a mesma estratégia do TM2:
# Redimensionando a imagem para pelo menos 224x224 e recortando do centrar
size = (224, 224)
image = ImageOps.fit(image, size, Image.LANCZOS)
```

# SOLUÇÃO PROPOSTA UTILIZANDO RNA

```
image_array = np.asarray(image) # Transforma a imagem em um array numpy

normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1 # Normaliza a imagem

image_array = np.asarray(image) # Transforma a imagem em um array numpy

data[0] = normalized_image_array # Carrega a imagem no array

model.layers[0] # acessa a primeira camada:

# percorre as camadas uma a uma e imprime o nome de cada camada no console:
for layer in model.layers:
    print(layer.name)

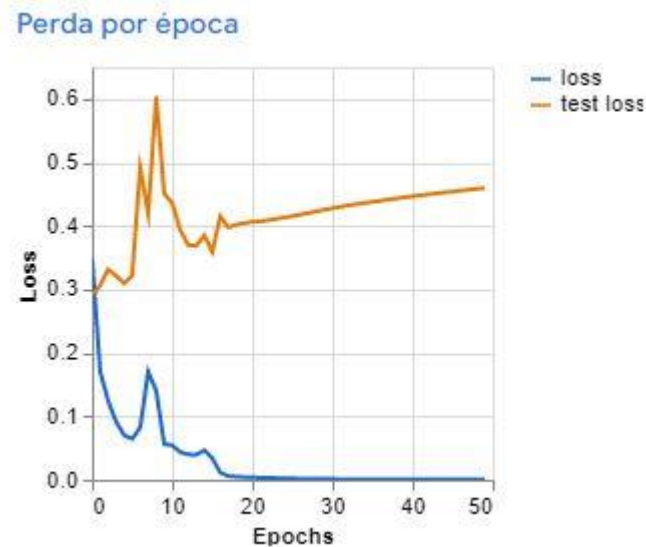
# Executa a inferência
prediction = model.predict(data)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

print('Class:', class_name, end='')
print('Confidence score:', confidence_score)

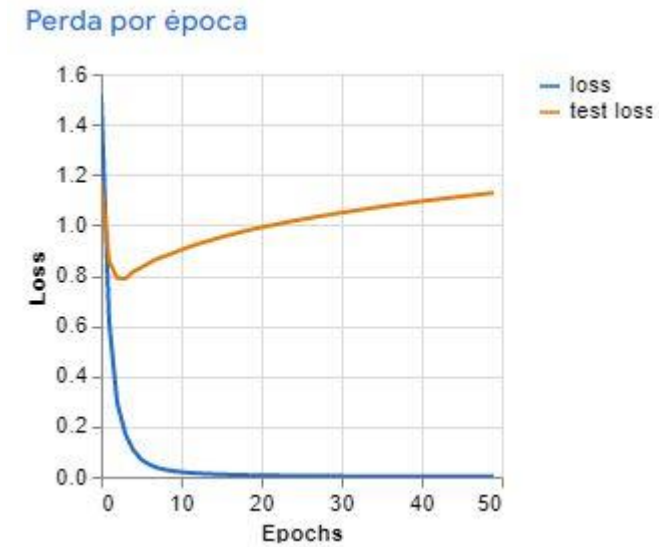
img = cv2.imread("01.jpg")
cv2_imshow(img)
```

# SOLUÇÃO PROPOSTA UTILIZANDO RNA

- Resultados dos treinamentos
  - Perda por época



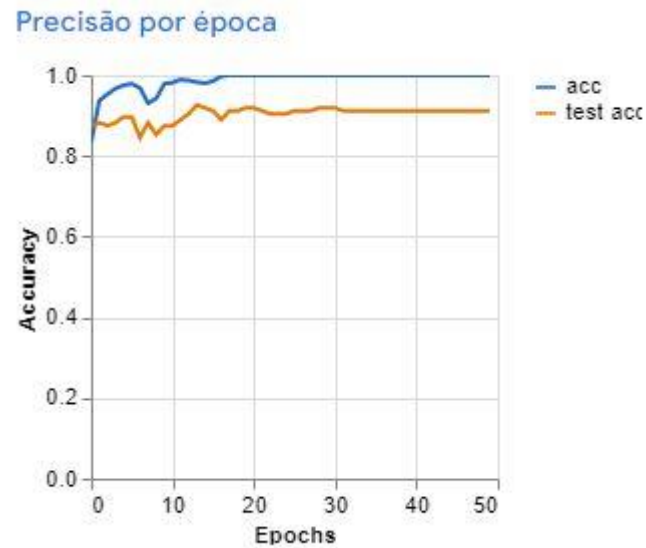
Modelo A



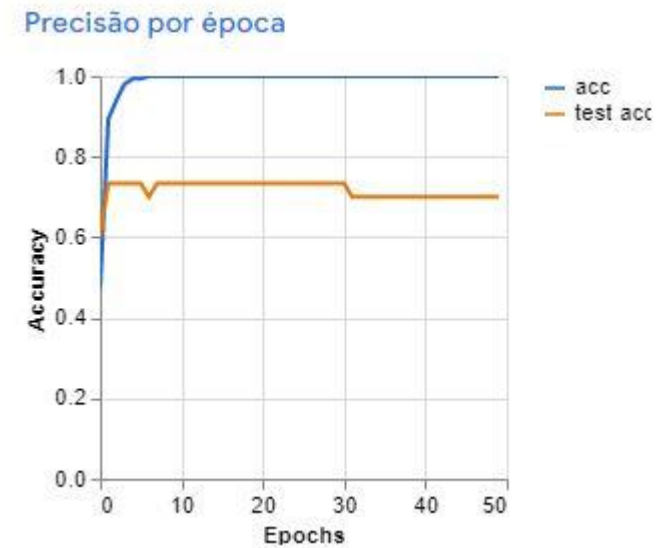
Modelo B

# SOLUÇÃO PROPOSTA UTILIZANDO RNA

- Resultados dos treinamentos
  - Precisão por época



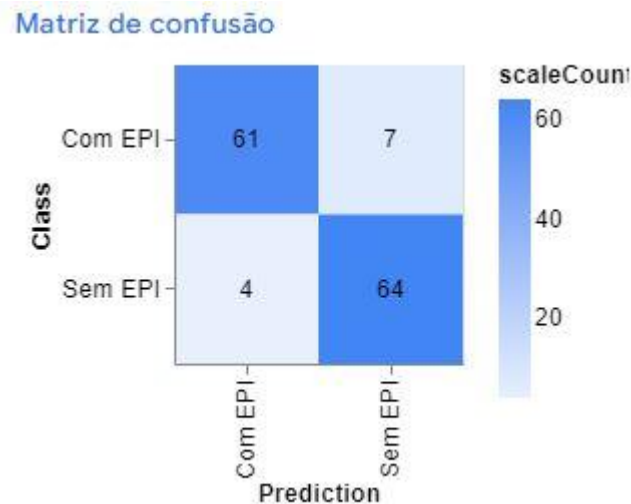
Modelo A



Modelo B

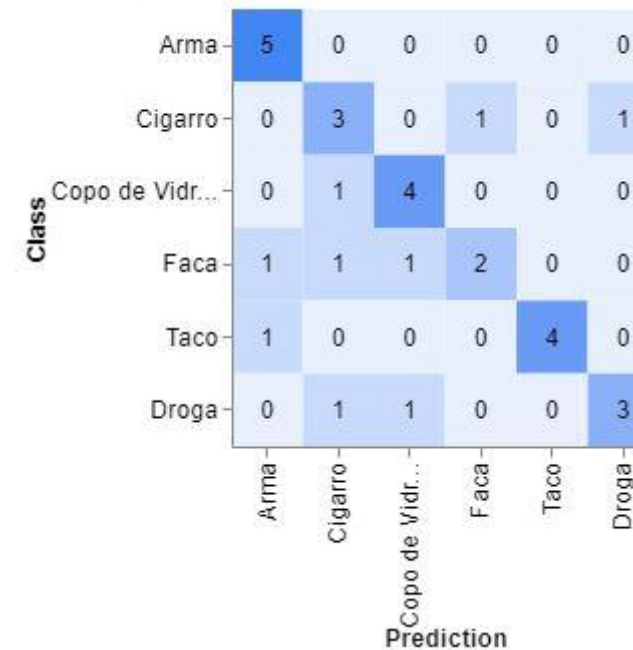
# SOLUÇÃO PROPOSTA UTILIZANDO RNA

- Resultados dos treinamentos
  - Matriz de confusão



Modelo A

Matriz de confusão



Modelo B

# RESULTADOS

- Teste modelo A

Classe	Pontuação de Confiança	Precisão
Com Máscara	100%	0.91
Sem Máscara	99%	0.91

```
print('Class:', class_name, end='')  
print('Confidence score:', confidence_score)
```

```
img = cv2.imread("02")  
cv2_imshow(img)
```

```
sequential_5  
sequential_7  
1/1 [=====] - 1s 915ms/step  
Class: 1 Sem Máscara  
Confidence score: 0.99989355
```



Fonte: Colab.

# RESULTADOS

- Teste modelo B

Classe	Pontuação de Confiança	Precisão
Arma	100%	1.00
Cigarro	99%	0.60
C. Vidro	95%	0.80
Droga	86%	0.60
Faca	98%	0.40
Taco	82%	0.80

```
print('Class:', class_name, end='')  
print('Confidence score:', confidence_score)
```

```
img = cv2.imread("02")  
cv2_imshow(img)
```

```
sequential_1  
sequential_3  
1/1 [=====] - 1s 1s/step  
Class: 0 Arma  
Confidence score: 0.9616441
```



Fonte: Colab.

# CONCLUSÃO

- Propusemos usar uma plataforma livre de maquina de ensinável. Na qual as pessoas poderão implementar o aprendizado de maquina e determinar os parâmetros CNN para máxima exatidão, precisão e resultado de sensibilidade sem altera todo modelo.



# TRABALHO FUTURO

- **Desenvolvimento de modelos híbridos:** A combinação de diferentes arquiteturas de redes neurais, como CNNs, RNNs e Transformers, pode melhorar a precisão no reconhecimento de ações complexa.
- **Melhoria da qualidade de vídeos e a robustez dos modelos:** Métodos de pré-processamento de vídeos para melhorar a qualidade da imagem e reduzir ruídos, assim como o uso de Redes Adversárias Generativas (GANs) para criar dados de treinamento mais variados e realistas.

OBRIGADO