

# **Redes Neurais e Deep Learning**

## Leitura de Placas de Veículos (OCR para Trânsito)

Efrain Marcelo, Pietro Silva, Willian Moura  
Redes Neurais e Deep Learning - Prof. Adrião

### O que são Redes Neurais com Deep Learning?

São modelos inspirados no cérebro humano que aprendem a identificar padrões complexos em dados.

O Deep Learning usa múltiplas camadas (deep = profundo) para processar informações em etapas.

#### **Exemplo**

*Como uma criança que aprende a reconhecer letras e números em uma placa — só que aqui é feito por uma máquina.*

### O que é OCR?

OCR (Optical Character Recognition) = Reconhecimento Óptico de Caracteres

O sistema “lê” textos em imagens, como placas de carro, usando visão computacional.

### Aplicado em:

- Trânsito e fiscalização (Lombadas eletrônicas, multas, entre outros)
- Estacionamentos automáticos (Natal Shopping utiliza esta tecnologia)
- Pedágios eletrônicos (Utilizado em pedágios e por empresas como “Sem Parar”, “Veloe”, entre outros)
- Controle de acesso em condomínios (Abertura de portão automático ao reconhecer a placa do carro cadastrado)



### Exemplo:

Um exemplo popular é a utilização em aparelhos celulares para scanner de texto.

Como funciona o sistema de leitura de placas com Deep Learning?

### Etapas principais:

#### 1) Captura da imagem

- Câmera registra o veículo passando.

#### 2) Detecção da placa

- Uma rede neural detecta onde está a placa (usando, por exemplo, YOLOv5).

#### 3) Segmentação dos caracteres

- A imagem é cortada para separar letras e números.

#### 4) Reconhecimento dos caracteres (OCR)

- Outra rede neural classifica cada caractere com base em um banco de dados.



### Exemplo prático

#### Tecnologias envolvidas:

- CNN (Convolutional Neural Network): usada para detectar e reconhecer caracteres.
- YOLO (You Only Look Once): identifica rapidamente a posição da placa na imagem.
- Tesseract OCR (opcional): motor OCR treinável da Google.

#### **Aplicação real:**

*Câmeras de monitoramento urbano identificam veículos em tempo real e comparam com bancos de dados de veículos roubados.*

### Benefícios da aplicação

- Agilidade: leitura instantânea
- Precisão: mesmo com placas sujas ou mal iluminadas
- Economia: reduz a necessidade de fiscalização manual
- Segurança: automação em pedágios, estacionamentos e controle urbano



### Desafios

- Placas ilegíveis (danificadas, sujas)
- Iluminação ruim ou excesso de reflexo
- Variações regionais de placas
- Treinamento do modelo com dados variados e rotulados



[Matéria completa](#)

### Conclusão

A leitura automática de placas com redes neurais deep learning representa um avanço importante na automação do trânsito e da segurança pública.

Combinando visão computacional, OCR e redes neurais, é possível criar sistemas inteligentes que veem, entendem e agem em tempo real.

### Código

```
import cv2
import pytesseract
import numpy as np
from tensorflow.keras.models import load_model

# Caminho para o executável do Tesseract (ajuste conforme seu sistema)
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe' # para Windows

# Carrega imagem
img = cv2.imread('placa.jpg')
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Pré-processamento: conversão e filtro
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (5, 5), 0)
_, thresh = cv2.threshold(blur, 100, 255, cv2.THRESH_BINARY)

# Detectar contornos (simples para placas)
contours, _ = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

# Suponha que a maior área é a placa
plate_img = None
for cnt in contours:
    x, y, w, h = cv2.boundingRect(cnt)
    aspect_ratio = w / float(h)
    if 2 < aspect_ratio < 6 and w > 100:
        plate_img = img[y:y+h, x:x+w]
        break
```

continuação >

```
# Suponha que a maior área é a placa
plate_img = None
for cnt in contours:
    x, y, w, h = cv2.boundingRect(cnt)
    aspect_ratio = w / float(h)
    if 2 < aspect_ratio < 6 and w > 100:
        plate_img = img[y:y+h, x:x+w]
        break

# Mostra imagem da placa detectada
if plate_img is not None:
    cv2.imshow("Placa Detectada", plate_img)

# OCR com Tesseract
placa_texto = pytesseract.image_to_string(plate_img, config='--psm 8')
print("Texto reconhecido (Tesseract):", placa_texto.strip())

# Aqui você poderia aplicar um modelo Keras pré-treinado, ex:
# modelo = load_model('modelo_placa.h5')
# Entrada: imagem segmentada, normalizada e redimensionada
# Resultado: caracteres identificados

cv2.waitKey(0)
cv2.destroyAllWindows()
else:
    print("Nenhuma placa detectada.")
```

**GitHub:**

<https://github.com/UB-Mannheim/tesseract/wiki>