

## Trabalho Final

### Módulo – Deep Learning I

**Descrição:** o presente trabalho destina-se a avaliação da aprendizagem dos conteúdos ministrados durante o módulo de Deep Learning I, aplicando técnicas de regressão com redes neurais convolucionais e redes neurais artificiais.

**Objetivo:** praticar os conteúdos estudados durante o módulo por meio da implementação aplicada em solução de regressão a partir de um conjunto de dados utilizando a técnica de redes neurais convolucionais.

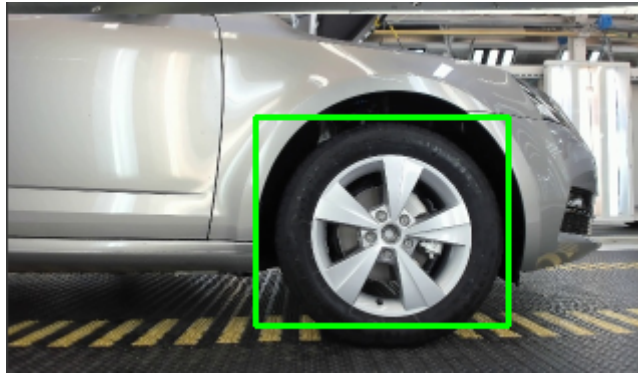
**Entrega:** a entrega deve ser realizada até o dia 11/03/2023 às 23:59 pela plataforma Google Classroom. Caso deseje versionar o código, o mesmo deve ser colocado na plataforma github e o link do repositório deve ser enviado na plataforma Google Classroom.

#### **Problema:**

Imagine que você foi designado para resolver um problema de detecção de objetos utilizando regressão, e para isso você deve utilizar redes neurais para resolver o problema.

Trata-se de um problema de detecção de um único objeto por imagem. Ou seja, o objeto alvo aparece uma única vez por imagem.

O objeto alvo é a roda de um carro e você deverá definir a marcação de onde esse objeto se localiza na imagem, conforme Figura 1.



**Figura 1.** Roda de veículo detectada por rede neural. Fonte: Growdev.

## Dataset

Os dados utilizados no problema são imagens de uma base de dados disponível neste [link](#). Contudo, apenas uma parte do dataset é utilizado e está disponível nesse segundo [link](#). Existem duas pastas:

- *images*: pasta que contém as imagens a serem processadas.
- *annotations*: pasta que contém os arquivos com as coordenadas das boxes.

## Imagens

As imagens contidas na pasta *images* possuem tamanho 640 x 360. Esse tamanho pode ser modificado de acordo com a necessidade de execução, ou seja, para melhorar o desempenho ou reduzir a utilização de memória e consequentemente possibilitar a utilização de mais imagens para treinamento e/ou teste.

Se desejar reduzir o tamanho das imagens você pode usar o método *resize* da biblioteca *opencv* (*cv2*).

## Anotações

As caixas estão anotadas em arquivos *txt* dentro da pasta *annotations*. O nome dos arquivos é o mesmo para imagens e anotações, apenas alterando-se as extensões, por exemplo:

CWD1500\_train\_f000\_camB.jpg

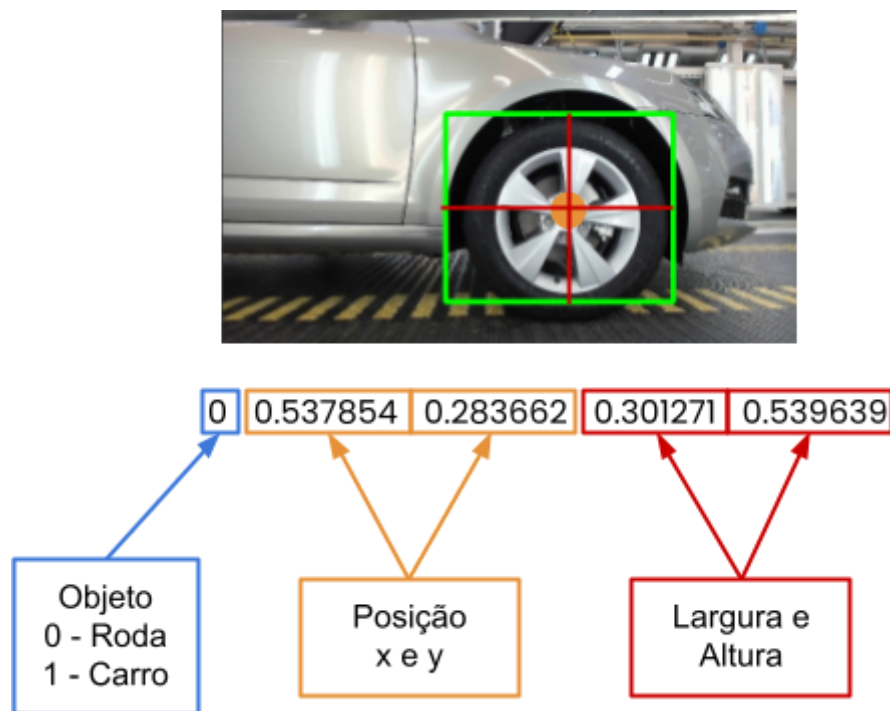
CWD1500\_train\_f000\_camB.txt

Dentro de cada arquivo de anotação existem as coordenadas das boxes de cada imagem respectiva. Um exemplo pode ser visto a seguir.

Existem 5 valores em cada linha, o primeiro indica o tipo de objeto que a caixa marca, valor 1 é carro, valor 0 é roda. Para esse problema vamos treinar o algoritmo para reconhecer a roda apenas.

Os 4 números seguintes indicam as coordenadas de localização da roda no formato YOLO. Ou seja, os dois primeiros números são as coordenadas x e y do centro do objeto e as duas últimas são a largura e altura do objeto. A Figura 2 ilustra como funciona essa declaração.

<classe><x><y><largura><altura>



**Figura 2.** Formato de anotações YOLO. Fonte: Growdev.

As coordenadas já estão normalizadas entre 0 e 1, portanto não importa em que resolução está a imagem, você conseguirá reconstruir a box nela.

### **Processamento**

Para detectar as rodas, você deve criar uma rede neural capaz de realizar uma regressão para determinar as coordenadas de onde se localiza o objeto (nesse caso, a roda) na imagem.

Utilize redes convolucionais para ajudar na precisão do modelo.

Separe as imagens de modo que 20% seja para teste e 80% para treinamento. A quantidade total, fica a seu critério.

Imagens que não possuem o objeto alvo (roda) aparecendo, devem ser ignoradas e não devem ser levadas em conta no treinamento e nem no teste. Ao final você deve aplicar a rede às imagens de teste e exibir a caixa de marcação nessas imagens.

### **Formato de entrega:**

- Você deverá entregar o código completo do programa.
- Não envie os dados utilizados.

### **Dicas**

- Procure organizar o código de forma que ele fique bem parametrizado.
- Procure balancear bem a quantidade de imagens, o tamanho das imagens e o tamanho da rede, para que não ocorra falta de memória (RAM ou de GPU).

Bom trabalho!