Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT213
Aluno:

Relatório do Laboratório 9 - Detecção de Objetos

1 Breve Explicação em Alto Nível da Implementação

Para a implementação da rede, seguiu-se os passos propostos para as camadas. Destaca-se o skip realizado na rede como particularidade ao modelo.

1.1 Sumário do Modelo

Sumário comprovando funcionamento.

```
"""Model: "ITA_YOLO"
_________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=========================================================================================
input_1 (InputLayer)         [(None, 120, 160, 3)]     0           []

conv_1 (Conv2D)              (None, 120, 160, 8)       216         ['input_1[0][0]']

norm_1 (BatchNormalization)  (None, 120, 160, 8)       32          ['conv_1[0][0]']

leaky_relu_1 (LeakyReLU)     (None, 120, 160, 8)       0           ['norm_1[0][0]']

conv_2 (Conv2D)              (None, 120, 160, 8)       576         ['leaky_relu_1[0][0]']

norm_2 (BatchNormalization)  (None, 120, 160, 8)       32          ['conv_2[0][0]']

leaky_relu_2 (LeakyReLU)     (None, 120, 160, 8)       0           ['norm_2[0][0]']

conv_3 (Conv2D)              (None, 120, 160, 16)      1152        ['leaky_relu_2[0][0]']

norm_3 (BatchNormalization)  (None, 120, 160, 16)      64          ['conv_3[0][0]']

leaky_relu_3 (LeakyReLU)     (None, 120, 160, 16)      0           ['norm_3[0][0]']

max_pool_3 (MaxPooling2D)    (None, 60, 80, 16)        0           ['leaky_relu_3[0][0]']

conv_4 (Conv2D)              (None, 60, 80, 32)        4608        ['max_pool_3[0][0]']

norm_4 (BatchNormalization)  (None, 60, 80, 32)        128         ['conv_4[0][0]']

leaky_relu_4 (LeakyReLU)     (None, 60, 80, 32)        0           ['norm_4[0][0]']

max_pool_4 (MaxPooling2D)    (None, 30, 40, 32)        0           ['leaky_relu_4[0][0]']
```

```
conv_5 (Conv2D)                 (None, 30, 40, 64)   18432       ['max_pool_4[0][0]']

norm_5 (BatchNormalization)     (None, 30, 40, 64)   256         ['conv_5[0][0]']

leaky_relu_5 (LeakyReLU)        (None, 30, 40, 64)   0           ['norm_5[0][0]']

max_pool_5 (MaxPooling2D)       (None, 15, 20, 64)   0           ['leaky_relu_5[0][0]']

conv_6 (Conv2D)                 (None, 15, 20, 64)   36864       ['max_pool_5[0][0]']

norm_6 (BatchNormalization)     (None, 15, 20, 64)   256         ['conv_6[0][0]']

leaky_relu_6 (LeakyReLU)        (None, 15, 20, 64)   0           ['norm_6[0][0]']

max_pool_6 (MaxPooling2D)       (None, 15, 20, 64)   0           ['leaky_relu_6[0][0]']

conv_7 (Conv2D)                 (None, 15, 20, 128)  73728       ['max_pool_6[0][0]']

norm_7 (BatchNormalization)     (None, 15, 20, 128)  512         ['conv_7[0][0]']

leaky_relu_7 (LeakyReLU)        (None, 15, 20, 128)  0           ['norm_7[0][0]']

conv_skip (Conv2D)              (None, 15, 20, 128)  8192        ['max_pool_6[0][0]']

conv_8 (Conv2D)                 (None, 15, 20, 256)  294912      ['leaky_relu_7[0][0]']

norm_skip (BatchNormalization)  (None, 15, 20, 128)  512         ['conv_skip[0][0]']

norm_8 (BatchNormalization)     (None, 15, 20, 256)  1024        ['conv_8[0][0]']

leaky_relu_skip (LeakyReLU)     (None, 15, 20, 128)  0           ['norm_skip[0][0]']

leaky_relu_8 (LeakyReLU)        (None, 15, 20, 256)  0           ['norm_8[0][0]']

concat (Concatenate)            (None, 15, 20, 384)  0           ['leaky_relu_skip[0][0]',
 'leaky_relu_8[0][0]']

conv_9 (Conv2D)                 (None, 15, 20, 10)   3850        ['concat[0][0]']

==================================================================================================
Total params: 445,346
Trainable params: 443,938
Non-trainable params: 1,408
--------------------------------------------------------------------------------------------------
None"""
```

1.2 ALgoritmo de deteção

Resumidamente, a implementação das classes foi feita a seguir, seguindo o orientado.

```
def detect(self, image):
        image = self.preprocess_image(image)
        output = self.network.predict(image)
```

```
        ball_detection, post1_detection, post2_detection = self.process_yolo_output(output)
        return ball_detection, post1_detection, post2_detection

def preprocess_image(self, image):
        image = cv2.resize(image, (160, 120), interpolation=cv2.INTER_AREA)
        image = np.array(image)
        image = image / 255.0
        image = np.reshape(image, (1, 120, 160, 3))
        return image

def process_yolo_output(self, output):

                (...)

        # Ecntrando o indice de maxima probabilidade
        tb = np.amax(output[:,:,0])
        arr2D = output[:,:,0]
        coord = np.where(arr2D == np.amax(arr2D))
        ib = coord[0]
        jb = coord[1]
        # Aplicando o índice
        txb = output[ib,jb,1]
        tyb = output[ib,jb,2]
        twb = output[ib,jb,3]
        thb = output[ib,jb,4]

                (...)

        # ball, output
        pb = sigmoid(tb)
        xb = coord_scale*(jb+sigmoid(txb))
        yb = coord_scale*(ib+sigmoid(tyb))
        wb = 640*pwb*np.exp(twb)
        hb = 640*phb*np.exp(thb)

        return ball_detection, post1_detection, post2_detection
```

2 Figuras Comprovando Funcionamento do Código

2.1 Detecção de Objetos com YOLO

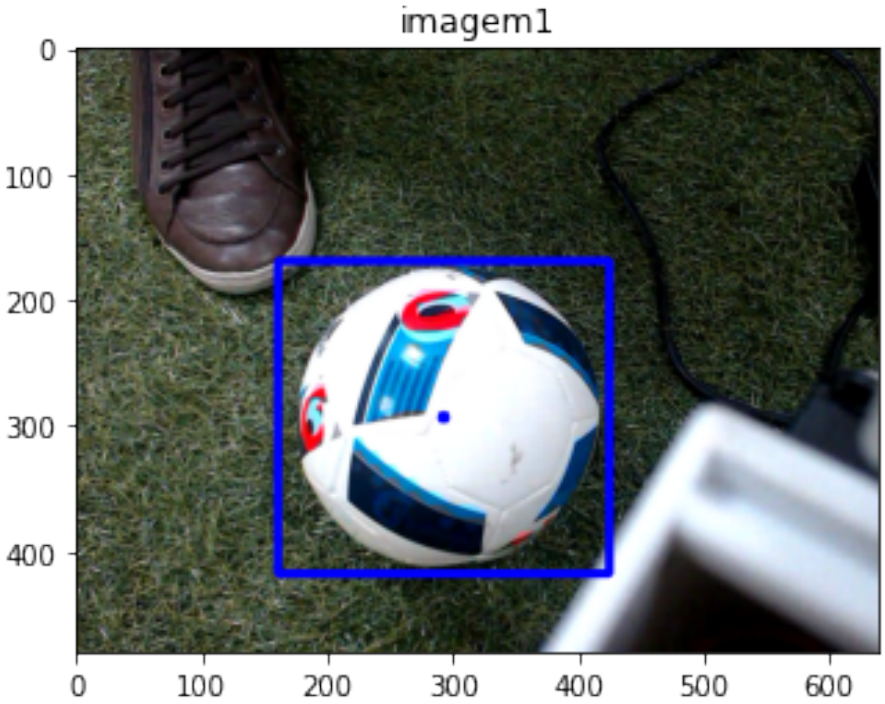

Figura 1: Bola encontrada e localizada pela rede neural.

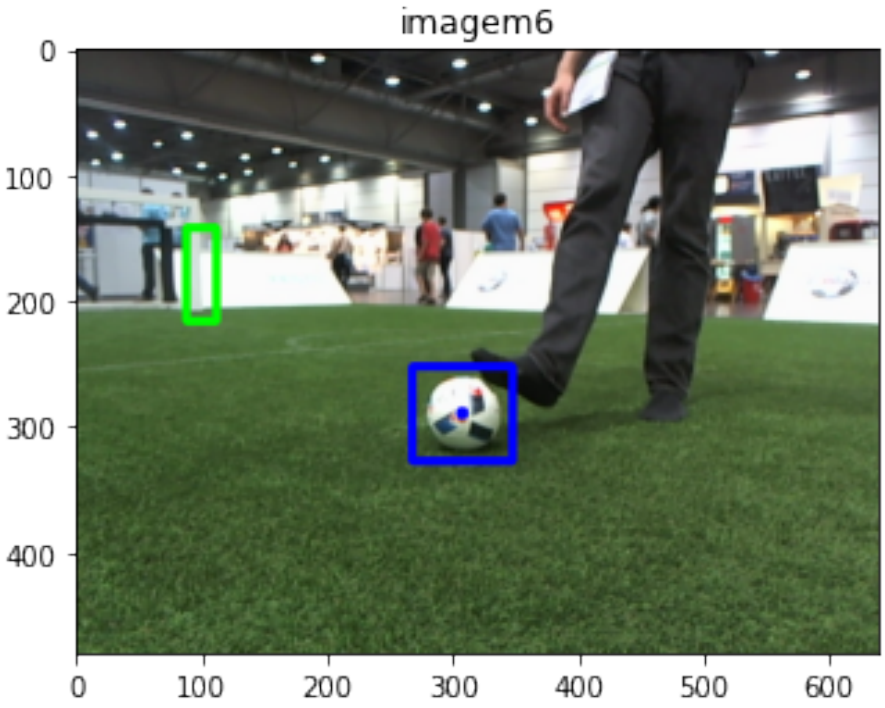

Figura 2: Bola detectada e trave ao fundo.]

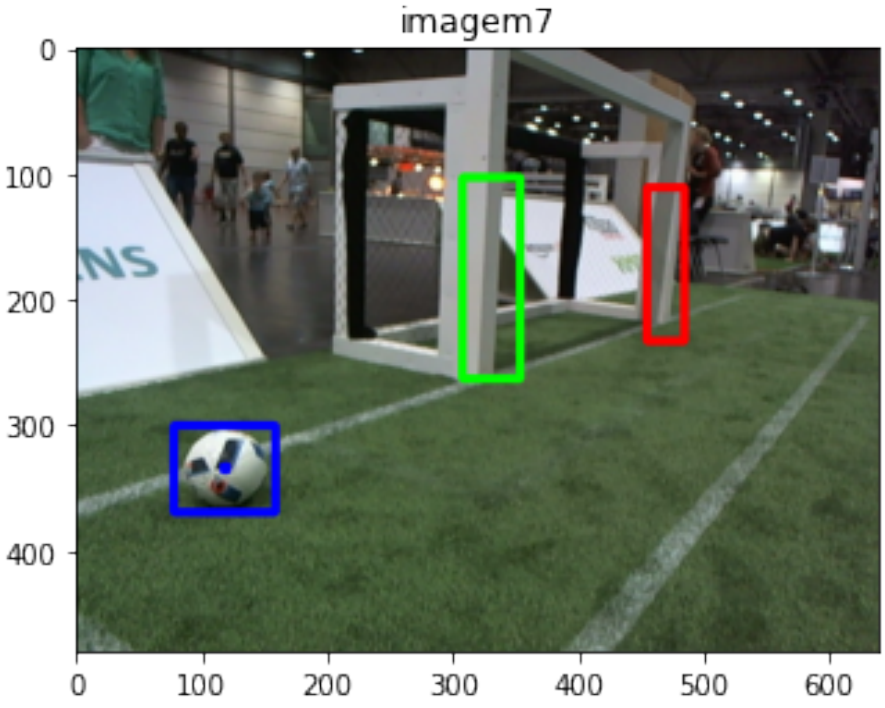

Figura 3: Bola e duas traves detectadas.

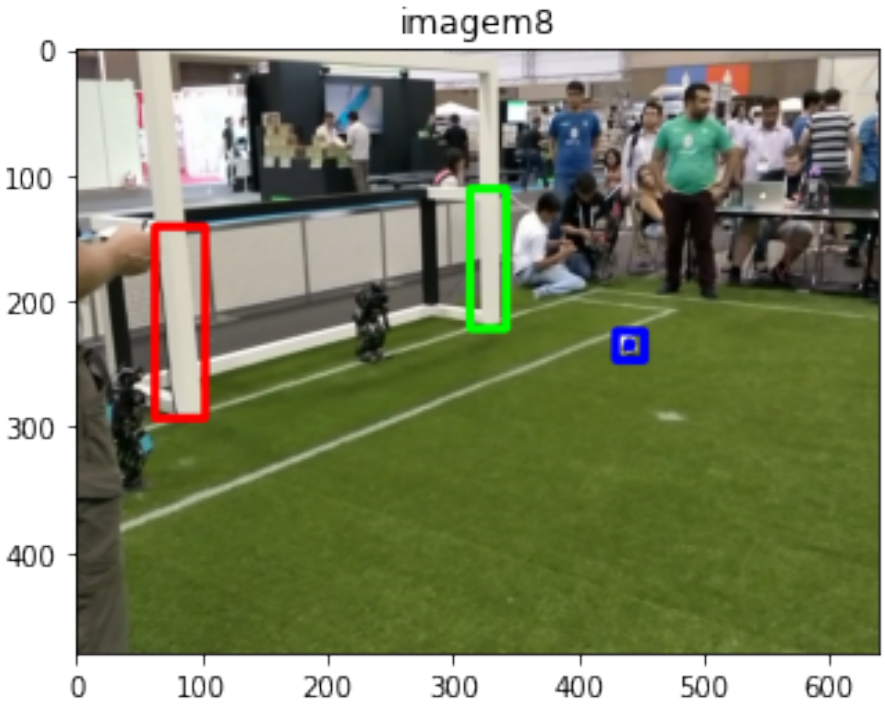

Figura 4: Bola e duas traves detectadas, mais distante.

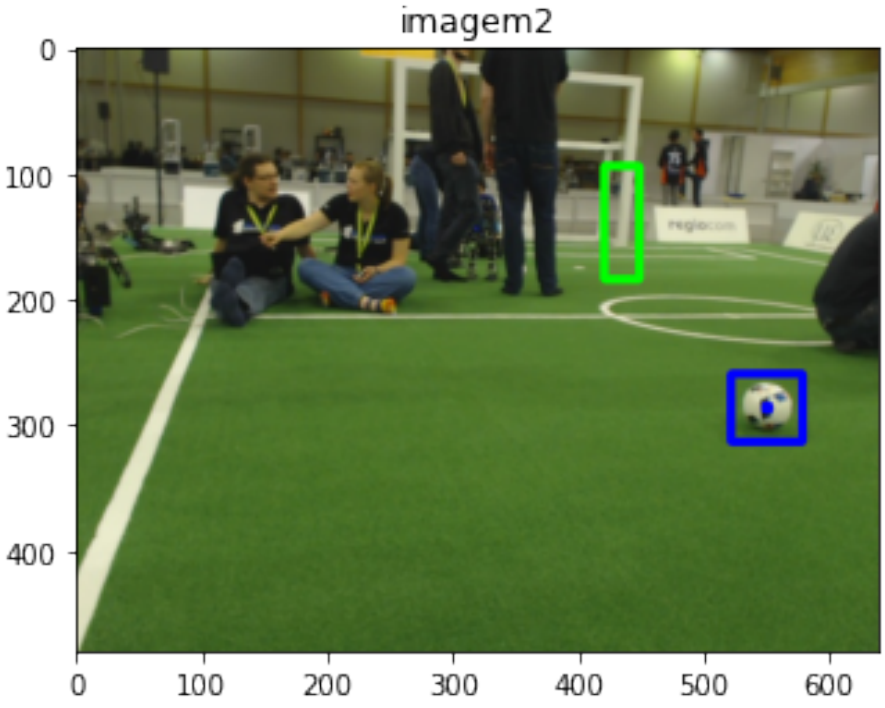

Figura 5: Bola e trave detectada com pessoas no campo (que não foram detectadas).

3 Discussão

A implementação da rede em cadamadas obteve sumário igual ao proposto e registrado no roteiro.

A implementação correta do pré- e pós-processamento mostram-se de fundamental importância. Durante erros na impelmentação, ao trocar p_w com p_h, os centros dos objetos eram detectados, porém os formatos se mostravam deformados.

Por fim, a localização se mostrou efetiva, com os exemplos citados chamando atenção apenas no tamanho das traves ser limitado (não selecionando toda a altura da trave, por exemplo).