

Relatório do Laboratório 9 - Detecção de Objetos

1 Breve Explicação em Alto Nível da Implementação

O objetivo deste laboratório é implementar e aplicar a arquitetura de rede neural convolucional YOLO (You Only Look Once) para realizar detecção e classificação de objetos em tempo real. Especificamente, a tarefa consiste em identificar e localizar com precisão a bola e as traves em imagens capturadas pela câmera de um robô humanoide, cenário típico de aplicações em robótica autônoma e visão computacional para jogos de robôs (robot soccer). A rede YOLO se destaca por sua capacidade de realizar a detecção de múltiplos objetos em uma única inferência, combinando alta acurácia com baixa latência, sendo, portanto, adequada para sistemas embarcados que exigem processamento em tempo real. Durante o experimento, o modelo treinado é utilizado para prever, a partir das imagens de entrada, as coordenadas e probabilidades de detecção dos objetos de interesse, permitindo sua posterior utilização por módulos de tomada de decisão e controle do robô. A estrutura da rede está representada abaixo:

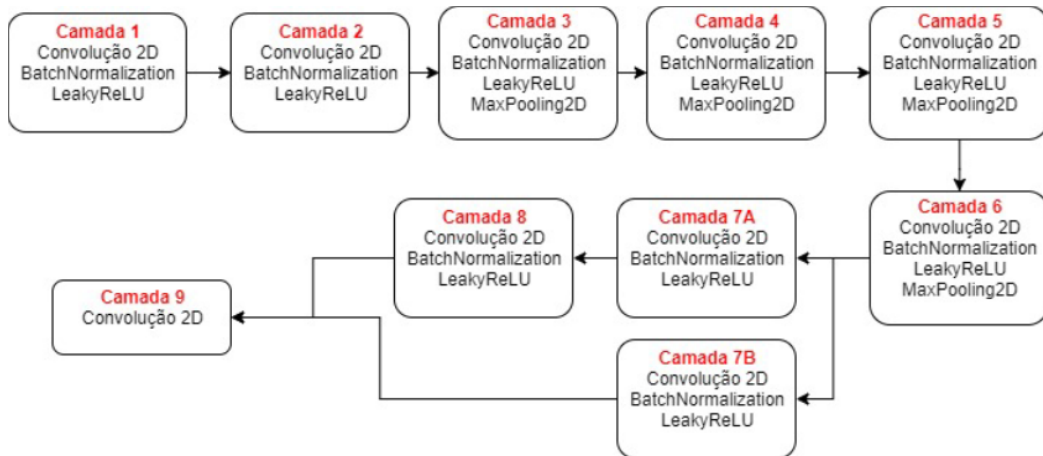


Figura 1: Evolução da Acurácia por Época nos Conjuntos de Treinamento e Validação.

Tabela 1: sumário da arquitetura da rede usada para detecção de objetos no futebol de robôs.

Camada	Tipo	Saída	Parâmetros
input_1	InputLayer	(None, 120, 160, 3)	0
conv_1	Conv2D	(None, 120, 160, 8)	216
norm_1	BatchNormalization	(None, 120, 160, 8)	32
leaky_relu_1	LeakyReLU	(None, 120, 160, 8)	0
conv_2	Conv2D	(None, 120, 160, 8)	576
norm_2	BatchNormalization	(None, 120, 160, 8)	32
leaky_relu_2	LeakyReLU	(None, 120, 160, 8)	0
conv_3	Conv2D	(None, 120, 160, 16)	1152
norm_3	BatchNormalization	(None, 120, 160, 16)	64
leaky_relu_3	LeakyReLU	(None, 120, 160, 16)	0
max_pool_3	MaxPooling2D	(None, 60, 80, 16)	0
conv_4	Conv2D	(None, 60, 80, 32)	4608
norm_4	BatchNormalization	(None, 60, 80, 32)	128
leaky_relu_4	LeakyReLU	(None, 60, 80, 32)	0
max_pool_4	MaxPooling2D	(None, 30, 40, 32)	0
conv_5	Conv2D	(None, 30, 40, 64)	18432
norm_5	BatchNormalization	(None, 30, 40, 64)	256
leaky_relu_5	LeakyReLU	(None, 30, 40, 64)	0
max_pool_5	MaxPooling2D	(None, 15, 20, 64)	0
conv_6	Conv2D	(None, 15, 20, 64)	36864
norm_6	BatchNormalization	(None, 15, 20, 64)	256
leaky_relu_6	LeakyReLU	(None, 15, 20, 64)	0
max_pool_6	MaxPooling2D	(None, 15, 20, 64)	0
conv_7	Conv2D	(None, 15, 20, 128)	73728
norm_7	BatchNormalization	(None, 15, 20, 128)	512
leaky_relu_7	LeakyReLU	(None, 15, 20, 128)	0
conv_skip	Conv2D	(None, 15, 20, 128)	8192
conv_8	Conv2D	(None, 15, 20, 256)	294912
norm_skip	BatchNormalization	(None, 15, 20, 128)	512
norm_8	BatchNormalization	(None, 15, 20, 256)	1024
leaky_relu_skip	LeakyReLU	(None, 15, 20, 128)	0
leaky_relu_8	LeakyReLU	(None, 15, 20, 256)	0
concat	Concatenate	(None, 15, 20, 384)	0
conv_9	Conv2D	(None, 15, 20, 10)	3850

1.1 Sumário do Modelo

Model: "ITA_YOLO"			
Layer (type)	Output Shape	Param #	Connected to
input_layer_2 (InputLayer)	(None, 120, 160, 1)	0	-
conv_5 (Conv2D)	(None, 120, 160, 64)	1,728	input_layer_2[0][0]
norm_5 (BatchNormalizatio...	(None, 120, 160, 64)	256	conv_5[0][0]
leaky_relu_5 (LeakyReLU)	(None, 120, 160, 64)	0	norm_5[0][0]
max_pool_5 (MaxPooling2D)	(None, 120, 160, 64)	0	leaky_relu_5[0][0]
conv_6 (Conv2D)	(None, 120, 160, 64)	36,864	max_pool_5[0][0]
norm_6 (BatchNormalizatio...	(None, 120, 160, 64)	256	conv_6[0][0]
leaky_relu_6 (LeakyReLU)	(None, 120, 160, 64)	0	norm_6[0][0]
max_pool_6 (MaxPooling2D)	(None, 120, 160, 64)	0	leaky_relu_6[0][0]
conv_7A (Conv2D)	(None, 120, 160, 128)	73,728	max_pool_6[0][0]
norm_7A (BatchNormalizatio...	(None, 120, 160, 128)	512	conv_7A[0][0]
leaky_relu_7A (LeakyReLU)	(None, 120, 160, 128)	0	norm_7A[0][0]
conv_7B (Conv2D)	(None, 120, 160, 128)	73,728	max_pool_6[0][0]
conv_8 (Conv2D)	(None, 120, 160, 256)	294,912	leaky_relu_7A[0][0]
norm_7B (BatchNormalizatio...	(None, 120, 160, 128)	512	conv_7B[0][0]
norm_8 (BatchNormalizatio...	(None, 120, 160, 256)	1,024	conv_8[0][0]
leaky_relu_7B (LeakyReLU)	(None, 120, 160, 128)	0	norm_7B[0][0]
leaky_relu_8 (LeakyReLU)	(None, 120, 160, 256)	0	norm_8[0][0]
concat (Concatenate)	(None, 120, 160, 384)	0	leaky_relu_7B[0][0] leaky_relu_8[0][0]
conv_9 (Conv2D)	(None, 120, 160, 16)	3,856	concat[0][0]
Total params: 487,376 (1.86 MB)			
Trainable params: 486,096 (1.85 MB)			
Non-trainable params: 1,280 (5.00 KB)			

Figura 2: sumário da arquitetura da rede YOLO.

2 Figuras Comprovando Funcionamento do Código

2.1 Detecção de Objetos com YOLO

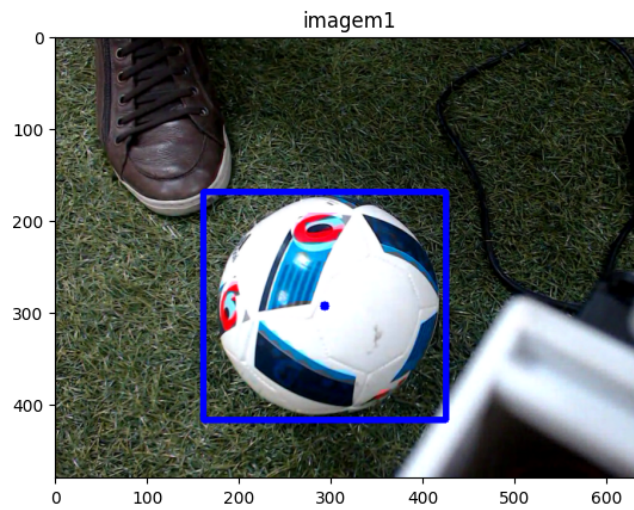


Figura 3: exemplo 1 de detecção da rede.

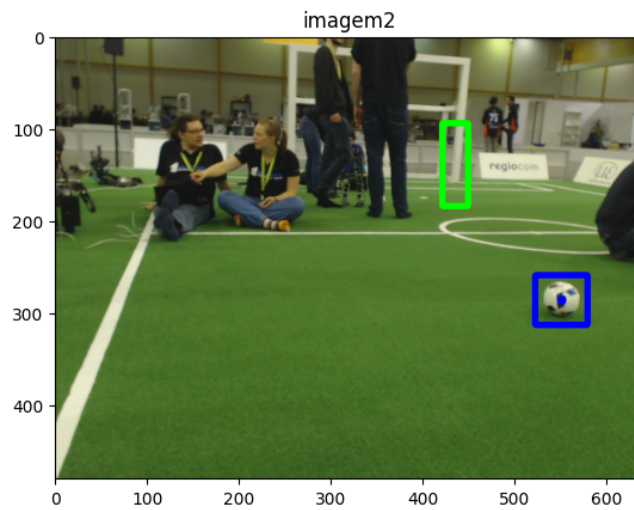


Figura 4: exemplo 2 de detecção da rede.

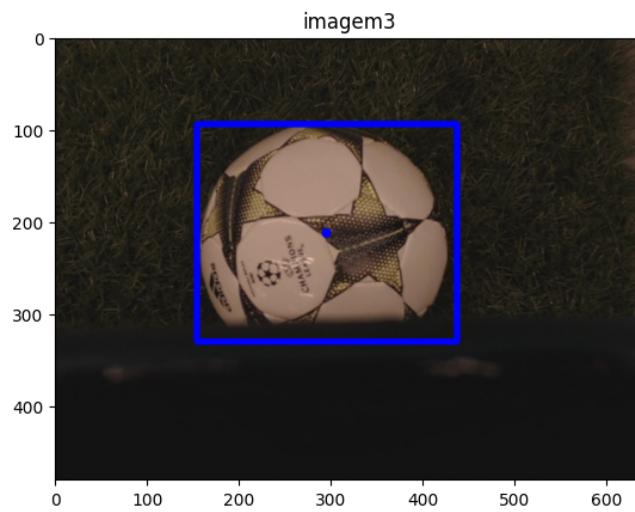


Figura 5: exemplo 3 de detecção da rede.

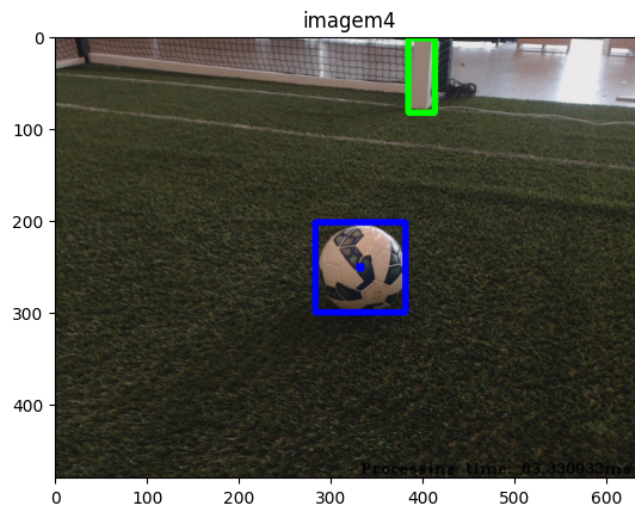


Figura 6: exemplo 4 de detecção da rede.

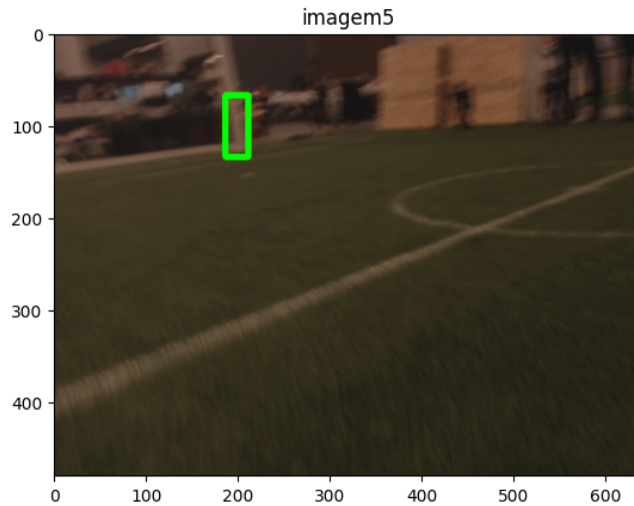


Figura 7: exemplo 5 de detecção da rede.

3 Discussão

A rede YOLO demonstrou excelente desempenho na tarefa de detecção e classificação de objetos de interesse, sendo capaz de identificar com precisão tanto a bola quanto as traves no ambiente analisado. Sua arquitetura baseada em convoluções profundas, combinada com funções de ativação LeakyReLU e normalização em batch, permitiu uma extração eficiente de características visuais mesmo em resoluções reduzidas. O uso de camadas de pooling e operações de concatenação também contribuiu para preservar informações espaciais relevantes ao longo da rede. Além disso, a capacidade da YOLO de realizar detecção em tempo real a partir de uma única etapa de regressão para múltiplas bounding boxes a torna extremamente adequada para aplicações em robótica, como no caso de robôs humanoides em ambientes dinâmicos. Assim, os resultados obtidos validam a eficácia do modelo proposto para cenários de visão computacional embarcada e reforçam o potencial da YOLO como solução robusta e eficiente para sistemas autônomos de percepção visual.