

# Desenvolvimento de um Sistema de Visão para Detecção de Objetos Embarcado em um Braço Robótico

Igor F. Soares<sup>1</sup>, Anfranserai M. Dias<sup>2</sup>

<sup>1</sup>Lab. de Robótica e Controle (LARC) – Universidade Estadual de Feira de Santana (UEFS)  
Avenida Transnordestina, S/N, Novo Horizonte  
Feira de Santana – BA, Brasil – 44036-900

<sup>2</sup>Departamento de Tecnologia (DTEC) – Universidade Estadual de Feira de Santana (UEFS)  
Avenida Transnordestina, S/N, Novo Horizonte  
Feira de Santana – BA, Brasil – 44036-900

ifs54@hotmail.com, anfranserai@ecomp.uefs.br

**Abstract.** The present work describes the development of a vision system for object detection and measurement to be embedded in a robotic arm, using the ESP32-CAM module for image acquisition and a distance sensor. The transfer learning technique was applied with the EfficientDet-Lite model for object detection. Additionally, an object measurement model was developed based on images, which relates pixels to the actual dimensions of the object through linear regression. The system was trained with images of lego pieces, achieving an AP of 81.92% on the validation set and 61.86% on the test set. The measurement model showed an RMSE of less than 10 mm when measuring object sizes at distances of up to 340 mm, with performance degradation at greater distances. This limitation is acceptable, as the manipulator used in the project has a maximum reach of 390 mm.

**Resumo.** O presente trabalho descreve o desenvolvimento de um sistema de visão para detecção e mensuração de objetos a ser embarcado em um braço robótico, utilizando o módulo ESP32-CAM na aquisição de imagens e um sensor de distância. Foi aplicada a técnica de transferência de aprendizado com o modelo EfficientDet-Lite para a detecção de objetos. Além disso, foi desenvolvido um modelo de mensuração de objetos a partir das imagens, que relaciona pixels a dimensões reais do objeto por meio de regressão linear. O sistema foi treinado com imagens de peças de lego, obtendo um AP de 81,92% no conjunto de validação e 61,86% conjunto de teste. Por sua vez, o modelo de mensuração apresentou RMSE menor que 10 mm na medição do tamanho de objetos a distâncias de até 340 mm, com queda de desempenho em distâncias maiores. Esta limitação é adequada, pois o manipulador usado no projeto tem o alcance máximo de 390 mm.

## 1. Introdução

Robô pode ser definido como uma máquina controlada por computador, que opera no mundo físico, percebendo seu ambiente e executando ações para atingir metas específicas [Matarić 2014]. Robôs autônomos são máquinas capazes de realizar tarefas em ambientes não estruturados, sem controle humano explícito ou contínuo sobre seus movimentos.

Eles têm se tornado aliados importantes, assistindo humanos em diversas funções, desde o auxílio em atividades diárias e industriais até a atuação em cenários desafiadores, como a exploração espacial e o atendimento em desastres [Rice et al. 2023].

Para que um robô execute suas tarefas de maneira autônoma e eficiente, é fundamental que ele tenha a capacidade de interpretar o ambiente ao seu redor, algo que é viabilizado pelo uso de sensores. A visão computacional permite o desenvolvimento de sensores versáteis, adequados para diferentes finalidades. Essa área faz uso intensivo de algoritmos e técnicas de Inteligência Artificial, que, em conjunto com os sensores, analisam o contexto ao redor e tomam decisões de forma independente para alcançar o objetivo predefinido [Szeliski 2011].

A Visão Computacional abrange várias linhas de pesquisa, entre elas a busca em imagens. Uma das tarefas dessa área é a detecção de objetos, que consiste em localizar a presença de objetos de uma classe específica em uma imagem digital. Para realizar essa tarefa, são utilizados algoritmos de reconhecimento de padrões, frequentemente aplicando técnicas de aprendizado de máquina. Esses métodos analisam exemplos e ajustam os parâmetros de modelos adaptativos para melhorar a precisão na detecção [Silva 2018].

Dentro do contexto da robótica, a visão computacional é utilizada de diversas maneiras. Por exemplo, o trabalho de Ali et al. (2018) integra um sistema de visão computacional ao robô Scrbot, para que ele possa determinar as especificações de um objeto com base em sua cor, tamanho e formato. O processamento desses dados habilita o robô a manipular o objeto. Na pesquisa, foram usados variados softwares como MatLab, Visual Basic e Scrbase [Ali et al. 2018].

Outro trabalho propõe um método de detecção de objetos para o posicionamento e reconhecimento mais precisos por um robô de manipulação. No trabalho é aplicada a calibração *Eye-In-Hand* para determinar a pose tridimensional dos objetos, e métodos de otimização são utilizados para aprimorar o modelo de detecção de objetos YOLOv5. Para a implementação, foi utilizada uma plataforma composta por um robô xArm, uma câmera Intel RealSense D415 e um servidor [Song et al. 2021].

Por fim, pode-se destacar o DOFBOT, um braço robótico comercial voltado para a educação, que oferece uma ampla gama de funções tanto autônomas quanto manuais. Este robô é capaz de responder a gestos humanos, reconhecer e rastrear faces, identificar cores, reconhecer objetos, além de escrever e desenhar. Assim como os outros projetos mencionados, o DOFBOT utiliza uma câmera, sensores e, principalmente, softwares focados em visão computacional para alcançar os resultados e funcionalidades desejados [Yahboom 2024].

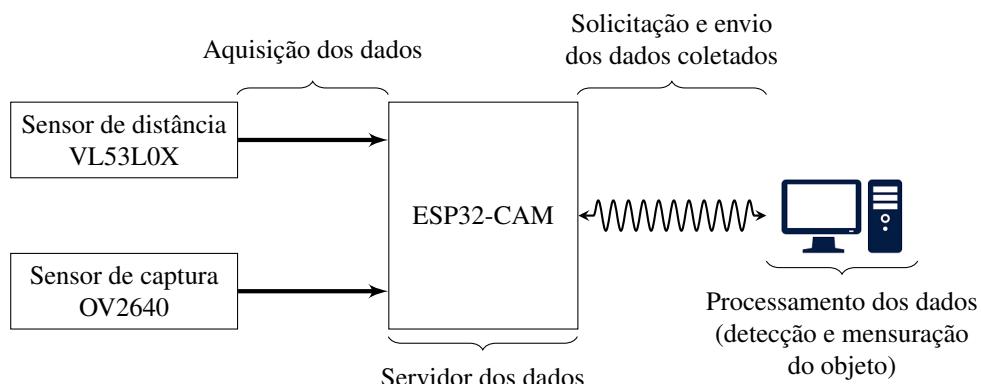
O presente trabalho descreve um sistema de visão para a detecção de objetos a ser embarcado em um braço robótico. O sistema de visão precisa detectar o objeto, identificar a distância em que o objeto se encontra do braço e mensurar o tamanho (altura e largura) do objeto, com o intuito de determinar se a garra acoplada ao braço é capaz de segurá-lo.

A câmera precisa ser leve para ser posicionada na garra. Esse foi um dos motivos que levou à escolha de uma ESP32-CAM. Além disso, ela pode ser programada para processar a imagem captada [Mehendale 2022]. Contudo, durante o desenvolvimento, observou-se que, apesar das vantagens oferecidas pelo ESP32-CAM, o modelo de detecção implementado exigia mais recursos do que o dispositivo era capaz de fornecer.

Isso resultou na necessidade de transferir o processamento para um dispositivo mais robusto. O artigo descreve a análise das limitações e o desempenho do modelo de detecção, destacando as adaptações necessárias para lidar com as restrições de *hardware*.

## 2. Metodologia

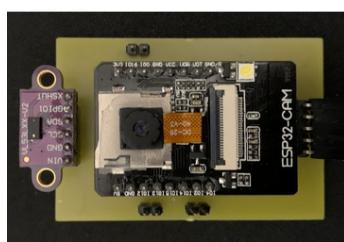
O sistema foi projetado utilizando como base o microcontrolador ESP32 para aquisição das imagens do sistema de visão e para obter a distância até o objeto. Foi utilizado o módulo ESP32-CAM, que conta com um sensor de captura de imagens já acoplado ao microcontrolador. Além disso, o ESP32 é embarcado com Wi-Fi, processador de dois núcleos de 32 bits e um conjunto de pinos de E/S que possibilitam a conexão com diversos sensores, como o sensor de distância. Ademais, suas características compactas e seu baixo custo favorecem seu acoplamento ao braço robótico. O sistema proposto é formado por dois módulos principais. O módulo de aquisição e o módulo de inferência e mensuração. A Figura 1 mostra a arquitetura do sistema.



**Figura 1. Arquitetura do sistema de visão**

### 2.1. Módulo de Aquisição

O módulo de aquisição foi embarcado no ESP32-CAM e funciona como um servidor, fornecendo as imagens e a distância do objeto para o módulo de inferência e mensuração. Para a aquisição das imagens, foi utilizado o sensor OV2640 de 2 MP, que permite resoluções de até 1600x1200 pixels. Já para a obtenção da distância, foi utilizado o sensor a laser VL53L0X, capaz de medir distâncias de até 2 metros. A Figura 2 ilustra a construção do módulo.



**Figura 2. Sistema de aquisição**

Apesar do sensor de captura permitir uma resolução de 1600x1200, optou-se por utilizar a resolução 640x480 pixels. Isso porque em resoluções menores a identificação

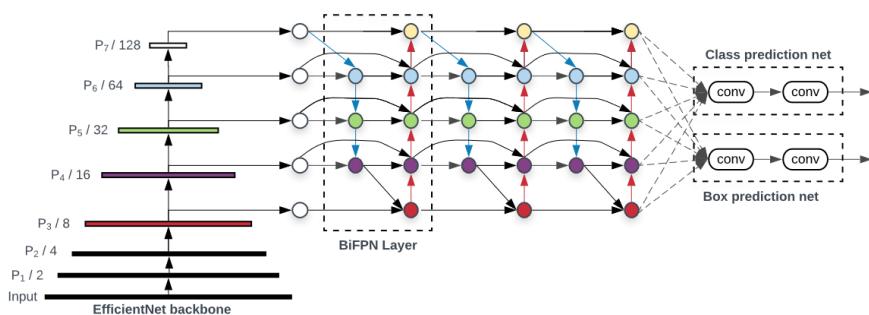
dos objetos era dificultada, já na resolução máxima a taxa média transferência das imagens para o módulo externo era de 1 quadro por segundo, com quedas bruscas ao passo que o microcontrolador aquecia. Enquanto que na resolução de 640x480 a taxa média foi de 2 quadros por segundo, mas mantendo-se estável mesmo com o aquecimento do dispositivo.

## 2.2. Módulo de Inferência e Mensuração

A principal atribuição deste módulo é efetuar a identificação do objeto na imagem recebida do módulo de aquisição por meio de um Modelo de Detecção de Objetos. Caso o objeto seja identificado, suas dimensões na imagem (largura e altura) em pixels são obtidas e submetidas ao Modelo de Mensuração de Objetos que, em conjunto com a distância que o objeto se encontra, determina o seu tamanho real.

## 2.3. Modelo de Detecção de Objetos

O modelo de detecção de objetos foi construído com base no modelo de arquitetura EfficientDet-Lite, utilizando a técnica de transferência de aprendizado. O desenvolvimento do modelo foi feito por meio da biblioteca TensorFlow e exportado para o formato Tensorflow Lite. O EfficientDet-Lite é um modelo derivado do EfficientDet treinado com o *dataset* COCO 2017 otimizado para o Tensorflow Lite e desenvolvido para dispositivos móveis [Kamal et al. 2021]. O EfficientDet é um modelo de detecção de objetos que utiliza o EfficientNet como rede principal combinada com a *Bidirectional Feature Pyramid Network* (BiFPN). A BiFPN atua como uma rede de características, que recebe as características dos níveis 3 a 7 da rede principal e aplica a fusão de características bidirecionais de cima para baixo e de baixo para cima. O resultado dessas características são inseridos na rede de classes e caixas para produzir a previsão das classes e as caixas delimitadoras dos objetos na imagem [Tan et al. 2020]. A arquitetura do EfficientDet pode ser vista na Figura 3.



**Figura 3. Arquitetura do modelo EfficientDet. Adaptado de [Tan et al. 2020]**

A técnica de transferência de aprendizado permite o aproveitamento de pesos aprendidos por um modelo treinado com um grande conjunto de dados para uma determinada tarefa, na construção de um outro modelo que execute uma tarefa similar. Essa abordagem possibilita treinar modelos com conjunto de dados menores e com menos recursos computacionais.

A avaliação do modelo foi feita seguindo as métricas definidas pelo método *Common Objects in Context* (COCO) e com um conjunto de teste de 236 imagens. O método

COCO define um conjunto de métricas de avaliação de um modelo de detecção de objetos, tais quais: *Average Precision* (AP), com o limite de corte de 0,5 à 0,95 com passo de 0,05;  $AP^{IoU=0,50}$  e  $AP^{IoU=0,75}$  com respectivos limites de corte de 0,5 e 0,75; AP Across Scales que mede o AP por tamanho de objetos (pequenos, médios e grandes); *Average Recall* (AR), para o número de detecção de objetos por imagem e; AR Across Scales, para o número de detecção de objetos de diferentes tamanhos (pequenos, médios e grandes) por imagem [COCO 2015].

## 2.4. Modelo de Mensuração de Objetos

O modelo de mensuração foi desenvolvido para estimar a altura e largura reais dos objetos detectados. O modelo foi construído com base na relação entre o tamanho de um objeto no mundo real (em mm) e na imagem (em pixels), a uma distância conhecida. Deste modo, foram capturadas imagens com o ESP32-CAM dos objetos a uma distância conhecida em relação ao módulo de aquisição. Foi efetuado um conjunto de capturas para 20 distâncias distintas, iniciando em 50 mm e variando de 50 em 50 mm até atingir a distância de 1000 mm.

A partir das imagens coletadas foram medidas as alturas e larguras médias em pixels dos objetos para cada uma das distâncias. Deste modo, conhecendo as alturas e larguras reais das peças, foi possível obter, por meio de regressão linear, uma equação que estima a relação mm/pixel para a altura e outra equação que estima o valor mm/pixel para a largura em função da distância em que a peça se encontra do sistema.

## 3. Resultados

O primeiro passo para realização dos teste foi a construção de um conjunto de imagens (*dataset*). Optou-se por utilizar um conjunto de peças de lego como objeto de interesse. Essa escolha foi motivada pela capacidade das peças de lego serem combinadas em diversos arranjos, possibilitando a criação de objetos com diferentes tamanhos. O *dataset* foi construído capturando imagens dos objetos com o ESP32-CAM, em diferentes condições, como baixa e alta iluminação, além de diferentes orientações dos objetos. A distância entre o objeto e o ESP32-CAM variou de 50 mm a 1200 mm. A Figura 4 apresenta uma amostra do conjunto de dados de teste.



**Figura 4. Imagens presentes no conjunto de dados de teste**

O *dataset* conta com 1020 imagens, onde 90% foi destinada ao treinamento do modelo e 10% para validação.

O módulo de inferência e mensuração foi executado em um computador convencional, equipado com um processador Intel Core i3-6100U e 4 GB de RAM.

### 3.1. Escolha do Modelo de Detecção de Objetos

O EfficientDet-Lite dispõe de 5 modelos de detecção de objetos, como mostrado na Tabela 1. Cada modelo apresenta uma precisão média e um tamanho diferente, sendo o EfficientDet-Lite0 o menor, porém menos preciso. Contudo, por ser menor, o EfficientDet-Lite0 tem uma latência média reduzida se comparado com os outros modelos da família. Devido a ideia inicial de executar a inferência diretamente no microcontrolador, o EfficientDet-Lite0 foi o modelo mais adequado devido ao seu tamanho reduzido.

**Tabela 1. Modelos EfficientDet-Lite**

Modelo de Arquitetura	Tamanho (MB)	Precisão Média <sup>1</sup> (%)
EfficientDet-Lite0	4,4	25,69
EfficientDet-Lite1	5,8	30,55
EfficientDet-Lite2	7,2	33,97
EfficientDet-Lite3	11,4	37,7
EfficientDet-Lite4	19,9	41,96

<sup>1</sup> No conjunto de dados de validação COCO 2017.

Dentro deste panorama, o modelo foi treinado, porém ao tentar embarcá-lo no ESP32-CAM, surgiu um problema de limite de memória do microcontrolador. Onde, ao converter o modelo Tensorflow Lite em uma matriz de char na linguagem C, o tamanho da matriz excedia o valor de memória disponível no dispositivo. Mesmo treinando o modelo com um conjunto de imagens de resolução menor não foi possível obter um modelo pequeno suficiente para ser embarcado no microcontrolador. Visto isso, a detecção de objetos e mensuração dos seus tamanhos precisaram ser executadas em um módulo externo.

### 3.2. Detecção de Objetos

O treinamento do modelo de detecção de objetos foi feito utilizando 20 *epochs* e um *batch size* de 4. Isso significa que a rede neural percorre o conjunto de dados 20 vezes e que é necessário processar 4 amostras antes dos parâmetros internos da rede serem atualizados. Após o treinamento, o modelo foi exportado para o formato do Tensorflow Lite. A Tabela 2 mostra os valores de desempenho do modelo de acordo com as métricas COCO para os dois conjuntos de dados.

**Tabela 2. Desempenho do modelo**

Métricas Avaliadas												
AP	AP <sup>IoU=0,50</sup>	AP <sup>IoU=0,75</sup>	AP <sup>small</sup>	AP <sup>medium</sup>	AP <sup>large</sup>	AR <sup>max=1</sup>	AR <sup>max=10</sup>	AR <sup>max=100</sup>	AR <sup>small</sup>	AR <sup>medium</sup>	AR <sup>large</sup>	
Conjunto de Dados de Validação												
81,92	95,69	92,89	52,35	79,81	92,05	76,75	85,18	85,18	59,23	84,26	93,40	
Conjunto de Dados de Teste												
61,86	87,41	70,62	34,05	68,28	77,61	64,33	69,05	69,54	46,23	73,88	84,06	

Na avaliação do modelo com o conjunto de dados validação foi obtido um AP de 81,92%. Ademais, o modelo apresentou uma precisão média para objetos grandes de 92,05%, enquanto que para objetos pequenos e médios esse valor foi de 52,35% e 79,81% respectivamente. Isso indica que o modelo apresentou um melhor desempenho na detecção de legos que ocupavam áreas maiores nas imagens. Já com o conjunto de dados de teste, foi obtido um AP de 61,86%. Assim como no caso do conjunto de dados de validação, o modelo apresentou um melhor desempenho na detecção de legos que ocupavam áreas maiores na imagem, obtendo uma precisão média neste caso de 77,61%.

Enquanto que na detecção legos que ocupavam áreas pequenas e médias a precisão média foi, respectivamente, de 34,05% e 68,28%. Nota-se que o modelo apresentou um pior desempenho com o conjunto de dados de testes se comparado com o conjunto de dados de validação. Essa discrepância pode ter sido causada devido ao fato do conjunto de dados de teste apresentarem imagens em situações (exposição, iluminação ou posições) em que o *dataset* não foi capaz de suprimir para o treinamento do modelo.

### 3.3. Mensuração de Objetos

Para realização deste teste, foram capturadas imagens de dois conjuntos de peças de legos, um vermelho e outro amarelo, ambos com largura de 31,5 milímetros e altura de 21 milímetros. Duas cores distintas foram utilizadas para facilitar a identificação das bordas das peças nas imagens, assim foi possível obter um valor médio da altura e largura em pixels para cada uma das distâncias do sensor de captura em que o objeto foi posicionado.

A avaliação do sistema de mensuração foi feita com o sistema acoplado ao modelo de detecção de objetos desenvolvido. Foi avaliada a raiz do erro médio quadrático (RMSE), que fornece a diferença entre os valores medidos pelo modelo e os valores reais das peças. Para o conjunto de legos de altura 11 mm e largura 31,5 mm o RMSE da altura foi de 2,73 mm e da largura 2,22 mm. Já para o segundo conjunto, o RMSE da altura foi de 2,46 mm e da largura de 2,39 mm.

Além disso, foi avaliado qualitativamente o comportamento do modelo para objetos em perspectiva. Para isso, foram submetidos ao modelo três conjuntos de legos de tamanhos distintos, dispostos de modo que suas faces não estivessem em paralelo ao plano da imagem. Considerou-se que a largura e altura de referência da peça seria a face de maior destaque na imagem. A Figura 5 apresenta dois resultados para objetos do conjunto 1 (31,5 mm de largura e 11 mm de altura).



**Figura 5. Estimativa do modelo para o conjunto 1 em perspectiva**

O conjunto 1 foi posicionado próximo ao centro da imagem. Neste caso, o modelo apresentou boas estimativas do tamanho real da peça, como visto na Figura 5. Contudo, com a rotação do conjunto, o modelo levou em consideração que a lateral da peça fazia parte da face frontal, gerando um efeito compensatório da rotação, visto que o sistema não leva em consideração a profundidade dos objetos e considera que tudo na imagem encontra-se no mesmo plano.

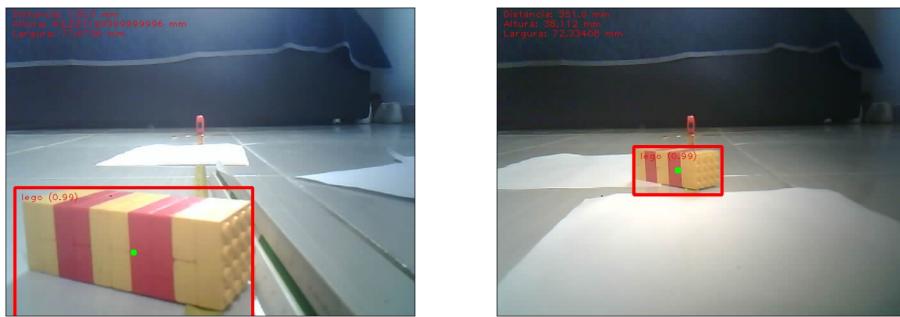
O conjunto 2 utilizou objetos de 31,5mm de altura e 31,5 mm de largura. A Figura 6 exemplifica os resultados obtidos. Para o conjunto 2, diferente do conjunto 1, os valores estimados da peça apresentaram uma diferença significativa dos valores reais,

essencialmente a largura que apresentou valores maiores que o real. Neste caso, o efeito compensatório acabou fazendo com que a peça apresentasse valores maiores de largura.



**Figura 6. Estimativa do modelo para o conjunto 2 em perspectiva**

No conjunto 3 o sistema foi testado com objetos de 98 mm largura e 31,5 mm de altura, como ilustrado pela Figura 7.



**Figura 7. Estimativa do modelo para o conjunto 3 em perspectiva**

Os valores estimados para a largura da peça foram menores que o real para o conjunto 3. Neste caso, o efeito compensatório não foi capaz de compensar o valor de largura da face da peça que foi ocultada com a rotação do objeto.

Por fim, percebe-se que o modelo apresenta comportamentos distintos para objetos em perspectiva. Portanto, modelo é adequado para trabalhar apenas com objetos cujas faces estejam paralelas ao plano da imagem do sistema de visão.

### 3.4. Sistema completo

A análise de performance do sistema final foi feita utilizando dois blocos distintos de peças de legos: bloco 1 com altura de 98,0 mm e largura de 31,5 mm; bloco 2 com altura de 59,0 mm e largura de 31,5 mm. Os blocos foram posicionados a diferentes distâncias do sistema de que modo que, para cada distância, um conjunto de aquisições foram feitas e submetidas ao sistema para a execução das inferências.

O bloco 1 foi posicionado inicialmente a 220 mm do sistema e sua distância foi variada em 40 mm até atingir a distância de 540 mm. Não foi possível avaliar para distâncias menores porque o objeto não era enquadrado na imagem devido a proximidade com o sensor de captura. Já para distâncias maiores que 540 mm, o sensor de distância não foi capaz de efetuar as medidas até o objeto, isso se deve às características do sensor, que apesar de

ser capaz de medir distâncias até 2 metros, para objetos com áreas reduzidas sua aptidão é afetada. A Tabela 3 mostra o desvio padrão amostral, o RMSE e distância média obtida pelo sensor de distância para cada uma das distâncias a que o bloco 1 foi submetido.

**Tabela 3. Desempenho do sistema final para o bloco 1 (98,0 mm x 31,5 mm)**

Distância (mm)	Desvio padrão (mm)		RMSE (mm)	
	Altura	Largura	Altura	Largura
220	0,88	0,20	5,89	3,41
260	0,72	0,43	1,88	3,06
300	1,46	0,38	4,35	3,67
340	1,26	0,36	4,06	3,31
380	0,67	0,62	12,45	2,98
420	1,25	0,56	13,26	3,06
460	0,99	0,55	20,90	1,72
500	1,44	0,89	21,56	1,80
540	1,88	0,79	25,69	5,36

A partir de 380 mm o RMSE da altura apresenta valores elevados (acima de 10 mm). Isso é um reflexo das limitações do sensor de distância para obter distâncias até objetos de tamanho reduzido. Para distâncias maiores que 340 mm o sensor apresentou valores distorcidos da realidade, prejudicando o desempenho do sistema.

O bloco 2 foi posicionado inicialmente a 140 mm do sistema e sua distância foi variada em 40 mm até atingir a distância de 540 mm. A Tabela 4 mostra o desvio padrão amostral, o RMSE e a distância média obtida pelo sensor de distância para cada uma das distâncias a que o bloco 2 foi submetido.

**Tabela 4. Desempenho do sistema final para o bloco 2 (59,0 mm x 31,5 mm)**

Distância (mm)	Desvio padrão (mm)		RMSE (mm)	
	Altura	Largura	Altura	Largura
140	0,68	0,34	1,92	0,91
180	1,43	0,51	1,36	2,76
220	0,39	0,39	1,57	2,40
260	0,87	0,41	3,83	2,27
300	0,51	0,51	8,69	0,62
340	0,59	0,57	7,66	0,94
380	0,69	0,39	10,08	1,13
420	0,76	0,64	12,11	2,66
460	0,98	0,69	14,32	2,99
500	0,76	0,67	17,71	4,91
540	0,93	0,71	20,55	6,44

Neste caso, assim como ocorreu com o bloco 1, para distâncias menores o objeto não era enquadrado na imagem. Acima de 380 mm o RMSE da altura para o bloco 2 apresentou valores elevados. Novamente o sensor apresentou valores muito distintos dos reais para distância acima 340 mm. Deste modo, o sistema se torna limitado na sua distância de atuação, garantindo o melhor desempenho até distâncias de 340 mm.

## 4. Conclusões e Trabalhos Futuros

Este trabalho teve como objetivo desenvolver um sistema de visão computacional a ser embarcado em um braço robótico para a detecção e mensuração de objetos, utilizando o módulo ESP32-CAM e um sensor de distância. O sistema de detecção de objetos construído é capaz de cumprir as atribuições para as quais foi desenvolvido, alcançando uma acurácia de 81,92% no conjunto de validação e 61,86% no conjunto de teste.

No entanto, o sistema apresentou limitações, como a dependência de um dispositivo externo para o processamento, devido às restrições de memória do ESP32-CAM, o que impedi a execução das inferências diretamente na plataforma embarcada.

Dessa forma, em trabalhos futuros, o sistema pode ser aperfeiçoado com a construção de um novo modelo que possa ser embarcado no microcontrolador. Além disso, outros sensores de distância podem ser empregados, visando permitir que o sistema atue em uma gama maior de situações.

## Referências

- Ali, M. H., Aizat, K., Yerkhan, K., Zhandos, T., and Anuar, O. (2018). Vision-based robot manipulator for industrial applications. *Procedia computer science*, 133:205–212.
- COCO (2015). Detection evaluation metrics. <https://cocodataset.org/#detection-eval>. [Online]. Acesso em 17 de agosto de 2023.
- Kamal, M. U. A., Suciati, N., and Hidayati, S. C. (2021). Calculation of handwriting mathematics expressions on mobile devices using efficientdet-lite0 and reverse polish notation. *The Journal on Machine Learning and Computational Intelligence (JMLCI)*, 1(1).
- Matarić, M. J. (2014). *Introdução à robótica*. Editora Blucher.
- Mehendale, N. (2022). Object detection using esp 32 cam. *SSRN Electronic Journal*.
- Rice, C., Taylor, J., Widom, J., Zegart, A., and Lin, H. (2023). *The Stanford Emerging Technology Review 2023: A Report on Ten Key Technologies and Their Policy Implications*. Stanford University.
- Silva, G. R. (2018). Detecção de objetos em imagens utilizando técnicas de aprendizagem profunda. *Trabalho de Conclusão de Curso. Universidade Federal de Santa Catarina, Florianópolis*.
- Song, Q., Li, S., Bai, Q., Yang, J., Zhang, X., Li, Z., and Duan, Z. (2021). Object detection method for grasping robot based on improved yolov5. *Micromachines*, 12(11):1273.
- Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*. Springer London, 1 edition.
- Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet: Scalable and Efficient Object Detection.
- Yahboom (2024). Dofbot-jetson repository. [http://www.yahboom.net/study/Dofbot-Jetson\\_nano-1](http://www.yahboom.net/study/Dofbot-Jetson_nano-1). Acesso em 30 de setembro de 2024.