

Nicolas Garcia CC - Dom Helder
Pesquisa Científica

- Braço robótico com visão computacional.

. Hoje, a área da robótica está em constante ascensão. As novas práticas da economia moderna demandam linhas de produção e elementos com alta percepção sensorial, aqui entra a visão computacional, que por sua vez é capaz de em tempo real visualizar um objeto e em uma velocidade muitas vezes superior a capacidade humana, analisar e interpretar certo objeto com perfeição e executar determinada ação. No caso de uma esteira de controle de qualidade, a máquina pode analisar um objeto com problema e removê-lo da esteira.

Detecção de objetos:

Alguns softwares comumente utilizados para detecção de objetos são: OpenCV, FOMO e YOLOv8, que podem se comunicar com um microcontrolador através da linguagem Python.

. OpenCV: A biblioteca por si só, não possui uma alta precisão na detecção de pequenos objetos para uso no mundo real. Para utilizá-la da melhor forma, geralmente utiliza-se o modelo YOLOv8 para a detecção e a biblioteca OpenCV para cuidar de outras funções, como redimensionar imagens, apresentar resultados em caixas delimitadas, adicionar sobreposições de quadros por segundo, entre outras funções.

. YOLOv8: Detecta vários objetos simultaneamente, com precisão, porém, necessita de um hardware potente.

. FOMO: Detecta uma grande variedade de objetos com precisão, como pessoas, insetos, objetos em esteiras de produção, entre outros. Ideal para projetos com baixo consumo energético, e microcontroladores.

Com base nos dados apresentados no gráfico a seguir, é possível concluir que todas as bibliotecas têm um uso muito parecido, sendo as principais diferenças relacionadas ao(s) requisitos para execução, desempenho/performance e precisão de detecção.

- Gráfico para comparação das bibliotecas:

Aspect	OpenCV	YOLOv8	FOMO
What it is	CV library	Modern object detection & segmentation model	Lightweight edge detection model
Main Use	General CV tasks	High-accuracy detection, segmentation, pose	TinyML detection for small objects
Pros	Versatile, hardware acceleration, great for pre/post-processing	Very accurate, fast, easy training pipeline, flexible tasks (detect, segment, pose)	Ultra-low latency, runs on microcontrollers, very efficient
Cons	Not a detection model by itself	Needs GPU or strong CPU, too heavy for microcontrollers	Simpler models, lower accuracy for complex/large objects
Best For	Image processing, camera input, using other models	Real-time detection & segmentation on PC, GPU, Jetson	Small embedded devices with tight RAM/CPU limits
Hardware Needs	Runs anywhere	Needs good GPU/CPU	Runs on MCUs (ESP32, Arduino)

. Dessa forma é possível concluir que para um ambiente de desenvolvimento em que será utilizado um notebook e sua própria câmera para se comunicar com o braço robótico, o sistema de detecção mais adequado seria o YOLOv8. Devido a capacidade de utilização do hardware do notebook (característica que não é encontrada no sistema FOMO, por ser voltado a hardware com baixo consumo energético), YOLOv8 retorna com mais precisão e velocidade as movimentações da câmera.

Câmera:

Dentro da visão computacional é comum encontrar informações sobre dois termos mais utilizados: "Eye-in-hand", em que a câmera é fixada no braço robótico, e o termo "Eye-to-hand", que é a câmera em um ponto fixo fora do braço.

Existem vantagens e desvantagens entre os dois métodos, a escolha vai depender do uso que será atribuído ao braço. No caso da opção "Eye-in-hand" algumas vantagens são por exemplo: mais precisão no cálculo da distância de objetos em relação ao braço, perspectiva dinâmica. Já algumas desvantagens do método são: imagem tremida, menor campo de visão e maior complexidade de calibração da posição da câmera.

Microcontroladores:

Para definir uma ligação entre a câmera e os motores do braço existem várias opções de microcontroladores, que são uma ótima opção visando baixo custo, uma grande comunidade ativa e diversos conteúdos gratuitos relacionados ao tema.

Em geral para projetos relacionados à visão computacional e motores, é comum a utilização de microcontroladores como Arduino Nano, Uno, e ESP32 que é um eficiente microcontrolador de baixo custo com Wi-Fi e Bluetooth integrados.

Referências:

. Revista Thêma et Scientia - Visão computacional aplicada em um braço robótico

<https://themaetscientia.fag.edu.br/index.php/RTES/article/view/807>

. Cruzeiro do Sul - Braço robótico de baixo custo controlado por visão computacional pode revolucionar setores críticos

<https://noticias.cruzeirosuleducacional.edu.br/braco-robotico-de-baixo-custo-controlado-por-visao-computacional-pode-revolucionar-setores-criticos/>

. FOMO: Object detection

<https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices>

. YOLOv8

<https://yolov8.com/>

. OpenCV

<https://opencv.org/>