

Paralelismo em contador de multidões

Samuel Gonçalves de Araujo e Larissa Batista Maciel

Introdução

e Configuração

Este projeto aplicou paralelismo à contagem automática de pessoas em imagens, usando YOLOv5 para reconhecer cabeças. A partir de uma única imagem, foram geradas 988 cópias para simular carga real.

Com o módulo multiprocessing do Python, dividimos o conjunto entre múltiplos processos para acelerar o processamento. O objetivo foi comparar o desempenho com 1, 2, 4, 6, 8, 12 e 16 processos, medindo tempo, *speedup* e eficiência.

Configuração da máquina:
Intel Core i7-12700 (2.10GHz), 16 GB RAM

Problema e Solução

A detecção de pessoas em grandes volumes de imagens é uma tarefa computacionalmente custosa, mesmo com modelos avançados como o YOLOv5. Para contornar esse gargalo, foi adotada uma abordagem baseada em paralelismo.

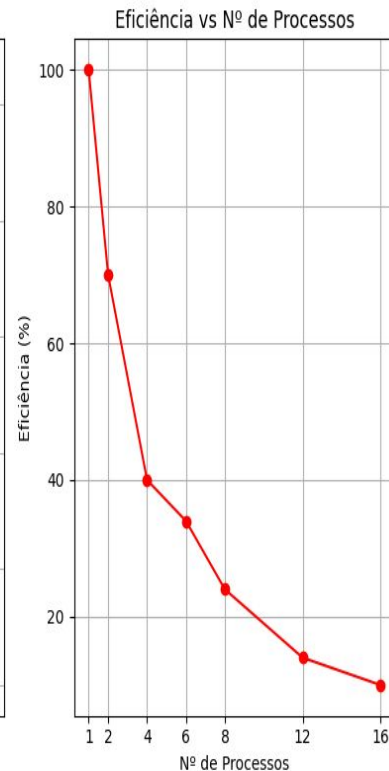
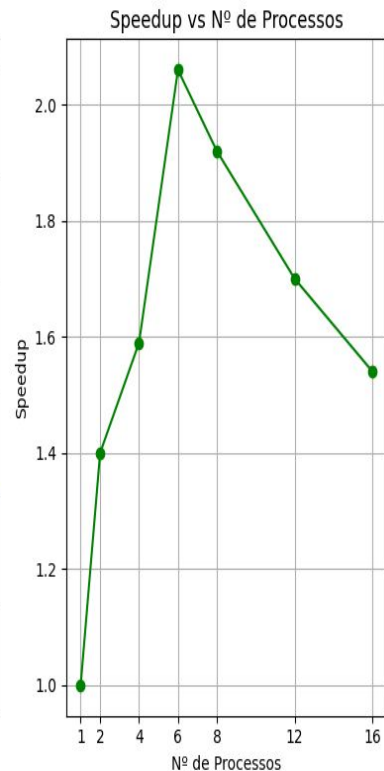
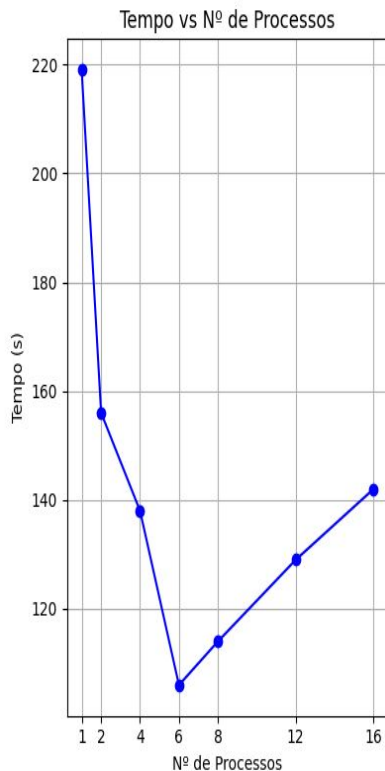
Inicialmente, tentou-se detectar corpos inteiros, mas a obstrução em multidões reduziu a eficácia. A solução foi utilizar a detecção de cabeças, que apresentou melhores resultados, apesar de limitações em rostos mais distantes.

A imagem original foi replicada 988 vezes para simular carga real. Em cada imagem, foram detectados 222 objetos (cabeças). Essas imagens foram divididas entre os processos, que executaram a contagem de forma paralela, acelerando significativamente o processamento.



Resultados

Nº de Processos	Tempo (s)	Speedup	Eficiência (%)
1	218,75	1,00	100,00
2	155,70	1,40	70,00
4	137,67	1,59	39,75
6	105,94	2,06	34,33
8	114,21	1,92	24,00
12	128,63	1,70	14,17
16	141,94	1,54	9,63



Conclusão

Melhor resultado com 6 processos:

105,94 segundos – *speedup* de 2,06

A partir de 8 processos, o desempenho caiu.

Causas possíveis:

- Sobrecarga de gerenciamento
- Gargalo no disco (I/O)
- Custo de sincronização entre processos

A paralelização do processamento com YOLOv5 mostrou-se eficiente para reduzir o tempo de execução em tarefas intensivas, especialmente em cenários com grande volume de imagens.

Com 6 processos, alcançamos o melhor desempenho, otimizando o tempo sem comprometer a estabilidade do sistema.

Entretanto, o uso excessivo de processos gerou efeitos contrários, como aumento no tempo total, devido à sobrecarga, concorrência de recursos e limitações físicas da máquina.

A aplicação de paralelismo exige análise criteriosa. Nem sempre mais processos significam melhor desempenho. Encontrar o ponto de equilíbrio entre velocidade e eficiência é essencial para aproveitar ao máximo os recursos computacionais.