

**Deteção, reconhecimento e alerta de objetos em vídeo**

Licenciatura em Engenharia Informática

Marco António da Silva Ferreira

Leiria, « mês » de 2024



**Deteção, reconhecimento e alerta de objetos em vídeo**

Licenciatura em Engenharia Informática

Válter Santos Pedrosa

Trabalho de Projeto da unidade curricular de Projeto Informático realizado sob a orientação do Professor Doutor Paulo Costa e do Professor Doutor Filipe Neves

Leiria, « mês » de 2024

# Dedicatória

Inserir aqui a dedicatória. Trata-se de um elemento **facultativo**.

Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória.

Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória.

# Agradecimentos

Inserir aqui os agradecimentos. Trata-se de um elemento **facultativo**.

Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos.

Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos.

# Resumo

Inserir aqui o resumo. Trata-se de um elemento **obrigatório**.

Deve começar sempre numa página ímpar. Se ocupar um número par de páginas (p. ex. 2), deve ajustar-se o texto para que a próxima secção (abstract) se inicie numa página ímpar. O resumo deve acabar com a lista de palavras-chave.

**No resumo deve dar-se nota das principais ideias do trabalho (objetivos e conclusões).**

Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo.

Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo.

**Palavras-chave:** deteção, yolo, reconhecimento, IA.

# Abstract

Please insert here the abstract in English. This is a **mandatory** element.

The abstract should always start in an odd page. If the length is a multiple of two, the text should be adjusted in order to the next section start also in an odd page. The abstract should end with a list of keywords.

Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English.

Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English.

**Keywords:** maximum of 6 words separated by “,”

Índice

Trata-se de um elemento **obrigatório**. Nota: **o índice nunca figura do índice.**

[Dedicatória ii](#_Toc167634553)

[Agradecimentos iii](#_Toc167634554)

[Resumo iv](#_Toc167634555)

[Abstract v](#_Toc167634556)

[Lista de Figuras x](#_Toc167634557)

[Lista de tabelas xii](#_Toc167634558)

[Lista de siglas e acrónimos xiii](#_Toc167634559)

[1. Introdução 1](#_Toc167634560)

[2. Análise teórica 2](#_Toc167634561)

[2.1. Representação de uma imagem digital 2](#_Toc167634562)

[2.2. Processamento de imagens 2](#_Toc167634563)

[2.2.1. Tipos de processamento de imagens 2](#_Toc167634564)

[2.2.1.1. Visualização 3](#_Toc167634565)

[2.2.1.2. Reconhecimento 3](#_Toc167634566)

[2.2.1.3. “Sharpening” e restauração 3](#_Toc167634567)

[2.2.1.4. Reconhecimento de padrões 3](#_Toc167634568)

[2.2.1.5. Recuperação 4](#_Toc167634569)

[2.3. Deteção de objetos 4](#_Toc167634570)

[2.3.1. Métodos tradicionais 4](#_Toc167634571)

[2.3.1.1. Viola-Jones Detector 5](#_Toc167634572)

[2.3.1.2. Histogram of Oriented Gradients Detector 5](#_Toc167634573)

[2.3.1.3. Deformable Part-based Model 5](#_Toc167634574)

[2.3.2. Métodos deep learning 5](#_Toc167634575)

[2.3.2.1. Redes Neuronais Convolucionais (CNN) 5](#_Toc167634576)

[2.3.2.1.1. One-stage detectors 5](#_Toc167634577)

[2.3.2.1.2. Two-stage detectors 6](#_Toc167634578)

[3. Metodologias e tecnologias 7](#_Toc167634579)

[3.1. Metodologia de trabalho 7](#_Toc167634580)

[3.1.1. Método tradicional 7](#_Toc167634581)

[3.1.2. Metodologias ágeis 7](#_Toc167634582)

[3.1.2.1. Scrum 9](#_Toc167634583)

[3.1.2.2. XP 10](#_Toc167634584)

[3.1.2.3. Kanban 11](#_Toc167634585)

[3.1.2.4. Scrumban 12](#_Toc167634586)

[3.1.3. Metodologia de trabalho escolhida 12](#_Toc167634587)

[3.2. Análise de requisitos 13](#_Toc167634588)

[3.2.1. Mockups 13](#_Toc167634589)

[4. Testes 18](#_Toc167634590)

[4.1. Teste de margem ideal para definir posição do objeto 18](#_Toc167634591)

[4.1.1. Introdução 18](#_Toc167634592)

[4.1.2. Condições do teste 18](#_Toc167634593)

[4.1.3. Resultados obtidos: 19](#_Toc167634594)

[4.1.3.1. Class “backpack” 19](#_Toc167634595)

[4.1.3.2. Class “sports ball” 21](#_Toc167634596)

[4.1.3.3. Class “Person” 22](#_Toc167634597)

[4.1.4. Conclusão: 23](#_Toc167634598)

[4.2. Teste da arquitetura para recolher frames de dispositivos 24](#_Toc167634599)

[4.2.1. Introdução 24](#_Toc167634600)

[4.2.1.1. Grab & Retrieve 24](#_Toc167634601)

[4.2.1.2. Parallel 24](#_Toc167634602)

[4.2.1.3. Open & Close 24](#_Toc167634603)

[4.2.1.4. Open & Close threading 24](#_Toc167634604)

[4.2.2. Condições de teste 25](#_Toc167634605)

[4.2.3. Resultados obtidos 25](#_Toc167634606)

[4.2.3.1. Grab & Retrieve 25](#_Toc167634607)

[4.2.3.2. Parallel 25](#_Toc167634608)

[4.2.3.2.1. Parallel 5 sec 26](#_Toc167634609)

[4.2.3.2.2. Parallel 10 sec 27](#_Toc167634610)

[4.2.3.3. Open & Close 27](#_Toc167634611)

[4.2.3.4. Open & Close threading 28](#_Toc167634612)

[4.2.4. Discussão dos Resultados 28](#_Toc167634613)

[4.2.5. Conclusão 29](#_Toc167634614)

[5. Conclusões ou Conclusão 30](#_Toc167634615)

[Bibliografia ou Referências Bibliográficas 31](#_Toc167634616)

[Anexos 32](#_Toc167634617)

[Glossário 33](#_Toc167634618)

# Lista de Figuras

Elemento a figurar, **quando aplicável**.

[Figura 3.1 - Mockup início 13](#_Toc167634619)

[Figura 3.2 - Mockup layout horizontal 14](#_Toc167634620)

[Figura 3.3 - Mockup adicionar dispositivo 14](#_Toc167634621)

[Figura 3.4 - Mockup adicionar dispositivo IP 15](#_Toc167634622)

[Figura 3.5 - Mockup definições dispositivo 15](#_Toc167634623)

[Figura 3.6 - Mockup emitir alerta para um dispositivo 16](#_Toc167634624)

[Figura 3.7 - Mockup dispositivo expandido 16](#_Toc167634625)

[Figura 3.8 - Mockup página alertas 17](#_Toc167634626)

[Figura 4.1 - Gráfico cantos backpack a 5M 19](#_Toc167634627)

[Figura 4.2 - Gráfico cantos backpack a 10M 20](#_Toc167634628)

[Figura 4.3 - Gráfico cantos backpack a 15M 20](#_Toc167634629)

[Figura 4.4 - Gráfico cantos bola a 5M 21](#_Toc167634630)

[Figura 4.5 - Gráfico cantos bola a 10M 21](#_Toc167634631)

[Figura 4.6 - Gráfico cantos bola a 12M 22](#_Toc167634632)

[Figura 4.7 - Gráfico cantos person a 5M 22](#_Toc167634633)

[Figura 4.8 - Gráfico cantos person a 35M 23](#_Toc167634634)

[Figura 4.9 - Gráfico cantos person a 70M 23](#_Toc167634635)

[Figura 4.10 - Gráfico performance parallel 25](#_Toc167634636)

[Figura 4.11 - Gráfico performance parallel 5 segundos 26](#_Toc167634637)

[Figura 4.12 - Gráfico performance parallel 10 segundos 27](#_Toc167634638)

[Figura 4.13 - Gráfico performance open & close 27](#_Toc167634639)

[Figura 4.14 - Gráfico performance open & close threading 28](#_Toc167634640)

# Lista de tabelas

Elemento a figurar, **quando aplicável**.

[Tabela 1.1 - Texto ilustrativo da tabela 1. 3](#_Toc92389035)

# Lista de siglas e acrónimos

Elemento a figurar, **quando aplicável**.

|  |  |
| --- | --- |
| ESTG | Escola Superior de Tecnologia e Gestão |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Cuidados na elaboração da lista de siglas e acrónimos:

* Ordenação alfabética;
* Apenas as que sejam relevantes para a leitura do texto.

Adicionar mais entradas à tabela, caso seja necessário (a tabela não tem contornos, mas está no texto).

# Introdução

Introdução deve conter resumidamente os seguintes elementos:

* O objeto do trabalho (o tema);
* A justificação ou a pertinência do tema;
* Os objetivos do trabalho (gerais e específicos, perguntas a responder ou hipóteses a testar);
* Os métodos e as técnicas utilizados;
* Estrutura do trabalho.

# Análise teórica

O propósito deste projeto é a criação de uma aplicação desktop para deteção de objetos com Inteligência Artificial (IA) e alerta.

O ambiente escolhido para o desenvolvimento foi python pela sua simplicidade e vasta oferta de bibliotecas e frameworks úteis na área de machine e deep learning, nomeadamente o YOLO.

## Representação de uma imagem digital

Uma imagem é representada pelas suas dimensões (altura e largura) consoante o número de pixéis. Um pixel é um ponto na imagem que tem uma tonalidade, opacidade ou cor específica. Normalmente é representado de uma das seguintes maneiras:

* Grayscale – Um pixel é um inteiro compreendido entre 0 e 255 em que 0 é completamente preto e 255 completamente branco.
* RGB – Um pixel é composto por 3 inteiros compreendidos entre 0 e 255 em que cada um representa a intensidade de vermelho (R), verde (G) e azul (B).
* RGBA – É uma extensão de RGB que acrescenta um campo alfa, que reflete a opacidade do pixel.

## Processamento de imagens

O processamento de imagens consiste no processo de transformar uma imagem numa forma digital e executar certas operações para extrair informação útil sobre a mesma. Este processo requere sequências fixas de operações que são executadas a cada pixel de uma imagem sendo que o processador da imagem executa a primeira sequência de operações na imagem, pixel a pixel. Quando completa, começa a executar a segunda operação e assim sucessivamente. O valor de saída destas operações pode ser calculado a cada pixel da imagem.

### Tipos de processamento de imagens

Existem 5 tipos principais de processamento de imagens.

### Visualização

Este primeiro tipo de processamento envolve a conversão de dados na imagem para um formato visual que possa ser facilmente interpretado por humanos. Existem várias técnicas possíveis a ser utilizadas, nomeadamente melhoramento de contraste, segmentação ou criação de imagens tridimensionais a partir de dados de imagens bidimensionais. Um bom exemplo de aplicação desta técnica é na medicina, onde são aplicadas técnicas de visualização em MRI’s (Magnetic Resonance Imaging) para obter um diagnóstico ou explorar conectividade no cérebro de um indivíduo.

### Reconhecimento

No contexto de processamento de imagens, o reconhecimento refere-se à identificação e categorização de objetos ou padrões específicos dentro de uma imagem. Isso pode envolver a deteção de rostos humanos, identificação de caracteres de texto ou reconhecimento de objetos em cenas complexas. São utilizadas técnicas avançadas de machine learning e redes neurais convolucionais (CNNs) frequentemente para melhorar a precisão e a eficácia dos sistemas de reconhecimento de imagens. É neste tópico que este projeto se insere.

### “Sharpening” e restauração

“Sharpening” e restauração de imagens são processos usados para melhorar a qualidade de imagens que tenham sido distorcidas, quer seja por ruído, desfoque ou compressão. “Sharpening”, por um lado, foca em realçar detalhes e bordas na imagem, tornando os elementos mais nítidos e claros. A restauração, por outro, tenta recuperar a imagem original corrigindo distorções e removendo ruídos. Estes processos utilizam, geralmente, técnicas matemáticas e algoritmos de filtragem.

### Reconhecimento de padrões

O reconhecimento de padrões envolve a análise de imagens para identificar e categorizar padrões ou características específicas que podem ser repetidas ou significativas. Isto pode incluir a identificação de texturas, formas, ou regularidades em conjuntos de dados de imagem. Esta área é fundamental em aplicações como análise de imagens médicas, inspeção de qualidade de produtos em fábricas e análise de solo em imagens de satélite. Algoritmos de machine learning são frequentemente utilizados para treinar sistemas para reconhecer e interpretar esses padrões com alta precisão.

### Recuperação

A recuperação é uma técnica de computer vision que envolve encontrar imagens semelhantes a uma original numa grande base de dados. A recuperação de imagens é crucial em áreas como bancos de dados multimídia, sistemas de vigilância e arquivos digitais, onde o acesso rápido e preciso a imagens relevantes é essencial.

Existem duas técnicas para realizar a recuperação de imagens, CBIR e TBIR. CBIR significa Content-Based Image Retrieval e retrata que o modelo compara a imagem fornecida com as da base de dados através do conteúdo visual e as suas características intrínsecas, ou seja, analisa a cor, textura, forma entre outros atributos. TBIR significa Text-Based Image Retrieval e esta abordagem baseia-se em descrições textuais e metadados associados às imagens para realizar a busca. As imagens são indexadas com base em palavras-chave, legendas, anotações ou outras informações textuais fornecidas.

## Deteção de objetos

Como expresso anteriormente, este projeto enquadra-se na categoria de reconhecimento dentro dos vários tipos de processamento de imagens. Com este projeto pretendemos identificar objetos selecionados em imagens que são frames extraídos de câmaras conectadas à nossa aplicação. Existem dois tipos de processamento de imagens em computer vision: classificação de imagens e deteção de objetos. Os algoritmos de classificação apenas identificam a melhor classe que caracteriza a imagem designando uma “label” à imagem inteira baseado no seu conteúdo predominante, enquanto os algoritmos de deteção - implementados neste projeto - identificam quantos objetos de uma determinada categoria estão na imagem e onde cada objeto se encontra. Existem vários métodos relacionados à deteção de objetos, sendo que estes podem pertencer a abordagens tradicionais ou deep learning.

### Métodos tradicionais

Os métodos tradicionais de deteção de objetos envolvem geralmente extração de características e algoritmos de machine learning como Viola-Jones Detector, Histogram of Oriented Gradients (HOG) Detector e Deformable Part-based Model (DPM). Estas abordagens tradicionais contribuíram significativamente para o desenvolvimento de métodos de deteção de objetos, oferecendo “insights” na consideração de diferentes características e desafios de objetos.

### Viola-Jones Detector

Este algoritmo utiliza "Haar-like features" que são características formadas por elementos retangulares que capturam diferenças na intensidade dos píxeis. Estas características são combinadas em secções em cascata, onde cada estágio descarta rapidamente as áreas da imagem que não contêm elementos de interesse. As localizações que passam por todos os estágios são consideradas como contendo um objeto. A eficiência deste método reside na capacidade de filtrar rapidamente a maioria das áreas negativas (sem objeto), analisando detalhadamente apenas as áreas promissoras, o que o torna adequado para aplicações em tempo real.

### Histogram of Oriented Gradients Detector

Este método (HOG) funciona calculando gradientes de intensidade em células locais, que capturam orientações de bordas e contornos na imagem. Para cada célula são gerados histogramas dos gradientes, que são normalizados em blocos sobrepostos para garantir robustez contra variações de brilho e contraste. O vetor de atributos resultante representa a distribuição das arestas em toda a imagem e é utilizado como entrada para um classificador, como o SVM, para detetar objetos específicos.

### Deformable Part-based Model

O Deformable Part-based Model (DPM) é um método avançado de deteção de objetos que modela um objeto como uma coleção de partes interconectadas, onde cada parte pode mover-se ligeiramente em relação às outras. Este método utiliza um modelo hierárquico em que a aparência de cada parte e as relações espaciais entre as partes são capturadas por descritores de características, como o HOG. O DPM envolve um processo de aprendizagem que determina a melhor configuração das partes e as suas deformações permitidas, criando um modelo flexível que pode lidar com variações de pose, escala e forma do objeto. Esta técnica é particularmente eficaz para a deteção de objetos em cenários complexos, onde a rigidez dos modelos tradicionais anteriores pode falhar.

### Métodos deep learning

Os métodos de deep learning têm ganho significativamente maior atenção e sucesso nos últimos anos. As Redes Neuronais Convolucionais (CNNs) são uma tecnologia chave no deep learning para deteção de objetos. Estas conseguem aprender a extrair automaticamente características relevantes de imagens e aprender padrões complexos que são representativos de diferentes categorias de objetos. Existem dois tipos de arquiteturas de deteção de objetos baseadas em CNNs, “one-stage detectors” e “two-stage detectors”.

### Redes Neuronais Convolucionais

Rede neuronal convolucional ou CNN (do inglês Convolutional Neural Network) é um algoritmo de Deep Learning .O pré-processamento exigido numa CNN é muito menor quando comparado a outros algoritmos de classificação. Enquanto em métodos primitivos os filtros são desenhados manualmente (hand-engineered), as CNN, com treino suficiente, têm a habilidade de aprender estes filtros/características. A arquitetura de uma rede neuronal convolucional é análoga à do padrão de conectividade entre neurónios no cérebro Humano e foi inspirada pela organização do córtex visual.

### One-stage detectors

Os One-stage detectores são desenhados para prever coordenadas da bounding box (caixa que delimita um objeto detetado) e probabilidades de classe para vários objetos numa única passagem pela rede. Estes modelos são conhecidos pela sua simplicidade e eficiência, uma vez que eliminam a necessidade de uma etapa separada para elaboração de propostas. O objetivo destes detectors é examinar amostras densas de possíveis localizações de objetos e prever a presença de um objeto e a sua bounding box associada numa única fase (single-shot). Por realizarem a deteção numa passagem única, estes algoritmos, são mais rápidos na deteção, o que os torna ideais para aplicações com deteção em tempo real. Esta abordagem sacrifica ligeiramente a precisão dos modelos para se focar na sua rapidez, e tendem a ter dificuldade a detetar objetos pequenos e tratar instâncias de objetos com variações de tamanho significativas. Os one-stage detectors mais populares são YOLO, SSD, RetinaNet e CenterNet.

### Two-stage detectors

Os two-stage detectors, ao contrário do one-stage, seguem um processo de duas etapas. Na primeira etapa, estes detetores geram um conjunto de propostas de regiões candidatas a localizações com objetos. Estas propostas são depois refinadas e classificadas na segunda etapa. Two-stage detectors tendem a ter uma precisão maior, mas a ser mais lentos quando comparados com os one-stage detectors, devido à etapa adicional de elaboração de propostas. Os two-stage detectors mais populares são R-CNN, SPPNet, Fast R-CNN, Faster R-CNN, FPN e S2ANet.

# Metodologias e tecnologias

## Metodologia de trabalho

Quando se fala em desenvolvimento de software, existem duas maneiras de abordar a sua gestão e planeamento de modo a tornar a gestão de projeto mais eficiente e eficaz. Por um lado, temos as metodologias tradicionais, como o modelo cascata, que apostam numa abordagem sequencial e planeamento detalhado desde o início do projeto. Enquanto, em contraste, as metodologias ágeis dão prioridade à flexibilidade, colaboração e adaptação contínua ao longo do projeto.

### Método tradicional

A abordagem tradicional de gestão de projetos, também conhecida como método cascata, segue um processo linear e sequencial onde só se passa à fase seguinte quando a anterior é completada. É uma abordagem estruturada e orientada a planos com foco no planeamento completo e documentação antecipada. Todos os projetos, independentemente da sua natureza, seguem o mesmo ciclo de vida que incluem as fases de viabilidade, planeamento, design, construção, teste, produção e suporte. O projeto é planeado todo no início e não há qualquer mudança de requisitos, uma vez que nesta abordagem assume-se que os requisitos são fixos e o tempo e custo variáveis. Como tal, não é adequado a projetos de grande escala em que os requisitos para todo o projeto não se conseguem definir numa fase inicial.

### Metodologias ágeis

Ao contrário do método anterior, o foco das metodologias ágeis é o trabalho em equipa, colaboração com o cliente e flexibilidade. É uma abordagem iterativa que se concentra mais na incorporação do cliente e lançamentos contínuos em cada iteração do desenvolvimento do projeto. O conceito chave desta abordagem é que aprofunda a evolução das mudanças e esforço colaborativo para obter resultados em vez de um processo predefinido. O planeamento adaptativo é a principal característica das metodologias ágeis que convence muitos gestores de projeto.

Na seguinte tabela podemos ver a comparação de alguns aspetos comuns a ambas as abordagens.

|  |  |  |
| --- | --- | --- |
| Parâmetros | Metodologias Ágeis | Gestão de projeto tradicional |
| Flexibilidade e adaptabilidade | Ênfase na flexibilidade e adaptabilidade | Ênfase no planeamento e previsibilidade |
| Requisitos do projeto | Prioriza a satisfação do cliente e software funcional | Prioriza o seguimento de um plano rigoroso e atender aos requisitos do projeto |
| Comunicação | Encoraja comunicação e colaboração cara-a-cara | Confia na documentação escrita. Comunicação formal |
| Fases de projeto e planificação | Sprints curtos para planeamento e entrega. Permite alterações e ajustes ao longo do projeto | Fases de projeto mais longas para planeamento e entrega. Tem um plano rígido que e difícil alterar |
| Funcionalidade de equipa | Equipas multifuncionais | Equipas com funções divididas |
| Organização | Equipas auto-organizadas | Equipas organizadas hierarquicamente |
| Estrutura organizacional | Iterativa | Linear |
| Escala de projeto | Grande escala | Escala pequena e média |
| Modelo de desenvolvimento | Modelo de entrega evolutivo | Modelo de ciclo de vida |
| Requisitos do utilizador | Inputs interativos | Claramente definidos antes de implementação |
| Envolvimento do cliente | Alto | Baixo |

Fonte: <https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management> (com ajustes)

Optamos por seguir uma metodologia ágil por não saber de antemão os requisitos que vai ter a nossa aplicação, uma vez que esta abordagem permite-nos desenvolver a aplicação de maneira mais flexível e colaborativa. Vamos então analisar as principais metodologias ágeis e a forma como abordam a gestão dos projetos.

### Scrum

O scrum é a framework mais utilizada de todas as metodologias ágeis e caracteriza-se por etapas ou ciclos de desenvolvimento, conhecidos como sprints e pela maximização do tempo de desenvolvimento de um produto de sofware em direção ao Product Goal.

Uma equipa de Scrum é composta por 3 elementos com papeis bem definidos, o Product Owner, o Scrum Master e a equipa de desenvolvimento. O PO é o elemento que liga o cliente à equipa de desenvolvimento, assumindo a função garantir que a visão do cliente é entendida pela equipa de desenvolvimento e definindo os critérios de sucesso. É este que define o que deve ser feito, e em que ordem deve ser feito, atribuindo valor aos requisitos. O SM é referido como o facilitador do projeto uma vez que este certifica que as boas práticas da metodologia Scrum são seguidas, verifica relações saudáveis entre o PO e os programadores e elimina quaisquer obstáculos que impeçam uma colaboração produtiva. Por fim, a equipa de desenvolvimento, como o nome indica, é o conjunto de membros que trabalha em conjunto para criar, estar e lançar versões incrementais do produto final.

Além da equipa Scrum, esta metodologia conta com os seguintes eventos Scrum: Daily Scrum, Sprint, Sprint planning, Sprint review e Sprint retrospective.

O Daily Scrum consiste numa curta reunião diária que ocorre todos os dias no mesmo sítio à mesma hora. Nesta reunião a equipa avalia o trabalho realizado no dia anterior e planeia o que será feito nas próximas 24 horas.

A Sprint representa o período em que o trabalho deve estar concluído, não maior que um mês. Uma nova sprint começa quando a anterior acaba.

A Sprint planning meeting marca o início de uma nova sprint. Nesta reunião, toda a equipa Scrum participa e define objetivos, sendo que no final pelo menos um incremento de software utilizável deve ser produzido.

No fim de cada sprint, acontece a sprint review em que a equipa scrum mostra o trabalho realizado ao PO e stakeholders.

Por fim, a sprint retrospective, dá lugar à equipa para discutir o que correu bem e o que pode ser melhorado em relação à última sprint com o objetivo de atingir uma melhoria contínua.

No Scrum existem ainda 2 backlogs. O Product backlog que contém a lista das tarefas para sprints futuras e s Sprint Backlog que inclui as tarefas a realizar na sprint atual.

### XP

XP é outra metodologia ágil bastante presente no mundo dos projetos de desenvolvimento de software. Tem como objetivo criar sistemas de alta qualidade, assente nos princípios de proximidade com o cliente, testes constantes e ciclos de desenvolvimento curtos. Os 5 componentes básicos do XP são: comunicação, simplicidade, feedback, respeito e coragem. Esta abordagem aposta na velocidade e simplicidade com ciclos curtos de desenvolvimento e menos documentação.

No XP o cliente deve participar ativamente do processo de desenvolvimento. Tudo precisa da comunicação com o cliente. Este deve receber o melhor resultado possível a cada sprint, ver o progresso no sistema e ser informado de mudanças de planos para que possa dar a sua opinião.

Outra prática fundamental no XP é o pair programming. O pair programming consiste numa programação a pares num único computador em que um tem o papel de escrever o código enquanto o outro está sempre ao lado a rever. Desta maneira, erros e detalhes que possam passar despercebidos a um programador, o outro consegue cobri-los diminuído a probabilidade de falhas.

Para perceber mais facilmente o que difere o XP do Scrum podemos consultar a seguinte tabela:

|  |  |
| --- | --- |
| Scrum | Extreme Programming |
| Trabalho em iterações (sprints) podem ir até 1 mês | Trabalho em iterações de apenas 1-2 semanas |
| Modelos Scrum não permitem mudanças no seu cronograma ou diretrizes. | Permite mudanças nos cronogramas definidos |
| Ênfase na auto-organização | Ênfase em fortes práticas de engenharia |
| Equipa determina a sequência na qual o produto é desenvolvido | Equipa segue uma ordem de prioridade pré-determinada |
| A framework Scrum não está completamente descrita. Para adotá-la é necessário completá-la com métodos de trabalho, como XP ou Kanban | Pode ser aplicado diretamente a uma equipa. O XP é também conhecido pelas suas características “Ready-to-apply” |
| Não enfatiza as práticas de engenharia de software que os programadores devem usar | Enfatiza as técnicas de programação que os programadores devem seguir para garantir um melhor resultado |
| Requer que os programadores estejam conscientes da adoção de métodos de engenharia para garantir um melhor progresso ou qualidade | É muito rígido na adoção de métodos de engenharia, como pair programming, design simples, reestruturação, para garantir um melhor progresso ou qualidade. |
| O Product Owner define a prioridade das tarefas de acordo com os requisitos | O cliente define a prioridade das tarefas e analisa os lançamentos. |
| Existe flexibilidade para a equipa ajustar a prioridade das tarefas se necessário | A equipa não pode alterar a prioridade dos requisitos definidos pelo cliente |
| Valores: - Abertura  - Foco  - Compromisso | Valores: - Comunicação  - Simplicidade  - Feedback |
| Menor envolvimento do cliente no projeto | Maior envolvimento do cliente no projeto |

Fonte: https://www.geeksforgeeks.org/difference-between-scrum-and-xp/

### Kanban

A metodologia kanban visa uma melhoria contínua, flexibilidade no gerenciamento de tarefas, e fluxo de trabalho aprimorado. Por ser uma abordagem ilustrativa, o progresso de todo o projeto é facilmente compreendido em poucos segundos.

O Kanban, ao contrário do scrum, não possui um prazo definido para quando as tarefas devem ser concluídas. Em vez disso, esta metodologia é gerida pela prioridade dos requisitos num quadro Kanban. Este quadro é dividido em três estados para uma tarefa: pendente (to do), em desenvolvimento (doing) e concluída (done). As tarefas/requisitos são representadas por cartões visuais que transitam entre os estados até serem finalizadas. Assim, a equipa consegue ter uma noção visual do desenvolvimento do projeto.

### Scrumban

O Scrumban, como o nome sugere, é uma metodologia ágil que combina elementos do Scrum e Kanban. Esta abordagem envolve aplicar os princípios de visualização do fluxo de trabalho e processos flexíveis do Kanban a uma equipa da Framework Scrum. Porém, este princípio, remove alguns dos aspetos mais rígidos do Scrum e deixa as equipas formar uma abordagem personalizada para o desenvolvimento do projeto.

O Scrunban adota um quadro similar ao quadro Kanban na medida em que é a ferramenta principal de fluxo de trabalho com a diferença que permite adicionar as colunas necessárias ao quadro para marcar a fase de progresso para cada requisito.

O Scrum, define limites de tempo e tarefas por cada sprint, o contrário acontece no Kanban que se foca no fluxo de trabalho contínuo. No Scrumban é estabelecido um limite para a quantidade de trabalho que a equipa pode realizar num momento e este limite é o total de cartões no quadro.

Enquanto no Scrum cada tarefa é atribuída a um membro específico da equipa, no Scrumban o foco é estabelecer a ordem de prioridade das tarefas no quadro e cabe à equipa distribuir as tarefas a realizar pelos diversos membros.

Embora esta abordagem não tenha todas as reuniões típicas da estrutura Scrum, no Scrumban pode haver reuniões curtas durante as sprints para a equipa discutir planos e desafios encontrados.

### Metodologia de trabalho escolhida

Das metodologias ágeis apresentadas, excluímos já de partida o XP por não termos o cliente sempre presente no desenvolvimento da aplicação nem condições para aplicar o pair programming que esta abordagem requere. Das restantes, Scrum, Kanban e Scrumban decidimos escolher a última (Scrumbam) por oferecer o melhor de cada uma das metodologias em que esta se baseia enquanto descarta os aspetos menos favoráveis das mesmas. Deste modo conseguimos aproveitar não só a flexibilidade que o Kanban nos permite, mas também os princípios de planeamento Scrum, permitindo-nos adaptar nosso processo de desenvolvimento de acordo com as necessidades do mesmo.

Para aplicar a metodologia ágil Scrumban, realizamos uma reunião semanal com o Product Owner e Scrum Master, interpretados pelos professores orientadores. Nesta reunião discutimos o trabalho desenvolvido na última sprint que acaba nessa mesma reunião e determinamos o que deve ser feito para a próxima. Utilizamos a framework Jira Software da Atlassian que nos permite criar Issues que representam as tarefas/requisitos que a nossa aplicação tem ou vai ter. Aqui também conseguimos criar o quadro kanban que nos proporciona uma noção visual do desenvolvimento do trabalho e onde podemos colocar os Issues previamente criados. Conseguimos ainda criar e terminar as várias sprints ao longo do processo de desenvolvimento.

## Análise de requisitos

### Mockups

Uma imagem com texto, multimédia, captura de ecrã, software

Descrição gerada automaticamente

Figura 3.1 - Mockup início

Uma imagem com texto, captura de ecrã, multimédia, Software de multimédia

Descrição gerada automaticamente

Figura 3.2 - Mockup layout horizontal

Uma imagem com texto, software, Software de multimédia, multimédia

Descrição gerada automaticamente

Figura 3.3 - Mockup adicionar dispositivo



Figura 3.4 - Mockup adicionar dispositivo IP



Figura 3.5 - Mockup definições dispositivo



Figura 3.6 - Mockup emitir alerta para um dispositivo

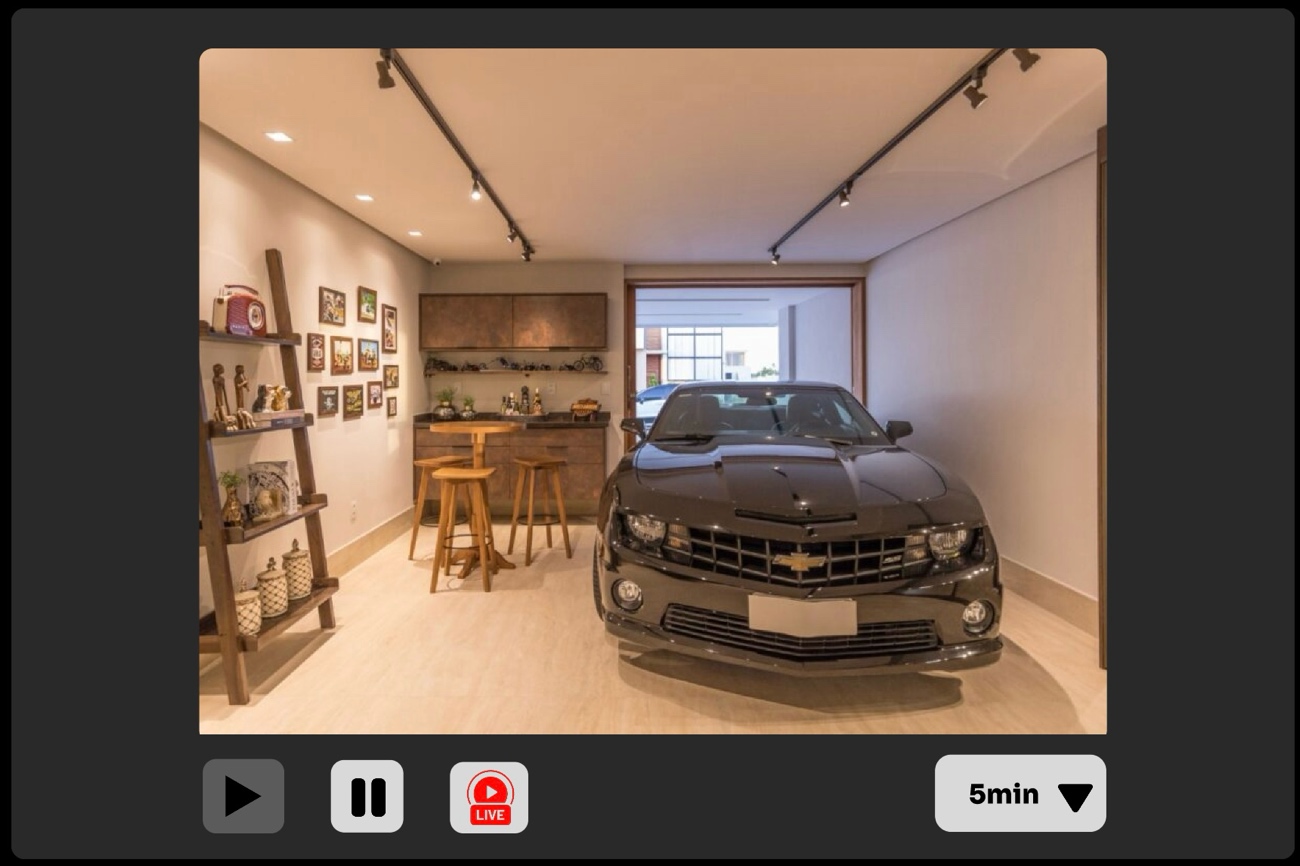


Figura 3.7 - Mockup dispositivo expandido

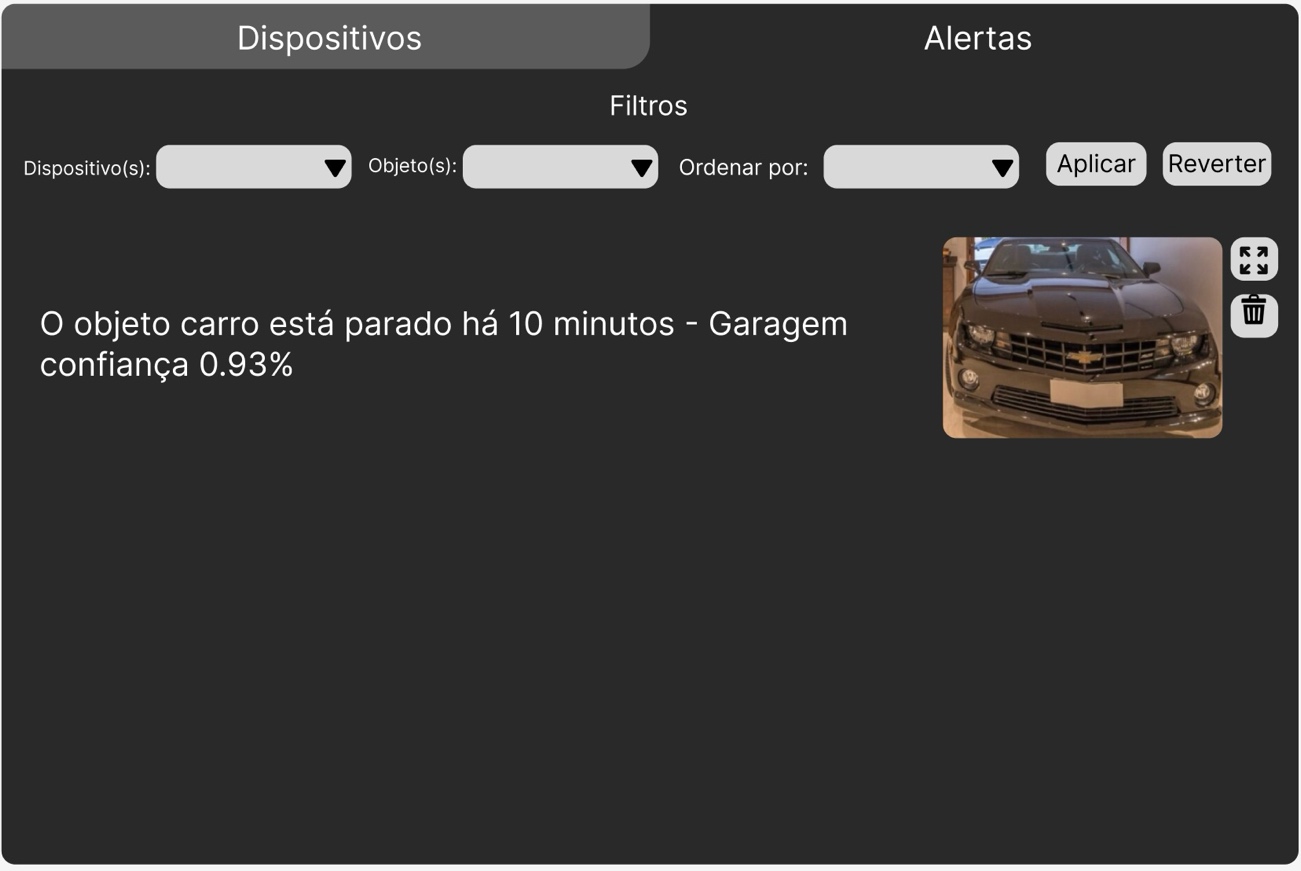


Figura 3.8 - Mockup página alertas

# Testes

## Teste de margem ideal para definir posição do objeto

### Introdução

O YOLO, após detetar um objeto, desenha uma bounding box que delimita o objeto detetado. Mesmo que o input não se altere, no caso deste projeto em que o input são camaras de vídeo, significa objeto na mesma posição, esta box pode variar ligeiramente devido a vários fatores. Estes fatores compreendem: variabilidade de rede, anchor boxes ou Non-Maximum Supression (NMS). Variabilidade de rede porque, a rede YOLO, como qualquer rede neuronal de deep lerning, tem variabilidade inerente nas suas previsões, devido à variação de valores inicias de peso, processos de otimização e descendente estocástico de gradiente (Stochastic Gradient Descent - SGD). Entende-se por Anchor boxes as formas e tamanhos predefinidos pelo algoritmo usadas para prever as bouding boxes. Se as anchor boxes não corresponderem exatamente à forma e tamanho dos objetos na imagem, pode também levar a variações nas bounding boxes. Por fim, NMS é uma técnica de pós-processamento usada pelo YOLO para filtrar as bounding boxes previstas. Neste processo é escolhida a box com maior grau de confiança e suprime-se as boxes sobrepostas com menor grau de confiança. O limite, isto é, o valor de confiança mínimo necessário para que uma bounding box seja considerada válida, usado nesta técnica pode afetar que bounding boxes são selecionadas, resulta em variações nas previsões finais.

Posto isto, submetemo-nos à criação de um cenário de testes para definir qual a margem ideal a definir para compensar a variação da posição da bounding box em objetos parados. Deste modo garantimos uma deteção de movimento mais rigorosa, e consequentemente alertas de segurança mais precisos e confiáveis.

### Condições do teste

Para avaliar qual a margem que confere a melhor exatidão dos resultados, foram simulados 17 cenários de testes variando os objetos a detetar e a distância a que se encontram do dispositivo de gravação. A posição da camara manteve-se a mesma durante todo o procedimento bem como o cenário onde se encontraram os objetos a detetar. Este cenário consistia num dia de sol, céu limpo com presença de sombra onde os objetos foram detetados.

Os cenários de teste foram os seguintes:

* Mochila (backpack) a 5,10 e 15 metros
* Bola (sports ball) a 5,10 e 12 metros
* Pessoa (person) a 5,10,15,20,25,30,35,40,50,60 e 70 metros

### Resultados obtidos:

Os gráficos seguintes representam, à esquerda, a distância entre os cantos inferior esquerdo (canto 1) do frame anterior e o atual e a distância do canto superior direito (canto 2) entre o frame anterior e atual à direita.

### Class “backpack”

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 4.1 - Gráfico cantos backpack a 5M

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 4.2 - Gráfico cantos backpack a 10M

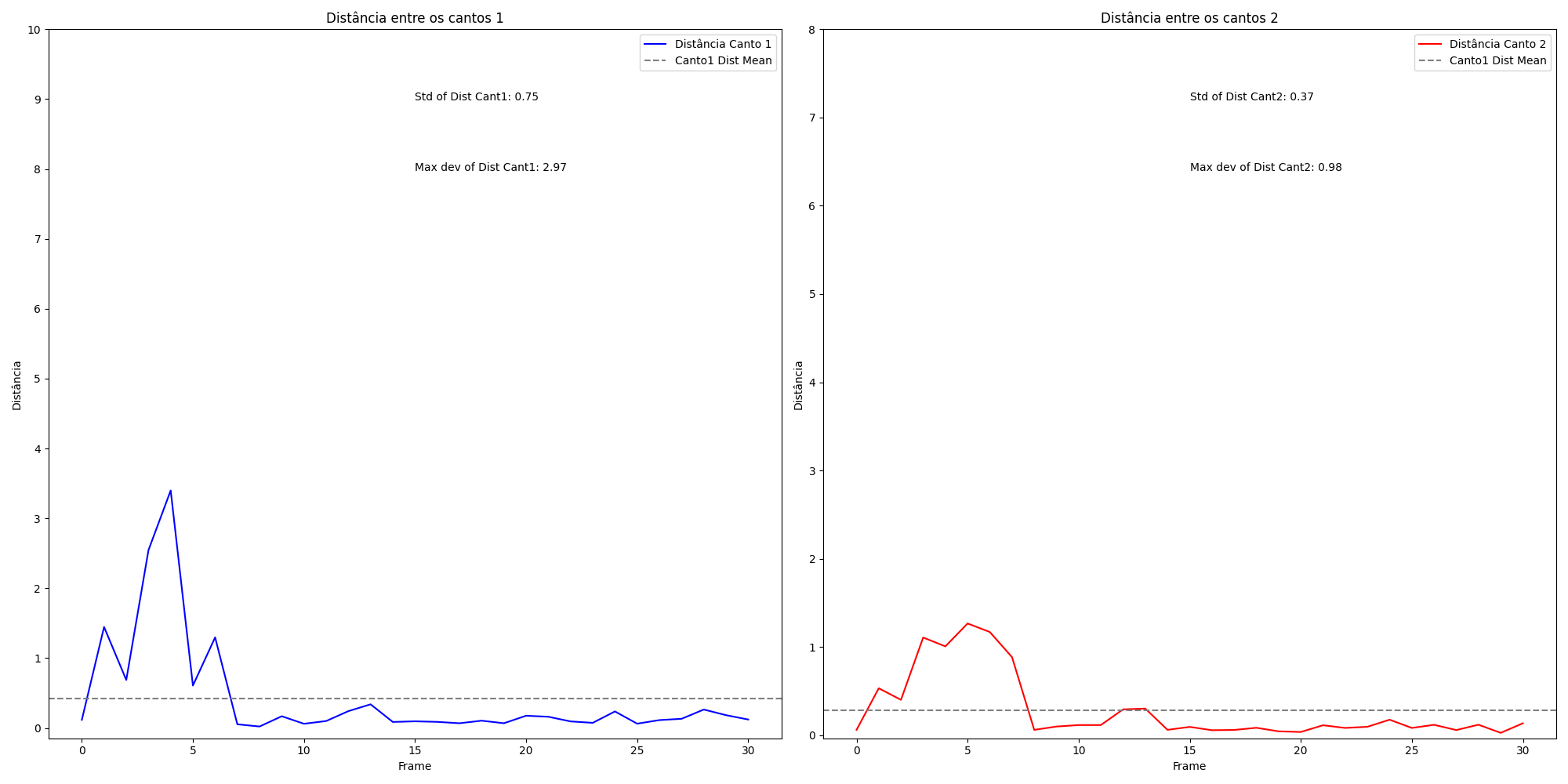


Figura 4.3 - Gráfico cantos backpack a 15M

### Class “sports ball”

Uma imagem com diagrama, file, texto, Gráfico

Descrição gerada automaticamente

Figura 4.4 - Gráfico cantos bola a 5M

Uma imagem com texto, file, Gráfico, diagrama

Descrição gerada automaticamente

Figura 4.5 - Gráfico cantos bola a 10M

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 4.6 - Gráfico cantos bola a 12M

### Class “Person”

Uma imagem com file, texto, diagrama, Gráfico

Descrição gerada automaticamente

Figura 4.7 - Gráfico cantos person a 5M

Uma imagem com file, diagrama, texto, Gráfico

Descrição gerada automaticamente

Figura 4.8 - Gráfico cantos person a 35M

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 4.9 - Gráfico cantos person a 70M

### Conclusão:

Após teste dos vários cenários, seguidos da análise e discussão dos resultados obtidos, consideramos 4 px como margem ideal a aplicar para desprezar as possíveis variações das bounding boxes geradas pelo modelo de deteção usado em objetos parados. Isto deve-se ao facto dos maiores desvios obtidos serem na ordem dos 3px, sendo o máximo deles 3.15 no caso de teste backpack a 5 metros. Sendo 4 o próximo inteiro, 4px cobre todos os desvios que obtivemos na fase de testes.

## Teste da arquitetura para recolher frames de dispositivos

### Introdução

Serve o presente teste para testar qual a arquitetura que melhor se adequa à nossa aplicação e aos seus requisitos. Para isso temos 4 modelos para teste sendo eles:

### Grab & Retrieve

Neste modelo, após inicializarmos as câmaras com o videCapture de cada dispositivo, criamos uma thread de captura por dispositivo. Esta thread apenas agarra (grab) continuamente frames do dispositivo para posterior descodificação e análise do frame. Na thread principal, de X em X segundos (definido pelo utilizador) chamamos a função retrieveFrames que percorre o vetor das câmaras previamente inicializadas e recolhe (retrieve) o último frame agarrado (grabbed) pela thread de captura do dispositivo associado descodificando-o. De seguida enviamos cada frame para a função YOLO para o processar.

### Parallel

Nesta arquitetura, para cada dispositivo de captura selecionado, é criada uma thread que lê, descodifica e envia o frame ao YOLO para detetar objetos na imagem.

### Open & Close

Neste padrão (ausente de threads) cada dispositivo de captura é inicializado, captura o frame, processa-o para deteção através do YOLO e fecha o dispositivo, seguindo para o próximo continuamente, N vezes, sendo N o número de dispositivos selecionado.

### Open & Close threading

Neste modelo, inicializa-se os dispositivos e cria-se uma thread por dispositivo para ler um frame. Após todas as threads terem lido um frame, cria-se uma thread por frame para o processar no YOLO. Finalmente, depois das threads terminarem o processamento, fecham-se os dispositivos inicializados no início.

### Condições de teste

Para comparar os vários modelos de implementação, utilizamos a biblioteca “psutil” do python que nos permite avaliar a % de CPU usage bem como a % de RAM usage durante todo o processo. Para avaliar os cenários, utilizamos as mesmas 5 câmaras, das quais 2 locais e 3 por IP e definimos 40 iterações para avaliar o tempo de execução.

### Resultados obtidos

### Grab & Retrieve

Neste padrão não foi possível concluir os testes porque ao usar câmaras IP em vez de locais no videoCapture, o retrieve não consegue descodificar (retrieve) o frame agarrado (grabbed) anteriormente, pelo que excluímos das opções.

### Parallel

Uma imagem com texto, file, diagrama, captura de ecrã

Descrição gerada automaticamente

Figura 4.10 - Gráfico performance parallel

Ao analisar a percentagem elevada de CPU usage nesta arquitetura, acrescentamos dois novos casos de teste para tentar diminuir a % de CPU usage sacrificando tempo de execução. Para isso, adicionamos um time.sleep de 5 e 10 segundos.

### Uma imagem com texto, file, captura de ecrã, diagrama Descrição gerada automaticamenteParallel 5 sec

Figura 4.11 - Gráfico performance parallel 5 segundos

### Uma imagem com texto, file, diagrama, Gráfico Descrição gerada automaticamenteParallel 10 sec

Figura 4.12 - Gráfico performance parallel 10 segundos

### Uma imagem com texto, file, Gráfico, diagrama Descrição gerada automaticamenteOpen & Close

Figura 4.13 - Gráfico performance open & close

### Open & Close threading

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 4.14 - Gráfico performance open & close threading

### Discussão dos Resultados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Método | Tempo execução | Máx – Init CPU % | Máx – Init RAM % | Mean – Init CPU % | Mean – Init RAM % |
| Parallel | 1m 44s | 100 | 10,80 | 85,88 | 10,03 |
| Parallel 5 sec | 5m 24s | 100 | 8,50 | 34,55 | 7,67 |
| Parallel 10 sec | 7m 39s | 35,20 | 8,30 | 20,46 | 4,11 |
| Open & Close | 17m 58s | 28,30 | 19,50 | 9,86 | 13,53 |
| Open & Close threading | 10m 25s | 32,40 | 13 | 18,14 | 8,77 |

Através dos dados retirados podemos criar a seguinte tabela:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Menor tempo execução | Maior tempo  Execução | Menor CPU usage | Maior CPU usage | Menor RAM usage | Maior RAM usage |
| Parallel | Open & Close | Open & Close | Parallel | Parallel 10 sec | Open & Close threading |

Neste projeto, priorizamos o menor custo computacional em relação ao tempo de execução. Dito esto, excluímos o modelo Parallel pois utiliza o esforço máximo do CPU e o Open & Close threading por representar o maior esforço sobre a RAM.

Entre o Parallel 5 sec e o Parallel 10 sec, o último representa o melhor comparativamente porque demora 1,42 vezes mais para concluir o teste mas utiliza menos 1,69 vezes o CPU e 1,87 vezes menos a RAM.

O Open & Close é o que apresenta menor custo para o processador (9,86%) mas demora demasiado tempo em relação ao Parallel 10 sec que utiliza 20,46% do processador, o que representa 2,08 vezes mais que o anterior, mas acaba o teste 2,35 vezes mais cedo, o que representa um rácio maior de tempo / CPU usage, pelo que também exclui o Open & Close em comparação ao Parallel 10 sec.

### Conclusão

Após análise e discussão dos resultados obtidos, escolhemos a arquitetura Parallel 10 sec como a mais adequada para a nossa aplicação por ser o mais equilibrado quando aplicamos os nossos critérios.

# Conclusões ou Conclusão

Inserir aqui as conclusões ou conclusão. Trata-se de um elemento **obrigatório**.

A conclusão:

* Deve ser sucinta;
* Não deve conter informações ou ideias novas;
* Deve permitir concluir se se atingiram os objetivos enunciados na introdução.

Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão.

Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão.

# Bibliografia ou Referências Bibliográficas

Inserir aqui a bibliografia ou referências bibliográficas. Trata-se de um elemento **obrigatório**.

Notas: o sistema a adotar para a apresentação das referências bibliográficas e as suas citações deve:

* Respeitar uma norma estabelecida;
* Seguir as práticas mais disseminadas na área em causa;
* Ser empregue de modo uniforme em todo o documento.

Bibliografia – quando se coloca toda a bibliografia consultada;

Referências bibliográficas – quando se faz referência apenas à bibliografia citada.

# Anexos

Elemento a figurar, **quando aplicável**.

# Glossário

Elemento a figurar, **quando aplicável**.