Universidade do Minho
Escola de Engenharia
Departamento de Informática

Catarina Freitas da Cruz

Framework para análise de comportamentos de objetos interativos em vídeo jogos

January 2022

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Catarina Freitas da Cruz

Framework para análise de comportamentos de objetos interativos em vídeo jogos

Master dissertation
Integrated Master's in Informatics Engineering

Dissertation supervised by
Paulo Novais
André Pimenta

January 2022

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

ABSTRACT

With exponential growth both in the area of Artificial Intelligence and videogames, the creation of platforms that help players has become fundamental. The creation of an analytical tool that analyzes human behavior, opens the door to more dynamic, competitive and fair games.

The analysis of a player's screen allows identifying, detecting and tracking movements of certain objects, in real time, and can help or monitor them. Whatever the case, it is first necessary to identify and detect the objects visualized, through Object Detection algorithms. Then, once the object has been identified, it is possible to predict its next location, as well as track its movement, using Object Tracking algorithms.

It is possible to analyze the player's screen by interleaving object tracking with object detection, either when it disappears from view, or to obtain confirmation that the correct object is being followed, thus helping the player.

This dissertation aims to develop a model capable of identifying the movement of a given object, in real time, in a game environment, using Machine Learning and Computer Vision techniques, more specifically methods of Object Detection and Object Tracking.

The practical environment will be developed using the OpenCV python library, which has a diverse range of computer vision algorithms available and also allows the parallel use of CPU and GPU for the optimization of these same algorithms.

KEYWORDS Artificial Intelligence, Machine Learning, Computer Vision, CNN, Object Detection, Object Tracking, OpenCV

RESUMO

Com um crescimento exponencial tanto na área da Inteligência Artificial como dos vídeo-jogos, a criação de plataformas que auxiliam os jogadores passou a ser fundamental. A criação de uma ferramenta analítica que analise comportamentos humanos, abre portas a jogos mais dinâmicos, competitivos e justos.

A análise do ecrã de um jogador permite identificar, detetar e rastrear movimentos de determinados objetos, em tempo real, podendo ter o intuito de o ajudar ou de o vigiar. Seja qual for o caso, é necessário, primeiramente, identificar e detetar os objetos visualizados, através de algoritmos de *Object Detection*. De seguida, já identificado o objeto, é possível prever a sua próxima localização, bem como rastrear o seu movimento, utilizando algoritmos de *Object Tracking*.

Intercalando o rastreamento com a deteção de objetos, quer quando este desaparece de vista, quer para obter confirmação que se está a seguir o objeto correto, é possível assim analisar o ecrã do jogador para o poder ajudar.

Esta dissertação tem como objetivo desenvolver um modelo capaz de identificar o movimento de um determinado objeto, em tempo real, no ambiente de um jogo, utilizando para isso técnicas de *Machine Learning* e *Computer Vision*, mais especificamente métodos de *Object Detection* e *Object Tracking*.

O ambiente prático será desenvolvido utilizando a biblioteca OpenCV para Python, que tem ao dispor um diverso leque de algoritmos de *Computer Vision* e ainda permite a utilização paralela de CPU e GPU para a otimização destes mesmos algoritmos.

PALAVRAS-CHAVE Inteligência Artificial, Machine Learning, Computer Vision, CNN, Object Detection, Object Tracking, OpenCV

CONTEÚDO

LISTA DE FIGURAS

SIGLAS

AI - Artificial Intelligence
ANN - Artificial Neural Networks
CDF-CSR - Discriminative Correlation Filter with Channel and Spatial Reliability
CNN - Convolutional Neural Network
CPU - Central Processing Unit
CSRT - Channel and Spatial Reliability Tracking
CV - Computer Vision
FPS - Frames Per Second
GOTURN - Generic Object Tracking Using Regression Networks
GPU - Graphics Processing Unit
IA - Inteligência Artificial
ID - Identity Document
KCF - Kernelized Correlation Filters
KNN - K-Nearest Neighbors
LSTM - Long short-term memory
MAP - Mean Average Precison
MIL - Multiple Instance Learning
ML - Machine Learning
MOOSE - Minimum Output Sum of Squared Error
PLN - Processamento de Linguagem Natural
RCNN - Region-based Convolutional Neural Networks
RGB - Red Green Blue
RNN - Recurrent Neural Network
ROLO - Recurrent YOLO
ReLU - Rectified Linear Unit
SORT - Simple Online Real-time Tracker
SSD - Single Shot Detector
SVM - Support Vector Machines
TLD - Tracking, Learning, and Detection
VC - Visão por Computadores
YOLO - You Only Look Once

Parte I

MATERIAL INTRODUTÓRIO

1

INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A indústria dos jogos continua, desde a sua criação, a ser uma área em expansão e, nos últimos anos, têm-se cruzado cada vez mais com a área de Inteligência Artificial, quer para auxiliar o jogador sugerindo dicas ou recomendações baseadas nas jogadas deste, quer para tornar o ambiente mais competitivo através da criação de bots.

No entanto, a visão por computadores tem sido pouco explorada nesta área. Muitos jogos parecem ter uma IA muito avançada, contudo, na maioria das vezes, é apenas uma ilusão. Normalmente, a IA embutida nos jogos tem uma vantagem relativamente aos jogadores, uma vez que conseguem aceder a todos os estados de jogo possíveis, incluindo aqueles indisponíveis ao oponente humano.

Este projeto teve início com a empresa `Anybrain`, criada por alumni da Universidade do Minho em 2015. Desde então, utilizam inteligência artificial em prol da sociedade, em especial da indústria de vídeo-jogos. São uma empresa que adora vídeo-jogos, mas que sabe que estes só fazem sentido quando são divertidos, competitivos e, acima de tudo, justos. Acreditam que qualquer pessoa pode jogar e que as comunidades *online* e de *esports* merecem um bom ambiente e fazem de tudo para o poder melhorar.

Os desportos eletrónicos (esports) estão em crescimento a nível mundial, e devido à pandemia provocada pela covid-19, ganharam ainda mais destaque. Existem cada vez mais jogos e torneios, e com isso cada vez mais pessoas aderem a plataformas relacionadas, seja para visualizar torneios, seja para participar neles. Por isso é fundamental, melhorar a experiência tanto de quem participa como de quem assiste.

O principal produto da `Anybrain` consiste numa plataforma, desenvolvida com técnicas de *Machine Learning*, que aprende automaticamente, em tempo real, sem qualquer fricção e com mais de 99% de precisão, a maneira como um jogador interage com um jogo. Esta plataforma adapta-se automaticamente a diferentes plataforma e a qualquer jogos e deteta proativamente fraudes e comportamentos anormais num jogador.

Assim sendo, este tema surge com o objetivo de desenvolver um modelo capaz de identificar, em tempo real, o movimento de um determinado objeto num ambiente de um jogo, sem acesso a in-game data, ou seja totalmente independente de quem desenvolve o jogo, utilizando para tal técnicas de *Machine Learning* e *Computer Vision*. Este modelo poderá, no futuro, integrar a plataforma *Anybrain*

servindo de base para auxílio de um jogador, deteção de alguma infração cometida pelo mesmo, ou até facilitar no treino de um jogo, visto que permite a monitorização, em tempo real, de jogos de esports.

1.2 MOTIVAÇÃO

Durante muito tempo, sonhou-se em criar máquinas com características idênticas às humanas, nomeadamente com inteligência para que estas pudessem pensar e agir como humanos. Uma das ideias mais fascinantes era a de dar visão aos computadores, para que estes pudessem interpretar o mundo à sua volta. Todavia, construir máquinas que possam ver e interpretar é um problema interessante, mas notoriamente complexo de resolver. Para o ser humano, que está constantemente a receber informação visual, reconhecer um rosto ou um determinado objeto parece uma tarefa fácil, contudo, para uma máquina, que tem dados limitados, a simples catalogação de uma imagem é uma tarefa bastante complexa.

Graças aos avanços no poder computacional e nas áreas de inteligência artificial (IA) a fição tornou-se finalmente realidade. *Computer vision* tornou-se finalmente um sub-campo de destaque de inteligência artificial. E dentro desta área também a deteção e o rastreamento de objetos têm evoluído constantemente. Para além de detetar um objeto numa imagem estática, rastrear o seu movimento e prever a sua trajetória, em tempo real, são conhecimentos cada vez mais aplicados nas diversas áreas, desde detetar automóveis que tenham cometido infrações, ao auxílio na medicina, passando também pelos vídeo-jogos.

Também os vídeo-jogos sofreram uma grande popularização nos últimos anos, principalmente com o auxílio do *streaming*, atraindo milhares de visualizadores a um novo mundo. Passou a ser ainda mais fundamental o conforto tanto para os jogadores como para os visualizadores, havendo um crescimento exponencial de ferramentas ligadas a este fim e ainda mais em conjunto com o inteligência artificial e *machine learning*. Criaram-se *bots*, ferramentas analíticas, plataformas de *streaming*, entre muitas outras aplicações destinadas a jogos.

Com o auxílio de *computer vision* e *machine learning* é possível desenvolver algoritmos que integrem plataformas que ajudem a melhorar a experiência de vídeo-jogos, bem como a analisar comportamentos humanos. No final, o mais desafiante é integrar aplicações ou produtos já existentes com tecnologias que recorrem a *computer vision* e, consequentemente, melhorá-los.

1.3 OBJETIVOS

Para o desenvolvimento desta dissertação, o primeiro passo passa por analisar ferramentas já existentes de identificação, deteção e rastreamento de objetos, bem como perceber e explorar modelos de *machine learning* passíveis a ser usados.

Após uma primeira fase de análise e exploração dos conceitos base, o passo seguinte passa por desenvolver uma aplicação/agente capaz de analisar a informação visual processada no ecrã de um

jogador. De seguida, pretende-se desenvolver um algoritmo de reconhecimento de diferentes objetos. Após desenvolvido o mesmo é necessário treiná-lo repetidamente para obter bons resultados. Como último objetivo, será criado um sistema de reconhecimento e rastreamento de objetos no ecrã de um utilizador em tempo real.

Em suma, serão seguidos os seguintes passos:

- Estudo dos conceitos base;
- Estudo teórico de ferramentas e algoritmos;
- Criação de uma aplicação/agente que recolha informação contida no ecrã;
- Análise e exploração de ferramentas e algoritmos de identificação e deteção de objetos;
- Análise e exploração de ferramentas e algoritmos de rastreamento de objetos;
- Criação de algoritmo de reconhecimento de diferentes objetos;
- Treino de algoritmo de reconhecimento de diferentes objetos;
- Desenvolvimento de software para análise de objetos;

1.4 ESTRUTURA DO DOCUMENTO

O presente documento é relativo à pré-dissertação e encontra-se dividido em três capítulo. No primeiro e atual capítulo encontra-se a introdução, onde é abordado a contextualização, motivação e objetivos da dissertação em estudo.

O segundo capítulo é referente ao estado da arte, no qual são apresentados os conceitos base e essenciais ao desenvolvimento do projeto. Em particular, são abordados os temas de inteligência artificial, *machine leanring* e *computer vision*, explorando ainda a biblioteca `OpenCV`. São ainda referenciados os principais algoritmos de *Object Detection* e *Object Tracking*, bem como é feita uma avaliação comparativa da performance destes.

No último capítulo é efetuado um ponto de situação onde se apresentam as conclusões sobre a implementação do problema em estudo. Ainda nesse capítulo, é delineado o trabalho futuro da dissertação, sendo apresentado um diagrama de Gantt para demonstrar a distribuição temporal das tarefas a realizar.

2

ESTADO DA ARTE

2.1 INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

Artificial Intelligence (AI), ou em português Inteligência Artificial(IA), é o resultado de uma procura contínua com o objetivo de criar máquinas que façam tudo o que os humanos conseguem, desde ouvir, ver, perceber, pensar e até exibir emoções.

Esta ideia surgiu em 1950, quando o cientista da computação Alan Turing propôs um teste para avaliar a inteligência de uma máquina: "*If a machine can trick humans into thinking it is human, so then it has intelligence.*", sugeriu Alan Turing. Esta ideia manteve-se até 1995, altura em que se realizou a primeira conferência sobre *Artificial Intelligence*, onde se introduziu pela primeira vez este termo. McCarthy, juntamente com Alan Turing, Allen Newell, Herbert A. Simon e Marvin Minsky, ficaram assim conhecidos como os pais fundadores da AI. Desde aí desenvolveram-se máquinas quer com o intuito de competição, começando com o Deep Blue, que em 1997 derrotou o campeão mundial de xadrez Garry Kasparov até 2017 quando o AlphaGo venceu o campeão mundial Ke Jie no complexo jogo de tabuleiro Go, quer com o intuito de auxiliar o ser humano nas tarefas mais mundanas, desde a Roomba que foi o primeiro aspirador róbotico que aprendeu a navegar e limpar casas até à Siri e Alexa, assistentes virtuais com uma interface de voz.

IA tem sub-campos em várias áreas como processamento de linguagem natural (PLN), robótica, aprendizagem máquina e aprendizagem profunda (machine learning (ML) e Deep Learning), sistemas especialistas, fala ou reconhecimento de voz, automação inteligente e visão por computadores (VC). Gollapudi (2019)

2.1.1 *Machine Learning*

Machine Learning (ML) é um tipo de inteligência artificial que permite criar máquinas capazes de desenvolver inteligência, uma vez que estas máquinas, utilizando apenas a sua própria experiência, isto é, dados históricos, são capazes de desenvolver um algoritmo de previsão de resultados sem serem explicitamente programadas para tal.

O objetivo destes algoritmos é produzir um resultado na forma de uma regra que possa ser generalizado. *Machine Learning* é caracterizada pela forma como os algoritmos aprendem e prevêem, sendo normalmente subdivididos em quatro abordagens básicas Ayodele (2010):

- aprendizagem supervisionada/*supervised learning*, visto que usa dados catalogados para a previsão dos resultados;
- aprendizagem não supervisionada/*unsupervised learning*, uma vez que não é fornecido ao algoritmo qualquer dado catalogado ou rotulado;
- aprendizagem semi-supervisionada/*semi-supervised learning*, combinação das duas primeiras abordagens, ou seja, apenas parte dos dados estão catalogados;
- aprendizagem por reforços/*reinforcement learning*, baseado em recompensar comportamentos desejados e/ou punir comportamentos indesejados

Alguns exemplos bastante comuns da utilização de algoritmos de *machine learning* são mecanismos de recomendação, deteção de fraude, filtragem de spam, deteção de ameaças de malware, entre outros.

2.2 INTRODUÇÃO À VISÃO POR COMPUTADOR

Computer Vision (CV) é o campo de estudo que procura desenvolver técnicas que permite computadores, dispositivos ou máquinas em geral verem, compreenderem e interpretarem conteúdo de dados visuais digitais, como fotografias, vídeos ou até mesmo uma representação gráfica de um local ou de um mapa de intensidade de calor.

Embora o auge da visão por computador tenha sido recentemente, não é uma nova área científica. Um dos primeiros e mais importantes artigos que influenciaram *computer vision* foi publicado no final dos anos 1950 por dois neuro-fisiologistas. O artigo intitulado "Receptive fields of single neurons in the cat's striate cortex"descreve as propriedade da resposta dos neurónios do córtex visual de um gato, chegando à conclusão que existem neurónios simples e complexos no córtex visual primário e que o processamento visual começa com estruturas simples como bordas e linhas.

Apesar deste artigo não ter nada a ver com a área de engenharia ou teste de softwares, foi essencial para se criar a estrutura base dos algoritmos, uma vez que *computer vision* funciona da mesma maneira que a visão humana. Similarmente a um humano a tentar decifrar uma imagem distante, CV primeiro discerne arestas e formas simples e, só de seguida é que completa a informação das suas previsões após várias iterações.

Após este artigo, os próximos marcos na história da CV foi a invenção de um scanner digital de imagens, que permitiu aos computadores digitalizarem e adquirirem imagens, e ainda a transformação de imagens em matrizes de números.

Só após estas ferramentas base serem descobertas e desenvolvidas é que se pode avançar com a implementação de algoritmos de *computer vision*. Esta tecnologia necessita de analisar inúmeros dados

repetidamente, para conseguir gradualmente distinguir pequenas diferenças até conseguir reconhecer imagens. *Computer vision* implementa técnicas de *machine learning*, *deep learning*, *convolutional neural network*(CNN) ou *recurrent neural network* (RNN) e, nalguns casos específicos, implementa técnicas de processamento de linguagem natural para a análise de texto extraído de imagens. A CNN é usada para interpretar uma só imagem, enquanto que a RNN é usada, de maneira semelhante, para relacionar imagens em vídeo.

Com os avanços na área de *deep learning*, a construção de métodos para classificar imagens, detetar objetos, fazer rastreamento e manipulação de imagens, tornou-se cada vez mais simples e precisa. Hoje em dia, existem inúmeras aplicações que utilizam *computer vision*, desde aplicações que permitem encontrar um objeto ou uma pessoa num vídeo, compreender movimento e padrões num vídeo, até aumentar ou diminuir o tamanho, brilho ou nitidez de uma imagem. Demush (2019)

2.2.1 *Desafios da Visão por Computador*

Computer vision parece um problema fácil de resolver, uma vez que crianças e mesmo animais conseguem ver e interpretar o mundo que os rodeia. Todavia, permanece um problema por resolver, em parte devido à limitada compreensão da visão biológica e em função da complexidade da perceção da visão relativamente a um mundo físico dinâmico e infinitamente variável.

Analisar uma simples imagem digital pode ser bastante complicado, uma vez que esta pode apresentar diversa informação e conteúdo diverso. Uma imagem pode retratar um simples objeto, possuir uma descrição, descrever um modelo multidimensional, entre outros. A interpretação destas imagens influencia a precisão dos mecanismos de *computer vision*. Brownlee (2019)

Para além disso, os dados visuais digitais podem ser adquiridas através de várias fontes, como webcams, câmaras fotográficas ou de filmagem, gravadores de vídeo, scanners, entre outros.

De seguida, são exibidas uma série de imagens problemáticas que tornam o processo da interpretação de imagens complexo:

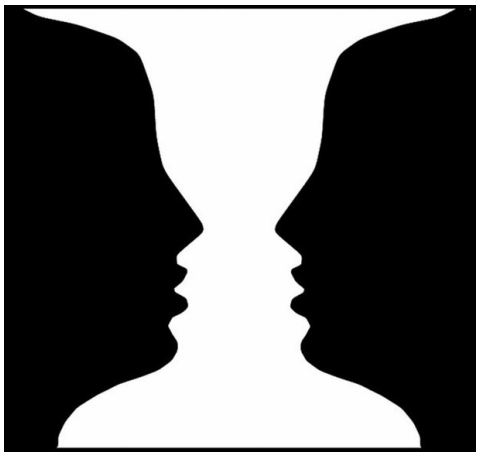


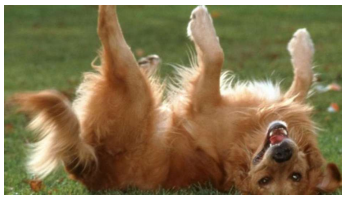
Figura 1: Ilusão: vaso ou rosto?

Tal como se pode observar na figura 1, existem imagens que até para o olho humano podem trazer problemas, visto que dependendo da perspetiva da pessoa que visualiza, isto é, na imagem representada tanto se pode observar um vaso como duas caras de perfil.

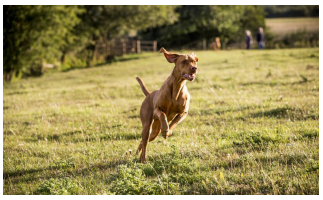
Podem ainda existir problemas com a resolução e qualidade de uma imagem, dificultando o foco de um objeto e a interpretação da imagem. Este problema surge principalmente em imagens com um fundo pouco contrastante em relação ao objeto principal como por exemplo em imagens com pouca iluminação, onde se torna difícil de visualizar o pretendido.

(a) Diferentes raças de cães

(b) Cão de costas



(c) Cão de pernas para o ar



(d) Cão a correr

Figura 2: Diversas figuras de cães

Por último, as figuras 2 são todas relativas a cães, todavia parecem totalmente diferente umas das outras, isto deve-se ao facto de haver várias raças de animais (sub-figura 2(a)), serem imagens de diferentes ângulos (sub-figuras 2(b), 2(c)) e caso um objeto esteja em movimento, pode apresentar uma forma ligeiramente diferente, como é o caso do cão a correr (sub-figura 2(d))

Para o ser humano, que cresceu a ver estas diferentes variações e está constantemente a receber informação visual, a catalogação destas imagens parece trivial, contudo para uma máquina, que tem dados limitados, esta catalogação é bastante complexa.

2.2.2 *Aplicações no mundo real*

Computer vision está em constante uso e progresso e como tal existem cada vez mais áreas e aplicações que utilizam esta tecnologia. Desde retail até segurança, saúde, agricultura e automação

industrial, são algumas das áreas industriais que mais têm beneficiado com os avanços tecnológicos. Meel (2021)

Indústria automóvel

O recente desenvolvimento dos carros autónomos trouxe uma revolução na maneira como a inteligência pode ser incorporada nos automóveis para gerir congestionamentos, acidentes rodoviários e manutenção pró-ativa dos veículos. Os veículos são equipados com sensores e câmaras que conseguem recolher imagens e dados sobre o ambiente envolvente. Com isto, os veículos tornam-se capazes de detetar sinalização na via como sinais de limite de velocidade, avisar o condutor quando este se encontra estacionado numa zona proibida, encontrar um lugar de estacionamento livre e até dar indicações ao condutor para chegar a um determinado local ou até identificar obstáculos perigosos no meio da estrada.

Para além disso, tem havido um uso intensivo de tecnologias de *computer vision* em cidades inteligentes para a identificação de veículos que cometeram infrações como excesso de velocidade, desobedecer um sinal vermelho num semáforo ou um sinal de STOP, andar em contra-mão, entre outros. Esta tecnologia também pode ser utilizada para detectar a ocupação de um parque de estacionamento utilizando apenas as câmaras de vigilância.

Indústria biomédica e da saúde

Similarmente à indústria automóvel, a indústria da saúde tem adotado cada vez mais técnicas de *machine learning* e *computer vision* para auxiliar e facilitar muitos dos processos envolventes na análise de um paciente. Através do reconhecimento de imagens é possível detetar ligeiras diferenças entre imagens cancerígenas e não cancerígenas, auxiliando o médico a diagnosticar precocemente tumores e cancro através de dados de exames de ressonância magnética. É ainda possível distinguir se o cancro é maligno ou benigno permitindo um tratamento muito mais personalizado e eficiente. Da mesma forma, armazenar os dados de análise e obter perceções de relatórios digitais de saúde de pacientes pode melhorar significativamente a precisão e eficácia dos tratamentos.

Ademais, existem robôs capazes de realizar cirurgias complexas com precisão e eficiência. Em casos de cirurgias, é ainda possível prever a quantidade de perda sanguínea evitando transfusões de sangue desnecessárias.

Com a situação pandémica devido ao coronavírus, o reconhecimento do uso de máscara e equipamento de proteção tornou-se essencial e indispensável para limitar a disseminação do virús. Por esse motivo, empresas privadas como a Uber implementaram um mecanismo de reconhecimento facial especialmente para os motoristas e estafetas. Este mecanismo obriga os trabalhadores a tirarem uma selfie com a máscara antes de começarem o seu percurso. Assim, é confirmado que os todos os funcionários cumprem as normas de segurança tornando o transporte público muito mais seguro nestes tempos pandémicos Nunes (2020).

Agricultura

A monitorização dos animais com *computer vision* e *machine learning* é fundamental para a agricultura. Os sistemas de visão inteligente têm como objetivo analisar o comportamento animal e aumentar a produtividade, saúde e bem-estar dos animais e, deste modo, influenciar a produção e os benefícios económicos da indústria. Para tal são utilizadas câmaras para vigiar e monitorizar a saúde de animais específicos, isto é, porcos, gado ou aves, conseguindo assim uma análise muito mais personalizada.

Por outro lado, aplicações de *computer vision* permitem ainda acompanhar o crescimento de plantas de forma contínua e não destrutiva, facilitando examinar detalhadamente as necessidades nutricionais das plantas. Em comparação com as operações manuais, é possível detetar mudanças subtis devido à desnutrição e muito mais precocemente, fornecendo uma base confiável e precisa para uma regulação atempada.

É possível ainda identificar áreas menos férteis em termos de crescimento e até mesmo áreas áridas como também reconhecer rapidamente e precisamente insetos e conta-los com o objetivo de controlar e prevenir eventuais pragas.

Indústria do mercado de revenda

Tanto lojas físicas como virtuais conseguem utilizar técnicas de *machine leanring* e *computer vison* para facilitar e melhorar a experiência dos clientes. Um exemplo bastante conhecido é a Amazon, que implementou *computer vision* para identificar produtos visualmente semelhantes com preços ligeiramente diferentes e sugere artigos baseados com a comparação entre gostos de diferentes clientes. Para além disso, aconselha os vendedores acerca do posicionamento de um determinado produto.

Relativamente a lojas físicas é possível, utilizando câmaras de vigilância previamente instaladas, contar pessoas anonimamente e analisar o tempo médio gasto destas em diferentes áreas, o tempo de espera em filas e avaliar metodicamente a qualidade do serviço. Esta análise tem como objetivo melhorar a distribuição e organização da loja, detetar e evitar longas filas e contar o número de clientes na loja, para impedir que o limite máximo seja atingido. Como não podia deixar de ser, com a pandemia covid-19 foi ainda necessário ter em conta o distanciamento entre pessoas. Para isso foram utilizados detetores de distância para medir o distanciamento entre clientes e funcionários.

Desporto

Por último, sendo a área desportiva bastante popular a nível mundial, é também uma área de interesse para os algoritmos de *computer vision*. Através de vídeos de jogos de equipas é possível calcular e determinar a pose e o movimento dos diversos jogadores, bem como analisar o desempenho, postura, identificar pontos fracos e melhorar os potenciais de cada jogador. Este estudo personalizado permite auxiliar os treinadores e os jogadores a analisarem os jogos de forma rápida e concisa, permitindo realizar esta análise entre jogos ou partes de um mesmo jogo.

Em contrapartida, os sistemas de auto-treinamento também ganharam destaque. Sendo possível corrigir pequenos defeitos de postura e desempenho, é possível criar um plano de treino personalizado e progredir até certo ponto sem a ajuda de outrem.

2.2.3 *Imagens e as suas características*

Ao contrário de nós, os computadores vêm uma imagem como uma matriz 2D, ou seja, as imagens digitais são constituídas por uma matriz de pixels. O Pixel é a menor unidade que compõe uma imagem e cada imagem é composta por um conjunto de pixels, onde o valor de cada um representa a intensidade da luz naquele determinado local da imagem.

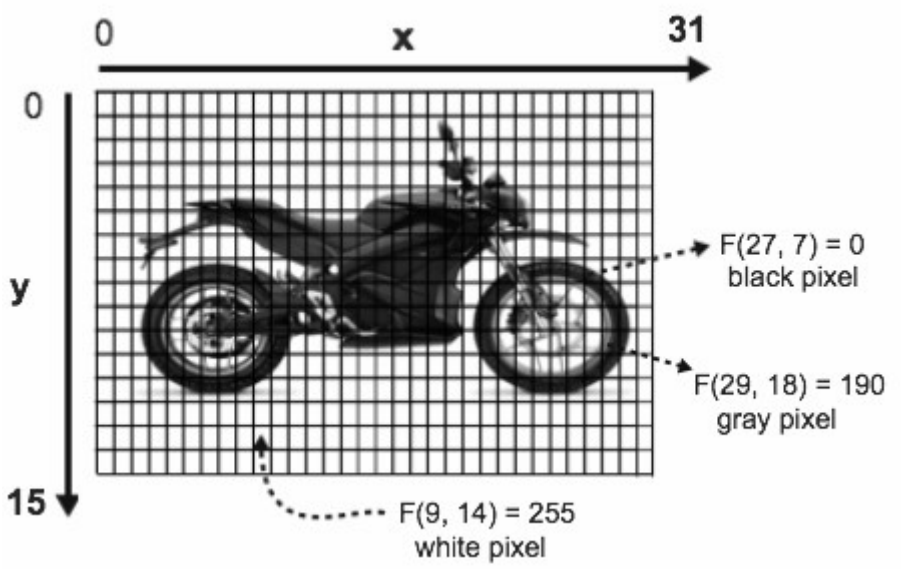


Figura 3: Matriz de uma imagem em tons de cinza

A imagem 4 representada acima tem um tamanho de 32 x 16, o que significa que as dimensões da imagem são 32 pixels de largura e 16 pixels de altura. No total, a imagem possui 32*16 = 512 pixels. Nesta imagem a preto e branco, cada pixel contém um valor que, como já foi referido, representa a intensidade da luz no pixel específico. Estes valores podem variar entre 0 e 255, onde o 0 representa a cor preta e o 255 a cor branca. Todos os valores entre estes dois números representa um tom de cinza, onde quanto mais baixo for o valor mais escuro é a sua tonalidade. Todas as imagens que são guardadas neste formato também são chamadas de *channel* ou canal.

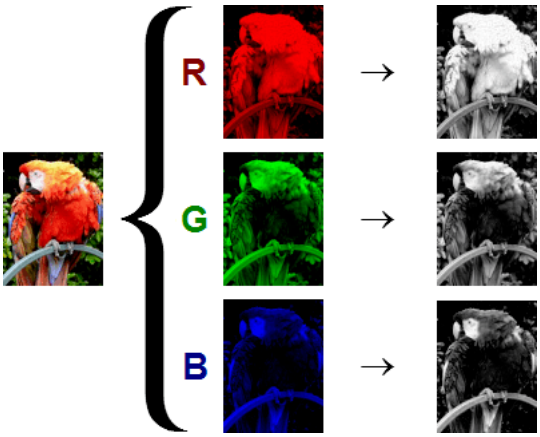


Figura 4: Junção das 3 matrizes RGB

Para imagens a cores, em vez de estas serem representadas por uma única matriz, são representadas por três matrizes: uma que representa a intensidade do vermelho, uma para o verde e outra para o azul, ou seja, uma por cada tonalidade RGB (red, green blue). Cada uma destas matrizes têm novamente valores que variam de 0 a 255, onde cada um desses números representa a intensidade dos pixels, isto é, os tons de vermelho, verde e azul, respetivamente. No final, estas matrizes são sobrepostas para que a imagem apresente todas as cores possíveis.

Assim, uma imagem digital a cores, quando vista por um computador, tem a forma N x M x 3, onde N é o número de pixels de altura, M seria o número de pixels de largura e 3 representa o número de canais, sendo cada um dos canais para cada uma das cores vermelho, verde e azul. Desta forma, de modo similar às imagens com tons de cinza, as imagens coloridas guardadas neste formato são chamadas de *Three-Channel*, isto é, três canais.

2.2.4 *OpenCV*

OpenCV significa "open source computer vision"e é uma biblioteca construída para fornecer uma suporte para aplicações de *computer vision* e *machine learning*, usado tanto a nível académico como comercial e contém um leque de funções de processamento de imagens de baixo nível e algoritmos de alto nível, como deteção de rosto, deteção e correspondência de características e rastreamento de objetos.

Foi uma iniciativa que começou no *Intel Research Lab* com o intuito de agilizar todo o processo de percepção da máquina e possibilitar o uso intensivo de CPU (Central Processing Unit). Foi concebido como uma forma de disponibilizar universalmente a infraestrutura de *computer vision* e como tal possui mais de 2500 algoritmos otimizados, com um conjunto abrangente dos mais recentes algoritmos.

Em 2010, foi adicionado, ao OpenCV, um novo módulo, que ainda se encontra em desenvolvimento. Este módulo permite a utilização do GPU (Graphics Processing Unit), sem a necessidade de programação direta a nível do GPU. Os utilizadores podem assim tentar combinar simultâneamente

o processamento do GPU com os cálculos do CPU, com o objetivo de minimizar as sobrecargas de transferência de dados e aumentar o desempenho geral.

OpenCV suporta as linguagens de C/C++, Python, MATLAB e Java e pode ser usada para construir aplicações de *computer vision* para qualquer sistema operativo tanto para computadores como para dispositivos móveis, isto é, suporta Windows, Linux, macOS, Android e iOS. ope (a)

2.3 DETECÇÃO E RECONHECIMENTO DE OBJETOS

A deteção de objetos é um dos problemas fundamentais de *computer vision*, servindo de base para muitas áreas de *computer vision* como *image captioning*, *instance segmentation*, *object tracking* entre outros A deteção de objetos, em conjunto com classificação e localização de objetos, serve para determinar a posição de um ou mais objetos numa determinada imagem digital e seguidamente catalogar cada objeto encontrado, permitindo responder a questões como "Que objetos estão onde?"

2.3.1 *Convolutional Neural Network (CNN)*

Convolutional neural networks (CNN) são um tipo de artificial neural networks(ANN), por isso, são constituídas por neurónios, pesos e *biases*. Fornecem uma abordagem escalável para a classificação de imagens e tarefas de reconhecimento de objetos, aproveitando os princípios da álgebra linear, especificamente a multiplicação de matrizes, para identificar padrões dentro de uma imagem. Todavia, requerem poder computacional bastante elevado, sendo muitas vezes necessário treinar modelos utilizando processamento gráfico (GPUs).

Uma ANN contém uma camada de input, uma ou mais camadas ocultas (*hidden layers*) e uma camada de output. Cada camada é constituída por neurónios, e cada neurónio liga-se a pelo menos um da camada seguinte, passando-lhe informação. Cada neurónio possui ainda um peso associado que varia consoante a informação recebida. Se este peso estiver acima de um determinado valor de *threshold*, então o neurónio será ativado e enviará informação para a camada seguinte. Caso contrário, nenhuma informação será enviada para a próxima camada. Education (2020b)

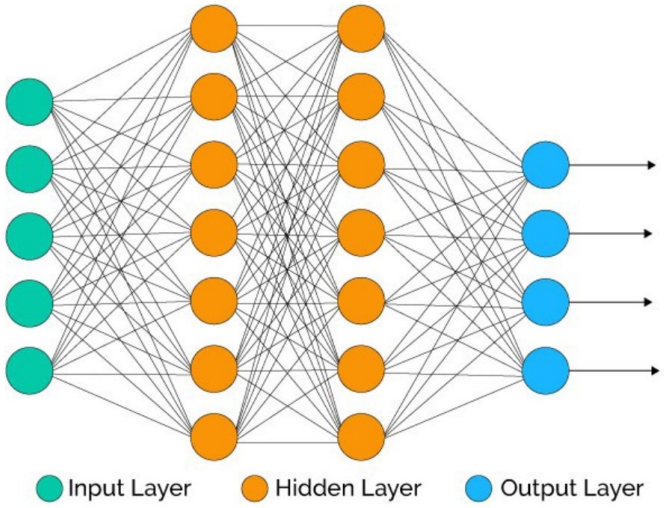


Figura 5: *Artificial Neural Network* Medisetti (2021)

A única diferença de uma CNN é que assume-se que esta rede recebe uma imagem como input, permitindo assim otimizar a mesma. Apresenta três camadas principais: *convolutional layer*, *pooling layer* e *fully-connected layer*.

A camada convolucional é a camada principal que dá o nome à rede e que requer maior poder computacional. Recebe como parâmetros os dados relativos a uma imagem e um filtro, que consiste normalmente numa matriz 3x3, e devolve um mapa de ativação (feature map, activation map ou convolved feature). Nesta camada, o filtro é então aplicado a uma região da imagem e é calculado o produto escalar entre os pixels de imagem recebida no input e o filtro. O resultado final é exibido no output, tal como suguere a imagem abaixo. Depois, o filtro desloca-se, normalmente uma unidade, repetindo o processo até que toda a imagem seja percorrida. O output final, também chamado de mapa de ativação, é então o conjunto dos produtos escalares da imagem e do filtro.

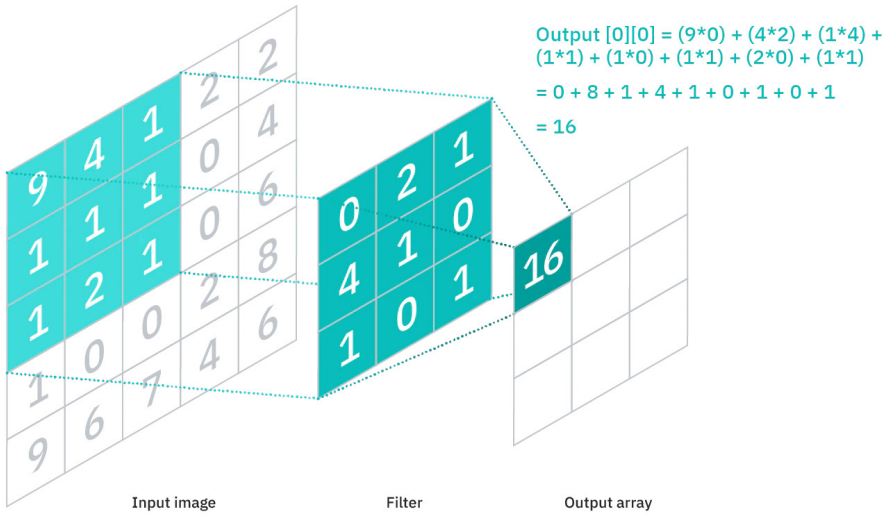


Figura 6: *Convolutional layer* Education (2020a)

Após cada operação de convolução, a CNN aplica uma transformação de Unidade Linear Retificada (*Rectified Linear Unit* - ReLU) ao mapa de ativação, para introduzir não linearidade no modelo. O número de camadas de convolução não é limitado, podendo seguir outra camada após a camada de convolução inicial.

Na camada de *Pooling* (agrupamento) é realizada uma redução ao número de parâmetros de input, sendo este normalmente o mapa de ativação devolvido pela camada de convolução. Nesta camada é aplicada uma função de agregação aos valores recebidos apresentado os resultados na forma de matriz. Há dois tipos principais de *pooling*:

- `Max pooling`: seleciona o pixel com o maior valor (mais usado)
- `Avarege pooling`: calcula o valor médio dos pixeis dentro de cada campo

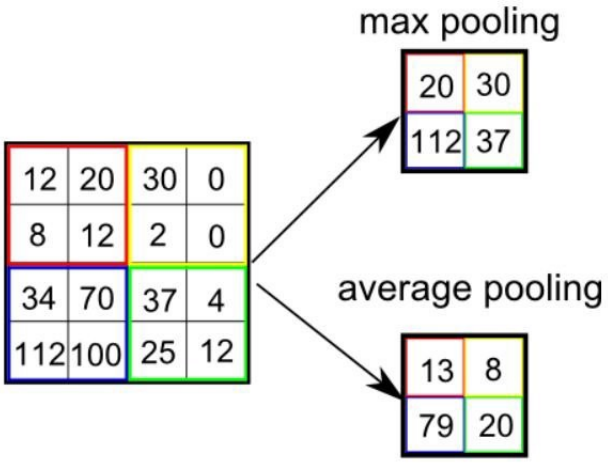


Figura 7: Tipos de *Pooling* Saha (2018)

Apesar de se perder muita informação nesta camada, os benefícios compensam. Permite reduzir a complexidade, melhorar a eficiência e limita o risco de *overfitting*.

Na *full-connected layer*, tal como o nome indica, os neurónios estão totalmente ligados, ou seja, cada neurónio na camada de output está diretamente ligado a todos os neurónios da camada anterior. É nesta camada que se realiza a tarefa de classificação com base nas características extraídas das camadas anteriores. O resultado final apresentado é uma probabilidade entre 0 e 1 relativo à certeza que o algoritmo tem sobre a classificação da imagem.

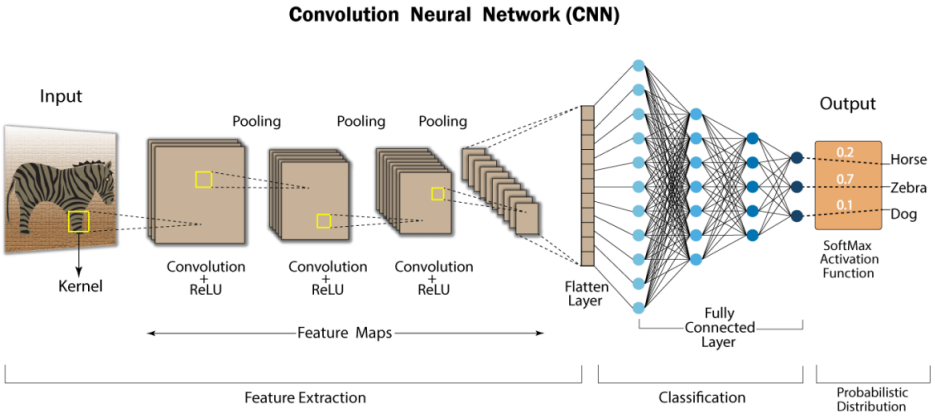


Figura 8: *Convolutional Neural Network* Swapna

2.3.2 *Algoritmos de Detecção de Objetos*

Region-based Convolutional Neural Networks (RCNN)

Nos modelos de redes neuronais convolucionais baseadas em região, a imagem inserida é primeiro dividida em regiões, normalmente em cerca de duas mil regiões, com o objetivo de extrair os recursos mais essenciais. O processo de seleção das características mais significativas pode ser computado com a ajuda de um algoritmo de busca seletiva que alcança os recursos regionais mais importantes, garantindo que as sub-segmentações escolhidas sejam as ideais. Assim que o algoritmo de busca seletiva é concluído, o próximo passo é extrair os recursos e realizar previsões, aplicando um rede neuronal convolucional a cada uma das regiões. A etapa final do R-CNN é fazer as previsões apropriadas para a imagem e catalogar cada região delimitadora.

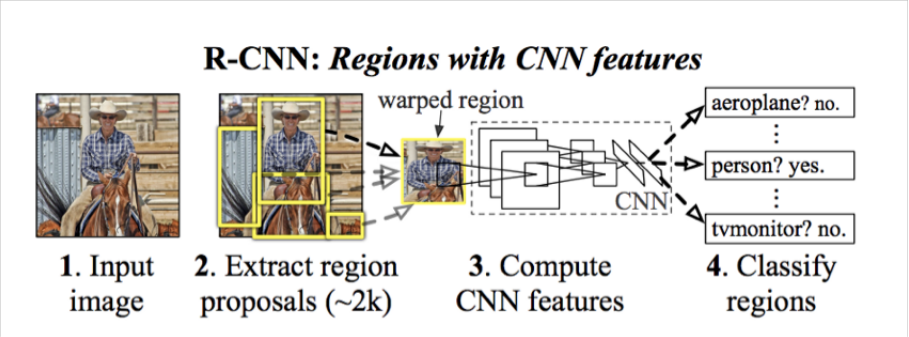


Figura 9: *RCNN: Regions with CNN features*

Apesar de produzir resultados corretos, o procedimento geral de extração de todas as regiões propostas é extremamente lento. Na globalidade, o modelo R-CNN apresenta não só uma baixa taxa de treino, como também um elevado tempo de previsão.

Em 2015, o *Fast R-CNN* foi desenvolvido com a intenção de reduzir significativamente este tempo de treino. Enquanto o R-CNN original calcula independentemente os recursos da rede neural em cada uma das regiões, o Fast R-CNN executa a rede neural uma única vez para toda a imagem. Existem outras versões do modelo R-CNN sendo a que apresenta melhor desempenho a nível de velocidade e *accuracy* o modelo Faster-RCNN, devolvendo o resultado ótimo em cerca de 0.2 segundos.

Single Shot Detector (SSD)

Single Shot Detector (SSD) é um método que deteta e classifica múltiplos objetos numa imagem usando apenas uma única rede neural profunda. Começa por diferencia o output através de caixas delimitadoras *default* de diferentes tamanhos e proporções para de seguida prever a categoria do objeto, combinando múltiplas características para permitir a catalogação de objetos de vários tamanhos.

Este método é relativamente fácil de treinar e integrar em sistemas que requerem um componente de deteção. Comparativamente a outros métodos de previsão de objetos com caixas delimitadores, o SSD apresenta melhor valores de *acurracy* mesmo para imagens menores.

You Only Look Once (YOLO)

Ao contrário da maior parte dos algoritmos de deteção de objetos, que utilizada regiões para localizar um objeto dentro de uma imagem, YOLO é um sistema, em tempo real, que, similarmente a SSD, apenas utiliza uma única rede neural para analisar a imagem toda de uma única vez.

YOLO utiliza uma única rede convolucional para simultaneamente prever e classificar múltiplas caixas delimitadoras. E uma vez que este processo é realizado num único passo é cerca de mil vezes mais rápido que R-CNN e cem vezes que o Fast R-CNN. Comparativamente a SSD, apresenta valores exatos ligeiramente piores, mas ultrapassa na velocidade. O modelo base processa imagens, em tempo real, a 45 fps (frames per second), enquanto que a versão versões mais recentes ultrapassam os 155fps com exatidão bastante elevada comparativamente a outros métodos.

A maior desvantagem do algoritmo YOLO é que este exibe algumas dificuldades a detetar objetos relativamente pequenos, devido às restrições espaciais do algoritmo.

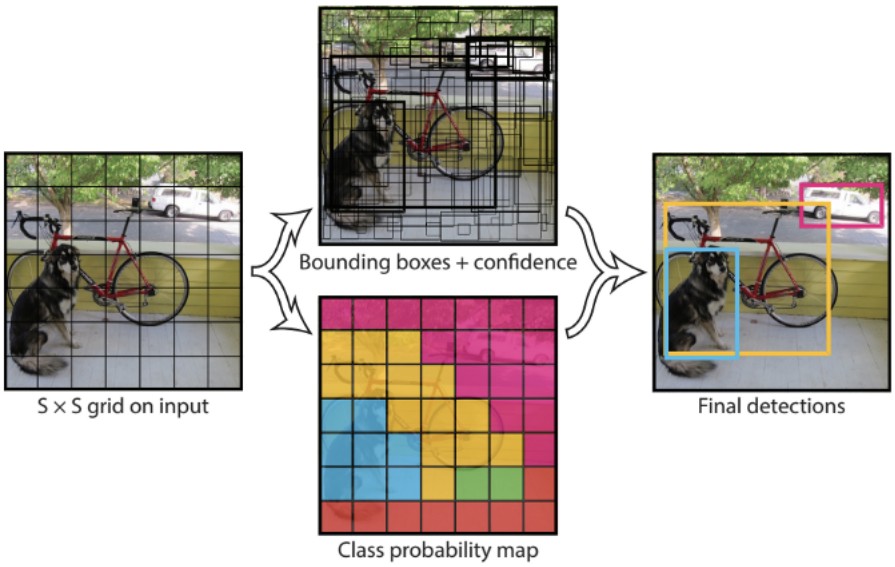


Figura 10: You Only Look Once (YOLO)

Comparação entre os vários algoritmos

Microsoft's Common Objects in Context (COCO) é um dataset usado para deteção, segmentação e catalogação de objetos em grande escala. É utilizado para avaliar o desempenho de métodos de *computer vision*, e é interpretado automaticamente em tempo real.

O *dataset* apresenta um formato específico em JSON que dita como é que os rótulos e os metadados devem ser guardados numa imagem. Contém mais de 80 objetos de diferentes categorias, com as respetivas caixas delimitadoras. Ainda possui mais de duzentas mil imagens e 250 mil imagens de pessoas rotuladas e com pontos chaves para a deteção de imagens.

A métrica de avaliação primária para modelos de deteção de objetos é o valor médio das precisões médias de cada classe (mean average precison - MAP). Esta avaliação tem em conta caixas delimitadoras pequenas, médias e grandes, bem como limites variados para o número de deteções por imagem. É possível ainda analisar os vários modelos tendo em conta os fps e o tempo que demoraram a detetar os vários objetos.

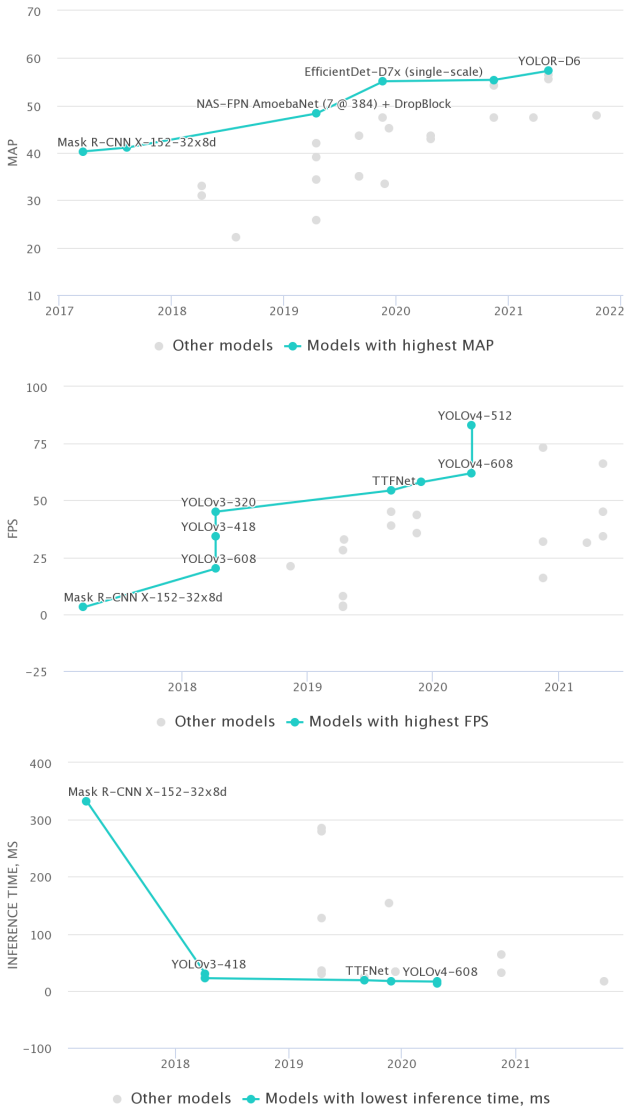


Figura 11: Real-Time Object Detection on COCO

Segundo a plataforma COCO, dos algoritmos apresentados anteriormente, YOLO é o que apresenta melhor performance. Existem várias versões do mesmo algoritmo, o melhor modelo, comparativamente a MAP, é `YOLOR-D6` com 57.3 MAP para 34 fps. O que apresenta maiores valores de fps, bem como menor tempo de inferência é o modelo `YOLOv4-512` com 43 MAP, 83 fps e 12ms. Contudo, temos de ter em conta, que este modelo apresenta valor de treino extra comparativamente aos outros algoritmos e daí obter valores melhorados.

2.4 ANÁLISE DE MOVIMENTO E RASTREAMENTO DE OBJETOS

O rastreamento de um objeto é o processo de localizar e descobrir a posição exata de um objeto enquanto este está em movimento, isto é, em frames consecutivos de um vídeo.

O primeiro passo no rastreamento de um objeto é deteta-lo, criando uma caixa delimitadora à volta do objeto e posteriormente cataloga-lo. De seguida, é atribuido um ID único a cada objeto encontrado. Para cada frame de um vídeo é necessário detetar novamente os objetos presentes na imagem e de seguida compara-los com as informações previamente guardadas para saber se se tratam de um objeto novo ou um já antes detetado.

2.4.1 *Deteção de Objetos vs Rastreamento de Objetos*

Para a análise de um vídeo em tempo real, normalmente, os algoritmos de rastreamento são mais rápidos que simplesmente a utilização de algoritmos de deteção. Isto deve-se ao facto de que, quando se realiza um rastreamento de um objeto, este já é conhecido nos frames anteriores. Desta forma a aparência já é conhecida, bem como a sua localização, direção e velocidade do movimento que apresenta. Esta informação facilita a procura de um objeto num novo frame de um vídeo, enquanto que um algoritmo de deteção começa do zero a cada novo frame.

Apesar dos algoritmos de rastreamento beneficiarem de informações extras, também podem perder o controlo de um objeto quando este se desloca rapidamente ou quando o objeto fica atrás de um obstáculo por longos períodos de tempo. Nestes casos, o algoritmo de rastreamento acumula sucessivamente mais erros e a caixa delimitadora do objeto afasta-se cada vez mais do mesmo. Nestes casos, é necessário que um algoritmo de deteção seja executado para corrigir os erros para reencontrar o objeto perdido.

2.4.2 *Desafios no rastreamento de objetos*

O rastreamento de objetos é um método bastante desejado e é esperado apresentar resultados precisos num curto espaço de tempo. Todavia, existem alguns casos em que o rastreamento de objetos encontra algumas entraves na resolução do seu trabalho.

- **Oclusão do Objeto:** Em muitos casos, o objeto a seguir desaparece por breves períodos de tempo, ou porque fica ofuscado por outro objeto ou simplesmente porque desaparece do alcance da câmara. Nestes casos é difícil detetar e atualizar o objeto nas imagens futuras.
- **Velocidade:** Quando o movimento de um objeto é demasiado rápido, a câmara pode não detetar o objeto com nitidez suficiente para que o algoritmo consiga detetar o objeto e persegui-lo.
- **Forma:** Caso o objeto a ser detetado não possua uma forma rígida e constante, isto é, se apresentar mudanças na sua forma, haverá uma falha na deteção e consecutivamente no rastreamento do objeto.

- **Falsos Positivos:** quando há vários objetos semelhantes é difícil distinguir com elevada certeza os vários objetos relativamente ao seu ID. Ou seja, o algoritmo pode perder o objeto atual e começar a rastrear um novo objeto semelhante.

Estes desafios podem interromper repentinamente o programa ou dar uma previsão completamente errada da localização do objeto a perseguir.

2.4.3 *Técnicas de deteção de objetos para o Rastreamento*

O rastreamento de objetos usa técnicas de deteção de objetos que são aplicadas a frames consecutivos de um vídeo. Visto que um vídeo é constituído por com conjunto de frames consecutivos, as caixas delimitadoras de cada objeto necessitaram de ser ajustadas em cada um desses frames.

Uma informação a ter em conta, é que para o rastreamento mostrar resultados assume-se que o objeto a ser rastreado está presente em todos os frames do vídeo. Isto é necessário para que se possa comparar a posição do objeto nos vários frames consecutivos.

Diferenciação de Frames

Frames são imagens de um vídeo capturadas a intervalos regulares e cada frame é constituído por um conjunto de pixeis. A mínima variação num píxel indica a variação num frame.

A diferenciação de frames é uma técnica usada para medir a diferença entre dois frames de um vídeo, tendo em conta a posição de um ou mais objetos. Para determinar o movimento de um objeto , primeiro é calculado a diferença entre pixels em imagens consecutivas.

Visto que nem todas as alterações nos pixels são devido a movimentação de objetos, é implementado um limite. Ou seja, para diferenciar ruído ou até mudanças na luminosidade é imposto um valor mínimo de mudança, ignorado todos os valores inferiores a esse.

A diferenciação de frames é um método que funciona relativamente bem, sobretudo com imagens de fundo estático. Apresenta valores de *accuracy* elevados e necessita de pouco poder computacional.

Subtração de Fundo

A subtração de plano de fundo é uma técnica de pré-processamento importante. O seu principal objetivo é separar o plano de fundo do primeiro plano num vídeo. Uma vez que a maior parte dos objetos se encontram no primeiro plano, a localização destes torna-se muito mais rápida, principalmente em câmaras estacionárias.

Todavia, no mundo real este método torna-se um pouco mais complicado. Com a iluminação e mudança da mesma, surgem sombras e movimento das mesmas. Tal como as sombras também podem existir mais objetos que estejam incluídos no plano de fundo e sejam removidos, tornando a sua localização muito mais difícil e muitas vezes incorreta. O método de substração de fundo é mediano relativamente a *accuracy* e tempo computacional.

Optical Flow

Optical flow corresponde à movimentação de objetos entre imagens consecutivas, quer devido à movimentação da câmara ou do objeto na imagem em si. Este método cria um vetor para cada pixel que se movimenta de um frame para outro.

Para esta técnica funcionar assume-se que a intensidade dos pixeis entre frames apenas se altera se houver movimento e que existe movimentação semelhante entre pixeis adjacentes. Para usar *optical flow* é preciso poder computacional relativamente alto e os resultados obtidos apenas conseguem *accuracy* moderada.

2.4.4 *Métodos de Rastreamento de Objetos*

O rastreamento de objetos começou com alguns algoritmos clássicos de *machine learning*, como KNN e SVM. Estas abordagens são relativamente boas para a previsão de objetos mas necessitam muitas vezes de informações extraídas manualmente, por exemplo, é necessário muitas vezes as imagens estarem manualmente catalogadas para estes algoritmos serem mais eficientes.

Desta maneira, os algoritmos destinados ao rastreamento de objetos foram introduzidos com o objetivo de melhorar a *accuracy* e a eficiência dos algoritmos já existentes. Assim, estes algoritmos realizam já a extração das características da imagem que necessitam.

GOTURN

Generic Object Tracking Using Regression Networks ou GOTURN utiliza uma abordagem baseada em regressão para rastrear objetos. Basicamente, o modelo regride diretamente para localizar o objeto com apenas uma única passagem *feed-forward* pela rede.

Por outras palavras, a rede recebe dois inputs. Recebe o objeto encontrado no frame anterior e consecutivamente a região de procura no frame atual. Desta forma, a rede compara as duas imagens e encontra facilmente o objeto na imagem atual.

É um modelo que treina de forma offline, ou seja, apenas é utilizado depois de completar o treino. Este tipo de algoritmos apresentam melhores resultados do que os online, uma vez que podem ser treinados para lidar com rotações, mudanças do ponto de vista, mudanças de iluminação entre outros desafios complexos. Assim, conseguem aprende uma relação genérica entre o movimento e a aparência de objetos e podem ser usados para rastrear objetos para os quais não foram treinados.

ROLO - Recurrent YOLO

ROLO é uma combinação de redes neuronais recorrentes e do método de deteção de objetos YOLO. ROLO combina assim dois tipos de redes neuronais, uma é CNN, usada para extrair informações espaciais, e outra é uma rede LSTM, que calcula a trajetória do objeto.

A cada iteração, as informações espaciais e as caixas delimitadores dos objetos, extraídas pela CNN, são enviadas ao LSTM.

DeepSORT

DeepSORT é uma junção do *Simple Online Real-time Tracker* (SORT) com *deep learning network*. Apesar de ser ser um algoritmo online, o que o torna lento e pouco adequado para devolver resultados em tempo real, é bastante eficaz contra oclusão de objetos.

O algoritmo SORT é composto por três fases principais. Primeiramente, na fase inicial é detetado o objeto. De seguida, é feita uma previsão da localização deste no estágio seguinte. Por último, é feita uma otimização calculando assim a posição correta.

O DeepSORT agrupa SORT com técnicas de *deep learning*, permitindo estimar a localização de um objeto e rastreá-lo com maior precisão.

2.4.5 *Rastreamento de Objetos no OpenCV*

OpenCV oferece um conjunto de algoritmos pré-construídos desenvolvidos explicitamente para fins de rastreamento de objetos. De seguida apresentam-se alguns dos rastreadores disponíveis no OpenCV Rosebrock (2018); Deshpande et al. (2020):

- `BOOSTING`: é baseado no algoritmo de *machine learning* AdaBoost e existe há mais de 10 anos. É treinado em tempo de execução aprendendo os exemplos positivos e negativos do objeto a ser rastreado. É lento e não apresenta resultados muito confiáveis, mesmo para alguns dados relativamente mais superficiais.

- `Multiple Instance Learning (MIL)`: semelhante ao BOOSTING, todavia, para além de utilizar a localização do objeto para o rastrear, também utiliza parte da vizinhança deste. Apresenta melhor precisão que o BOOSTING.

- `Kernelized Correlation Filters (KCF)`: baseia-se no conceito de que vários exemplos positivos têm grandes regiões sobrepostas.

- `Discriminative Correlation Filter with Channel and Spatial Reliability (DCF-CSR) ou Channel and Spatial Reliability Tracking (CSRT)`: usa um mapa de confiança espacial para auxiliar na localização de objetos. Proporciona alta precisão para fps baixos (25 fps)

- `MedianFlow`: calcula os deslocamentos do objeto em tempo real e mede o erro e a diferença entre as duas posições. O principal objetivo é minimizar o erro e detetar falhas no rastreamento e para poder selecionar as trajetórias mais confiáveis.

- `Tracking, Learning, and Detection (TLD)`: este algoritmo segue o objeto frame a frame e localiza em todos os frames a posição do objeto, tendo por base o que aprendeu no frame anterior, corringo simultaneamente se necessário.

- `Minimum Output Sum of Squared Error (MOOSE)`: Suporta escalas, deformações não rígidas, mudanças de iluminação e oclusões e consegue ainda reencontrar o objeto assim que este reaparece em cena. Relativamente a desempenho, é um pouco pior que GOTURN

- `GOTURN`: é baseado numa abordagem *deep learning*, visto que utiliza *convolutional neural networks*. Apresenta *accuracy* elevada e suporta deformações, mudanças de iluminação e de ponto de visto, contudo não lida bem com oclusões.

2.4.6 *Rastreamento de Objetos com OpenCV*

Segundo estudos como Dardagan et al. (2021) e Janku et al. (2016), que avaliam diferentes algoritmos de rastreamento de objetos disponíveis na biblioteca OpenCV, foi possível fazer uma comparação entre os mesmos tendo em conta o nível médio de exatidão e precisão, bem como o sucesso a analisar características mais concretas como iluminação, oclusão, deformação de objetos entre outros.

Estes estudos concluíram que MOSSE e Medianflow são os melhores algoritmos para o rastreamento de objetos em tempo real, visto que conseguem rastrear mais de 100 objetos simultaneamente. Relativamente à exatidão e precisão, CSRT é o melhor algoritmo seguido pelo MIL e Boosting. KCF, Medianflow, MOOSE e TLD apresentam os piores resultados. Assim, Boosting ou MIL são os algoritmos que balanceie ambos os aspetos de precisão e rapidez.

É possível ainda melhorar estes algoritmos através da combinação de algoritmos simultaneamente, Singh et al. avalia a performance dos algoritmos CSRT e KCF simultaneamente em detrito dos mesmos algoritmos individualmente, obtendo melhores resultados, tanto a nível de accuracy como de fps na junção dos algoritmos.

Deve-se ter em especial atenção o facto de que o OpenCV continua em progresso e como tal já existem versões mais recentes, com novos algoritmos, desde que estes estudos foram realizados. A versão mais recente de OpenCV, atualmente, é 4.5.5 lançada a 30 de Dezembro de 2021. ope (b) Esta nova versão tem o algoritmo GOTURN, que possui potencial para ser ainda melhor que os outros algoritmos já existentes e analisados até ao momento, contudo ainda não foram realizados estudos sobre o mesmo no ambiente de OpenCV. Para além disso, OpenCV permite a utilização paralela do GPU com o CPU, possibilitando otimizar qualquer algoritmo.

3

COCLUSÕES E TRABALHO FUTURO

Neste relatório foi possível identificar os projetos e conceitos essenciais ao desenvolvimento da plataforma a criar. Foram analisados os conceitos base de inteligência artificial, *machine learning* e *computer vision*, bem como foram aprofundado os principais algoritmos de *object detection* e *object tracking*. É feita uma análise comparativa entre os diferentes algoritmos, para ser possível retirar as vantagens e desvantagens de cada um. Foram ainda identificadas as principais dificuldades relativas à análise de imagens e vídeos, sendo necessário uma análise cuidada para ultrapassar estes problemas.

Para trabalho futuro, primeiramente será investigado e desenvolvido um agente capaz de analisar a informação visual processada no ecrã do jogador. Este mecanismo pode ser feito por duas abordagens, ou por *screenshots* constantes ao ecrã ou por *livestream*, ou seja, ou por uma série de imagens separadas consecutivas, sendo analisada imagem a imagem, ou gravação constante do ecrã do jogador produzindo um vídeo. Serão analisadas as duas abordagens para concluir qual a que requer menos poder computacional e que devolve melhores resultados em tempo-real.

De seguida, será aplicado e treinado um algoritmo de deteção e reconhecimento de objetos, utilizando a biblioteca OpenCV para Python. Uma vez que o objetivo da plataforma é devolver o resultado em tempo-real, e permitir simultaneamente o utilizador realizar outras atividades, então o algoritmo tem de ser o mais rápido possível não utilizando elevado poder computacional. Para isso será necessário um algoritmo *offline*, isto é, que seja treinado previamente, para ser o mais leve possível, a termos de poder computacional, e de preferência com resultados precisos. Assim, será necessário experimentar os vários algoritmos explorados para este caso em concreto, e descobrir e treinar o algoritmo ideal.

Por último, será finalmente construída a plataforma que rastreia um dado objeto no ecrã do jogador, utilizando para isso o algoritmo treinado.

Paralelamente à realização destas tarefas, será escrita a dissertação final explicando detalhadamente todo o processo desenvolvido, os resultados encontrados, bem como as dificuldades sentidas. Ao longo de todo o trabalho, todas as soluções propostas serão refletidas e analisadas detalhadamente e em caso de alguma imprecisão serão investigadas outras alternativas possíveis, de modo a alcançar os objetivos pretendidos.

De seguida é apresentado o planeamento da dissertação proposto, indicando as tarefas concluídas até ao momento:

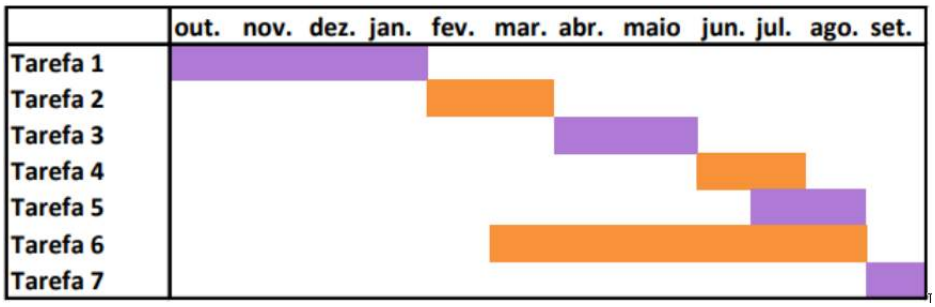


Figura 12: Calendarização

- `Tarefa 1`: Estado de Arte e escrita da pré-dissertação (4 meses) ✓
  - `Tarefa 1.1`: Contexto geral da área de *Machine Learning* e *Computer Vision* ✓
  - `Tarefa 1.2`: Procura de algoritmos da área de *Machine Learning* e *Computer Vision* ✓
- `Tarefa 2`: Criação de uma aplicação/agente capaz de analisar a informação visual processada no ecrã do jogador (2 meses)
- `Tarefa 3`: Criação e treino de um algoritmo de reconhecimento de diferentes objetos (2 meses)
- `Tarefa 4`: Criação de um sistema de reconhecimento e rastreamento de objetos no ecrã do jogador (2 meses)
- `Tarefa 5`: Construção de uma framework para análise de comportamentos de objetos interativos em vídeo jogos (2 meses)
- `Tarefa 6`: Escrita da dissertação (6 meses)
- `Tarefa 7`: Preparação para a defesa final (1 mês)

O presente relatório de pré-dissertação apresenta os resultados da realização da `tarefa 1`, assim como um panorama global de uma análise complementar que auxiliará o término das tarefas restantes. Pode-se assim concluir que está a ser cumprido o plano de trabalho inicialmente proposto.

BIBLIOGRAFIA

About, a. URL https://opencv.org/about/.

Releases, b. URL https://opencv.org/releases/.

Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3:19–48, 2010.

Jason Brownlee. A gentle introduction to computer vision, Mar 2019. URL https://machinelearningmastery.com/what-is-computer-vision/.

Nađa Dardagan, Adnan Brđanin, Džemil Džigal, and Amila Akagic. Multiple object trackers in opencv: A benchmark. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 1–6. IEEE, 2021.

Rostyslav Demush. A brief history of computer vision (and convolutional neural networks), Feb 2019. URL https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8a

H Deshpande, A Singh, and H Herunde. Comparative analysis on yolo object detection with opencv. *International Journal of Research in Industrial Engineering*, 9(1):46–64, 2020.

IBM Cloud Education. What are convolutional neural networks?, Oct 2020a. URL https://www.ibm.com/cloud/learn/convolutional-neural-networks.

IBM Cloud Education. Neural networks, Aug 2020b. URL https://www.ibm.com/cloud/learn/neural-networks.

Sunila Gollapudi. *Learn computer vision using OpenCV: with deep learning CNNs and RNNs*. Apress, 2019.

Peter Janku, Karel Koplik, Tomas Dulik, and Istvan Szabo. Comparison of tracking algorithms implemented in opencv. In *MATEC Web of Conferences*, volume 76, page 04031. EDP Sciences, 2016.

Yashwanth Medisetti. Neural networks in everyday life, Feb 2021. URL https://medium.com/analytics-vidhya/neural-networks-in-everyday-life-ca2b7cb37052.

Vidushi Meel. 83 most popular computer vision applications in 2022, Mar 2021. URL https://viso.ai/applications/computer-vision-applications/.

Diogo Ferreira Nunes. Uber obriga motoristas, passageiros e estafetas a usarem máscara. *Diário de Notícias*, May 2020. URL https://www.dn.pt/dinheiro/uber-obriga-motoristas-passageiros-e-estafetas-a-usarem-mascara-12193002.html.

Adrian Rosebrock. Opencv object tracking, Jul 2018. URL https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/.

Sumit Saha. A comprehensive guide to convolutional neural networks, Dec 2018. URL https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1

Suryansh Singh, Akshat Mittal, and Manas Gupta. Comparing various tracking algorithms in opencv.

KE Swapna. Convolutional neural network: Deep learning. URL https://developersbreach.com/convolution-neural-network-deep-learning/.