Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

Sistema multimodal para análise automática do desempenho desportivo de atletas

Miguel Jesus

Orientador(es) | José Saias

Orlando de Jesus Fernandes

Évora 2022

Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

Sistema multimodal para análise automática do desempenho desportivo de atletas

Miguel Jesus

Orientador(es) | José Saias

Orlando de Jesus Fernandes

Évora 2022

A dissertação foi objeto de apreciação e discussão pública pelo seguinte júri nomeado pelo Diretor da Escola de Ciências e Tecnologia:

Presidente	|	Lígia Maria Ferreira (Universidade de Évora)
Vogais	|	José Saias (Universidade de Évora) (Orientador) Vitor Beires Nogueira (Universidade de Évora) (Arguente)

Évora 2022

À minha avó Esmeraldina

Agradecimentos

A toda a família Icon Jiu Jitsu Évora, e em especial ao meu professor Bruno Ferreira, um grande obrigado por me tornar um ser humano mais capaz e resiliente.

À equipa da qual faço parte no projeto Guardian da Talkdesk, por me mostrarem todos os dias que tenho algo a aprender e pelo tempo que me foi cedido para dedicar à dissertação.

Aos meus pais, Laura Gaspar e Carlos Jesus, por me terem tornado eternamente insatisfeito com a perfeição, ou falta dela.

Aos meus avós, Rosa, António, Esmeraldina e Severino, por me terem ensinado o valor do trabalho.

A grandes pessoas como Débora Marcão, Jaime Ferreira, Taís Azevedo e Rodrigo Roboredo por estarem sempre presentes e me relembrarem de quem sou, até quando eu próprio me esqueço.

Aos meus orientadores José Saias e Orlando Fernandes um enorme agradecimento pela disponibilidade e paciência dispensada ao garantirem que me mantinha sempre alinhado com os meus objetivos.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

GPS Global Positioning System

RGB Red Green Blue

FIFA Fédération Internationale de Football Association

NBA National Basketball Association

JS JavaScript

HTML Hyper Text Markup language

CSS Cascading Style Sheets

XML Extensible Markup Language

JSX JavaScript Syntax Extension

Sumário

Os sistemas multimodais são cada vez mais utilizados no contexto desportivo e este trabalho tem por objetivo demonstrar como a aprendizagem automática embebida na visão por computador aplicada ao contexto do boxe pode ajudar um praticante a melhorar a sua técnica e tempo de reação.

Foi implementada uma aplicação utilizando ReactJS, que através da câmara consegue extrair dados sobre a postura do atleta num referencial de 2 dimensões por meio do modelo Posenet e que são posteriormente fornecidos a uma Rede Neuronal para os classificar como "esquerda", "direita", "esquiva" ou "guarda".

Foram recolhidas um total de 31425 amostras, incluindo movimentos de todas as classes alvo de classificação, sendo que o conjunto de dados está dividido em: 46% treino, 27% validação e 27% teste. Após o treino concluiu-se que a Rede Neuronal com afinação de hiper-parâmetros para 20 épocas foi o modelo que obteve melhor desempenho com uma Medida-F Média e Exatidão de 95%. Posteriormente o mesmo foi testado com o conjunto de teste e obteve uma Medida-F Média e Exatidão de 97.7% .

A aplicação foi concebida no formato de um jogo em que o jogador terá de executar corretamente os movimentos que lhe serão pedidos de forma aleatória ao longo de 30 segundos. Tudo isto foi desenvolvido com o apoio de ferramentas de código aberto e sem necessidade de nenhum dispositivo adicional ao de um computador portátil ou tablet comum.

Palavras chave: Interação Multimodal, Visão por Computador, Inteligência Artificial, Classificação, Desempenho

Abstract

Multimodal system for automatic analysis of athletes sports peformance

Multimodal systems are becoming mainstream in the context of sport and this work has the objective to show how machine learning embedded with computer vision can be applied to the context of boxing can help a praticioner by improving his technique and reaction time.

A web application was implemented using ReactJS and uses the web cam to extract data about the posture of the athlete in a 2 dimensions referential, thanks to Posenet, that is posteriorly supplied to a Neural Network that uses it to classify each pose as a "right", "left", "dodge" or "guard".

A total of 31425 samples were collected, including all classes target of classification, divided in: 46% training, 27% validation and 27% test. After training, the Neural Network with the hyperparameter tuning for 20 epochs was the model with the best performance overall with an Average F-score and Accuracy of 95%. The same model was tested with the test set and obtained an Average F-score and Accuracy of 97.7%.

The application was conceived in the format of a game in which the player will have to execute correctly the movements that will be demanded by the system in a random manner for 30 seconds. All this was developed with the support of open source tools and no need of any extra devices but a common laptop or a tablet.

Keywords: Multimodal Interaction, Computer Vision, Artificial Intelligence, Classification, Performance

1

Introdução

1.1 Motivação

Os avanços tecnológicos permitiram a atletas e treinadores passar a acompanhar padrões de movimento funcionais , carga de trabalho (interna e externa), extração de indicadores biomecânicos e vitais com o propósito de melhorar o desempenho e prevenir lesões [SLV+19] [inj19]. Nas duas últimas décadas os acessórios de vestuário eletrónico passaram de apenas dispositivos a sistemas de sensores [CSBR19] que usam os dados recolhidos para gerar um feedback com base na sua análise. GPS, acelerómetros,bandas cardíacas e sistemas de análise de vídeo são algumas das tecnologias mais utilizadas atualmente no alto rendimento desportivo eficácia demonstrada pelo seu uso em alguns desportos como atletismo, futebol, râguebi ou artes marciais. No entanto existe uma tentativa constante de fazer com que sejam cada vez mais precisas, menos invasivas no que respeita aos praticantes, com o intuito de não prejudicar o seu desempenho, portáteis, acessíveis e automatizáveis.

1.2 Objetivos

Esta dissertação tem como objetivo aplicar a visão por computador juntamente com técnicas de aprendizagem automática, para identificar padrões de movimento inerentes ao boxe sob a forma de um jogo onde é dada uma sequência, de golpes e poses, aleatória que o atleta deverá executar com rigor no tempo máximo de 30 segundos, sendo que a pontuação final é igual ao número de movimentos executados corretamente.

A aplicação faz apenas uso de tecnologias de código aberto e não requer nenhum hardware adicional ao de um computador portátil comum, visando assim priorizar a acessibilidade e portabilidade.

1.3 Principais Contribuições

Após a implementação da aplicação web, sob a forma de um jogo [ZCS+21], qualquer praticante da modalidade de boxe, poderá melhorar tanto o aspeto técnico [BZL+14], como a sua velocidade de reação, algo fundamental nas artes marciais.

O sistema implementado não carece de qualquer ação humana no que diz respeito à identificação dos movimentos específicos do boxe, sendo a identificação dos mesmos feita de maneira automática em tempo real através do uso de visão por computador para deteção e seguimento das ações do atleta.

A principal contribuição desta dissertação é demonstrar, ainda que de maneira modesta, o poder da aprendizagem máquina aliada à visão por computador no que diz respeito a deteção de padrões de movimento específicos que podem levar a uma melhoria de desempenho do utilizador num determinado desporto.

Neste caso o desporto escolhido foi o boxe pela complexidade motora que envolve, bem como pela necessidade de técnicas não invasivas de avaliação de performance que o mesmo requer por ser um desporto de contacto [PKV+16].

1.4 Estrutura da Dissertação

A dissertação encontra-se organizada em 5 capítulos. No capítulo número 2 é apresentado o Estado da Arte com especial atenção na Ciência de Dados e Visão por Computador englobando alguns conceitos importantes, bem como trabalhos relacionados nas áreas. O capítulo 3 engloba uma descrição detalhada das tecnologias utilizadas na implementação do projeto. No capítulo 4 encontra-se uma breve descrição das métricas de desempenho mais importantes no âmbito do trabalho, bem como a metodologia seguida, os resultados da sua avaliação e a discussão dos mesmos. Por fim, no capítulo 5 reside o balanço e o trabalho futuro a desenvolver.

2

Estado da Arte

2.1 Ciência de Dados

Ciência de Dados é o nome que se dá ao conjunto de métodos científicos , processos e sistemas para extrair conhecimento a partir de grandes quantidades de dados. Ao contrário da programação tradicional em que o computador recebe dados e regras aos quais os mesmo tem de obedecer e nos dá o resultado final, no âmbito de Ciência de Dados introduzimos os dados e o resultado esperado sendo que o objetivo é aferir as regras. Para isto utilizam-se técnicas e algoritmos de Aprendizagem Automática que permitem ao computador aprender sem que tivesse sido explicitamente programado para tal, através de reconhecimento de padrões. Atualmente representa um enorme valor acrescentado em áreas como a medicina, desporto, estatística ou economia.

2.1.1 Aprendizagem Automática

Aprendizagem Automática pode ser definida como a capacidade de ensinar um computador ou algoritmo a melhorar a sua abordagem progressivamente perante uma tarefa específica [Ten19]. Esta área de estudo está dividida em 3 categorias: Aprendizagem Supervisionada, Não Supervisionada e por Reforço.

Aprendizagem Supervisionada

Na Aprendizagem Supervisionada [Ten19] os modelos são treinado em dados previamente rotulados para encontrar regras através das quais se poderão rotular novos dados, ou seja, mapear um "caminho" entre as entradas e saídas. A classificação de um email como SPAM é um bom exemplo, pois é fornecida ao sistema uma grande quantidade de mensagens de correio-eletrónico previamente rotulado de acordo com o seu teor, para que possa filtrar e distinguir conteúdo fidedigno e de interesse para o utilizador de conteúdo malicioso ou publicidade em massa. O reconhecimento facial também é possível utilizando tipo de técnica. A partir de fotos publicadas em redes sociais, que tenham identificação dos intervenientes, os algoritmos recolhem exemplos de dados etiquetados e que podem utilizar para sugerir a identificação de um utilizador específico através da análise de uma foto onde o mesmo não tenha sido marcado [Ten19]. Alguns dos algoritmos mais populares neste ramo da Aprendizagem são as Redes Neuronais, Árvores de Decisão e os K-Vizinhos mais próximos.

Aprendizagem Não Supervisionada

No âmbito da Aprendizagem Não Supervisionada [Ten19] os algoritmos encontram padrões sem treino anterior entre dados não rotulados, agrupando os mais semelhantes entre si. Ou seja, estes algoritmos são capazes de ensinar-se automaticamente de modo a criar partições de dados semelhantes entre si, sem que seja preciso fornecer-lhes exemplos específicos para a tarefa. Com a quantidade de informação circulante que não está rotulada, estes algoritmos representam uma potencial mais-valia para a aquisição de conhecimento [Ten19]. Os sistemas de recomendação de produtos fazem uso deste tipo de Aprendizagem, para com base num histórico de pesquisas, sugerirem ao utilizador os anúncios de produtos em que o mesmo possa estar potencialmente interessado. Outra utilização de grande importância destes algoritmos é na deteção de fraudes bancárias através da análise do histórico de atividade dos clientes, permitindo identificar anomalias (comportamentos suspeitos) em tempo útil de modo a mitigar possíveis consequências negativas. Os algoritmos de Agregação, como o K-médias e a Agregação Hierárquica, são alguns dos mais utilizados no que a esta temática diz respeito.

Aprendizagem por Reforço

Inicialmente este tipo de algoritmos operam em ambientes desconhecidos e estão em constante adaptação estimulados por recompensas que podem tanto ser positivas como negativas, portanto pode dizer-se que ao contrário dos restantes tipos de aprendizagem, este caso específico é orientado ao ambiente. Em tarefas de Aprendizagem por Reforço o modelo faz uso de funções de recompensa para que possa iterativamente adotar um comportamento mais adequado aos novos dados que obtém, de maneira a maximizar a sua performance no longo prazo [HJK+09]. Este tipo de Aprendizagem é bastante utilizado nas áreas da neurociência e psicologia, para simular o modo como funciona o sistema de recompensa humano, ou até para tentar ter um melhor entendimento da tomada de decisão dos animais [AN18][HJK+09].

2.1.2 Transferência de Conhecimento

A Transferência de Conhecimento é um processo utilizado em aprendizagem automática que visa utilizar conhecimento previamente adquirido por um modelo para o aplicar numa tarefa para a qual não foi originalmente concebido. Esta técnica permite estender a utilidade e versatilidade dos modelos existentes o que pode poupar tempo de treino ou até mesmo ser uma solução válida para tarefas para a qual seria requerida

um conjunto volumoso de dados para treinar o modelo de raíz [DS18], como é o exemplo de tarefas de reconhecimento de voz, imagem [YCBL14] ou processamento de linguagem natural.

2.2 Visão por Computador

Este conceito baseia-se em ensinar um computador a processar uma imagem em formato digital, e a interpretá-la com recurso a algoritmos especializados, sendo possível extrair padrões e desta forma automatizar tarefas como identificação e seguimento de objetos, análise de movimento em vídeo e reconstrução de imagens [Dic20][com].

As imagens são formadas por píxeis, vetores com 3 dimensões (RGB), cada uma correspondente ao valor de vermelho, verde e azul presentes no mesmo variando de 0 a 255. Alguns algoritmos de aprendizagem automática fazem uso dos píxeis para identificar vértices, padrões ou mesmo classificar e detetar imagens a partir de padrões previamente aprendidos.

Hoje em dia com os avanços nas tecnologias móveis e redes sociais e o seu uso em massa temos uma grande quantidade de dados visuais disponíveis na internet, o que em conjunto com os progressos no poder de processamento dos computadores e algoritmos como as redes neuronais convolucionais fez com que seja possível que alguns sistemas de visão por computador sejam mais eficazes que a visão humana a detetar, categorizar e a reagir a estímulos visuais [Mih20].

2.3 Trabalhos Relacionados

A visão por computador e aprendizagem automática tem uma história relativamente recente no âmbito do desporto, no entanto tem sido largamente adotada e com diversos tipos de aplicação nesta área [cvS20][use20].

No caso do futebol existe o sistema Hawk-Eye, certificado pela FIFA, que usa 7 câmaras de alta velocidade, sendo capaz de identificar e classificar objetos em movimento semelhantes a uma bola baseado em características chave como a forma, cor e área, verificando se a mesma ultrapassou, ou não, a linha de golo com um erro de precisão de 1.5cm e uma velocidade de deteção de 1s.

Em 2011/2012 na Alemanha a Stemmer Imaging ajudou a Impire a desenvolver um sistema que utilizava 2 câmaras para o seguimento automático dos jogadores da Bundesliga, o que reduziu o número de operadores de câmara necessários para a tarefa.

No ténis este tipo de tecnologias são utilizadas para perceber a trajetória da bola que pode ser reconstruída através do seguimento da mesma a cada fotograma com a finalidade de perceber se aterrou dentro ou fora dos limites do campo.

Através da identificação e seguimento de objetos também podem ser obtidas estatísticas automaticamente sobre jogadores ajudando equipas na análise de desempenho dos mesmos, no recrutamento de novos talentos ou até no estudo dos adversários.

Em 2013 a NBA chegou a acordo com a STATS LLC para integrar o seu sistema de seguimento de atletas, SportVU, nas arenas de todas as equipas. Este sistema constituído por uma rede de 6 câmaras conseguia identificar em tempo real a posição dos jogadores e da bola 25 vezes por segundo, o que possibilitava recolher dados estatísticos individuais no que respeita a ressaltos, lançamentos e passes. Uma ferramenta que possibilitou tanto ao staff técnico, como a jogadores e até aos fãs ter um entendimento mais profundo do jogo.

A equipa de basquetebol Orlando Magic, pertencente à NBA, assinou, em 2019, um contrato de exclusividade com a empresa STATS para utilizar os dados recolhidos pelo seu sistema AutoSTATS com vista a otimizar as decisões no que diz respeito à contratação de jovens talentos. O AutoSTATS [Aut], é uma ferramenta de análise de vídeo que não requer a instalação de hardware adicional ,como conjuntos de câmaras especializadas, para seguimento de atletas e da bola, e junta à recolha das estatísticas individuais dos jogadores processamento de informação através de soluções de aprendizagem automática para gerar informação mais detalhada sobre o desempenho das equipas.

No estudo [KFSM17] foi utilizada uma câmara Swiss ranger SR400 time-of-flight produzida pela MESA Imaging AG com noção de profundidade diretamente por cima do atleta enquanto este fazia sombra com o objetivo de classificar automaticamente os diferentes tipo de golpes como ganchos, diretos e "uppercuts" com ambas as mãos. Primeiramente foi feita a identificação do torso do praticante e mais precisamente do pescoço, cotovelo e mão através do histograma de contornos a diferentes níveis de profundidade com o objetivo de conseguir extrair características como a direção do antebraço e o ângulo do cotovelo a fim de distinguir os diferentes golpes. Utilizando um classificador Multi-Class Support Vector Machine, treinado com 303 e testado com 302 amostras, foi possível obter uma precisão de 97% na classificação dos golpes de atletas previamente vistos pelo sistema e 96% em atletas nunca antes vistos pelo sistema.

O [PY06] visou a implementação de uma interface de reconhecimento de gestos para um jogo de computador chamado "O.J. Boxing" O reconhecimento de movimento e dos golpes são feitos por 2 câmaras distintas, uma à frente do jogador e outra por cima do mesmo, respetivamente. Para melhorar a identificação dos golpes o jogador utiliza luvas vermelhas de modo facilitar a identificação das mãos através da segmentação de cores após a imagem ter sido convertida num histograma de cores. Após identificação dos pontos de interesse, a distância relativa entre as duas luvas no eixo vertical é verificada para poder diferir entre um golpe de direita ou esquerda. Quanto à identificação do movimento é feita a partir da subtração de plano de fundo e posterior identificação do jogador com um rectângulo que o delimita em relação ao resto da imagem. A partir do deslocamento deste rectângulo entre o fotograma atual e o fotograma anterior pode ser obtida a direção do movimento do jogador, podendo ser ela centro-direita, direita-centro, centro-esquerda ou esquerda-centro. Utilizando este conjunto de técnicas as ações do jogador foram detetadas com uma precisão de 94% em média.

No trabalho [Kha21],a partir de dados extraídos por acelerómetros e giroscópios colocados em ambos os pulsos de um atleta de boxe, foi treinada uma rede neuronal com objetivo de categorizar os movimentos em "direto", "gancho", "uppercut" ou "movimento sem golpes". O conjunto de dados para treino da rede neuronal foi recolhido a partir de populações de atletas com diferentes tempos de prática do boxe de modo a incluir alguma variabilidade técnica dentro de cada movimento etiquetado para evitar overfitting. Este modelo conseguiu uma precisão de 92.93 $\pm 4.33\%$ na identificação de movimentos de atletas de elite e 91.89 $\pm 3.45\%$ com atletas de nível iniciante.

O projeto [HG18] atenta no reconhecimento de poses de atletas famosos no mundo do boxe. Foi utilizado um procedimento de transferência de conhecimento no qual tecnologia Open Pose, um modelo previamente treinado para identificação de poses humanas, para extração de pontos chave em fotos alusivas ao boxe (6000 amostras) e outros desportos (20000 amostras) com intuito de treinar uma rede neuronal para distinguir entre movimentos de "boxe" e "não-boxe". Com o conjunto de dados original este modelo conseguiu uma precisão de 89% a distinguir uma pose do boxe de uma pose associada a outro desporto.

O sistema yogAI [yog20] tem por base a identificação e classificação de poses de Yoga e podendo auxiliar o utilizador a nível postural no decorrer da prática. O objetivo será manter a pose requerida pela aplicação durante 30 segundos. O modelo Posenet é utilizado para fazer a deteção de poses humanas a partir de cada fotograma captado pela câmara do praticante e de seguida esse conhecimento traduzido num conjunto de 17 coordenadas a duas dimensões alimenta uma rede neuronal treinada previamente para classificar poses

de yoga.

A aplicação web Teachable Machine [Tea] permite criar modelos de Aprendizagem Automática de forma simples e rápida, sem ter que programar. É destinada à aprendizagem de todos os que queiram explorar o funcionamento dos algoritmos como Redes Neuronais sem conhecimento prévio da temática. Através deste sistema com uma interface simples e intuitiva o computador pode ser treinado para reconhecimento de poses, imagens ou sons, dando a possibilidade de experimentar com o modelo criado num ambiente gráfico que mostra a classificação após cada interação com o utilizador e o seu grau de confiança. É ainda possível exportar o modelo criado para JavaScript, fazendo uso da biblioteca Tensorflow.js.

3

Proposta

3.1 Objetivos

Como mencionado anteriormente o objetivo principal desta dissertação é a implementação de um sistema multimodal que utiliza visão por computador aliada a algoritmos de Aprendizagem Automática, mais precisamente Redes Neuronais, para identificar padrões de movimento específicos do boxe como golpes de "esquerda" e "direita", "esquiva" ou "guarda".

A partir da identificação das poses foi possível criar um jogo onde o praticante tem um tempo máximo de 30 segundos para executar corretamente o máximo de movimentos que lhe serão pedidos de forma aleatória pelo sistema, contribuindo de forma positiva para o apuro técnico do atleta, bem como para a melhoria da sua velocidade de reação [BCKC14] [KKhK+15].

No que toca a funcionalidades o sistema ajusta-se melhor a um praticante iniciante na modalidade, já que não faz uma distinção profunda dos vários tipos de golpe possíveis como "diretos", "ganchos" e "uppercuts", o que lhe dá uma menor complexidade no que respeita às sequências.

O sistema multimodal em questão foi implementado sob a forma de uma aplicação web, cuja interface pode ser controlada através do rato ou comandos de voz, e está alojada pelo GitHub Pages [pag], de forma

a garantir acessibilidade e portabilidade. Esta abordagem torna-se mais vantajosa pelo facto de a aplicação não requerer qualquer tipo de instalação, nem terá que existir uma versão diferente para cada plataforma, o que também facilita a implementação de futuras atualizações.

3.2 Tecnologias Utilizadas

3.2.1 JavaScript

JavaScript [js] é uma linguagem de scripting que permite embeber lógica numa página web. É importado diretamente no ficheiro HTML da página e permite a realização de tarefas avançadas que permitem ao utilizador interagir com a aplicação e mudar o estado da mesma, sendo suportado por qualquer browser moderno, inclusivé os móveis, sendo portanto uma boa escolha no que toca a acessibilidade e portabilidade.

3.2.2 HTML

É a linguagem que constitui todos os elementos visuais das páginas web como vídeos, imagens, botões, tabelas, texto, etc [htm].

O HTML é constituído por marcações e atributos.

As marcações são utilizadas para definir o começo e o fim de um elemento HTML.

Os atributos localizam-se dentro da marcação de abertura de um elemento HTML e contêm informação adicional que podem ser obrigatórios, opcionais ou padrão. Os atributos obrigatórios e opcionais alteram elementos específicos, enquanto que os padrão são atributos que podem ser aplicados a qualquer elemento.

3.2.3 CSS

CSS [css] é a linguagem utilizada para estilizar os elementos html. De uma forma simplista pode dizer-se que estabelece as regras segundas as quais cada elemento deve ser mostrado, desde a cor até à posição que ocupa no ecrã por exemplo.

Uma folha de estilos em CSS é constituída por regras. Cada uma dessas regras tem um seletor, que faz referência ao elemento em causa através da especificação do elemento em si, de atributos específicos como a classe ou o identificador único ou da posição do elemento relativamente a outros na árvore do documento.

3.2.4 ReactJS

ReactJS [rea][AML+20] é uma biblioteca Javascript de código aberto unicamente especializada em desenho de interfaces baseada em componentes visuais. Faz uso da sintaxe JSX, JavaScript XML, que permite embeber expressões JavaScript diretamente no código HTML.

Todos os componentes visuais React são funções JavaScript que podem, ou não, receber argumentos, aos quais se chama propriedades, e têm um método "render" implícito, para, como o próprio nome indica, fazer a renderização dos elementos HTML correspondentes.

Esta biblioteca tem como objetivo "dividir para conquistar", permitindo a desconstrução de uma aplicação

web em componentes simples e reutilizáveis para tornar mais ágil o desenvolvimento de interfaces de utilizador.

3.2.5 PoseNet

PoseNet é um modelo da biblioteca TensorFlow [tfm] capaz de identificar poses humanas em tempo real através da deteção de pontos-chave como olhos, orelhas, nariz, ombros, cotovelos, pulsos, ancas, joelhos e tornozelos [tfp18].

Este modelo é baseado numa Rede Neuronal Convolucional (MobileNet) [Puj20][HZC+17], desenhada para correr em dispositivos móveis pelo facto de ser leve e ter baixa latência, desenvolvida pela Google e que foi treinada no conjunto de dados frequentemente utilizado em tarefas de reconhecimento de imagem chamado ImageNet [Sar21].

Após uma deteção bem sucedida o modelo retorna um vetor de com um objeto formado pelos campos "pose" e "skeleton". O campo "pose" contém todos os pontos-chave identificados, as suas coordenadas (x e y) baseadas nas dimensões do vídeo, a confiança com que foi detetado cada um deles, bem como a confiança total da pose. O campo "skeleton" contém a informação necessária para unir os pontos articulares que formam o "esqueleto" do PoseNet [Nig19].

3.2.6 ml5.js

ml5.js [ml5] é uma biblioteca de Aprendizagem Automática em JavaScript que faz uso da biblioteca TensorFlow.js, abstraindo muitos dos seus detalhes técnicos, o que a torna acessível a uma população mais vasta de programadores.

Através da ml5.js podem ser importados modelos pré-treinados, como é o exemplo do PoseNet [Agr21], ou até mesmo treinar e personalizar os próprios modelos baseados em algoritmos de Aprendizagem Automática como Redes Neuronais ou K vizinhos mais próximos, entre outros.

Tudo isto permite que Aprendizagem Automática diretamente no browser, o que aumenta a acessibilidade e se torna ao mesmo tempo um ambiente familiar para o utilizador já que as aplicações web são largamente utilizadas nos dias de hoje, com a grande vantagem de todos os dados permanecerem no front-end da aplicação, o que faz com que surjam menos problemas relacionados com latência e garante segurança e proteção de dados.

3.2.7 Redes Neuronais

As Redes Neuronais [Ras16] são um algoritmo de Aprendizagem Automática popularmente utilizados em problemas de regressão ou classificação como o descrito nesta dissertação.

À unidade mais básica constituinte de uma rede neuronal chamamos neurónio, que não são mais que nós interligados formando uma rede a partir da qual podem trocar informação, para se ajustarem à realidade e serem capazes de perceber padrões, tal como acontece no cérebro humano. Cada neurónio contém uma função de ativação, que utiliza uma soma ponderada como entrada (função x em 3.1) e que baseada no valor das entradas, produz uma saída binária e um fator de peso. Dependendo do fator de peso assim será a importância da informação à saída do mesmo para os neurónios da camada seguinte, sendo que terá que ultrapassar um certo limiar para ser considerada.

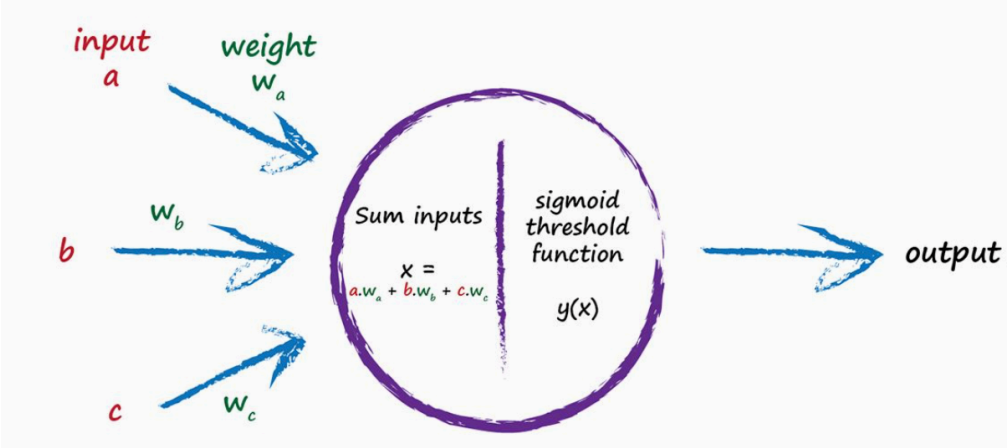

Figura 3.1: Representação do funcionamento de um neurónio. Fonte: [Ras16]

É necessário ter um conjunto de dados previamente rotulado a fim de treinar uma rede neuronal para uma tarefa de classificação, pois a mesma tem de saber antecipadamente qual a "resposta" a cada entrada, de modo a poder ajustar-se à realidade. A cada passagem dos dados pela rede neuronal a saída é comparada à resposta esperada e são feitas alterações aos pesos da rede de forma a minimizar uma função de perda. A uma passagem de dados pela rede chama-se iteração e à passagem de todo o conjunto de dados pela rede chama-se época e este ciclo pode estender-se tantas vezes quanto necessário até atingir o modelo ótimo.

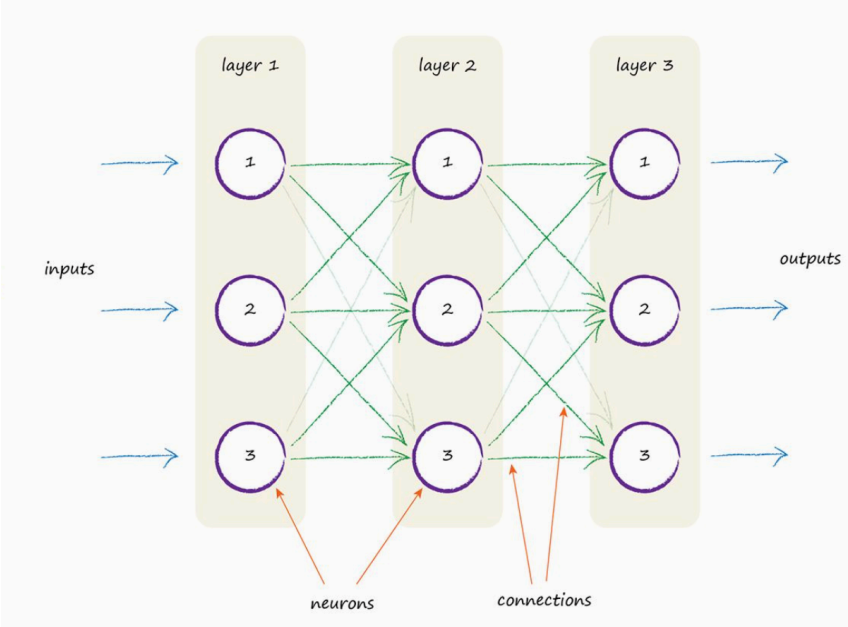

Figura 3.2: Representação de uma rede neuronal. Fonte: [Ras16]

3.2.8 p5.js

p5.js [p5] é uma biblioteca JavaScript para uso criativo que simplifica a interação do programador com o hardware do dispositivo no que diz respeito a colunas, microfone e câmara, bem como permite manipular elementos HTML e renderizá-los.

Pode dizer-se que o p5 é uma solução simples e eficaz de criar aplicações web interativas que envolvam sistemas multimodais.

3.2.9 annyang

A biblioteca JavaScript annyang [ann] permite que aplicações web sejam controladas por voz. Esta aplicação utiliza um serviço de reconhecimento de fala que fica à escuta durante a execução da aplicação. A transformação da fala em texto é feita e é usada como entrada para uma lista de comandos definida pelo utilizador com as respetivas funções inerentes a ativação de cada um. Esta biblioteca dá também a possibilidade ao programador de permitir que o utilizador da aplicação gere comandos de voz personalizados.

3.3 Arquitetura

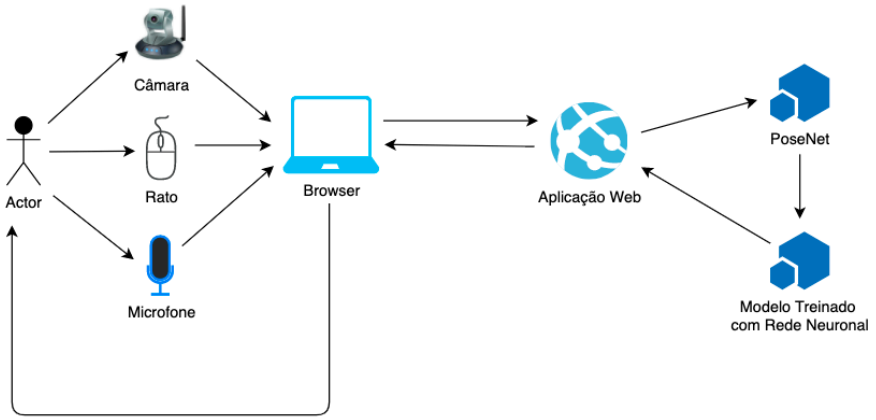

Figura 3.3: Representação da arquitetura do sistema

A arquitetura deste sistema multimodal assenta numa aplicação web construída em ReactJS. Foi desenhada uma interface multimodal simples e intuitiva com a qual o utilizador poderá interagir através de comandos de voz ou do rato para a iniciar ou parar a captura de vídeo. Com recurso à biblioteca p5.js foi criado um elemento HMTL "<video>", no qual será mostrado o vídeo em tempo real da câmara do utilizador com as dimensões 640x480, e um "<canvas>" exatamente com as mesmas dimensões e sobreposto ao anterior, onde é possível renderizar formas, texto e imagens. Logo que seja dada autorização ao browser para utilizar a câmara e o microfone, é despoletado um evento que ativa o modelo PoseNet e começa a procura de poses em de cada fotograma capturado.

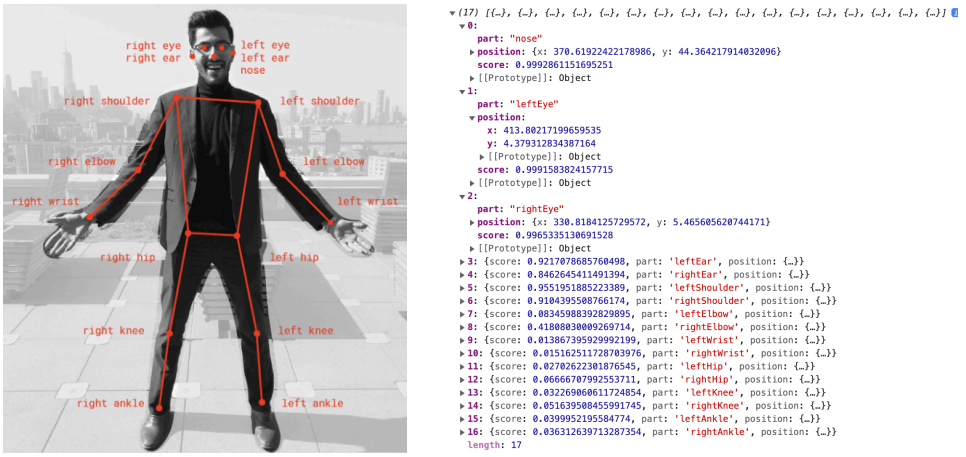

Figura 3.4: Vetor de 17 pontos-chave identificados pelo modelo PoseNet e respetiva legenda. Fonte: [key18]

Posteriormente cada coordenada do vetor de pontos-chave em 3.4 extraídos pelo PoseNet é utilizado como entrada em 3.5 para o modelo previamente treinado com uma Rede Neuronal para classificar a pose que está efetivamente a ser realizada como "esquerda", "direita", "esquiva" ou "guarda", notando que ambos os modelos Aprendizagem Automática estão localizados inteiramente no front-end da aplicação, de modo que não há necessidade de comunicar com um servidor externo para obter resultados [JSB+20].

```
let inputs = [];
for (let i = 0; i < pose.keypoints.length; i++) {
  let x = pose.keypoints[i].position.x;
  let y = pose.keypoints[i].position.y;
  inputs.push(x);
  inputs.push(y);
}
neuralNetwork.classify(inputs, getLabel);
```

Figura 3.5: Transferência de conhecimento do modelo PoseNet para a Rede Neuronal

Por fim a aplicação web é informada de que movimento foi efetivamente executado pelo utilizador e compara-a ao movimento requerido. No caso de existir correspondência o sistema passará a apresentar a novo movimento a ser executado através de texto no browser. Assim que clicado o botão "Fight & Stop" ou o comando de voz "fight" for executado o cronómetro que começa em 30 segundos iniciará uma contagem decrescente ao mesmo tempo da emissão de um sinal sonoro de uma campainha de ringue. Assim que o tempo chegar a 0 o campo "Score" no ecrã será atualizado com o número de movimentos corretamente executados de modo a avaliar o desempenho do jogador.

3.4 Implementação

3.4.1 Conjunto de Dados

Para a recolha do conjunto de dados foi seguido o procedimento seguinte:

- Primeiramente foram definidos os vários movimentos alusivos ao boxe que se queriam identificar.
- Categorizaram-se os movimentos em "esquerda", "direita", "esquiva" e "guarda" , representados pelas etiquetas "a", "b", "c", "d" respetivamente no sistema, como pode ser observado na Figura 3.6.
- Foi definido que cada um dos movimentos iria ser filmado na sua posição final como representado na Figura 3.6.
- Cada amostra é equivalente à pose identificada pelo modelo PoseNet num fotograma do vídeo filmado a uma taxa de 60 fotogramas por segundo.
- O alvo das filmagens foi um praticante iniciante da modalidade de boxe.
- O praticante usa como referência para o seu posicionamento o alvo no centro do ecrã, para o qual deve apontar sempre que executa uma "direita" ou uma "esquerda" bem como deve mantê-lo numa posição logo abaixo do seu queixo.
- Foram recolhidas amostras para treino, teste e validação. Tendo o conjunto total de dados tem 31425 amostras com a seguinte distribuição: 46% treino, 27% validação e 27% teste.
- Os conjuntos de treino, teste e validação foram recolhidos em momentos distintos para evitar que o actor permanecesse exatamente no mesmo local em que foi filmado e bem como tentar que introduzisse alguma variabilidade ao seu movimento e ao ângulo da filmagem para evitar o sobreajuste dos dados e aquando da avaliação ter uma melhor noção da capacidade de adaptação do modelo a possíveis variantes nas poses que correspondem a uma certa etiqueta [Gho19].
- A seleção de atributos foi baseada nas partes do corpo que serão visíveis aquando da filmagem, apenas membros superiores como demonstra 3.6 , excluindo à partida os pontos-chave dos pés, joelhos e ancas. Esta redução de dimensionalidade diminui o tempo de treino e a latência na resposta do modelo, bem como os recursos utilizados em termos de poder de processamento e memória.

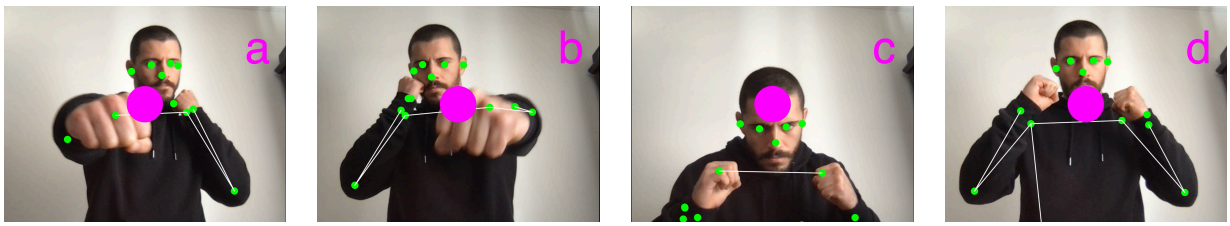

Figura 3.6: Treino da Rede Neuronal

Recolha de Dados para Treino

O conjunto de dados recolhido inclui as deteções de 11 pontos-chave de cada pose do praticante. Neste conjunto não figura nenhum dos pontos-chave localizado nos membros inferiores do actor em 3.4 (pés, joelhos e ancas).

Tabela 3.1: Conjunto de Dados de Treino

Etiqueta	Nº de Amostras
esquerda	3619
direita	3599
guarda	3607
esquiva	3610

Recolha de Dados para Validação

O conjunto de dados recolhido, inclui as deteções de 11 pontos-chave de cada pose do praticante. Neste conjunto não figura nenhum dos pontos-chave localizado nos membros inferiores do actor (pés, joelhos e ancas).

Tabela 3.2: Conjunto de Dados de Validação

Etiqueta	Nº de Amostras
esquerda	2137
direita	2129
guarda	2127
esquiva	2132

Recolha de Dados para Teste

O conjunto de dados recolhido, inclui as deteções de 11 pontos-chave de cada pose do praticante. Neste conjunto não figura nenhum dos pontos-chave localizado nos membros inferiores do actor (pés, joelhos e ancas).

Tabela 3.3: Conjunto de Dados de Teste

Etiqueta	Nº de Amostras
esquerda	2114
direita	2112
guarda	2106
esquiva	2125

3.4.2 Conjunto de Experiências

Algoritmo Selecionado

O algoritmo selecionado para ser treinado no conjuntos de dados recolhidos foi uma Rede Neuronal. A escolha deveu-se em grande parte à sua grande versatilidade de aplicação, facilidade a lidar com a alta dimensionalidade dos dados, o facto de as mesmas terem bons desempenhos em grandes conjuntos de

dados, o e a sua baixa latência principalmente de modo a que a modelo interferisse o menos possível na experiência do utilizador, de modo a não afetar a usabilidade e fluidez da aplicação.

```
const options = {
  inputs: 22,
  outputs: 4,
  task: "classification",
  debug: true,
  learningRate: 0.2,
  hiddenUnits: 16,
  layers: [
    {
      type: "dense",
      units: this.options.hiddenUnits,
      activation: "relu",
    },
    {
      type: "dense",
      activation: "softmax",
    },
  ],
};
```

Figura 3.7: Especificações da Rede Neuronal utilizada

A Rede Neuronal utilizada obedece às especificações expostas na Figura 3.7. Com um total de 22 entradas, coordenadas em 2 dimensões dos 11 pontos-chave do PoseNet considerados para o treino e 4 saídas, uma por cada movimento que se pretende classificar (direita, esquerda, esquiva e guarda). A taxa de aprendizagem é 0.2 e o número de neurónios na camada oculta da rede é 16, tal como o padrão estabelecido pela biblioteca ml5.js. A rede é constituída por 3 camadas, totalmente conectadas entre si (densas), a camada de entrada, uma camada oculta constituída por 16 neurónios com função de ativação ReLU e uma camada de saída com função de ativação Softmax.

Treino

Durante o treino do algoritmo foi seguido o protocolo seguinte:

- Para os 17 pontos-chaves recolhidos pelo PoseNet, a cada fotograma como demonstrado em 3.4, teremos uma coordenada em x e outra em y, o que dá um total de 34 entradas por pose detetada. No caso dos conjuntos de dados com redução de dimensionalidade (sem pés, joelhos e ancas), apenas temos 11 pontos-chave, o que dá um total de 22 entradas por pose detetada.
- Seguidamente procedeu-se à normalização de todos os conjuntos de dados como requerido pela biblioteca ml5.js.
- Utilizou-se a técnica de paragem antecipada [Ska18][TJA21] no treino da Rede Neuronal que consiste na realização de pseudo-testes com o conjunto de validação para avaliar o desempenho do modelo treinado com diferentes valores para o parâmetro das épocas, cessando assim o treino quando o

desempenho nas métricas escolhidas se começar a deteriorar, sinalizando um possível sobreajuste [Ska18].

- No treino foi utilizado o valor padrão de 32 para o tamanho do lote [ML18][Ben12] estabelecido pela biblioteca ml5.js.
- Foram treinados e validados ambos os modelos com as afinações de hiper-parâmetros seguintes:

Tabela 3.4: Valores de épocas com que o conjunto de dados foi treinado

Conjunto de dados	Nº de Épocas
treino	10; 20; 40; 60; 100

Identificação de Movimentos

Durante o tempo em que a câmara do jogador está ativa o modelo tenta classificar as poses identificadas pelo PoseNet a cada frame. A cada ação do jogador irá ser classificado um certo número de poses como mostra a Figura 3.8, dependendo do tempo de execução do movimento, do processador e ou placa gráfica utilizados no processamento da informação e da taxa de fotogramas por segundo da câmara web.

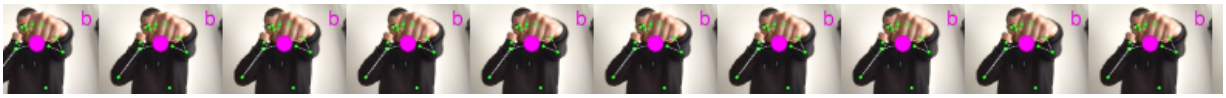

Figura 3.8: Exemplo de fotogramas pertencentes a uma única ação do utilizador

Isto significa que a cada movimento efetivo de "direita", "esquerda", "esquiva" ou "guarda" irá ser classificado um conjunto de poses, podendo nem todas corresponder exatamente àquelas com que o modelo foi treinado, causando alguma imprevisibilidade na identificação dos movimentos. Para mitigar esta situação foi definida uma heurística, que agrupa os resultados da classificação a cada 10 poses (número que poderá variar de acordo com as especificações do sistema) e percorre o conjunto de etiquetas à procura daquela com a maior cardinalidade e desta forma o sistema assume que o movimento executado pelo utilizador foi o correspondente a essa mesma etiqueta [BFGV16] [FGM$^+$15]. De modo a evitar alguns falsos positivos na identificação dos movimentos utilizando esta heurística convencionou-se que o sistema não poderá requerer a mesma posição duas vezes seguidas, pois dependendo da velocidade de movimentos e especificações da máquina do jogador poderia ter sido identificado o mesmo movimento várias vezes quando este só o executou apenas uma. Como tal o próximo movimento a ser efetuado é escolhido aleatoriamente de entre as etiquetas excluindo a última correspondente ao movimento anteriormente efetuado.

Especificações

O sistema foi desenvolvido e testado com as seguintes especificações:

- Sistema Operativo: Windows 10 Pro
- Browser: Google Chrome 96.0.4664.45
- Câmara: ASUS ROG EYE 1080p 60fps

- Placa Gráfica: NVIDIA RTX 3060
- AMD Ryzen 5900HS
- Mémória RAM: 16GB
- Ecrã: 14' 1080p

3.4.3 Interface do Utilizador

O sistema apresenta uma interface desenhada para ser simples e intuitiva. Aquando do arranque da aplicação a captura de vídeo inicia em tempo real num quadro com as dimensões 640x480 píxeis. Pode ser também obervado um temporizador que começa em 30 segundos, o campo "Score" com a última pontuação do utilizador o alvo (círculo rosa), em relação ao qual o jogador se deverá posicionar de modo a que fique por debaixo do queixo, e o movimento requerido pelo sistema que poderá ter o valor "a", "b", "c" ou "d" correspondentes a "esquerda", "direita", "esquiva" e "guarda", respetivamente.

Figura 3.9: Exemplo do estado inicial da aplicação

Para iniciar ou parar a contagem do tempo e movimentos com sucesso basta clicar do Botão "Fight | Stop" ou executar os comandos de voz "fight" ou "stop", que irão desencadear um sinal sonoro de uma campainha de ringue para alertar o utilizador. O botão "Draw & Erase" e os comandos de voz "draw" e "erase" têm como função desenhar ou apagar o esqueleto criado a partir dos pontos-chave identificados pelo PoseNet a cada frame da captura de vídeo.

Figura 3.10: Exemplo da estado da aplicação após serem dados os comandos "fight" e "draw"

A função particular do alvo é ajustar a posição do utilizador em relação à câmara para que os pontos-chave detetados pelo PoseNet não se distanciem demasiado daqueles com que o modelo foi treinado e testado, de modo a garantir um melhor desempenho e fluidez na experiência de jogo. A cada identificação correta de um movimento, a cor do movimento requerido pelo sistema muda para verde durante um frame e passa à posição seguinte que é dada de forma aleatória e não será igual à anterior. Caso o jogador não efetue um movimento de maneira correta ou o mesmo não seja identificado pelo sistema a pose requerida permanecerá inalterada. Após atingir o tempo máximo de 30 segundos, o utilizador poderá observar a sua pontuação e esta será igual ao número de movimentos que efetuou corretamente.

3.4.4 Publicação da Aplicação

Primeiramente foi gerada a versão de produção da aplicação, o que implica uma melhoria em termos de performance já que utiliza técnicas de caching de longo prazo para evitar que o browser carregue novamente o conteúdo de certos ficheiros sem que este tenha sido alterado, bem como suprime ainda avisos do React, que num ambiente de desenvolvimento são úteis, mas que acabam por tornar a biblioteca maior e mais lenta. De modo a permitir acessibilidade e para efeitos de demonstração a versão de utilizador da aplicação está publicada https://github.com/migjesus/boxing-ai-trainer e https://migjesus.github.io/boxing-ai-trainer/ e todo o código da mesma está localizado no repositório público https://github.com/migjesus/boxing-ai-trainer. Para correr a aplicação apenas será indispensável um computador com acesso à internet, uma câmara web e opcionalmente um microfone.

4

Resultados

Neste capítulo são introduzidas todas as métricas de acordo com a sua potencial relevância para a avaliação do modelo proposto em 3.4.2. Posteriormente são apresentados todos os resultados, para cada uma das métricas, alusivos à fase de treino e validação do modelo com afinação de híper-parâmetros, mais concretamente das épocas da Rede Neuronal. Por fim, após ter sido escolhido o modelo que melhor se ajusta ao problema de classificação recorrendo à técnica de paragem antecipada [TJA21], são expostos todos os resultados das métricas de desempenho relativos ao treino do mesmo.

4.1 Métricas Utilizadas

Em problemas de classificação, como o apresentado, a matriz de confusão é uma maneira útil de disposição dos dados para ter uma melhor ideia do desempenho do modelo. As linhas representam as classes reais e as colunas as classes previstas pelo modelo e com base nesta informação poderemos obter métricas como a Precisão, a Cobertura a Medida-F e a Exatidão [Mis18].

Tabela 4.1: Exemplo de uma matriz de confusão

		Previsto	
		Classe X	Classe Y
Real	Classe X	Verdadeiro Positivo	Falso Negativo
	Classe Y	Falso Positivo	Verdadeiro Negativo

A Precisão mostra-nos a proporção de identificações de positivos que estava efetivamente correta e é dada pela seguinte fórmula:

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos}+\text{Falsos Positivos}}$$

Um modelo é tanto mais preciso quanto menos falsos positivos identificar, o que acaba por ser uma mais valia no caso do sistema descrito na dissertação, pois a experiência do utilizador ficaria afetada caso fossem identificados movimentos que o mesmo não teria executado, podendo dar-se o caso de o mesmo executar um movimento incorreto e o sistema classificá-lo de forma contrária, permitindo o seu avanço na sequência.

Ao contrário da Precisão a Cobertura dá ênfase aos falsos negativos e mostra-nos que proporção de entre todos os positivos reais foi corretamente identificada e é dada pela seguinte fórmula:

$$\text{Cobertura} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos}+\text{Falsos Negativos}}$$

A Cobertura é tanto melhor quanto menos falsos negativos um modelo obtiver, o que no contexto descrito neste sistema multimodal também é uma métrica valiosa uma vez que irá ser requerida ao utilizador a execução de um movimento específico e caso não seja identificado, irá impactar a pontuação final já que o mesmo poderá perder segundos valiosos até que uma posição seja identificada.

A média harmónica da Precisão e da Cobertura, é dada pela Medida-F :

$$\text{Medida-F} = \frac{2\times\text{Precisão}\times\text{Cobertura}}{\text{Precisão}+\text{Cobertura}}$$

Já que tanto os falsos positivos como os falsos negativos podem ter ações impactantes a nível do sistema multimodal descrito, podendo implicar o avanço na sequência de movimentos sem que para tal o utilizador tenha executado o movimento correto ou a potencial perda de tempo a identificar um certo movimento que pode prejudicar a fluidez e a experiência de utilização, mas principalmente qualquer uma destas ocorrências terá impacto direto na pontuação, logo implicam perdas na capacidade que o sistema tem de avaliar o desempenho do jogador.

A Exatidão é a proporção de previsões corretas do modelo como mostra a fórmula:

$$\text{Exatidão} = \frac{\text{Verdadeiros Positivos}+\text{Verdadeiros Negativos}}{\text{Verdadeiros Positivos}+\text{Verdadeiros Negativos}+\text{Falsos Negativos}+\text{Falsos Positivos}}$$

Esta métrica também tem um carácter interessante para o projeto visto que apesar de os falsos positivos e falsos negativos terem um certo peso, com a heurística definida anteriormente em 3.4.2 estes últimos acabam por ser algo "suavizados" e a taxa de acerto do modelo acaba por ganhar alguma preponderância.

4.2 Resultados Empíricos

Os dados seguintes referem-se ao modelo produzido utilizando uma Rede Neuronal de 3 camadas, com diferentes valores para as épocas (10, 20 e 40 respetivamente) a partir do conjunto de dados de treino e posteriormente validados com o conjunto de dados de validação.

4.2.1 Treino e Validação

Rede Neuronal Treinada com 10 Épocas

Tabela 4.2: Matriz de Confusão da Rede Neuronal treinada com 10 épocas no conjunto de dados de validação

	esquerda	direita	esquiva	guarda
esquerda	1864	61	141	71
direita	0	1945	0	184
esquiva	0	0	2132	0
guarda	0	0	0	2127

O modelo treinado com 10 épocas apresenta uma exatidão total de 0.95 e a Medida-F para cada uma das suas classes é apresentada na Tabela 4.3.

Tabela 4.3: Métricas de desempenho da Rede Neuronal treinada com 10 épocas no conjunto de dados de validação

	Precisão	Cobertura	Medida-F
esquerda	1	0.87	0.93
direita	0.97	0.91	0.95
esquiva	0.94	1	0.97
guarda	0.89	1	0.94

Rede Neuronal Treinada com 20 Épocas

Tabela 4.4: Matriz de Confusão da Rede Neuronal treinada com 20 épocas no conjunto de dados de validação

	esquerda	direita	esquiva	guarda
esquerda	1858	41	167	71
direita	0	2021	0	108
esquiva	0	0	2132	0
guarda	0	0	0	2127

O modelo treinado com 20 épocas apresenta uma exatidão total de 0.95 e a Medida-F para cada uma das suas classes é apresentada na Tabela 4.5.

Tabela 4.5: Métricas de desempenho da Rede Neuronal treinada com 20 épocas no conjunto de dados de validação

	Precisão	Cobertura	Medida-F
esquerda	1	0.87	0.93
direita	0.98	0.95	0.96
esquiva	0.93	1	0.96
guarda	0.92	1	0.95

Rede Neuronal Treinada com 40 Épocas

Tabela 4.6: Matriz de Confusão da Rede Neuronal treinada com 40 épocas no conjunto de dados de validação

	esquerda	direita	esquiva	guarda
esquerda	1856	48	159	74
direita	0	1915	0	214
esquiva	0	0	2132	0
guarda	0	0	0	2127

O modelo treinado com 40 épocas apresenta uma exatidão total de 0.94 e a Medida-F para cada uma das suas classes é apresentada na Tabela 4.7.

Tabela 4.7: Métricas de desempenho da Rede Neuronal treinada com 40 épocas no conjunto de dados de validação

	Precisão	Cobertura	Medida-F
esquerda	1	0.87	0.93
direita	0.98	0.9	0.94
esquiva	0.93	1	0.96
guarda	0.88	1	0.94

Rede Neuronal Treinada com 60 Épocas

Tabela 4.8: Matriz de Confusão da Rede Neuronal treinada com 60 épocas no conjunto de dados de validação

	esquerda	direita	esquiva	guarda
esquerda	1836	50	176	75
direita	0	1950	0	179
esquiva	0	0	2132	0
guarda	0	0	0	2127

O modelo treinado com 60 épocas apresenta uma Exatidão total de 0.94 e a Medida-F para cada uma das suas classes é apresentada na Tabela 4.9.

Tabela 4.9: Métricas de desempenho da Rede Neuronal treinada com 60 épocas no conjunto de dados de validação

	Precisão	Cobertura	Medida-F
esquerda	1	0.86	0.92
direita	0.98	0.92	0.95
esquiva	0.92	1	0.96
guarda	0.89	1	0.94

Rede Neuronal Treinada com 100 Épocas

Tabela 4.10: Matriz de Confusão da Rede Neuronal treinada com 100 épocas no conjunto de dados de validação

	esquerda	direita	esquiva	guarda
esquerda	1809	74	174	80
direita	0	1871	0	258
esquiva	0	0	2132	0
guarda	0	0	0	2127

O modelo treinado com 100 épocas apresenta uma exatidão total de 0.93 e a Medida-F para cada uma das suas classes é apresentada na Tabela 4.11.

Tabela 4.11: Métricas de desempenho da Rede Neuronal treinada com 100 épocas no conjunto de dados de validação

	Precisão	Cobertura	Medida-F
esquerda	1	0.85	0.92
direita	0.96	0.88	0.92
esquiva	0.92	1	0.96
guarda	0.86	1	0.92

4.2.2 Seleção do Modelo

O modelo selecionado foi a Rede Neuronal treinada com 20 épocas pois apesar de apresentar um desempenho muito semelhante em termos de Medida-F Média e exatidão ao modelo treinado com 10 épocas a Medida-F para as classes "esquerda", "direita" e "guarda" melhoraram ligeiramente, embora o inverso ter acontecido relativamente à "esquiva". Portanto pode dizer-se que este é o modelo que oferece uma melhor relação entre as medidas-F para cada classe e a exatidão global, o que proporcionará mais deteções corretas dos movimentos, minimizando também os falsos positivos e os falsos negativos para garantir que a pontuação final do jogo seja demonstrativa do desempenho do jogador.

Tabela 4.12: Comparação de desempenho global dos vários modelos treinados no conjunto de dados de validação

	Medida-F Média	Exatidão
Rede Neuronal treinada com 10 épocas	0.95	0.95
Rede Neuronal treinada com 20 épocas	0.95	0.95
Rede Neuronal treinada com 40 épocas	0.94	0.94
Rede Neuronal treinada com 60 épocas	0.94	0.94
Rede Neuronal treinada com 100 épocas	0.93	0.93

Como ilustrado na Tabela 4.12 pode notar-se que o desempenho do modelo se começou a degradar a partir das 40 épocas em diante, face às experiências anteriores , tanto pelo valor da exatidão como pela Medida-F relativa a cada classe, o que pode significar que o modelo começou a sofrer um sobreajuste relativamente aos dados de treino e portanto aumentar o número de épocas ainda poderia deixá-lo mais tendencioso e menos adaptável a variabilidade nas poses, fisionomia do praticante, ângulo da filmagem e distância em relação à câmara, já que estes fatores influenciam a deteção dos pontos-chave de cada pose, tendo perdido alguma capacidade de generalização.

4.2.3 Teste do Modelo

Os dados seguintes referem-se ao modelo produzido utilizando uma Rede Neuronal com 20 épocas a partir do conjunto de dados de treino e testada com o conjunto de dados de teste.

Tabela 4.13: Matriz de Confusão da Rede Neuronal treinada com 20 épocas no conjunto de dados de teste

	esquerda	direita	esquiva	guarda
esquerda	2106	0	0	8
direita	0	1975	0	137
esquiva	0	0	2106	0
guarda	0	0	48	2077

O modelo treinado com 20 épocas apresenta uma exatidão total de 0.977 e a Medida-F para cada uma das suas classes é apresentada Figura 4.14.

Tabela 4.14: Métricas de desempenho da Rede Neuronal treinada com 20 épocas no conjunto de dados de teste

	Precisão	Cobertura	Medida-F
esquerda	1.000	0.996	0.998
direita	1.000	0.935	0.966
esquiva	0.978	1.000	0.989
guarda	0.935	0.977	0.956

Tabela 4.15: Visão global do desempenho da Rede Neuronal treinada com 20 épocas no conjunto de dados de teste

	Medida-F Média	Exatidão
Rede Neuronal treinada com 20 épocas	97.7%	97.7%

O modelo treinado com uma Rede Neuronal com ajuste de parâmetros para 20 épocas apresenta uma exatidão de 97.7% e uma Medida-F Média de 97.7% no conjunto de dados de teste.

4.2.4 Discussão

Através da análise de desempenho realizada na fase de treino do algoritmo, pode ser verificado que as especificações padrão da biblioteca ml5.js na Figura 3.7 se ajustam ao contexto do problema, pelo facto de a função de perda baixar o seu valor ao longo das épocas e convergir rapidamente para um mínimo próximo de 0. Este facto não garante melhoria de desempenho, pois a perda é apenas relativa ao conjunto de dados de treino, significando que o modelo a partir de um certo limiar de épocas poderá perder a sua capacidade de generalização [YSGR19] e começar a sofrer sobreajuste. A técnica de paragem antecipada [TJA21] do treino permite evitar o sobreajuste [Ska18] e consiste na realização de vários pseudo-testes com o conjunto de validação ao longo do treino da Rede Neuronal e como se observa na Figura 4.13 a partir das 40 épocas em diante o desempenho começa a deteriorar-se, sendo que os testes posteriores relativos aos valores de 60 e 100 épocas são puramente demonstrativos da perda progressiva da capacidade de generalização.

Entre os modelos de Redes Neuronais treinados, com o conjunto de dados de treino, e parametrização para 10, 20, 40, 60 e 100 épocas o que obteve melhor desempenho quando testado com o conjunto de dados de validação foi a Rede Neuronal treinada com 20 épocas apresentando uma Exatidão e Medida-F Média de 95% como mostra a Figura 4.13, tendo sido um resultado satisfatório e bastante válido uma vez que o conjunto de treino e validação não apresentam cardinalidades excessivamente diferentes, representando 47% e 27% do total dos dados respetivamente, e tanto um como outro foram gerados em momentos distintos o que possibilitou alguma variabilidade na posição do praticante de boxe em relação à câmara e consequentemente provocou variações ligeiras a nível da pose, garantindo assim uma maior imparcialidade entre conjuntos de dados.

O modelo escolhido para a implementação do projeto foi, portanto, a Rede Neuronal treinada com 20 épocas e posteriormente testada com o conjunto de dados de teste, obtendo uma Medida-F Média e Exatidão de 97.7% presentes na Figura 4.15, o que se verificou um bom resultado visto que os conjuntos de treino e teste não apresentam uma grande diferença em termos de número de amostras e ambos foram gerados sob condições ligeiramente diferentes, demonstrando boa capacidade de generalização.

Apesar de os conjuntos de dados terem sido gerados com alguma variabilidade [Gho19] introduzida a nível das poses, seja pela sua colocação em relação à câmara ou posicionamentos articulares ligeiramente diferentes durante a recolha de dados para uma das classes o desempenho da Rede Neuronal foi muito satisfatória, encontrando um bom equilíbrio entre a Exatidão e a Medida-F, o que garante não só muitas identificações corretas como uma pequena taxa de falsos positivos e falsos negativos que poderiam ter um impacto negativo na experiência de utilização da aplicação. Contudo não pode ser esquecido o facto de todo o conjunto de dados ter sido gerado com as poses relativas a uma só pessoa, o que faz com que seja invariavelmente tendencioso, perdendo alguma capacidade de generalização no que compete à classificação das poses de uma pessoa com uma fisionomia distinta, altura por exemplo, pois influencia bastante as coordenadas dos pontos-chave identificados e posição relativa à câmara. Idealmente e para uma maior

fiabilidade do sistema o conjunto de dados deveria ter sido gerado a partir da filmagem das poses de vários praticantes da modalidade de boxe para permitir maior resiliência do modelo perante variações na técnica ou fisionomia dos utilizadores. Provavelmente a avaliação de desempenho não apresentaria valores tão elevados, porém com um conjunto de dados grande o suficiente uma melhor ideia do verdadeiro potencial de generalização deste modelo poderia ter sido dada.

5

Conclusão

Este capítulo contém o balanço de todo o trabalho efetuado, mencionando todos os objetivos cumpridos, bem como as limitações da abordagem escolhida. São ainda descritos os trabalhos futuros no âmbito do tema selecionado, incluindo propostas para suprimir algumas limitações identificadas e a sugestão de diferentes contextos de aplicação para o sistema desenvolvido.

5.1 Balanço

Esta dissertação teve como objetivo a criação de um sistema multimodal para avaliar o desempenho de atletas. Depois de analisar o estado da arte concluiu-se que apesar de já existirem muitas soluções que permitem acompanhar o desenvolvimento de capacidades técnicas, táticas e físicas algo que este tipo de sistema ainda carece é de alguma portabilidade e acessibilidade. Motivos pelos quais o maior desafio deste projeto foi conseguir implementar um sistema apenas com ferramentas de código aberto e que utilizasse o mínimo de hardware especializado possível e que ainda assim garantisse alguma fiabilidade na extração de indicadores de desempenho.

O desporto escolhido como contexto de aplicação do sistema foi o boxe, pela complexidade de movimentos que envolve, o facto de ser uma modalidade de contacto e necessitar de técnicas de avaliação de desempenho pouco invasivas, de modo a não prejudicar o desempenho dos atletas.

Para garantir acessibilidade e portabilidade o sistema foi embebido numa aplicação web construída em ReactJS, que utiliza a câmara e o rato ou microfone como veículos de interação de modo a iniciar a captura de vídeo. Logo que a captura inicia um evento relativo ao modelo PoseNet é disparado, capaz de identificar 17 pontos-chave a 2 dimensões relativos a poses humanas em imagens, enviando estes mesmos dados pré-processados sujeitos a normalização (requerida pela biblioteca ml5.js) e seleção de atributos, excluindo todos os pontos articulares pertencentes aos membros inferiores. A transferência de conhecimento é feita em tempo real, e recebida por uma Rede Neuronal de 3 camadas, com a configuração padrão da biblioteca ml5.js, tal como acontece nos projetos [yog20] e [Tea] e classifica o conjunto de coordenadas em "esquerda", "direita", "esquiva" ou "guarda".

De momento modelos como o PoseNet ainda não estão suficientemente desenvolvidos para dar fortes indicadores biomecânicos no que toca ao alto rendimento, pois a suscetibilidade ao erro na identificação dos pontos-chave a cada fotograma ainda é suficiente para impactar o valor de qualquer indicador que se pretenda extrair. Em trabalhos relacionados como [KFSM17] e [Kha21] conseguem distinguir-se vários tipos de golpes de mão como ganchos, diretos e "uppercuts", pelo facto de recorrerem a uma câmara com noção de profundidade por cima do atleta e unidades de medida inerciais colocadas nos pulsos dos mesmos, respetivamente. Esta introdução de hardware especializado adicional permitiu uma noção tridimensional em ambos os projetos, aumentando a dimensionalidade dos dados, o que possibilitou diferenciar os golpes de maneira eficaz. Algo que não aconteceria no sistema implementado no âmbito desta dissertação por se considerarem apenas 2 dimensões, o que, potencialmente, faria com que existisse demasiada sobreposição entre as várias classes e impactaria negativamente a aprendizagem da Rede Neuronal. Devido a esta limitação de dimensionalidade, e pela diferença nos golpes ser pouco vincada, optou por fazer-se uma categorização mais generalizada em golpes de "direita" e "esquerda", tal como no projeto [PY06].

A Rede Neuronal para classificação das poses foi treinada a partir de um conjunto de dados gerado pela filmagem de um atleta de grau iniciante na modalidade de boxe dividido em treino, validação e teste com percentagens respetivas de 46%, 27% e 27% num total de 31425 amostras, sendo que todas as classes estão balanceadas. Durante o treino do algoritmo foi afinado o hiper-parâmetro relativo às épocas e verificou-se através da técnica de interrupção antecipada do treino que a Rede Neuronal treinada com 20 épocas seria a que melhor de ajustava ao problema com uma Exatidão e Medida-F Média de 95% uma vez que as medidas de desempenho escolhidas se foram degradando das 20 épocas em diante pela perda de capacidade de generalização do modelo. Posteriormente o desempenho do modelo escolhido foi avaliado ao classificar o conjunto de teste, obtendo uma Exatidão e Medida-F Média de 97.7%, verificando-se bastante satisfatório para a tarefa proposta. Contudo há que ter em conta a limitação que representa o facto de todo o conjunto de dados ter sido gerado pela mesma pessoa. Apesar de ter sido introduzida alguma variabilidade na recolha de dados, devido a pequenas alterações na pose relativa a uma certa classe, a fisionomia continua a ser um fator-chave no posicionamento dos pontos-chave identificados pelo PoseNet. Dada esta situação, num cenário ideal deveriam ter sido recolhidas amostras paras as 4 classes a partir de um maior número de participantes e preferencialmente de um grau de experiência mais avançado na modalidade para garantir que o modelo enraizaria bons padrões técnicos.

É preciso também distinguir a classificação de uma pose da classificação de um movimento [BFGV16]. Um movimento é um conjunto de poses rotuladas em "esquerda", "direita", "esquiva" ou "guarda" e com as especificações do sistema em que foi testado o projeto de dissertação são classificadas cerca de 60 poses por segundo. Posto isto, cada vez que o sistema requer um movimento, é utilizada uma heurística [FGM$^+$15] que considera que um movimento só foi efetivamente efetuado se o rótulo de maior cardinalidade encontrado nas últimas 10 classificações for aquele que lhe corresponde. Esta regra não

garante o desempenho da aplicação desenvolvida noutros sistemas com hardware inferior pelo facto de a taxa de fotogramas por segundo da câmara influenciar a quantidade de informação por unidade de tempo recebida pela aplicação, assim como o modelo do processador e da placa gráfica influenciam a latência da classificação das poses. Uma abordagem a este problema passa por variar o tamanho do conjunto de poses classificadas no qual se procura o rótulo com maior cardinalidade com base na taxa de fotogramas por segundo da câmara utilizada, já que é o fator com maior preponderância. Utiliza-se o tamanho 10 para conjuntos de poses capturadas por câmaras com taxa de 60 fotogramas por segundo e 5 para câmaras com uma taxa de 30 fotogramas por segundo por exemplo. Embora a heurística ajude a que seja feito um reconhecimento dos movimentos, também esconde algumas limitações do modelo como a incapacidade de classificar corretamente algumas poses transitórias correspondentes a uma única ação, sendo uma maneira de lidar com toda a imprevisibilidade para a qual o modelo não foi treinado.

O sistema descrito nesta dissertação assumiu a forma de um jogo com tempo máximo de 30 segundos no qual o utilizador tem de executar uma sequência de movimentos aleatória, e na qual 2 movimentos nunca se repetem seguidamente para evitar falsos positivos. Quando o tempo atinge o seu valor limite um ecrã é mostrado com a pontuação correspondente ao número de movimentos corretamente efetuados. Ainda que a aplicação não tenha sido exaustivamente testada por mais que um utilizador, exibe um comportamento bastante responsivo na identificação dos movimentos e sem problemas graves a nível de falsos positivos e falsos negativos que pudessem atrapalhar a extração da métrica pretendida.

Por fim, ressalva-se que, muito provavelmente, os bons resultados obtidos se devem a um sobreajuste dadas as limitações das tecnologias empregues e à pouca variabilidade a nível de fisionomias humanas presentes no conjunto de dados. Com a crescente versatilidade das tecnologias web e a otimização de modelos de Aprendizagem Automática para correr diretamente no browser, permitindo baixas latências, a perspetiva revela-se animadora para que num futuro próximo seja possível retirar indicadores de grande importância para a avaliação de métricas de desempenho de atletas utilizando apenas ferramentas de código aberto e hardware presente em qualquer computador portátil ou tablet comum.

5.2 Trabalhos Futuros

Apesar de o sistema em causa estar efetivamente implementado, é um trabalho longe de estar terminado, havendo muita margem para progressão. Ao conjunto de dados já existente poderiam ser adicionadas mais amostras de praticantes da modalidade de boxe com diferentes fisionomias para criar uma melhor capacidade de generalização do modelo proposto. Atualmente o sistema consegue apenas categorizar um movimento como "direita", "esquerda", "esquiva" ou "guarda", não diferindo os vários golpes de mão, no entanto utilizando o modelo BlazePose [tfb] seria possível obter valores tridimensionais para a posição de 33 pontos-chave [IGL21][BGR+20] que identificam uma pose humana e como tal, com este aumento de dimensionalidade, aumenta também a possibilidade de conseguir distinguir entre ganchos, diretos e "uppercuts", criando padrões mais bem definidos que ajudariam a aprendizagem da Rede Neuronal. Todo este trabalho poderia ser alargado a modalidades similares como o kickboxing [SOWK19] ou muay thai, incluindo a classificação de golpes com membros inferiores como frontais ou médios, por exemplo.

Outra aplicação que o sistema descrito nesta dissertação poderia ter era a de identificar uma sequência de posições em tempo real a partir da filmagem de um atleta a realizar um exercício de sombra em frente à câmara. Este tipo de exercício é muito utilizado por atletas de desportos de combate e consiste em visualizar mentalmente a luta e efetuar os golpes o mais próximo possível da realidade. A partir da identificação e classificação dos vários movimentos presentes ao longo da filmagem poderia ser traçado um "perfil" para o atleta, através da análise da árvore de sufixos relativa à sequência produzida pelos rótulos de cada movimentos por ele efetuado. Este estudo poderia ser potencialmente útil na criação de atributos para um

novo conjunto de dados que permitiria, por meio de técnicas de regressão, identificar um atleta com base nas suas poses e sequências de movimentos característicos.

Bibliografia

[Agr21] Raghav Agrawal. Posture detection using posenet with real-time deep learning project, 2021. https://www.analyticsvidhya.com/blog/2021/09/posture-detection-using-posenet-with-real-time-deep-learning-project/ ,Acedido:2021-11-6.

[AML+20] Anthony Accomazzo, Nate Murray, Ari Lerner, Clay Allsopp, David Gutman, and Tyler McGinnis. *Fullstack React ; The Complete Guide to ReactJS and Friends*. r40. Newline.co, 2020.

[AN18] Manisha Biswas (auth.) Abhishek Nandy. *Reinforcement Learning : With Open AI, TensorFlow and Keras Using Python*. Apress, 1 edition, 2018.

[ann] annyang. https://www.talater.com/annyang/,Acedido:2021-10-28.

[Aut] Autostats. https://www.statsperform.com/, Acedido: 2021-9-15.

[BCKC14] Kara Blacker, Kim Curby, Gian Klobusicky, and Jason Chein. Effects of action video game training on visual working memory. *Journal of experimental psychology. Human perception and performance*, 40, 07 2014.

[Ben12] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *CoRR*, abs/1206.5533, 2012.

[BFGV16] Hakan Bilen, Basura Fernando, Efstratios Gavves, and Andrea Vedaldi. Action recognition with dynamic image networks. *CoRR*, abs/1612.00738, 2016.

[BGR+20] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. Blazepose: On-device real-time body pose tracking. *CoRR*, abs/2006.10204, 2020.

[BZL+14] Vikranth R. Bejjanki, Ruyuan Zhang, Renjie Li, Alexandre Pouget, C. Shawn Green, Zhong-Lin Lu, and Daphne Bavelier. Action video game play facilitates the development of better perceptual templates. *Proceedings of the National Academy of Sciences*, 111(47):16961–16966, 2014.

[com] What is computer vision. https://www.ibm.com/topics/computer-vision,Acedido:2021-6-10.

[CSBR19] Emily E Cust, Alice J. Sweeting, Kevin A. Ball, and Sam Robertson. Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance. *Journal of Sports Sciences*, 37:568 – 600, 2019.

[css] Css tutorial. https://www.w3schools.com/css/,Acedido:2021-11-2.

[cvS20] Computer vision in sport, 2020. https://www.sportperformanceanalysis.com/article/computer-vision-in-sport, Acedido: 2021-6-5.

[Dic20] Ben Dickson. What is computer vision, 2020. https://www.pcmag.com/news/what-is-computer-vision,Acedido:2021-6-10.

[DS18] Tamoghna Ghosh Dipanjan Sarkar, Raghav Bali. *Hands-On Transfer Learning with Python Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras*. Packt Publishing, 2018.

[EEU18] Ayşe Eldem, Huseyin Eldem, and Deniz Ustun. A model of deep neural network for iris classification with different activation functions, 2018.

[FGM+15] Basura Fernando, Efstratios Gavves, José Oramas M., Amir Ghodrati, and Tinne Tuytelaars. Rank pooling for action recognition. *CoRR*, abs/1512.01848, 2015.

[Gho19] Salma Ghoneim. 5 types of bias & how to eliminate them in your machine learning project, 2019. https://towardsdatascience.com/5-types-of-bias-how-to-eliminate-them-in-your-machine-learning-project-75959af9d3a0 Acedido:2021-10-5.

[HG18] Evan Huang and Cedric Fraces Gasmi. Applying computer vision and deep learning to the art of boxing. Deep Learning - Final Project, Stanford University, 2018.

[HJK+09] N. Huh, S. Jo, H. Kim, J. H. Sul, and M. W. Jung. Model-based reinforcement learning under concurrent schedules of reinforcement in rodents. *Learning and Memory vol. 16 iss. 5*, 16, apr 2009.

[htm] Html: Hypertext markup language. https://developer.mozilla.org/en-US/docs/Web/HTML,Acedido:2021-11-2.

[HZC+17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[IGL21] Valentin Bazarevsky Ivan Grishchenko and Na Li. High fidelity pose tracking with mediapipe blazepose and tensorflow.js, 2021. https://blog.tensorflow.org/2021/05/high-fidelity-pose-tracking-with-mediapipe-blazepose-and-tfjs.html, Acedido:2021-10-8.

[inj19] Computer vision and machine learning in sports analytics: Injury and outcome prediction, 2019. https://softarex.com/blog/computer-vision-and-machine-learning-in-sports-analytics-injury-and-outcome-predict ,Acedido:2021-6-5.

[js] Javascript. https://www.javascript.com/,Acedido:2021-8-3.

[JSB+20] Nozha Jlidi, Ahmed Snoun, T. Bouchrika, Olfa Jemai, and Mourad Zaied. Ptlhar: Posenet and transfer learning for human activities recognition based on body articulations, 2020.

[key18] Real-time human pose estimation in the browser with tensorflow.js, 2018. https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html, Acedido: 2021-8-7.

[KFSM17] Soudeh Kasiria, Clinton Fookesa, Sridha Sridharana, and Stuart Morgan. Fine-grained action recognition of boxing punches from depth imagery. *Computer Vision and Image Understanding*, 2017.

[KGC15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

[Kha21] Ilshat Khasanshin. Application of an artificial neural network to automate the measurement of kinematic characteristics of punches in boxing. *Applied Sciences*, 2021.

[KKhK+15] Yong-Hwan Kim, Dong-Wha Kang, Dong ho Kim, Hye-Jin Kim, Yuka Sasaki, and Takeo Watanabe. Real-time strategy video game experience and visual perceptual learning. *The Journal of Neuroscience*, 35:10485 – 10492, 2015.

[LSNH+09] Liang Lu, Reihaneh Safavi-Naini, Markus Hagenbuchner, Willy Susilo, Jeffrey Horton, Sweah Yong, and Ah Tsoi. Ranking attack graphs with graph neural networks, 2009.

[Mih20] Ilija Mihajlovic. Everything you ever wanted to know about computer vision, 2020. https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-a Acedido:2021-7-20.

[Mis18] Aditya Mishra. Metrics to evaluate your machine learning algorithm, 2018. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234,Acedido:2021-10-5.

[ML18] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *CoRR*, abs/1804.07612, 2018.

[ml5] ml5.js. https://ml5js.org/,Acedido:2021-9-12.

[Nig19] Kartik Nighania. Machine learning for everyone: Pose estimation in a browser with your webcam, 2019. https://betterprogramming.pub/machine-learning-for-everyone-pose-estimation-in-a-browser-with-your-webcam-65bb264 Acedido:2021-11-5.

[p5] p5.js. https://p5js.org/,Acedido:2021-11-3.

[pag] Github pages. https://pages.github.com/,Acedido:2021-7-20.

[PKV+16] Ewa Polak, Jerzy Kulasa, Antonio VencesBrito, Maria António Castro, and Orlando Fernandes. Motion analysis systems as optimization training tools in combat sports and martial arts. *Revista de Artes Marciales Asiáticas*, 10:105 – 123, 01 2016.

[Puj20] Abhijeet Pujara. Image classification with mobilenet, 2020. https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470, Acedido:2021-11-6.

[PY06] Ji-Young Park and June-Ho Yi. Gesture recognition based interactive boxing game. *International Journal of Information Technology*, 2006.

[Ras16] Tariq Rashid. *Make Your Own Neural Network*. Createspace Independent Publishing Platform, 2016.

[rea] Reactjs. https://reactjs.org/,Acedido:2021-9-15.

[Sar21] Arjun Sarkar. Understanding depthwise separable convolutions and the efficiency of mobilenets, 2021. https://towardsdatascience.com/understanding-depthwise-separable-convolutions-and-the-efficiency-of-mobilenets-6de Acedido:2021-11-6.

[SJB+21] Ahmed Snoun, Nozha Jlidi, Tahani Bouchrika, Olfa Jemai, and Mourad Zaied. Towards a deep human activity recognition approach based on video to image transformation with skeleton data. *Multimedia Tools and Applications*, 80(19):29675–29698, July 2021. https://link.springer.com/article/10.1007/s11042-021-11188-1 ; https://dblp.uni-trier.de/db/journals/mta/mta80.html#SnounJBJZ21.

[Ska18] Piotr Skalski. Preventing deep neural network from overfitting, 2018. https://towardsdatascience.com/preventing-deep-neural-network-from-overfitting-953458db800a,Acedido: 2021-10-8.

[SLV+19] Dhruv R. Seshadri, Ryan T. Li, James E. Voos, James R. Rowbottom, Celeste M. Alfes, Christian A. Zorman, and Colin K. Drummond. Wearable sensors for monitoring the internal and external workload of the athlete. *npj Digital Medicine vol. 2 iss. 1*, 2, jul 2019.

[SOWK19] Kasper Soekarjo, Dominic Orth, Elke Warmerdam, and John Kamp. Automatic classification of strike techniques using limb trajectory data, 04 2019.

[Tea] Teachable machine. https://teachablemachine.withgoogle.com/, Acedido: 2021-09-30.

[Ten19] Oliver Tensor. *Machine Learning: The Definitive Guide*. 3 Books in 1: Machine Learning for Beginners;Artificial Intelligence Business Applications;Artificial Intelligence and Machine Learning for Business. Independently published, 2019.

[tfb] tf-blazepose. https://github.com/vietanhdev/tf-blazepose,Acedido:2021-10-5.

[tfm] tfjs-models. https://github.com/tensorflow/tfjs-models/tree/master/pose-detection,Acedido:2021-11-4.

[tfp18] Real-time human pose estimation in the browser with tensorflow.js, 2018. https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html,Acedido: 2021-11-4.

[TJA21] Justin K. Terry, Mario Jayakumar, and Kusal De Alwis. Statistically significant stopping of neural network training. *CoRR*, abs/2103.01205, 2021.

[use20] Computer vision use cases in sports industry, 2020. https://requestum.com/computer-vision-in-sports,Acedido:2021-6-5.

[YCBL14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27:3320–3328, 2014.

[yog20] yogai, 2020. `https://github.com/cris-maillo/yogAI`, Acedido: 2021-10-21.

[YSGR19] Santosh Kumar Yadav, Amitojdeep Singh, Abhishek Gupta, and Jagdish Lal Raheja. Real-time yoga recognition using deep learning. *Neural Computing and Applications*, 31(12):9349–9361, 2019.

[ZCS+21] Ru-Yuan Zhang, Adrien Chopin, Kengo Shibata, Zhong-Lin Lu, Susanne M. Jaeggi, Martin Buschkuehl, C. Shawn Green, and Daphne Bavelier. Action video game play facilitates "learning to learn". *Communications Biology*, 4(1):1154, 2021.

Contactos:
Universidade de Évora
Escola de Ciências e Tecnologia
Colégio Luis António Verney, Rua Romão Ramalho, nº59
7000 - 671 Évora | Portugal
Tel: (+351) 266 745 371
email: geral@ect.uevora.pt