ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

M

MESTRADO
ENGENHARIA INFORMÁTICA

Deteção de Cyberbullying em Redes Sociais: Uma Abordagem Textual, Visual e Cognitiva

Carlos Daniel de Sousa e Silva

Doutor Ricardo Jorge da Silva Santos

Agradecimentos

A realização deste trabalho apenas foi possível graças ao apoio e contributo de diversas pessoas, às quais deixo os meus sinceros agradecimentos:

Ao professor Ricardo Santos, pelo seu acompanhamento e visão ao longo de todo o trabalho, assim como toda a ajuda prestada e conhecimento transmitido enquanto orientador.

Ao colega Ricardo Barbosa, pela sua essencial colaboração na realização de contribuições científicas, assim como a todos os restantes colegas de curso que contruibiram para que fosse possível atingir esta etapa.

A todas as pessoas que de uma forma ou de outra se mostraram disponíveis para ajudar em qualquer altura, desde amigos a entidades empregadoras.

À minha família, pelo constante apoio e incentivo durante todo o meu percurso académico.

Abstract

The adoption of online social platforms as a common space for the virtualisation of identities is also correlated with the replication of real-world social hazards in the virtual world. With this increase in online presence, some problems like bullying begin to intensify. Cyberbullying is a very common practice among young people nowadays, becoming much more present due to their constant online activity, especially in online social networks This work aims to present a proposal that combines textual, visual and cognitive analysis techniques on the content that is available online, in order to have an intelligent and autonomous system that can identify and classify a possible bullying situation. It intends to analyze the text of an online publication, verify the presence of people in multimedia content and identify individuals involved in a possible case of bullying, trying to mitigate future similar situations.

Keywords: *Cyberbullying; Machine Learning; Online Social Networks; Textual Analysis; Visual Analysis; Cognitive Analysis*

Resumo

A adoção das plataformas sociais online como um espaço comum para a virtualização das identidades está também correlacionada com a replicação de riscos sociais do mundo real no mundo virtual. Com este aumento da presença online, alguns problemas como o *bullying* começam-se a intensificar. *Cyberbullying* é uma prática muito comum entre os jovens, tornando-se muito mais presente devido à sua constante atividade online, especialmente através das redes sociais. Este trabalho tem como objetivo apresentar uma proposta que combine técnicas de análise textual, visual e cognitiva sobre o conteúdo que seja disponibilizado online, de modo a ter um sistema inteligente e autónomo que consiga identificar e classificar uma possível situação de *bullying*. Pretende-se analisar o texto de uma publicação online, verificar a presença de pessoas em conteúdo multimédia e identificar os indivíduos envolvidos num possível caso de *bullying*, procurando mitigar futuras situações semelhantes.

Palavras-chave: *Cyberbullying; Machine Learning;* Redes Sociais; Análise Textual; Análise Visual; Análise Cognitiva

Índice

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

POS – *Part of speech*

ML – Machine Learning

IA – Inteligência Artificial

API – *Application Programing Interface*

IoT – *Internet of Things*

CNN – Rede Neuronal Convolucional

SVM – *Support Vector Machine*

LSF – *Lexical Syntatic Feature-based*

k-NN – k-Nearest Neighbor

RNN – Rede Neuronal Recorrente

NLP – Processamento de Linguagem Natural

PCA – *Principal Component Analysis*

SVD – *Singular Value Decomposition*

SSD – Single Shot Detector

mAP – Mean Average Precision

RPN – Rede de Regiões Propostas

HOG – *Histogram of Oriented Gradients*

Capítulo 1

Introdução

Atualmente a internet está cada vez mais presente na vida de todos, essencialmente pela existência de inúmeras formas de estarmos conectados à mesma em qualquer momento e a partir de qualquer lugar. Além disso, esta existência de infinitas possibilidades de conexão leva a que desde muito cedo os jovens comecem a visitar as diversas plataformas, e a estarem sujeitos aos inúmeros perigos que estas apresentam apenas à distância de um clique. A interação que é possível entre diferentes pessoas leva a que possam surgir práticas de risco para a saúde das pessoas, especialmente no que toca a problemas mentais. Facilmente alguém poderá tentar ferir outros por via de insultos e ameaças, levando a pessoa ao extremo e a fazer com que esta muitas vezes acabe por cometer ações que não seriam as mais indicadas.

Com todo este fluxo de dados, existe uma possibilidade para análise dos mesmos de modo a procurar reduzir este tipo de situações, nada melhor que ter um sistema informático que seja eficaz o suficiente de cobrir diferentes episódios deste género. Os mais recentes desenvolvimentos nas áreas da inteligência artificial e *machine learning* apresentam várias técnicas para a construção de uma ferramenta que possa consultar, recolher e analisar dados deste género, indo de encontro ao combate destes problemas da internet e à sua redução. A intenção de tornar cada vez mais o mundo um pouco mais autónomo, como já acontece com várias aplicações em diferentes contextos, ajuda a perceber que talvez os sistemas inteligentes, que têm capacidade de tomar decisões autónomamente e que conseguem aprender ao longo do tempo com base na sua experiência, sejam a melhor opção para a criação de elementos de combate aos problemas da internet.

Com o presente trabalho espera-se desenvolver um modelo de sistema que seja autónomo e capaz de detetar a prática de *cyberbullying*, um problema que afeta muitos jovens nos dias de hoje, muito por força da sua presença nas redes sociais. Pretende-se também estudar um pouco os detalhes técnicos relativos a algumas ténicas de *machine learning* de forma a optar pelos melhores algoritmos e ferramentas para solucionar as necessidades do problema encontrado.

1.1 Estrutura

O presente documento encontra-se estruturado em 8 capítulos. Neste primeiro, é feita uma pequena introdução ao trabalho desenvolvido. O segundo capítulo detalha os perigos que se

encontram atualmente na internet e apresenta os conceitos base da prática de *cyberbullying*. No capítulo 3, são descritas algumas aplicações existentes no mundo real que foram implementadas com recurso a técnicas e mecanismos de inteligência artificial e *machine learning*. O quarto capítulo foca-se na apresentação dos principais componentes tecnológicos que podem ser usados no combate ao problema que este trabalho procura resolver, assentando-se essencialmente em algoritmos de *machine learning*. O quinto capítulo apresenta o problema identificado ao qual se pretende responder com o desenvolvimento de um modelo de deteção automática de uma situação de *cyberbullying*. O capítulo 6 detalha os principais pontos para a construção do modelo e a respetiva proposta de solução ao problema apresentado no capítulo anterior. Ao longo do capítulo 7, serão apresentados exemplos reais, e a forma como o modelo será capaz de ser aplicado nesses momentos. Para finalizar, no capítulo 8, são tiradas as conclusões relativas ao desenvolvimento deste trabalho e ao trabalho que se pretende fazer no futuro.

Capítulo 2

Perigos da Internet e Cyberbullying

Por via do constante crescimento do uso da internet e dos serviços que esta oferece, surge uma vasta quantidade de oportunidade para o crime na *web*, e como tal, toda a informação que seja apresentada para alertar os utilizadores para estes perigos deve ser tida em consideração. O *cyberbullying* é uma prática muito comum entre jovens tornando-se cada vez mais presente pelo facto de estes estarem constantemente *online*, especialmente por via das redes sociais. Deste modo, este capítulo tem como propósito dar a conhecer melhor este tipo de situações, assim como as técnicas e tecnologias já utilizadas para o seu combate de forma automática e em tempo real.

2.1 Introdução

A constante necessidade de se estar conectado com o resto do mundo, seja por curtas interações periódicas, ou por atividades prolongadas como interagir em plataformas sociais ou conversar com um cliente ou parceiro para concretizar negócios, faz com que seja inacreditável viver sem a internet.

A internet permite realizar um número infinito de tarefas, simplificando a vida de quem a utiliza, tornando-se numa mais valia para sociedade. Contudo, a internet não tem apenas pontos positivos. Além de se estar ao alcance de ataques como vírus ou *phishing*, está-se também sujeito a acreditar em informação enganosa ou a utilizar de forma indevida as redes sociais.

As crianças despendem algum tempo com as novas tecnologias, e ao serem utilizadores de internet de forma não controlada pelos seus pais ou responsáveis pode levar a que estas a usem de forma indevida, acedendo a conteúdo ao qual não deveriam ter acesso. O tempo passado na internet pode levá-las a uma maior distração, ao qual se poderá juntar a falta de prática de atividade física, insónias ou falta de criatividade pelo facto de terem muita informação disponível e recorrerem, por exemplo, ao *copy and paste* para a realização de um trabalho para a escola. Se nos focarmos numa vertente orientada para a utilização de redes sociais como *Facebook, Twitter, Youtube*, etc, entramos num conjunto de problemas como a falta da privacidade, o facto de as pessoas estarem mais habituadas à comunicação através da internet recorrendo a um *chat* poderá causar uma maior dificuldade na comunicação cara a cara, e a serem vítimas de gozo ou insulto, ou seja, *bullying*.

Bullying é uma dinâmica social complexa, motivada essencialmente por diferenças de domínio, capital social ou cultura [1]. O desejo de dominância, de aquisição e manutenção de capital social, são fatores motivadores principais para a iniciação e continuação da prática de *bullying*. Por exemplo, a falta de capital social por parte das vítimas poderá impedi-las de obter uma melhor posição social ou aquisição de determinado bem, o que a poderá levar ao desprezo por parte de outros. Além disso, a denominação utilizada pelos agressores, também conhecidos como *bullies*, para subjugar as vítimas resulta em humilhação intensa que tem efeitos negativos nessas pessoas, como raiva e depressão.

O *cyberbullying*, tal como o *bullying* tradicional, tem um profundo impacto negativo na vítima, especialmente quando se tratam de crianças e jovens, sofrendo significativamente ao nível emocional e psicológico, sendo que alguns casos chegam mesmo a terminar em suicídios trágicos. Então, o *cyberbullying* pode ser descrito como: quando se recorre à internet, a telemóveis ou a outros dispositivos tecnológicos para enviar texto ou imagens com o objetivo de ferir, humilhar ou embaraçar outras pessoas, sendo esta uma prática mais constante do que o *bullying* tradicional [2].

Ao contrário do *spam*, este tipo de ataque é mais pessoal, variado e contextual [3]. As imagens publicadas por um indivíduo numa rede social, o tipo de conteúdo partilhado, as ligações comentadas e a possibilidade da troca fácil de mensagem com qualquer outro utilizador, permitem que a prática de *bullying* seja cada vez mais frequente e constante do que nunca, e um perigo a ter em conta na sociedade.

2.2 Problemas da Internet

Da mesma forma que se encontram benefícios quando se utiliza a internet, tal como acontece em todos os cenários onde o ser humano está envolvido, também são encontrados alguns problemas e identificados certos riscos.

Com esta facilidade de ligação com o mundo que as tecnologias oferecem, a informação enviada que contenha dados privados, muitas vezes disponível através das redes sociais, ou a realização de transações para pagamentos, serão elementos que poderão interessar aos piratas da internet, abrindo assim uma área para o crime online, também conhecido como cibercrime.

Desde logo, o cibercrime estende-se com a existência da *Deep Web*, uma internet que é maior que a internet comum em milhares de unidades, acessível apenas com um browser específico, não regulamentada, sem taxas, e escondida numa pesquisa típica de internet, tendo como principal foco a comercialização de bens ou serviços de forma irregular ou criminosa [4], onde a forma de pagamento é quase exclusivamente através de *Bitcoins*.

Muitos dos problemas atuais encontrados na internet ocorrem através de vírus e ataques como o *phishing*, onde, por exemplo, os criminosos podem exigir um pagamento para enviarem um código para remover um qualquer vírus, ou fazer-se passar por determinada entidade para obter o dinheiro da vítima do ataque. Os chamados *hackers* podem tentar aceder a servidores de empresas para tentar roubar a sua informação para beneficiarem com ela, e em muitos dos casos conseguem obter informações pessoais se acederem a serviços de armazenamento de informação considerada individual, como o caso de ataques a plataformas como o *Dropbox* ou *iCloud*. Por exemplo, em 2014 a *iCloud* da *Apple* foi alvo de um ataque e foram disponibilizadas na internet mais de 500 fotos íntimas de várias celebridades, o que acabou por denegrir a imagem de muitas destas personalidades o que as poderia ter levado a perder protagonismo [5].

Focando novamente nas redes sociais, um dos perigos existentes prende-se com o roubo de identidade, que consiste em alguém capaz de se fazer passar por outra pessoa utilizando os seus dados pessoais ou imagens, ou quando consegue aceder e assumir o controlo de, por exemplo, uma conta de uma rede social de outro [6]. Para combater este tipo de ações, o ideal seria deixar de utilizar apenas uma simples palavra passe que pode ser facilmente divulgada, esquecida ou copiada, e apostar em sistemas de autenticação como reconhecimento facial, sensores biométricos ou até mesmo reconhecimento de voz. O roubo de identidade poderá ser detetado através de comportamentos estranhos por parte do utilizador, como a publicação de conteúdo que não é habitual por parte do utilizador, especialmente se pretender levar ao engano de outros ou expor demasiado alguns aspetos relativos a si [7].

Recentemente foram também divulgadas informações acerca da possibilidade de espionagem que possamos ser alvo ao estarmos conectados pela internet. *Edward Snowden* ex-elemento da *CIA* e da *NSA*, tornou públicos vários detalhes acerca de um programa de espionagem destas organizações, que seria capaz de aceder a qualquer câmara ou microfone de um dispositivo ligado à internet e captar essa imagem e som, sem que o utilizador se apercebesse que estava a ser observado sem consentimento [8]. Desde então é prática cada vez mais comum encontrar-se um autocolante colado nas câmaras dos computadores, de forma a evitar a captura de imagem por qualquer espião da internet.

Os perigos presentes na internet mais difíceis de controlar talvez sejam aqueles aos quais as crianças não deveriam ter acesso. Estas poderão ter acesso a conteúdo que incentive à violência, a imagens que exponham nudez, droga ou armas, ou serem atacadas por um vírus através de uma simples pesquisa no *Google*, do clique num anúncio num *site* de *streaming,* ou pela partilha de *links* nas redes sociais. Este tipo de curiosidade poderá levar a que estas

crianças se envolvam em situações indesejadas, podendo até mesmo aceder a conversas com estranhos por via destes conteúdos, e deste modo, prejudicarem o seu futuro.

Na internet começa a existir a prática de problemas da sociedade já existentes cara a cara, como o caso do *bullying*. O *bullying* consiste no processo de ameaça ou agressão de um indivíduo ou grupo de indivíduos para com outros, normalmente relacionado com alguma característica da sua vida, como por exemplo a sua cultura, e é mais comum entre jovens. Através da internet, apesar de não existir a possibilidade de confronto físico, a facilidade de constante comunicação, poderá levar a que determinado indivíduo seja mais sujeito á ameaça e humilhação pública.

No estudo *Net Children Go Mobile* [9], onde foram estudados os comportamentos das crianças na internet em vários países, são apresentados alguns dados relevantes. Dos riscos presentes na internet para as crianças, entre os 11 e 16 anos, identificaram que 5% dos inquiridos (cerca de 3500 na totalidade) já sofreram *bullying*, e outros 5% receberam mensagens de cariz sexual, no entanto apenas 3% se sentiram incomodados com a situação. Conhecer pessoas novas foi algo que foi reportado por 11% das crianças, sendo que nenhuma se sentiu incomodada por tal facto. Já o contacto com imagens que contenham nudez ou conteúdo pornográfico foi experienciado por 27%, e outro tipo de conteúdos relativos a ódio, automutilação, anorexia e drogas foram alcançados por 10%. O contacto com as tecnologias e a internet começa a verificar-se através da prática de jogos, que em muitos dos casos requerem ligação à internet, o que desde logo poderá permitir a ligação a jogadores desconhecidos e de qualquer parte do planeta. Segue-se a necessidade de recurso à *web* para pesquisa tendo em conta a realização de trabalhos escolares, seguindo- se depois a adesão a redes sociais, principalmente ao *Facebook, Twitter* e *Instagram*. Curiosamente, o *Facebook* apenas permite a criação de conta a partir dos 13 anos de idade [10], excetuando-se Espanha e Coreia do Sul onde a idade mínima para ingressar na rede social americana é de 14 anos, contudo, a validação é feita apenas com recurso à data de nascimento inserida, pelo que quem quiser criar conta poderá inserir informação errada para avançar com o processo.

Numa análise mais detalhada sobre Portugal, mas inserida ainda no âmbito do mesmo estudo [11], os autores identificaram que a idade do primeiro acesso à internet, de forma autónoma, é em média aos 8.6 anos de idade, que o seu primeiro telemóvel é obtido aos 9.2, e o seu primeiro smartphone aos 11.3. Notam ainda que se a criança se deparar com algum conteúdo estranho ao usar a internet primeiro informam a mãe (68%), depois o pai (53%), seguidos de irmãos (36%), amigos (32%) e professores (20%). Estes dados conseguem mostrar que as crianças começam a ter acesso à tecnologia de forma algo prematura, coincidindo maioritariamente com o período em que ainda frequentam o primeiro ciclo do ensino básico.

Recentemente surgiu a notícia de que o governo britânico pretende combater os riscos digitais e deverá convidar empresas como a *Google* ou o *Facebook* a pagar por isso de forma voluntária, através de programas de segurança na *web*, contra o tipo de perigos aqui descritos [12]. Estes tipos de ações seriam necessários para ajudar a evitar enfrentar estas situações, mesmo tendo em conta que nas escolas e na comunicação social se realizam cada vez mais programas para alertar e preparar as pessoas para estes casos, no entanto, mecanismos automáticos podem prevenir que muitas destas situações escalem sem controlo.

2.3 Cyberbullying

Um dos principais problemas da internet nos dias de hoje é a prática de *cyberbullying*, originado a partir do problema social *bullying*. Este problema consiste em atos de violência psicológica repetidos, praticados por um jovem ou grupos de jovens sobre outro, recorrendo às tecnologias, seja por via de aplicações na internet, quer seja diretamente por mensagens de texto ou chamada telefónica.

Ao contrário do *bullying* tradicional, o *bullying* através das tecnologias não leva ao contacto físico, a não ser que os envolvidos se cruzem em paralelo no dia a dia, como por exemplo, na escola. No entanto, tendo em conta que os jovens despendem muito tempo com os seus dispositivos tecnológicos, especialmente para consulta e atualização das suas redes sociais, esta prática de violência torna-se mais constante, mais difícil de identificar, e mais propício à humilhação com um maior número de pessoas alcançadas.

McClowry et al. [13] dividem o *bullying* em dois tipos: direto, que envolve ataques flagrantes contra um jovem alvo; indireto, quando envolve a comunicação com outros sobre o alvo (propagação de rumores). Referem ainda que o *bullying* pode ser físico, verbal ou relacional (excluindo alguém, por exemplo, negando amizade) e pode envolver danos à propriedade. Os rapazes têm tendência a comportamentos de *bullying* mais diretos, ao passo que as raparigas se envolvem mais em atos direcionados para o *bullying* indireto.

Mas o porquê de alguém tomar a iniciativa de atacar outro? Em muitos casos, o *bully* já foi vítima anteriormente, tornando-o numa pessoa mais furiosa e agressiva levando-o a querer "descarregar em alguém", ou então sente-se sozinho e precisa de atenção, tem problemas em casa sendo vítima de abusos físicos ou verbais, tem baixa autoestima e para se sentir melhor tenta rebaixar os outros. Pode também pretender ter mais popularidade e ataca as pessoas das quais sente ciúmes, pode ter um grande ego, achando-se melhor que os outros, e tem em muitos casos um grupo de segurança, para o caso de alguém ripostar, sentindo-se assim seguro [14]. Estes são os principais motivos e características para determinada pessoa recorrer ao *bullying*.

Muitas das vezes estes ataques estão ligados a tópicos sensíveis como: a raça e cultura, a sexualidade, a inteligência, aspeto físico, e sobretudo aspetos que as pessoas não podem mudar sobre elas mesmas [2].

Contudo, por vezes quem é alvo de mensagens ofensivas na internet é também quem as escreve, enviando-as a si próprio, sob pseudónimo. Os motivos variam, desde jovens que o fazem como uma forma de diversão, para verificarem a reação que os seus amigos vão ter quando as visualizarem, ou casos de indivíduos que estão deprimidos e que se querem obrigar a sentir ainda pior. Este comportamento é mais prevalente em adolescentes que não se identificam como heterossexuais e pessoas que tenham sido vítimas de *bullying* no passado. Os rapazes também têm mais probabilidade de enviar mensagens ofensivas a si próprios, normalmente como uma piada ou forma de conseguir a atenção de amigos ou até mesmo interesses amorosos [15].

Nos espaços informativos já começa a ser frequente a publicação de notícias relativas ao tema, o que poderá contribuir para alertar os pais e os jovens para os perigos da internet com especial foco no *bullying*. Num relatório divulgado pela UNICEF em novembro de 2017 [16], é apresentado um gráfico com os países da Europa e América do Norte com mais relatos de *bullying* (Figura 2.3).

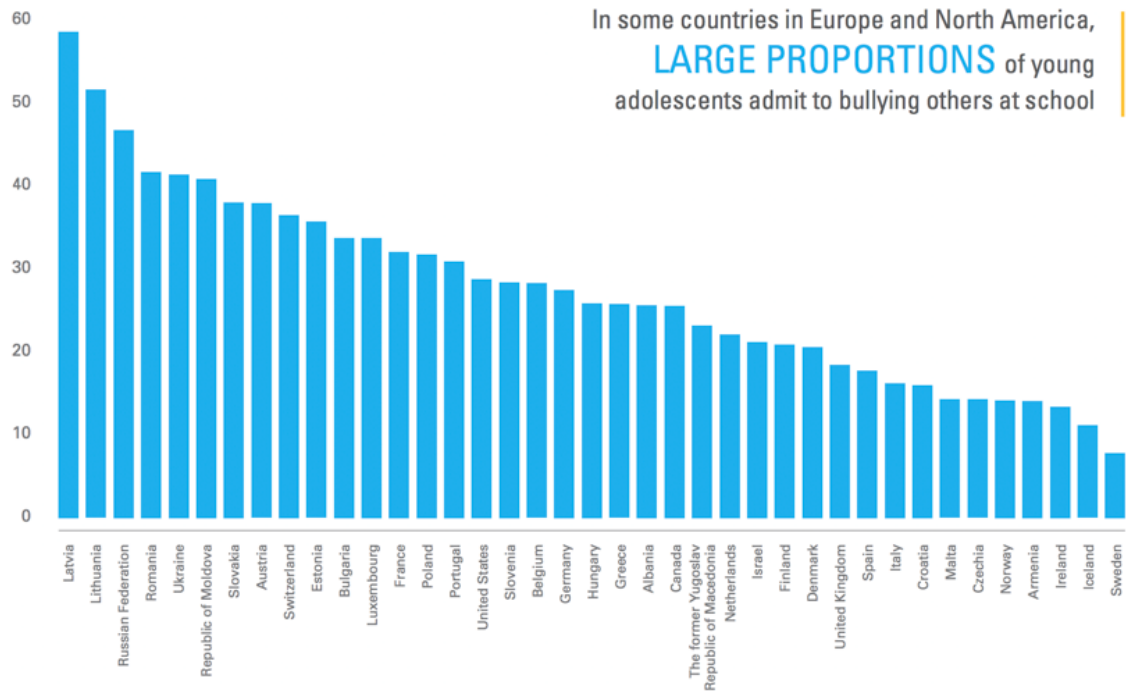

FIGURA 2.3: Percentagem de relatos de *bullying* na escola (2015)

Analisando o gráfico (dados de 2015) identifica-se que Portugal é o 15º país com mais relatos de *bullying* nas regiões apresentadas à data, estando à frente de Estados Unidos, o que pode

ser alarmante, pois é um país onde ocasionalmente ocorrem tiroteios nas escolas. Entre 31% e 40% dos adolescentes portugueses, com idades compreendidas entre os 11 e os 15 anos, disseram ter sido intimidados na escola pelo menos uma vez nos últimos dois meses.

Um artigo publicado pela *The Next Web* [17], pretende alertar para os perigos da internet, com especial foco no *cyberbullying* e no assédio *online*. Referem que a tecnologia fez maravilhas ao conectar pessoas da sociedade, mas infelizmente também tornou o *bullying* mais fácil, anónimo, mais permanente e, por vezes, mais fatal. A diferença entre *cyberbullying* e *cyber harassment* (assédio online) é a idade dos envolvidos. Quando o ameaçador e o ameaçado são ambos menores, é considerada uma situação de *bullying*. Quando ambos são adultos, é considerado assédio. Os motivos, em ambas as situações, variam amplamente, e podem incluir tédio, raiva ou prazer sádico ao prejudicar os outros.

O *cyberbullying* é também muito mais provável que seja realizado por alguém que a vítima conheça bem. As crianças têm sete vezes mais probabilidade de serem atacadas por amigos atuais, anteriores ou interesses românticos, do que por um estranho qualquer. Ao invés, mais de um terço dos adultos assediados na internet não conhecem a pessoa que os está a assediar, e pouco menos de um terço são assediadas por pessoas que escondem as suas identidades. Os estudantes homossexuais têm maior tendência a serem vítimas destes atos, assim como os estudantes não brancos. As raparigas são 2.6 vezes mais propensas a serem vítimas do que os rapazes, e as mulheres têm 2 vezes mais probabilidade de serem assediadas na internet. Nalguns casos, o agressor faz-se passar pela vítima, através de contas falsas, publicando conteúdo de forma a humilhá-la publicamente.

Este tipo de ataque pode parecer fácil de largar, contudo, a chantagem com a publicação de conteúdo que o agressor tenha recebido, nomeadamente informação mais privada ou fotos íntimas, e a ameaça física à família da vítima, pode levar a uma situação mais grave do que aparentava inicialmente. Tendo em conta que estar online é cada vez mais necessário quer seja pelo emprego, quer seja por trabalhos académicos, não permite que desligar apenas o computador seja uma opção para deixar de receber os ataques.

2.4 Trabalhos relacionados com o tema

Não podemos afirmar que este seja um tópico novo, já existem alguns trabalhos que outros investigadores realizaram tendo em vista a deteção e combate ao *cyberbullying*. Neste tópico serão apresentados alguns desses trabalhos de forma a ter uma melhor perceção do que já existe e quais os pontos a melhorar para alcançar uma solução para resolver este problema cada vez mais frequente.

O *cyberbullying é* um problema social sério especialmente entre adolescentes, e está definido como o uso de tecnologias de informação para prejudicar ou assediar outras pessoas de forma deliberada, repetida ou hostil. Com o surgimento das redes sociais, este fenómeno tornou-se mais prevalente. Huang et al. [18] utilizaram um conjunto de publicações do *Twitter* para identificar características sociais e textuais para criar um modelo composto para detetar automaticamente o *cyberbullying*. Foram desenvolvidos grafos relativos à rede social e definido um conjunto de características, de forma a se poder ver o contexto de "eu", "meus amigos" e o relacionamento entre eles, atribuindo pesos às arestas de forma a representar as interações entre os utilizadores. O estudo refere que as vítimas de *cyberbullying* podem ter uma autoestima significativamente mais baixa que o normal, e por isso, podem ser mais ativos nas redes sociais na procura de algo que os faça sentir melhor. Assim, uma das abordagens consiste em avaliar a popularidade e atividade dos utilizadores e o número de publicações entre eles. Do ponto de vista da análise do conteúdo textual, esta abordagem implica a verificação da densidade de recursos linguísticos como palavras insultuosas ("*asshole*", "*bitch*") e hieróglifos ("*5hit*", "*@ss*"), da frequência de letras maiúsculas, do número de pontos de exclamação e interrogação, e do número de *emojis*. É também importante analisar as *POS tags* existentes e procurar detetar texto como "*you are*" ou "*yourself*". Para classificação da informação recolhida, os autores aconselham o recurso ao *Weka* e a múltiplos algoritmos como *J48*, *Naive Bayes*, *BMO*, *Bagging*, e *Dagging*, de forma a conseguir alcançar os melhores resultados.

O estudo *Modeling the Detection of Textual Cyberbullying* [2] foca-se na análise de um conjunto de comentários de vídeos do *Youtube* ligados a tópicos sensíveis como raça e cultura, sexualidade, inteligência ou aspeto físico. Removem-se as *stop words*, as sequências não importantes de caracteres (por exemplo as repetições do último caracter em "*lollll*"), e os *links* para utilizadores (@*username*). Para a classificação de texto são realizadas duas experiências: treino de classificadores binários para verificar se uma instância pode ser classificada para mais do que um tópico sensível; utilização de classificadores multi-classe para classificar uma instância de um conjunto de tópicos sensíveis. Chegou-se à conclusão que os classificadores binários funcionam melhor no problema em questão. As ferramentas utilizadas foram o classificador *Naive Bayes* e *Support Vector Machines (SVM)*, *J48* e *JRip* como métodos de aprendizagem. Ao nível de características definiram a *TF-IDF*, uma medida de importância de uma palavra num documento tendo em conta a sua frequência ao longo do mesmo.

Os autores do estudo recorreram ainda a um léxico de forma a ficarem com uma lista de palavras que denotam efeito negativo, procurando também detetar *POS tags*, especialmente *bigrams* do tipo "*you are*" ou "*yourself*". No final, concluiu-se que as frases mais difíceis de

detetar foram aquelas que continham sarcasmo ou ironia, até porque estas normalmente não contêm as palavras negativas que se procuram para identificar o insulto. Como trabalho futuro, é indicada a pretensão de analisar o contexto da conversa e a resposta a comentários.

As dificuldades apresentadas no último estudo foram novamente expostas no trabalho *Detecting Offensive Language in Social Media to Protect Adolescent Online Safety* [19], onde os autores indicam que as abordagens textuais existentes à data da publicação não seriam capazes de detetar a frase "*you're a such crying baby*", pelo facto de esta não conter palavras incluídas em léxicos constituídos por conteúdo ofensivo. Outro problema associado é o facto de uma palavra poder ter vários significados. Foi proposta a criação de um LSF (*Lexical syntatic feature-based*), que permite avaliar o texto e os perfis de quem o escreve. Para isso, tem de se considerar linguagem ofensiva como expressões rudes ou grosseiras, expressões com teor sexual, e texto relacionado com temas como raça, religião, nacionalidade, entre outros. Algumas páginas *web* substituem partes de palavras grosseiras com asteriscos (*), o que por vezes não funciona como esperado, se este estiver escrito em idiomas diferentes daquele em que é efetuada a verificação. Por exemplo, no caso dessa verificação estar a ser feita em inglês, a palavra "assim", passará a ser apresentada como "***im", ou seja, neste caso estamos perante um falso positivo. O *Facebook* permite ao utilizador definir que palavras este considera como sendo rudes, de modo a que comentários com esse conteúdo lhe sejam apresentados como *spam* no futuro. Além de se detetar a ofensa ao nível das palavras, procura-se também detetá-la ao nível do utilizador, através da análise do seu histórico de conversas e respetivo contexto, procurando saber se é habitual o utilizador publicar tal conteúdo. Para as tarefas de classificação os autores optaram por recorrer aos mecanismos *Naive Bayes* e *Support Vector Machines*.

Rynolds, Kontostathis e Edwards [20], apresentam uma abordagem para detetar a prática de *cyberbullying* utilizando técnicas de *machine learning*. Para tal, foi criado um *dataset* com dados das publicações e dos utilizadores do *Formspring.me*, um *site* de *Q&A* muito frequentado por jovens, onde é fácil encontrar conteúdo relativo a *bullying*, muito por causa de ser possível publicar e responder a comentários através de perfis anónimos. Foi efetuado um processo de classificação manual com recurso ao *Amazon mechanical turk*, uma plataforma que permite que se coloquem tarefas e que alguém as realize por um custo reduzido, neste caso, 0.05 cêntimos por cada frase analisada. Especificamente neste estudo, foram apresentadas as questões: "a publicação contém *cyberbullying*?", "de 1 a 10 quão agressivo é? (0 caso não seja *bullying*)", "que frases ou palavras são indicativas de *bullying*?". A decisão final de cada análise seria aquela que no mínimo tivesse dois avaliadores de acordo. Os insultos identificados na classificação manual foram depois analisados em número (análise NUM) e em densidade (análise NORM), dividindo-os por 5 níveis de agressividade,

assim como o nível médio de agressividade da frase, com recurso ao *Weka*, *J48*, *JRip*, *IBK* e *SMO*. Fazendo a validação com dez repetições em cada ferramenta, concluiu-se que o *J48* e o *IBK* foram aqueles que alcançaram maior percentagem de acerto em ambas as análises.

O recurso a imagens para auxílio na deteção de *cyberbullying* poderá ser uma mais valia, tendo em conta que após a publicação de uma foto por parte de um indivíduo numa rede social, todos aqueles a quem este está ligado poderão responder com um insulto ou uma provocação. Além disso, a publicação de fotografias íntimas de outros indivíduos de forma a tentar humilhá-los em praça pública é outra das preocupações a ter em conta neste tipo de estudo. Lightbody et al. [3] afirmaram que a combinação de análise de sentimento com técnicas de processamento de imagens deve ser considerada uma plataforma apropriada para categorizar conotações textuais e visuais do conteúdo.

Com isto, os autores pretendem mostrar que não é apenas através de texto que o ataque poderá ser feito, pois o texto ofensivo poderá ser apresentado dentro de uma imagem, ou então, a ofensa pode ser procurada através da edição de uma fotografia. Foi referido que as imagens mais relevantes serão aquelas que possam conter nudez, evidências de edição e texto dentro da imagem. A existência de texto relacionado com a imagem ajuda a determinar o risco de negatividade do conteúdo e a categoria associada.

Como se pode ver na Figura 2.4, o objetivo do modelo apresentado pelos autores neste estudo passa por identificar o risco de existir qualquer tipo de conteúdo negativo nas imagens de uma rede social, e no caso de serem detetados perigos, alertar os pais de forma célere. A abordagem apresentada procurava identificar a presença de texto e fazer a sua análise de sentimento, tentando ainda detetar a presença do corpo humano nas imagens e, por exemplo, identificar o seu tom de pele. Caso fosse determinado que uma imagem era suscetível de ter alto risco, os pais deveriam ser imediatamente informados por MMS, e caso o risco fosse considerado moderado, o alerta seria efetuado por email. Não tendo sido identificado nada de relevante, proceder-se-ia para a análise da imagem seguinte sem despoletar qualquer alerta.

Outro dado a ter em conta para a análise da existência de *cyberbullying* é o género do ser humano, tal como nos foi apresentado pelos autores do artigo *Improved Cyberbullying Detection Using Gender Information* [21]. Foi referido que existem diferenças nas formas em como os rapazes e as raparigas fazem *bullying*. O género feminino tem tendência a usar um estilo de agressão mais relacional, por exemplo excluindo alguém de um grupo, e usando maioritariamente pronomes como "*I*", "*you*", "*she*". Já os rapazes recorrem a palavras mais grosseiras e expressões mais ofensivas, sendo que a nível de pronomes, os mais utilizados são "*a*", "*the*", "*that*". Construiu-se um classificador com uma *SVM* usando o *Weka* para analisar um *dataset* formado por 381000 publicações obtidas da rede social *myspace*, onde

34% foram disponibilizadas por mulheres e 66% por homens. Concluiu-se que os resultados obtidos foram melhores nos homens, talvez pela presença em maior número no *dataset* e pelo maior uso de palavras agressivas.

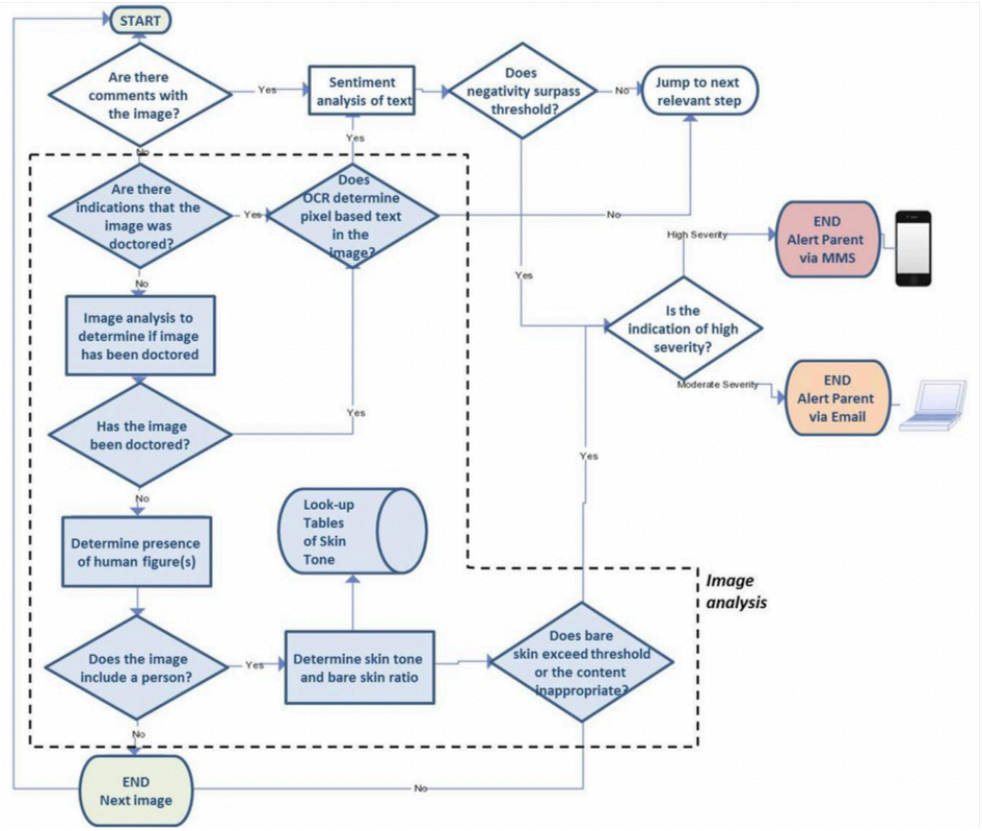

FIGURA 2.4: Modelo de deteção de risco de conteúdo negativo numa imagem

Zhao et al. apresentaram [22] um mecanismo para deteção automática de *cyberbullying* nas redes sociais através de um conjunto de características de *bullying* definidas. Primeiro, foi composta uma lista de palavras insultuosas. Depois, com base em *word embeddings*, estenderam esses recursos linguísticos para fazer a definição das características de *bullying* para a análise. Foram atribuídos diferentes pesos a cada característica com base na semelhança entre os *word embeddings*, concatenando-as posteriormente com características retiradas do *bag-of-words* e com características semânticas latentes para formar a representação vetorial final, com recurso ao *word2vec*. A lista de insultos utilizada era composta por 350 palavras que apresentam o insulto ou emoções negativas ("*nigga*", "*bitch*", "*fuck*", etc.). Utilizando os *word embeddings* foi verificada a similaridade entre palavras, por exemplo "*beef*" e "*pork*", através de pesos atribuídos a cada uma delas. Quando a representação final de cada mensagem da internet era obtida, utilizava-se um classificador linear com recurso a *SVM* para detetar a existência de *cyberbullying*. O *dataset* para teste foi composto por *tweets* que continham pelo menos uma das seguintes palavras "*bully*", "*bullied*" ou "*bullying*".

Estes são alguns dos trabalhos já elaborados tendo em vista o combate ao *cyberbullying* com recurso a mecanismos automáticos, tentando alcançar o objetivo de evitar situações com consequências mais graves e reduzir gradualmente esta problemática. A maioria dos estudos apresentados tem por base a análise textual, sendo que também foi possível relatar métodos que propõe o recurso a análise de imagens para combater este problema que está presente em grande número na internet.

2.5 Conclusão

Com este capítulo e com a pesquisa de trabalhos relacionados, foi possível ficar a conhecer um pouco melhor os perigos presentes na internet, com especial foco no *cyberbullying*. Este tipo de problemas é cada vez mais comum, muito por força da existência das redes sociais e da sua utilização por parte das crianças e jovens, pelo que, o presente trabalho pretende identificar as suas características principais de forma a ir de encontro a uma solução que seja capaz de reduzir este tipo de ataques e evitar tragédias que possam acontecer na sequência dos mesmos.

Foi possível ficar a conhecer quais os métodos utilizados pelos agressores para ameaçar as suas vítimas, o porquê de partirem para este tipo de ações, e de que formas podem terminar este tipo de casos. Através das abordagens aqui descritas conseguem-se identificar as principais palavras ou frases utilizadas para tentar ferir os outros, pelo que poder-se-ão reaproveitar algumas ideias para desenvolver uma solução.

Capítulo 3

Evolução da Inteligência Artificial

Há vários anos, sonhava-se com a possibilidade de as máquinas serem o mais parecidas com os seres humanos, e essencialmente através da inclusão de algumas ideias em filmes, era possível verificar o que se ambicionava alcançar com o desenvolvimento da tecnologia. Ora, pois, esse tempo chegou, e a inteligência artificial faz parte da presente onda de inovação, trazendo grandes mudanças na forma de relacionamento entre as pessoas e a tecnologia, levando a algumas alterações no comportamento da sociedade. A evolução das ferramentas com a introdução de capacidades, até então apenas humanas, é constante, e temos disponíveis vários mecanismos que nos conseguem imitar ou até mesmo substituir. A inteligência artificial aliada a *machine learning*, onde os sistemas possuem a capacidade de aprender sozinhos com base na sua experiência leva-nos para onde sempre quisemos chegar, de forma mais rápida, intuitiva, inteligente e com menor erro. O presente capítulo apresenta um pouco da evolução da inteligência artificial, com foco nas suas características, descrevendo ainda de que forma *machine learning* é importante para o desenvolvimento de mecanismos autónomos inteligentes.

3.1 Introdução

O mundo está em constante transformação, e a história conta-nos que o que ontem era novidade, hoje já é passado. O homem foi capaz de descobrir os elementos que seriam a alavanca para a construção de tudo aquilo que o poderia auxiliar na realização das suas tarefas, até chegarmos ao mundo como o conhecemos atualmente. Um mundo dominado pelas novas tecnologias, pela conectividade entre si, e pela capacidade de desenvolvimento e inovação que são cada vez mais velozes e conseguem ultrapassar todas as barreiras que surgem nos problemas que se propõe a resolver.

Ao nosso dispor temos um vasto leque de soluções para as dificuldades que nos surgem a cada segundo, seja uma aplicação para o telemóvel, um robô ou um *gadget*. A revolução industrial ficou marcada pelo surgimento de máquinas que tinham como objetivo auxiliar o trabalho do homem com a sua força, permitindo por exemplo, o trabalho com peças da engenharia constituídas por dimensões que o homem por si só não conseguiria transportar e movimentar [23].

Quando o ser humano realiza algum tipo de atividade, normalmente efetua, em simultâneo, esforços adicionais com o objetivo de melhorar o resultado da sua execução, ou seja, a sua performance na resolução de quaisquer tarefas está inseparavelmente relacionada com um processo de aprendizagem [24]. Além disso, uma pessoa não consegue estar concentrada e a produzir sempre ao mesmo ritmo, necessitando por vezes de fazer curtas pausas para recuperar energia, para se alimentar ou para satisfazer outras necessidades primárias. Por sua vez, o computador é apenas um executante de processos fornecidos pelo humano, podendo realizá-los de forma contínua.

Inteligência artificial e *machine learning* são dois temas cada vez mais populares. Uma grande parte do recente desenvolvimento e inovação tecnológica tem o seu foco virado para estas áreas com o propósito de dar às máquinas comportamentos inteligentes e autónomos, procurando oferecer-lhes capacidades para resolverem problemas mais semelhantes àqueles com que o homem se depara, que englobem diferentes tomadas de decisão e que, aprendendo ao longo do tempo, sejam melhores a cada dia um pouco à semelhança do que acontece com as pessoas.

Aos poucos, estas máquinas inteligentes começam a marcar uma maior presença na nossa sociedade, onde é cada vez mais normal encontrar alguém a falar sozinho com um *bot*, onde os computadores conseguem vencer os campeões mundiais em diversos jogos de tabuleiro, como o caso do xadrez, e até já existem carros que são capazes de conduzir sozinhos [25]. Tudo isto está a ser possível graças ao desenvolvimento que se tem feito nos campos da inteligência artificial e *machine learning*, indo ao encontro de uma sociedade mais automatizada, simplificada e com menor necessidade de intervenção da mão humana em muitas tarefas que sempre haviam sido realizadas manualmente.

3.2 Características Principais da IA

A inteligência artificial (IA) é a ciência relacionada com as máquinas que dispõem de capacidades de pensamento, sendo este um assunto de grande interesse público atualmente. Se é verdade que o avanço das tecnologias mais comuns está muito em voga, também é verdade que o desenvolvimento e a inovação por via da inteligência artificial está a ter um grande foco atraindo todas as atenções para si.

Para criar uma máquina que consiga “pensar”, é primeiramente necessário ter noção do que é o pensamento, para que se consiga entender um pouco melhor de que forma trabalha a mente humana, de modo a tentar replicá-la na máquina. A inteligência artificial é então uma abordagem para a compreensão de comportamentos, baseando-se no pressuposto de que a

inteligência pode ser analisada de uma forma melhor quando esta é simulada através de um computador [26].

O objetivo desta ciência passa por entender a inteligência humana e com isso produzir máquinas úteis para auxiliar o homem nas suas tarefas. Os pioneiros na sua investigação tinham o interesse em dar às máquinas a capacidade de possuírem comportamentos considerados inteligentes, como resolução de problemas, jogar xadrez ou provar teoremas em geometria e calcular os seus predicados [26].

O desenvolvimento realizado consegue apresentar um conjunto de vantagens aquando da comparação na realização de tarefas entre as máquinas e os humanos que, especialmente a nível empresarial, é algo que é tido em consideração. Estas máquinas não têm a necessidade de efetuar pausas, podendo trabalhar por diversas horas, mantendo uma performance constante e estando menos sujeitas ao erro, sendo que em alguns casos, até podem prever uma falha a tempo e travá-la, assumindo ainda que em tarefas de risco para a saúde e segurança, a máquina não necessitará de seguir as mesmas precauções que o humano precisa de tomar [27].

Os assistentes digitais permitem também às empresas reduzir custos com recursos humanos, interagindo estes com os clientes em sua substituição, da mesma forma que um humano o faria. A utilização de carros que conduzam sozinhos, permite que o condutor reaproveite o tempo de viagem para trabalhar ou descansar, algo que não seria possível fazer caso estivesse a controlar a viatura. Apesar destes pontos positivos, também têm de ser tidas em conta algumas desvantagens. Por exemplo, se as máquinas começarem a ter um desempenho superior ao do homem nas tarefas que lhe dizem respeito na sua profissão, é possível que, mais tarde ou mais cedo, a máquina o venha a substituir totalmente, algo que poderá levar ao aumento do desemprego. Além do mais, nem todas as empresas têm capacidade para optar por estes sistemas por serem bastante caros, quer no seu valor para aquisição, quer na sua manutenção e reparação. Outra desvantagem para as máquinas é o facto de, ao contrário dos humanos, não terem criatividade e não serem capazes de inovar no trabalho que realizam, o que as torna limitadas e dependentes de atualizações que possam melhorar as suas capacidades [27].

A inteligência artificial que é conhecida hoje pode ser denominada por IA fraca, tendo em conta que é destinada a executar uma tarefa limitada, por exemplo para fazer uma pesquisa na internet. Contudo, um objetivo a longo prazo para muitos investigadores na área passa por criar uma IA mais geral, que seja denominada por IA forte, que seja capaz de realizar quase todas as tarefas cognitivas que o ser humano consegue fazer, e deixar de se limitar a uma tarefa específica como resolver uma equação ou jogar um jogo [28]. Aos poucos está a ser

feita a ligação entre estes dois tipos de IA, pois estão já a ser desenvolvidos sistemas com capacidades cognitivas mais complexas que já permitem, por exemplo, ver carros a efetuarem viagens sem precisarem de condutor.

Um dos pontos que gera maior perturbação nesta área de investigação prende-se com a questão da segurança. Se se pretende que estes sistemas inteligentes estejam habilitados para fazerem tarefas que possam ter implicação na saúde, tem de ter total confiança de que estes não irão falhar, no entanto, o risco de as máquinas serem *hackeadas* é algo que gera preocupação. Além disso, um ponto muito discutido prende-se com a possibilidade de se conseguir evoluir de tal forma as capacidades cognitivas das máquinas, que os sistemas até comecem a ser melhores na execução dessas tarefas do que os seres humanos. Deste modo corre-se o risco de estes poderem assumir o controlo do planeta, algo que até dá aso a muitos argumentos para filmes. Apesar das máquinas poderem alcançar tal nível de inteligência, não é expectável que estas consigam ter emoções como amor ou ódio, pelo que partir para ações positivas ou negativas nunca será feito de forma intencional. Todavia, alguns sistemas poderão ser programados pelas mãos erradas, podendo por exemplo, serem desenvolvidas armas autónomas programadas para matar e concebidas para serem extremamente difíceis de desligar, podendo levar a uma guerra de inteligência artificial que atinja a população mundial [29]. Em adição, um sistema pode ser programado para fazer algo benéfico, mas pode ao mesmo tempo desenvolver, de forma não prevista, um método destrutivo para atingir o seu objetivo ao realizar uma tarefa. Pegando no exemplo do carro autónomo, que ao receber uma ordem para “levar-me até à estação, o mais rápido possível”, em vez de conduzir a pessoa pelo caminho que demore menos tempo, interprete que se está numa situação de risco, como uma perseguição, e efetue a viagem correndo vários riscos, como optar por rotas impossíveis, ultrapassar limites de velocidade e consequentemente aumentar a probabilidade de acidente por não cumprir as regras de trânsito [28].

Analisando todas estas situações, o caminho a seguir na investigação sobre inteligência artificial tem todas as condições para ser recheado de sucesso e um fator essencial para o desenvolvimento da sociedade.

3.3 Importância de Machine Learning

Com a quantidade de tráfego de informação e facilidade de acesso aos dispositivos, os sistemas armazenam uma quantidade de dados cada vez maior. Isto ocorre especialmente porque é fácil e barato comprar mais memória para armazenar tanta informação [30]. Tendo disponível tamanha quantidade de informação, o objetivo seguinte passa por identificar padrões nos dados de forma a ser possível extrair conhecimento dos mesmos, para

posteriormente ser criada uma base capaz de dar suporte à tomada de decisão que permita criar equipamentos capazes de aprenderem com a experiência ao longo do tempo.

Machine Learning é a aplicação da inteligência artificial que permite precisamente o desenvolvimento de sistemas que, com acesso aos dados de um determinado problema e mediante as instruções inicialmente inseridas, sejam capazes de irem aprendendo de forma autónoma ao longo do tempo através das suas execuções, podendo ser utilizada em diferentes contextos, como tomada de decisão, classificação, reconhecimento de sinais sensoriais, resolução de problemas, execução de tarefas, controlo ou planeamento. Por exemplo, numa aplicação que consiga ler e analisar texto, o sistema pode identificar se a pessoa que o escreveu está a apresentar uma queixa ou a felicitar alguém, e melhorar essa sua previsão à medida que vai enfrentando situações idênticas [31].

A principal vantagem do recurso a *machine learning* é que esta consegue resolver problemas reais complexos de forma robusta, dependendo de dados verdadeiros e não apenas de pura intuição, e ser capaz de se adaptar a novas situações à medida que o volume desses dados aumenta. Praticamente todos os problemas de aprendizagem podem ser formulados como mapeamentos (complexos) entre *inputs* e *outputs*, para se tentar aprender qual o melhor *output* que se consegue produzir para cada *input* dado [32].

M*achine learning* pode ser dividida em dois tipos principais: *supervised learning* e *unsupervised learning*. O primeiro é quando se pretendem usar os *inputs* para prever os valores dos *outputs*, por via de um algoritmo que aprende a partir de um *dataset* de treino. Já o segundo diz respeito a quando existem apenas valores de *input* e não se conhecem quaisquer valores de *output* correspondentes. Aqui, o objetivo passa por modelar a estrutura ou distribuição subjacente aos dados através de uma medida de qualidade, de modo a aprender mais sobre os mesmos [33]. Um bom modelo de aprendizagem é aquele que se generaliza bem para novos dados, ou seja, é capaz de se abstrair através da sua experiência de forma a detetar padrões subjacentes. A conceção e teste desses modelos é uma parte crucial na resolução de problemas com recurso a técnicas de *machine learning*.

As aplicações de processamento de linguagem natural (NLP) têm como principal objetivo entender a comunicação humana, seja por via da escrita ou da fala. Com a forte presença de dados disponíveis na internet e o constante aumento dos mesmos, especialmente através de conteúdo postado nas redes sociais, é fácil encontrar situações em que a análise textual possa fazer sentido, tal como para a deteção de *bullying*, de *clickbait* ou de conteúdo falso. Trabalhar os dados e colocá-los na estrutura pretendida poderá iniciar uma tarefa de classificação para decidir a que categoria um texto poderá pertencer. O objetivo da classificação é organizar e categorizar dados em classes distintas, classes essas que existem

em número finito. Um sistema de classificação desenvolvido com recurso a *machine learning* deverá ter uma base de conhecimento com dimensão e qualidade suficiente para dar suporte à decisão tomada, e, à medida que for executando mais análises, alargar essa mesma base e ser capaz de aprender ao mesmo tempo, pois quanto maior a frequência ou sequência de determinadas palavras ou frases, mais fácil será para classificar situações idênticas no futuro [34].

Para a construção de um modelo de classificação é necessário escolher o método adequado. Alguns dos métodos de classificação mais utilizados são conhecidos como árvores de decisão, onde cada nó implementa o teste a um atributo, cada ramo diz respeito a um valor para o atributo testado e cada folha atribui uma classificação; *support vector machines*, modelos de aprendizagem supervisionados que analisam os dados e reconhecem padrões muito por via de regressão linear [33]; teorema de *Bayes*, capaz de prever a probabilidade de cada registo pertencer a uma determinada classe, assumindo total independência entre atributos.

Para melhor se entender como funciona um sistema desenvolvido com recurso a *machine learning*, vamos imaginar um jogo de xadrez. O sistema deve ser programado de forma a que este conheça as regras do jogo. Seguidamente pode-se optar por definir um conjunto de partidas num *dataset* ou por colocar o sistema logo a jogar uma inúmera quantidade de partidas, de forma a treinar com esses dados para identificar quais serão os movimentos adequados nas diferentes situações [35]. O processo de treino pode ser visto como a fase em que a prática aumenta a experiência e o conhecimento, tal como acontece com os humanos, algo que neste caso em concreto permite que a qualidade do jogo da máquina aumente. Ao desenvolver e treinar o sistema, pode-se providenciar um mecanismo de pontos, e assim, quando a máquina ganha ou faz uma jogada positiva ser-lhe-ão atribuídos pontos, e da mesma forma ser-lhe-ão retirados quando o oposto suceder, sendo esta uma forma de aprendizagem para quais os movimentos adequados para o jogo que disputa. Neste tipo de jogo, o sistema deverá encontrar a melhor jogada (*output*) a partir da posição das peças no tabuleiro (*input*).

Machine Learning certamente será a melhor escolha para o desenvolvimento de sistemas que se proponham a executar tarefas de forma automática e inteligente, pois ao longo do tempo, estes irão melhorar cada vez mais a sua performance por via da sua experiência, que aumentará sucessivamente à medida que for enfrentando e resolvendo problemas.

3.4 Aplicações Práticas

São já várias as soluções que foram implementadas com recurso a técnicas de inteligência artificial combinadas com *machine learning* e que são mais uma demonstração da evolução que se tem vindo a notar nesta área.

A empresa americana *Tesla*, que produz veículos elétricos, tem já algumas versões de carros que têm a capacidade de conduzir sozinhos, sem que seja necessária a intervenção do humano. Através das diversas câmeras e sensores que os veículos possuem, o sistema inteligente instalado em cada carro analisa o que o rodeia em tempo real, de forma a se dirigir para o destino de forma segura, cumprindo as regras de trânsito e estando com atenção a situações adversas que possam surgir, especialmente por parte de ações de outros veículos ou peões [36]. A *Google* também aposta nesta área, com um projeto conhecido como *Waymo* [37], que diz respeito a um sistema implementado nos seus veículos muito semelhante ao que a *Tesla* possui. Em ambos os casos, o principal objetivo passa por reduzir a sinistralidade rodoviária por via dos acidentes que ocorrem nas estradas, em muito dos casos por desrespeito das regras, condução sobre o efeito de álcool, pelo uso do telemóvel ou por sonolência do condutor. Quando se tiver 100% de confiança num sistema destes, o condutor poderá, por exemplo, aproveitar os tempos de deslocação para descansar ou trabalhar, tal como atualmente o pode fazer num transporte público. Quanto ao nível mais técnico destes avanços tecnológicos não se conhece muito, sabendo-se apenas que a *Tesla* recorre a redes neuronais juntamente com a plataforma de computação para IA e *deep learning* da *Nvidia* [38] para reconhecimento dos padrões nas imagens e de sons para dar ordens ao veículo e este efetuar as suas ações.

A *Apple* tem implementado nos seus sistemas um assistente pessoal inteligente, de seu nome *Siri* [39]. A *Siri* é um *bot* com o qual se pode falar para pedir que este execute uma tarefa de modo a que não sejam necessários sucessivos cliques para a iniciar. Basta dizer a frase definida para que o *bot* seja ativado e depois pedir para, por exemplo, verificar como vai estar o tempo no dia seguinte. Com a quantidade de pessoas que utilizam este tipo de aplicação diariamente, os dados existentes aumentam em larga escala, e o sistema torna-se cada vez melhor naquilo que faz [40]. A *Google* e a *Microsoft* lançaram também os seus assistentes pessoais, o *Google Now* [41] e a *Cortana* [42], onde o seu propósito é em tudo semelhante à *Siri*. A principal dificuldade neste tipo de assistentes passa pelo reconhecimento do discurso, muito por via das diferentes pronúncias e idiomas, e pelo facto de se poder estar a referir à mesma coisa de formas diferentes. As tarefas que hoje conseguem realizar são consideradas simples, sendo que ainda não são capazes de tratar de operações mais complexas como agendar uma consulta no médico ou reservar um bilhete de avião [43].

Outro grande avanço na área está relacionado com robôs. Provavelmente na maioria dos casos em que se fale nesta ciência, a primeira imagem que venha à mente seja a de um robô que tenha ações muito idênticas ao homem. O robô *Sophia* [44] foi desenvolvido para ser o mais parecido com o ser humano, aprendendo e aumentando a experiência através das interações com as pessoas e objetos. Visto ter uma aparência muito próxima da realidade humana, à qual se junta um reportório de expressões faciais, faz com que este consiga simular o ser humano de uma forma cada vez mais realista. As capacidades deste robô levaram a que fosse o primeiro no mundo a receber o certificado de cidadão de um país, no caso a Arábia Saudita em outubro de 2017 [45]. Isto é um marco na história que inicia o fim de uma era de máquinas com mecanismos ruidosos e começo de uma era de máquinas que utilizam essencialmente o seu poder cognitivo. *Sophia* tem uma cara feminina, com câmeras colocadas nos olhos para consegui reconhecer caras que viu anteriormente, o que lhe permite cumprimentar alguém pelo nome. A sua face é composta por um silicone especial que é flexível e permite mostrar 62 expressões faciais que mostram sensações de alegria, nervos ou tristeza. Tem também um sistema de voz que lhe dá a possibilidade de comunicar, gesticulando tal como uma pessoa verdadeira. O facto já referido de ir aprendendo com a sua experiência, faz com que cada vez mais se sinta familiar com a cultura, emoções e estilos linguísticos dos seus interlocutores. Além disto, tem a possibilidade de fazer uma pesquisa na internet se alguém lhe colocar uma questão sobre um determinado tópico, e assim saber responder à mesma.

Um robô com estas características físicas e cognitivas poderá levar a que no futuro estes consigam substituir o humano em várias situações. Além disso, os robôs podem ser utilizados para simular situações reais, como por exemplo entrevistas de emprego, algo que atualmente existe apenas de forma virtual, para ajudar as pessoas a treinarem o seu discurso [46].

Estes são alguns dos principais exemplos das implementações que se conseguiram concretizar graças à existência da inteligência artificial, e como se pode verificar, o caminho a seguir permitirá o desenvolvimento de ainda mais mecanismos que venham a ser bastante úteis no dia-a-dia, oferecendo a possibilidade de paralelizar tarefas e conseguir aumentar a produtividade, disponibilidade e, ao mesmo tempo, a saúde.

3.5 Conclusão

Com a informação apresentada neste capítulo é possível identificar de que modo a inovação e o desenvolvimento no campo da inteligência artificial têm sido importantes para aumentar o leque de opções para auxiliarem o ser humano na realização de tarefas, que, até então, apenas se imaginariam possíveis de realizar de forma manual. Os exemplos apresentados

são já provas de que, à partida, se poderá confiar nas máquinas, e assim, delegar-lhes um conjunto vasto de tarefas, para substituírem ou auxiliarem o ser humano sempre que necessário.

Com recurso a técnicas de *machine learning,* os sistemas ficam a ter a capacidade de serem autónomos e melhorarem os seus desempenhos à medida que vão enfrentando um maior número de situações, tal como acontece com as pessoas, identificando assim, erros efetuados previamente, utilizando-os como pontos de aprendizagem.

Os sistemas existentes têm já uma boa base de suporte, o que poderá permitir o desenvolvimento de outros novos mecanismos que venham a ter grande influência em áreas como a saúde, por exemplo, para a deteção e prevenção de doenças. Existe ainda a preocupação alusiva à segurança que estes sistemas possam ter, imaginando o uso negativo que alguém mal intencionado lhe poderá inferir, assim como de que forma a evolução seria capaz de os levar até uma superiorização em relação ao ser humano.

Capítulo 4

Algoritmos de Machine Learning

Ao longo deste capítulo serão apresentados alguns dos algoritmos mais comuns para o desenvolvimento de sistemas autónomos e inteligentes característicos de *machine learning*. Com o conteúdo aqui presente pretende-se que se consiga identificar qual ou quais os algoritmos ideais para o desenvolvimento de um sistema inteligente que seja capaz de solucionar os problemas do dia a dia. Os algoritmos de *machine learning* dividem-se essencialmente entre dois grupos principais: *supervised learning* e *unsupervised learning*. Além destes, é ainda importante dar destaque aos conceitos de *deep learning* e *reinforcement learning.*

4.1 Supervised Learning

Supervised learning diz respeito ao grupo onde o processo de aprendizagem dos algoritmos é feito a partir dos dados de treino, funcionando quase com um professor a supervisionar o processo de aprendizagem dos alunos [47]. As suas principais tarefas dividem-se em regressão e classificação.

O principal objetivo de *supervised learning* é prever o *output* Y com a maior precisão possível quando são dados novos exemplos onde o *input* X é conhecido. Consegue fazer a previsão após ser treinado com um algoritmo e um *dataset* para o efeito (composto por *labeled training data* – dados já assinalados com as identificações da categoria correta a que pertencem), de forma a identificar padrões nos dados, sendo capaz de formar heurísticas. Assim, aplica-se o modelo desenvolvido a novos dados para calcular o *output* Y. Os atributos que são relevantes para calcular o resultado final da previsão são conhecidos por características (*features)* e podem ser numéricos ou categóricos.

Os dados podem ser divididos entre um *dataset* de treino e um de teste. O conjunto de treino tem os dados assinalados com as identificações para o modelo poder aprender através desses exemplos. Por sua vez, o conjunto de teste não tem identificações associadas de forma a que o modelo as tente prever pela primeira vez. Além disso, os valores de entrada devem ser diferentes dos existentes no conjunto de treino, de forma a evitar alguns problemas que serão apresentados ao longo deste capítulo, como o *overfitting*.

Nos problemas de *supervised learning,* começa-se com o *dataset* que contém exemplos de treino com as identificações corretas associadas. Por exemplo, um algoritmo de *supervised*

learning, quando aprende a classificar números manuscritos, possui milhares de fotografias de números escritos à mão, devidamente acompanhadas pelas identificações que indicam qual número que está representado na imagem. O algoritmo vai depois aprender a relação entre as imagens e os números associados, e aplicá-la para classificar novas imagens, que ainda não tenham identificação e que a máquina nunca tenha visto anteriormente.

4.1.1 Regressão Linear

A regressão tem como objetivo prever uma variável Y continuamente com base nos dados de *input* X, ou seja, o objetivo passa por aprender um modelo linear que possa ser usado para prever um novo valor para Y, dado um valor de X não conhecido anteriormente [48].

A regressão linear é um método paramétrico, ou seja, procura ver de que forma X ajuda a explicar o valor de Y. O modelo para prever Y é dado por:

$$\gamma = \beta_0 + \beta_1 * \chi + \in \tag{4.1.1}$$

β_0 é interceção na origem da reta, β_1 é o declive dessa reta e $\in$ (epsilon) é o termo de erro aleatório (positivo ou negativo) com média zero. Deve-se procurar aprender os parâmetros constantes (β_0 e β_1) do modelo, que minimizem o erro nas previsões do mesmo para qualquer valor de X. Para tal deve-se:

- definir uma função de perda que possa medir quão imprecisas são as previsões do modelo;
- encontrar parâmetros que minimizem a perda.

Um exemplo de um problema que consiga ser resolvido com recurso a regressão linear poderá ser o cálculo do vencimento que uma pessoa pode ter mediante o número de anos de experiência.

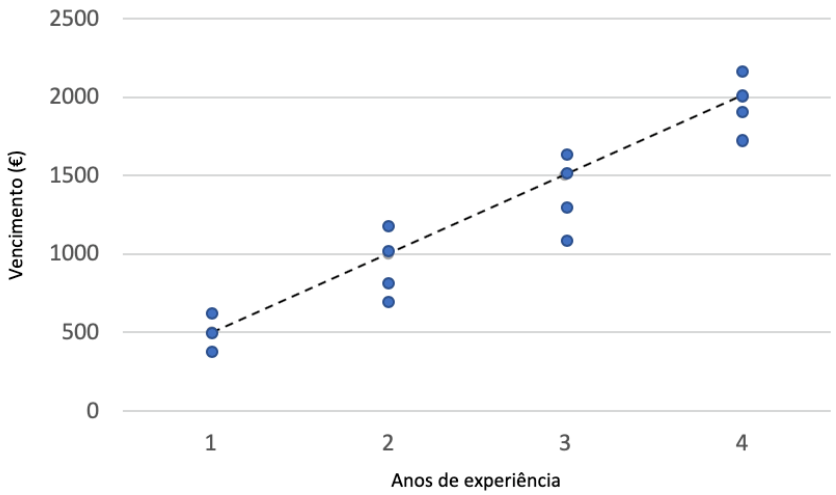

FIGURA 4.1.1: Exemplo de gráfico de análise de regressão linear

4.1.2 Gradiente Descendente

Gradiente descendente é um tópico que surge muitas vezes quando se fala em redes neuronais, sendo utilizado em *background* por bibliotecas de *machine learning* como *scikit-learn* e *tensorflow*. O seu objetivo passa por determinar o valor mínimo da função de perda do modelo, procurando obter uma melhor aproximação do mesmo a cada iteração. Imagine-se uma pessoa a caminhar por um vale com os olhos vendados, com o objetivo de chegar ao ponto mais fundo do mesmo. A melhor forma de chegar ao ponto pretendido talvez seja ir apalpando o terreno de forma verificar a inclinação do mesmo e optar pelo caminho que desça, sucessivamente, até se encontrar um lugar plano, que significaria que se tinha chegado ao ponto mais fundo do vale. Neste caso, a elevação no fundo do vale corresponde precisamente ao mínimo da função de perda. Este exemplo pode ser imaginado com recurso à figura 4.1.2 onde, partindo do ponto A, se deveria sucessivamente procurar o caminho mais indicado para chegar a B, que diz respeito ao valor mínimo [49].
A cada iteração tentam-se identificar os valores que minimizem a função, e vão-se calculando as diretivas parciais, que indicam quanto foi o aumento/decréscimo do total de perda para os valores inseridos.

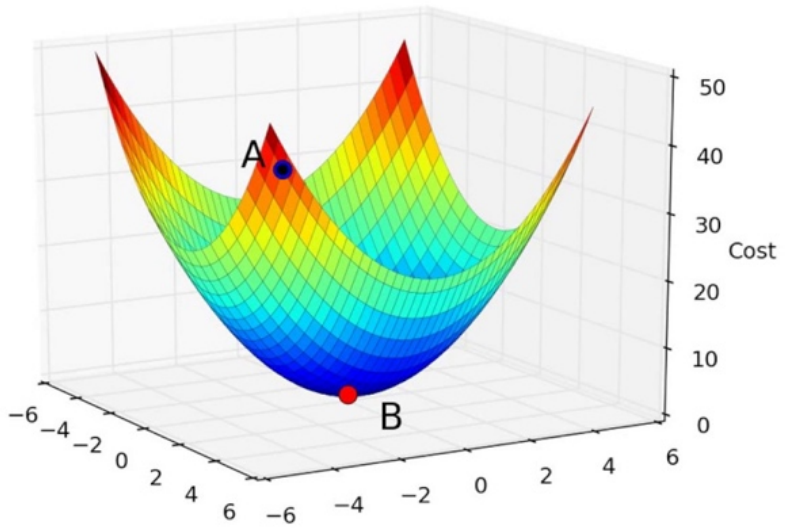

FIGURA 4.1.2: Exemplo de gráfico de gradiente descendente

4.1.3 Overfitting e Underfitting

Overfitting é um problema comum de *machine learning* e refere-se a aprender uma função que explica perfeitamente os dados de treino de onde o modelo foi gerado, mas que não generaliza bem os dados de teste que ainda não foram conhecidos. Acontece quando um modelo aprende em demasia a partir dos dados de treino, captando valores que podem apresentar algum ruído e variância. Isto torna-se especialmente problemático à medida que o

modelo vai ficando mais complexo, pois essas variâncias não se aplicarão a novos dados e a capacidade de generalização do modelo ficará afetada.

Underfitting é precisamente o oposto, ou seja, é quando o modelo não é complexo o suficiente para capturar a tendência subjacente nos dados, pelo que os resultados apresentados no treino não farão uma qualquer separação adequada [50].

Para combater o *overfitting* podem-se usar mais dados de treino ou usar regularização, ou seja, adicionar uma penalização na função de perda para que o modelo verifique mais ao pormenor cada característica dos dados. A figura seguinte permite verificar os casos onde existe *underfitting* (esquerda), por não se conseguir separar claramente as classes, onde existe *overfitting* (direita), visto que se separa ao pormenor os elementos das duas classes, e a separação apropriada, onde se consegue ter as duas classes particamente separadas corretamente na sua totalidade, ignorando um ou outro dado com maior variância ou ruído.

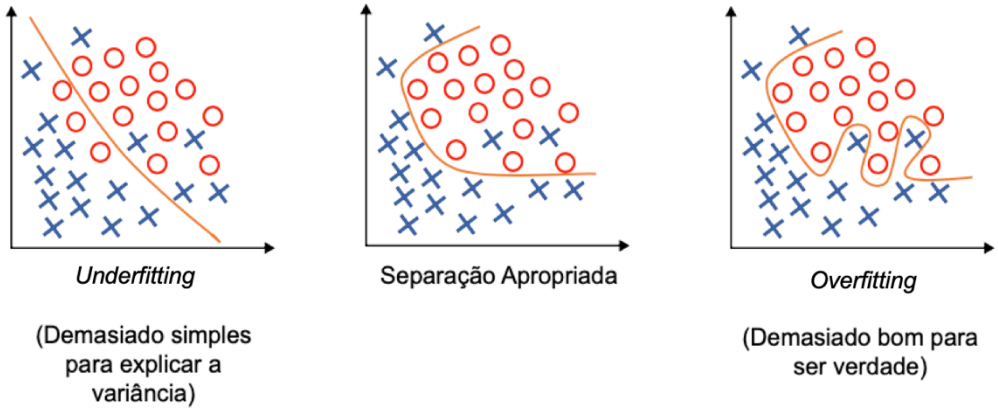

FIGURA 4.1.3: *Underfitting*, separação apropriada e *overfitting*

Neste ponto surgem alguns conceitos importantes:

- Bias – quantidade de erro introduzida pela aproximação de fenómenos do mundo real com um modelo simplificado;
- Variância – quanto muda o erro no teste do modelo, com base na variação nos dados de treino. Reflete a sensibilidade do modelo em relação às características do *dataset* em que foi treinado;
- Híper parâmetro lambda – uma definição geral do modelo que pode ser aumentada ou diminuída de forma a melhorar a performance;
- Cross-validation – técnica para avaliar modelos de *machine learning*. Divide os dados de input em K subconjuntos de dados e treina o modelo em todos, exceto em (K - 1), que é depois utilizado para avaliar o modelo. Repete-se o processo K vezes, com um subconjunto diferente em cada iteração [51].

Um bom modelo é aquele que tem um *bias* e variância baixos. O melhor valor do lambda deve ser decidido com recurso ao método de *cross validation*. Um valor mais alto para o lambda pode levar ao *overfitting.*

4.1.4 Classificação

Um ponto muito importante quando se fala de *machine learning* diz respeito à classificação. Este deverá ser o método ao qual se recorrerá caso se pretenda saber se um email é *spam* ou não, ou quem é a pessoa presente numa fotografia de uma rede social.

Classificação é a tarefa de atribuir um novo *input* à classe a que este deve pertencer através da análise das suas caraterísticas, e com base num modelo de classificação construído a partir de dados de treino assinalados com as respetivas identificações. As classes são sempre discretas, existem em número finito, não têm ordem e são identificadas por um nome. A precisão da classificação dependerá da eficácia do algoritmo a que se recorre, da forma como é aplicado e da quantidade de dados de treino úteis de que se dispõe [52].

FIGURA 4.1.4: Classificar e identificar a que classe de cores (cestos) pertence a bola

Quando se fala de classificação, é também importante conhecer o conceito de previsão. Previsão tem como objetivo prognosticar ou deduzir o valor contínuo de um atributo, baseado no valor de outros atributos. Com base no exemplo da figura 4.1.4, ao invés de classificar a que cesto colorido a bola pertence, pode-se tentar proceder à previsão do peso da mesma.

A construção de um classificador pode ser dividida em três etapas:

- Definição do modelo (fase de aprendizagem);
- Avaliação do modelo (estimar percentagem de correção ou precisão);
- Utilização do modelo (classificação ou previsão de novos objetos).

4.1.5 Classificador Naive Bayes

O classificador *Naive Bayes* é um método estatístico utilizado para a classificação de informação. Recorre à fórmula matemática de probabilidade condicional para calcular a probabilidade de cada registo pertencer a uma determinada classe. A classe com maior

probabilidade é considerada aquela a que o registo pertence. Este classificador assume que as diversas características não estão relacionadas entre si, pelo que a presença ou ausência de uma delas não influencia a presença ou ausência de qualquer outra. Por exemplo, um fruto pode ser considerado uma maçã se for vermelho, redondo e com cerca de 4 centímetros de diâmetro. Mesmo que estas características dependam umas das outras, ou da existência de outras características. O classificador *Naive Bayes* vai sempre considerar estas propriedades de forma independente para contribuir para a probabilidade de esse fruto ser uma maçã [53].

O classificador *Naive Bayes*:

- é rápido e altamente escalável;
- pode ser usado para classificação binária e multi-classe;
- possui um algoritmo simples que processa uma certa quantidade de cálculos para obter o resultado final;
- é bom para problemas de classificação de texto (popular em filtros de *spam*);
- é fácil de treinar num *dataset* pequeno.

4.1.6 Regressão Logística

Regressão logística é um método de classificação onde o modelo dá como output a probabilidade, entre 0 e 100%, de uma variável categórica Y pertencer a uma determinada classe. Embora seja frequentemente usada para classificação binária (duas classes), pode ser aplicada com um qualquer número de categorias.

Na utilização deste método deve ser definida uma probabilidade *cutoff*, ou seja, o limite mínimo para um resultado ser considerado positivo. Por exemplo, se o modelo considerar que a probabilidade de um mail ser spam é superior a 70%, atribui-se a classe “é spam”, caso seja inferior atribui-se a classe definida para essas situações, neste caso, “não é spam” [54]. Este limite depende da tolerância a falsos positivos e a falsos negativos. Por exemplo, se se pretender fazer um diagnóstico de cancro, tem de se ter uma tolerância muito baixa por falsos negativos, pois mesmo que exista uma possibilidade muito pequena de que o paciente tenha cancro, é necessário fazer mais testes para ter certezas do resultado. No caso de classificação de aplicações de empréstimo, para verificar se podem ser fraudulentas, a tolerância por falsos positivos deve ser maior, particularmente para pequenos empréstimos, uma vez que uma verificação mais aprofundada é dispendiosa e o valor de um pequeno empréstimo pode não valer os custos operacionais adicionais.

4.1.7 Support Vector Machines

Support vector machines (*SVMs*) são um conjunto de métodos de *machine learning* utilizados para classificação, regressão e deteção de valores atípicos. Tipicamente resolve os mesmos problemas que a regressão logística (classificação binária) e tem uma performance semelhante. Contudo, as *SVMs* são mais orientadas para análise geométrica [33].

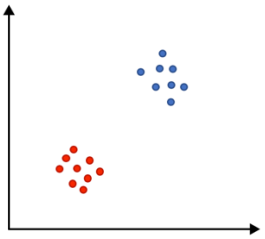

FIGURA 4.1.7.1: Separar os pontos por cores no espaço cartesiano

Considere-se agora um exemplo onde se pretende identificar quais os pontos vermelhos e quais os pontos azuis. Na figura 4.1.7.1 os pontos dos dados de treino estão colocados num espaço cartesiano e pretende-se avaliar os novos pontos que forem surgindo. Para tal, as *SVMs* utilizam uma linha de separação, ou um híper plano multidimensional caso se tenha mais do que duas dimensões, para dividir entre a zona vermelha e a zona azul, neste caso. A distância entre o ponto mais próximo de cada lado é conhecida por margem e a *SVM* tenta maximizar essa margem, desenhando a linha de forma a obter uma margem igual para cada classe. Quanto mais espaço disponível, menos pontos se classificarão de forma errada. A figura seguinte apresenta dois gráficos que fazem a distinção correta entre as duas classes, mas é possível verificar que o da esquerda é o que está mais correto, pelo facto de ter a margem maximizada.

FIGURA 4.1.7.2: Como desenhar corretamente a linha de separação

Para a linha estar desenhada corretamente, esta deve estar a separar os dados de forma limpa e a maximizar a margem. As linhas mais finas que se conseguem ver na figura acima em ambos os gráficos, mostram os pontos mais próximos do híper plano e são conhecidas como vetores de suporte. Se não se conseguir separar os dados claramente, tenta-se

adicionar uma terceira dimensão que poderá ser o suficiente para se conseguir desenhar a linha de separação.

4.1.8 K-Nearest Neighbor

O algoritmo K-Nearest Neighbor (k-NN) tem como objetivo atribuir uma identificação a um ponto X, através do cálculo da média ou moda das identificações dos k pontos mais próximos. Ao contrário dos algoritmos anteriores, este é um método não paramétrico, ou seja, não pode ser caracterizado por um conjunto de parâmetros fixos pelo que a sua estrutura é estritamente especificada a partir dos dados [55].

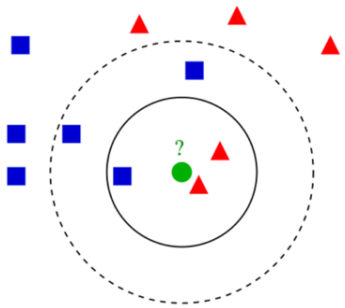

FIGURA 4.1.8: Exemplo do k-NN

Considerando a figura 4.1.8, pretende-se saber se o círculo verde é um quadrado azul ou um triângulo vermelho. Olhando para os pontos vizinhos mais próximos (*nearest neighbors*):

- se tiver em conta o mais próximo (k = 1), supõe-se que seja triângulo vermelho;
- se olhar para os 3 mais próximos (k = 3), supõe-se que seja um triângulo vermelho, pois esta é a moda;
- se analisar os 5 mais próximos (k = 5), supõe-se que seja um quadrado azul, pois esta é a moda.

A decisão pelo valor a calcular, entre média e moda, será definida consoante o tipo de variáveis. Se as variáveis forem contínuas olha-se para os k pontos mais próximos e acha-se a média dos seus valores. Se as variáveis forem categóricas, acha-se a moda.

Um exemplo real que este tipo de algoritmo poderia resolver seria prever o preço de uma casa, calculando a média dos preços das casas geograficamente mais próximas.

Para utilizar este algoritmo será necessário armazenar os dados de treino (com as características pretendidas), ordenar esses mesmos dados de treino pela sua similaridade e achar a média ou moda dos k *nearest neighbors*. As distâncias a calcular a partir do ponto de teste para identificar os k *nearest neighbors* podem ser:

- distância euclidiana – uma linha reta;
- distância manhattan – uma linha quebrada, que é apresentada como que um trajeto através dos blocos de uma cidade (exemplo: utilizada para um modelo de cálculo de tarifas como o da Uber).

O valor de k a usar pode ser decidido após teste com vários valores diferentes recorrendo a *cross-validation*. Um k mais alto irá prevenir o *overfitting*, mas por outro lado, se o valor de k for demasiado alto, o modelo acabará por ser bastante inflexível (*underfitting*). Se k for igual ao total de ponto dos dados, o modelo irá classificar todos os dados de teste com a média ou moda dos dados de treino.

O algoritmo aqui apresentado poderá ter várias aplicações no mundo real, nomeadamente em situações de classificação, por exemplo para deteção de fraude, onde o modelo pode ser atualizado virtualmente e instantaneamente com novos exemplos de treino desde que se adicionem mais dados, para que este conheça os novos métodos de fraude que possam surgir. Pode ainda ser aplicado em questões que tradicionalmente são resolvidas com recurso a regressão, como para prever preços de casas. Por exemplo, se a casa do vizinho mais próximo for de baixo valor, então é provável que a casa em análise também o seja. Então, *K-NN* é bastante útil em domínios onde a proximidade física importa. Além destes casos, pode também ser utilizado para introdução de dados de treino em falta. Se uma das colunas de um ficheiro *csv* tiver valores em falta, podem-se gerar dados para essa coluna a partir da média ou moda dos valores que esta já possui.

4.1.9 Árvores de Decisão

As árvores de decisão podem ser vistas como uma sequência de perguntas. A primeira divisão da árvore na sua raiz será a primeira questão a colocar na sequência de questões, pois pretende-se separar os dados da forma mais limpa possível, maximizando o ganho de informação a partir de cada divisão. Cada nó da árvore implementa um teste a um atributo e cada folha atribui uma classificação [56]. Este algoritmo acaba por funcionar um pouco da mesma forma como os hospitais fazem ao processo de triagem. Faz-se uma sequência de perguntas para se ir percebendo qual o diagnóstico, e não se avança logo para uma radiografia para se ver se o paciente tem um osso partido, por exemplo.

Existem formas de quantificar o ganho de informação de modo a que se consiga avaliar cada possível divisão dos dados de treino e maximizar esse mesmo ganho em cada uma das divisões. Desta forma, pode-se prever cada identificação ou valor da maneira mais eficiente possível.

Neste ponto surge o conceito de entropia. Entropia é o cálculo do total de desorganização num conjunto. Se os valores estão misturados, existe muita entropia. Se for possível dividir claramente os valores, então não há entropia. Para cada divisão do nó pai, pretende-se que os nós filhos sejam o mais puros possível, ou seja, que se minimize a entropia.

Imagine-se agora um caso onde se pretende jogar uma partida de ténis num campo ao ar livre, sendo que o jogo só se realizará se estiverem boas condições atmosféricas. Neste caso, o tempo é o grande determinante para a decisão final, como tal, faz sentido usar esta característica logo na primeira divisão, visto que é aquela que conseguirá dar um maior ganho de informação.

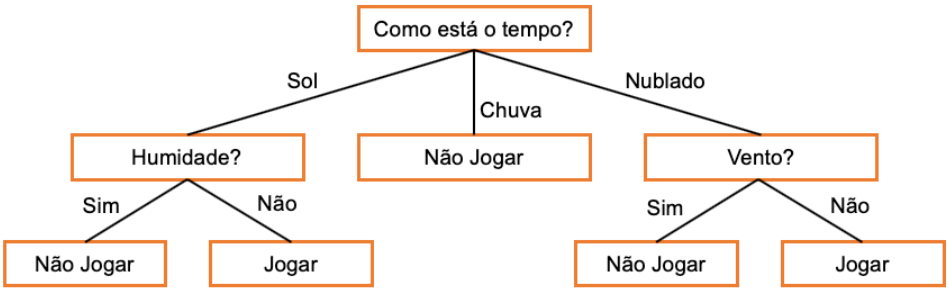

FIGURA 4.1.9: Exemplo de árvore de decisão

Analisando a figura 4.1.9 verifica-se que a primeira divisão dos diferentes tipos de tempo que se pode fazer sentir. Seguidamente, verificam-se outras características de cada um dos estados meteorológicos de modo a decidir se se deve jogar ou não. Este exemplo está a resolver um problema de classificação entre as duas classes: “jogar” e “não jogar”.

As árvores de decisão são eficazes porque são fáceis de ler e poderosas mesmo com dados confusos. São também uma boa opção para lidar com dados mistos (numéricos ou categóricos). Contudo, têm um custo computacional elevado para a realização do processo de treino, acartam um risco de *overfitting,* e não permitem voltar atrás depois de terem sido feitas divisões.

4.1.10 Random Forest

Random forest é um meta estimador que agrega múltiplas árvores de decisão e junta-as para obter uma previsão mais eficaz e estável. Apenas um subconjunto aleatório de características é tido em consideração para as divisões em cada nó. Isto assegura que o modelo conjunto não dependa muito de qualquer característica individual e faça uso justo de todas as características potencialmente preditivas. Além disso, cada árvore desenha uma amostra a partir do *dataset* original quando gera as suas divisões, adicionando um elemento de aleatoriedade extra para evitar o *overfitting* [57].

Estas modificações previnem que as árvores sejam demasiado correlacionadas. *Random Forests* são um excelente ponto de partida para modelar processos, visto que estas tendem a ter uma maior performance com alta tolerância para dados menos limpos e podem ser úteis para identificar quais as características que realmente importam.

Uma analogia para o funcionamento deste algoritmo pode ser um indivíduo a consultar vários amigos sobre qual deverá ser o destino ideal para uma viagem. Vai consultando diferentes pessoas e fazendo diferentes perguntas, para depois analisar toda a informação colecionada e tomar a decisão final.

4.2 Unsupervised Learning

O objetivo de *unsupervised learning* passa por encontrar uma estrutura subjacente num *dataset*, sumarizar e agrupar da forma mais útil, e representar os dados num formato comprimido. Ao contrário do que acontece com *supervised learning*, aqui os algoritmos não recebem os dados já identificados, sendo que estes são obrigados a encontrar a estrutura com base nos *inputs* [58].

Unsupervised learning é frequentemente utilizada para pré-processar dados, normalmente para os comprimir de forma a preservar o seu significado antes de os usar numa rede neuronal ou num algoritmo de *supervised learning*.

Os seus métodos podem ser aplicados nos seguintes casos:

- uma plataforma de publicidade segmenta a população de um país em grupos mais pequenos com uma demografia e hábitos de compras similares, podendo atingir o seu objetivo com anúncios mais relevantes;
- *zomato* agrupa as suas listas de restaurantes por tipo para os utilizadores poderem pesquisar e encontrar o que pretendem com maior facilidade;

Ao contrário de *supervised learning*, nem sempre é fácil encontrar métricas sobre a qualidade do desempenho que o algoritmo terá. A performance é geralmente subjetiva e específica do domínio.

4.2.1 Clustering

O objetivo do *clustering* é criar grupos de pontos de dados de modo a que esses pontos sejam distintos em diferentes clusters, enquanto que os pontos dentro de um *cluster* são semelhantes. Por exemplo, poderá agrupar textos que falem sobre o mesmo assunto e separar textos que retratem conteúdos diferentes.

4.2.1.1 K-means Clustering

Com *k-means clustering*, pretende-se agrupar os pontos de dados num total de k grupos. Um maior k cria grupos mais pequenos e com maior granularidade. Já por sua vez, um k menor significa a criação de grupos maiores e com menor granularidade. O *output* do algoritmo será um conjunto de identificações, atribuindo cada ponto de dados a um dos k grupos. Neste algoritmo, a forma como estes grupos são definidos é feita através da criação de um *centroid* para cada grupo. Um *centroid* pode ser visto quase como o coração do *cluster*, que captura os pontos mais próximos de si e adiciona-os ao *cluster*. Os passos do *clustering k-means* são os seguintes:

1) definir os *centroids*, e inicializá-los de forma aleatória;

2) encontrar o *centroid* mais próximo e atualizar as atribuições do *cluster*. Atribuir cada ponto de dados a um dos k *clusters*. Cada ponto de dados é atribuído ao *centroid* do *cluster* mais próximo, normalmente calculado com recurso à distância euclidiana;

3) mover os *centroids* para o centro dos seus *clusters*. A nova posição de cada *centroid* é calculada pela posição média de todos os pontos no *cluster.*

Os passos 2 e 3 devem ser repetidos até que o *centroid* pare de se mover muito em cada iteração [59].

4.2.1.2 Hierarchical Clustering

É parecido com o *clustering* regular, no entanto este tenta construir uma hierarquia de *clusters*. Isto pode ser útil quando se pretende flexibilidade no número de *clusters* que se quer ter. Em termos de *outputs* a partir do algoritmo, além das atribuições do *cluster* também se pode construir uma árvore que oferece informação sobre as hierarquias entre eles. Pode-se então escolher o número de *clusters* que se pretende a partir desta árvore. Os passos do *hierarchical clustering* são os seguintes [60]:

1) começar com N *clusters*, um para cada ponto de dados;

2) fazer a junção dos dois *clusters* que estão mais próximos um do outro;

3) voltar a computar as distâncias entre *clusters;*

4) repetir os passos 2 e 3 até que se obtenha um *cluster* de N pontos de dados*;*

4.2.1.3 Redução de Dimensionalidade

Redução da dimensionalidade trata-se da tentativa de reduzir a complexidade dos dados enquanto se mantém a maior parte da estrutura relevante possível, acabando por ser um conceito algo parecido com compressão. Ter uma imagem simples de 128 x 128 x 3 pixéis (comprimento, largura, valor RGB), diz respeito a 49152 dimensões de dados. Se se conseguir reduzir a dimensionalidade do espaço dessas imagens sem se estragar muito o conteúdo significativo das mesmas, a redução da dimensionalidade foi feita corretamente. As principais técnicas em prática são a *principal component analysis* (*PCA*) e a *singular value decomposition* (*SVD*) [61].

Na primeira técnica (*PCA*), assumindo um gráfico cartesiano, é possível modificar a posição inicial dos eixos base para reduzir a dimensão do espaço. Estes vetores base são chamados de componentes principais, e o subconjunto que se seleciona constitui um novo espaço que é mais pequeno em dimensionalidade do que o original, mas mantém a maior parte da complexidade dos dados possível. Para selecionar os componentes principais mais significantes, verifica-se a quantidade de variância dos dados que se capturam e ordenam-nos por métrica. Além disso, *PCA* remapeia o espaço em que os dados existem para torná-lo mais compreensível, sendo que a dimensão transformada é menor do que a dimensão original. Fazendo uso das primeiras dimensões do espaço, é possível começar a entender melhor a organização do conjunto de dados.

A segunda técnica (*SVD*) é uma computação que permite decompor uma grande matriz num produto de três matrizes mais pequenas. Imagine-se uma imagem. Ao se reduzir a sua dimensão, ainda se terá imagens com qualidade suficiente, pois descartam-se as partes das matrizes que têm os seus valores multiplicados por zero.

Este é objetivo da redução de dimensionalidade: reduzir a complexidade, mantendo a estrutura.

4.3 Deep Learning e Redes Neuronais

Deep Learning é um subconjunto de inteligência artificial e de *machine learning* que usa redes neuronais artificiais de várias camadas por serem capazes de oferecer uma precisão de última geração em tarefas como deteção de objetos, reconhecimento de voz, tradução de idiomas e muitas outras. Difere das técnicas tradicionais pois consegue aprender de forma automática representações de dados como imagens, vídeo ou texto, sem introduzir regras codificadas manualmente ou conhecimento de domínio humano. As suas arquiteturas altamente flexíveis podem aprender diretamente a partir de dados em bruto e podem aumentar a sua eficácia de

previsão quando dispõe de ainda mais dados. É responsável por muitos avanços recentes na IA em produtos como carros autónomos, assistentes de voz, entre outros [62].

Tal como em situações anteriores, *deep learning* tem também o objetivo de aprender uma função matemática que mapeie um *input* X e um *output* Y. Apesar de ter menos perdas de dados no teste, a complexidade do mundo real pode por vezes tornar a função mais complicada. Em problemas de linguagem natural, o maior tamanho de determinado vocabulário pode significar muitas características. As técnicas que foram apresentadas em pontos anteriores conseguem cumprir bem as tarefas quando os dados com os quais se está a trabalhar não são demasiado complexos. Contudo, não existe uma forma clara de poderem ser generalizados para cenários que envolvem problemas de *computer vision* ou disputa de jogos, que apresentam muita informação visual para cada um dos pixéis e que requerem a tomada de uma decisão baseada em cenários complexos com muitos futuros possíveis senão mesmo infinitos. Então, *deep learning* é uma boa escolha para aprender uma função, especialmente em situações onde os dados são complexos.

As redes neuronais estão inseridas neste contexto. São conhecidas como aproximadores universais de funções pois são capazes de aprender qualquer função com uma única camada oculta (*hidden layer*) [63]. Considere-se por exemplo um problema de classificação de uma imagem, que diz respeito ao *input*, e o *output* será uma classe (cão, gato, tigre, etc.). Na figura 4.3 pode-se ver graficamente uma rede neuronal que poderia resolver este problema. Neste caso, está-se perante uma enorme equação matemática com milhões de termos usados e muitos parâmetros. O *input* X é uma imagem de escala de cinzentos representada por uma matriz com os brilhos de cada pixel. O *output* Y é um vetor de probabilidades para as classes, ou seja, a probabilidade de cada classe ser a identificação correta. A maior probabilidade corresponderá à classe correta.

As camadas no meio estão a fazer um conjunto de multiplicações de matrizes para permitir que a rede aprenda uma função não linear. Curiosamente, pode-se utilizar um gradiente descendente, da mesma maneira que se usa em regressão linear, para treinar estes parâmetros de forma a minimizar perdas.

Imagine-se uma rede neuronal como uma série de portas, uma depois da outra, e pense-se numa pessoa como *input* para essa mesma rede. De cada vez que a pessoa abrir uma porta, tornar-se-á numa pessoa diferente e, quando abrir a última porta, a pessoa será o *output* da rede. Cada porta, neste caso, representa uma camada que transforma o *input* de alguma forma para produzir um *output*.

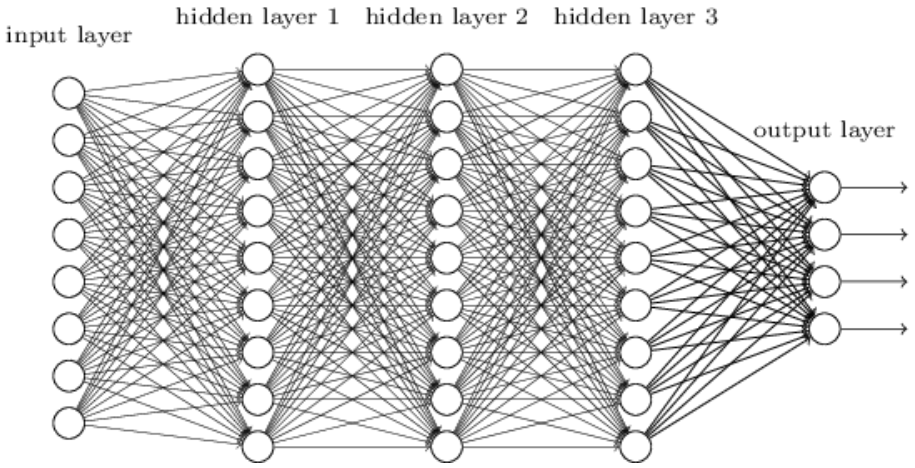

FIGURA 4.3: Exemplo de rede neuronal

Estas redes começaram a tornar-se mais profundas por volta 2006, quando surgiu o conceito de *deep learning*, no entanto, o foco de investigação nesta área apenas tem vindo a aumentar mais recentemente. Segundo o diretor de IA da *Tesla*, Andrej Karpathy, existem quatro fatores que impedem o recurso a inteligência artificial [64]:

- falta de poder de computação (Lei de Moore, *GPUs*, *ASICs*);
- qualidade dos dados (num formato interessante, não apenas algo disponível na internet);
- pouca diversidade de algoritmos (ideias de investigação);
- fraca infraestrutura (Linux, TCP/IP, *Git*, *AWS*, *Tensorflow*, etc.).

Na última década, o potencial de *deep learning* começa a ser desbloqueado por avanços relacionados com os primeiros dois pontos, como a evolução do *hardware* que permite outra capacidade de computação e pela existência de uma maior quantidade de dados relevantes e relativamente bem estruturados na *web*. Inevitavelmente, estes avanços vão levar a progressos nos últimos dois pontos. Os algoritmos começar a ser cada vez mais explorados e a passar do papel para o computador sendo que é relativamente fácil encontrar várias implementações dos mesmos e de algumas *frameworks* em sites como o *Github*, existindo ainda algumas demos que estão disponíveis para exemplo.

4.3.1 Neurónios e Camadas Ocultas

O cérebro humano consiste em biliões de neurónios interligados por sinapses. Se milhares de *inputs* de sinapses despoletarem para um neurónio, então esse neurónio também vai despoletar uma ação. Este processo é conhecido como pensamento. Para replicar este processo em computadores, é necessário recorrer a *machine learning* e redes neuronais [65].

Ao ler as palavras da listagem 4.3.1 não se está a examinar todas as letras de todas as palavras, ou todos os pixéis que compõe cada letra para derivar o significado das palavras. Está-se sim, a extrair detalhes e a agrupar coisas em conceitos de nível superior: palavras, frases, expressões e parágrafos.

```
Oru abiilty to exaimne hgiher-lveel fteaures is waht aollws us to
unedrtsand waht is wirtten in tihs prhase
```

LISTAGEM 4.3.1: Frase com palavras distorcidas

A mesma coisa acontece à visão, não apenas nos humanos, mas também nos animais em geral. Os cérebros são compostos por neurónios que se acionam ao emitir sinais elétricos para outros neurónios após serem ativados.

As redes biológicas do ser humano estão organizadas de uma forma hierárquica, assim, certos neurónios acabam por detetar características não muito específicas do que o rodeia, mas sim características mais abstratas. A ideia por detrás de uma rede neuronal é imitar a estrutura do cérebro humano com camadas de neurónios artificiais. A figura acima,

Retomando o exemplo da classificação de uma imagem, imagine-se que se pretende treinar uma rede neuronal para atribuir a identificação correta, assumindo um conjunto finito de identificações possíveis. As abordagens que recorrem a modelos lineares não envolvem camadas de abstração, e apenas combinam todas as diferentes orientações dos pixéis de cada tipo de imagem numa desfocagem média. Isto implicaria, por exemplo, ter de analisar todas as fotos de cada objeto, em diferentes posições, de forma a calcular os valores médios de cada característica para depois se conseguir fazer as comparações entre elas e atribuir a classificação final.

Então, em vez de aprender um simples modelo linear relacionando o *input* com o *output*, a rede neuronal deve conter camadas ocultas intermédias para aprender características cada vez mais abstratas, o que permite não perder todas as nuances dos dados complexos. A camada de *input* deve ter o brilho dos pixéis de uma imagem e a última camada deve ser um *output* com um vetor de probabilidades por classe. Os neurónios artificiais nas camadas ocultas vão aprender a detetar conceitos abstratos, ou seja, conceitos que em última instância possam ser mais úteis para capturar informação e minimizar perdas na precisão dos *outputs* da rede. Contudo, à medida que se adicionam mais camadas, os neurónios começam a representar características cada vez mais abstratas e difíceis de entender. A figura 4.3.1.1 apresenta um exemplo de uma camada que procura verificar se uma imagem contém uma face, através da procura das suas características principais.

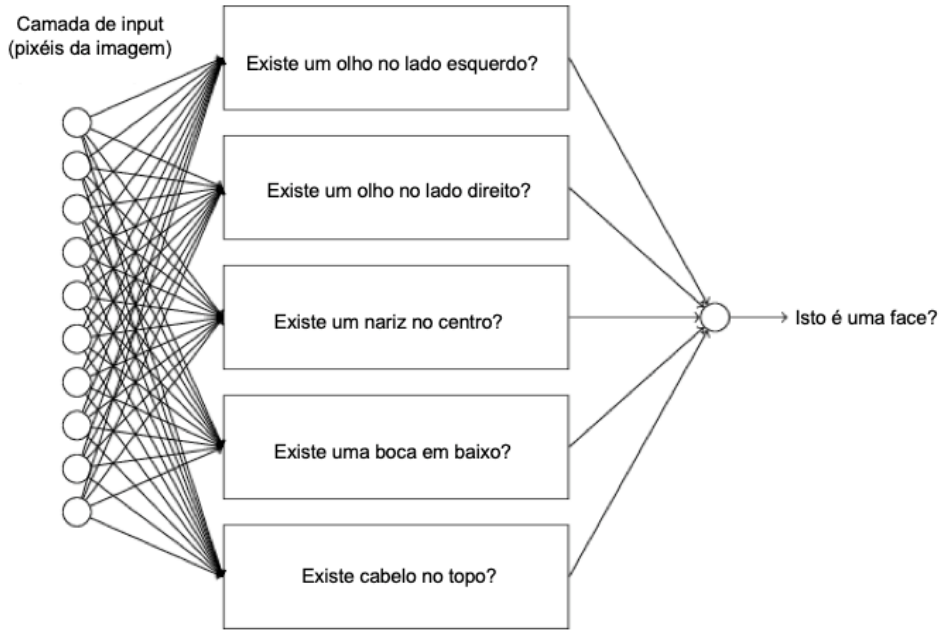

FIGURA 4.3.1.1: Ilustração do funcionamento de uma camada

D*eep learning* é muitas vezes referenciada como um método de caixa preta, onde basicamente se tenta obter um *output*, mediante o *input*, mas nunca se chega a entender realmente o que acontece lá dentro durante a execução. Por sua vez, regressão linear é interpretável porque se decidem quais as características que se devem incluir no modelo. As redes neuronais são mais difíceis de interpretar porque as características são aprendidas de forma autónoma e não explicadas textualmente, estando tudo na lógica da máquina.

Raramente é necessário fazer implementações de raiz de todas as partes da rede neuronal pois existem bibliotecas e ferramentas que podem tornar este desenvolvimento mais simples. Entrem muitas podem-se destacar: *Tensorflow, Torch, Caffe* ou *Theano*.

4.3.2 Redes Neuronais Convolucionais

As Redes Neuronais Convolucionais (*CNN*) são projetadas especificamente para usarem imagens como *input* e são eficazes na execução de tarefas de *computer vision,* contudo, podem também ser aplicadas em problemas de processamento de linguagem natural. São utilizadas primariamente para classificar imagens, agrupá-las por similaridade (como pesquisa por imagem) e fazer reconhecimento de objetos num determinado cenário. Dispõe de algoritmos que conseguem identificar rostos, indivíduos, sinais rodoviários e muitos outros aspetos visuais [66].

Este tipo de redes não entende as imagens do mesmo modo que os humanos. As *CNNs* entendem as imagens como volumes, como por exemplo, objetos tridimensionais em vez de planos medidos apenas por altura e largura. Isto acontece porque as imagens digitais têm um *encoding* RGB que com essas três cores misturadas produzem um espectro que os humanos

conseguem entender. Por sua vez, a rede consome essas imagens como três camadas separadas de cores empilhadas umas por cima das outras. Assim, para cada intensidade de vermelho (R), verde (G) e azul (B) em cada pixel é gerado um número que é depois utilizado para a comparação em diversas matrizes de um certo tamanho, de forma a identificar os principais padrões gerando vários mapas de características que depois serão distribuídos pelas camadas da rede. Aqui chega o conceito de convolução. Pode-se pensar na convolução como uma espécie de multiplicação sofisticada usada no processamento de sinais.

4.3.3 Redes Neuronais Recorrentes

As Redes Neuronais Recorrentes (*RNN*) são úteis no que toca a processar dados sequenciais, como som, dados de *time series* ou linguagem natural escrita. Diferem dos outros tipos de rede na medida em que retornam um *feedback*, em que o *output* da iteração n-1 é enviado de volta para alimentar a rede de forma a influenciar o resultado da etapa n, e assim sucessivamente para cada etapa que se siga. Por exemplo, se uma rede é exposta a uma palavra, letra por letra, e se pretende adivinhar qual a letra seguinte, a primeira letra da palavra irá ajudar a rede a determinar qual a será a segunda letra [67].

4.4 Reinforcement Learning

Em *supervised learning* foi possível verificar que os dados de treino vêm com uma chave para a resposta disponibilizada a partir de uma espécie de supervisor, mas não é apenas dessa forma que é possível aprender. Em *reinforcement learning* não existe essa chave, e o seu agente tem que decidir como atuar para executar a tarefa a que se propõe. Na ausência de dados de treino, o agente tem de aprender com base na experiência. Recolhe os exemplos de treino (“esta ação foi boa”, “esta ação foi má”) através do método de tentativa erro à medida que tenta executar a tarefa com o objetivo de maximizar a recompensa que pode alcançar.

O melhor contexto para entender o funcionamento de *reinforcement learning* é em jogos com um objetivo claro e um sistema de pontos. Imagine-se um jogo (figura 4.4) onde existe um rato à procura de uma melhor recompensa, que neste caso será um queijo no final de um labirinto (+100 pontos), ou uma pequena recompensa que será água colocada ao longo do mesmo labirinto (+1 ponto). Entretanto, pretendem-se evitar lugares que transmitam um choque elétrico (-10 pontos) e façam reduzir os ganhos das recompensas.

Depois de alguma exploração, o rato acaba por encontrar três pontos de água juntos, perto da entrada do labirinto, e perde todo o seu tempo a tirar partido dessa descoberta, acumulando constantemente essas pequenas recompensas que dizem respeito aos pontos da água e nunca mais explora o resto do labirinto em busca de uma recompensa melhor. Isto

significa que o rato terá de optar por perder os pontos relativos à água ou os pontos relativos ao queijo que está no final do labirinto. Uma simples estratégia para a exploração seria que o rato escolhesse a melhor opção conhecida (80% do tempo), mas ocasionalmente fosse explorar o labirinto numa nova direção aleatoriamente, mesmo que se esteja a afastar da recompensa conhecida.

FIGURA 4.4: Exemplo de jogo que pode ser vencido com *reinforcement learning*

Esta estratégia é conhecida por *epsilon-greedy*, onde a épsilon é a percentagem de tempo em que o agente escolhe uma ação aleatoriamente em vez de optar pela ação que é mais provável de maximizar a recompensa dado o que se conhece até ao momento, neste caso, 20%. Normalmente começa-se com um valor alto de exploração (maior valor para a épsilon). Ao longo do tempo, e à medida que o rato aprende mais sobre o labirinto e sobre que ações produzem a melhor recompensa, faria sentido reduzir de forma constante a épsilon, visto que o rato se limita a explorar o que conhece. É importante ter em conta que a recompensa não é sempre imediata. Ora veja-se neste exemplo, onde existe um longo caminho no labirinto e vários pontos de decisão que se têm de percorrer antes de se chegar ao queijo. O agente observa o ambiente, toma uma ação para interagir com o mesmo e recebe uma recompensa que pode ser positiva ou negativa [49].

Esta situação do rato a andar pelo labirinto pode ser formalizada como um processo de decisão de *Markov* [68], que especifica a probabilidade de transição de estado para estado, e inclui:

- **conjunto finito de dados** – posições possíveis para o rato dentro do labirinto**;**
- **conjunto de ações disponíveis em cada estado** – (“para a frente”, “para trás”) num corredor, e (“frente”, “trás”, “esquerda”, “direita”) nos cruzamentos;

- **transições entre estados** – se se for para a esquerda num cruzamento, acabar-se-á numa nova posição. Isto pode ser um conjunto de probabilidades que liguem a mais do que um estado possível;
- **recompensas associadas com cada transição** – a maioria das recompensas a cada movimento valem zero, sendo apenas positivas se se alcançar um ponto que tenha água ou o queijo, e negativas se se encontrar um choque elétrico;
- **fator de desconto Y** – quantifica a diferença na importância entre as recompensas imediatas e as futuras. Se Y = 7 e existe uma recompensa de 3, a 2 passos de distância, o valor presente para essa mesma recompensa é igual a 7^2 x 3;
- **inexistência** – quando o estado atual é conhecido, o histórico de viagens do rato através do labirinto pode ser apagado porque o atual estado de *Markov* contém toda a informação útil do histórico.

Com esta informação é possível formalizar o objetivo do rato. Está-se a tentar maximizar a soma das recompensas a longo prazo, optando pela melhor ação em cada estado.

Uma solução possível para o problema poderia ser gerada com recurso a *Q-learning*. É uma técnica que avalia qual a ação que se deve tomar com base numa função que determina o valor de se estar num determinado estado e se executar uma certa ação nesse mesmo estado. Dispõe de uma função *Q* que recebe como *input* um estado e uma ação, e retorna a recompensa esperada para essa ação (e subsequentes, se existirem) nesse estado. À medida que se explora mais o centro do labirinto, *Q* dá uma melhor aproximação ao valor de uma ação num determinado estado. Assim que se tem o valor estimado para cada par ação-estado, dados pela função, pode-se decidir qual a ação seguinte a tomar de acordo com a estratégia ação-seleção. No exemplo do rato, pode-se então tentar descobrir o valor de cada posição do labirinto e o valor das ações (direções a avançar) em cada posição, e escolher o que o rato deve fazer em cada momento.

Outra possível solução dá-se pelo nome de *policy learning*. É uma alternativa mais direta onde se aprende uma função que faz um mapeamento direto entre o estado e a melhor ação correspondente nesse estado, ou seja, quando se observa o estado X, a melhor coisa a fazer é a ação Y. Considerando o exemplo de um carro autónomo, se este deteta um stop e se está a menos de 30 metros do mesmo, deve começar a travar, senão, deve continuar. Assim, vai ser aprendida uma função que vai maximizar a recompensa esperada.

4.5 Conclusão

Com este capítulo foi possível ficar a conhecer os diferentes tipos de algoritmos que existem para implementar uma solução inteligente que seja capaz de resolver problemas complexos

de forma autónoma e com capacidade de ir melhorando o seu desempenho à medida que efetua mais execuções. As opções variam um pouco na sua forma de pensar, contudo a sua possível combinação permitirá a mais avanços no desenvolvimento de novas aplicações que tenham o objetivo de resolver diversas tarefas que até então, apenas o ser humano era capaz de efetuar.

Capítulo 5

Apresentação do Problema

O presente capítulo tem como objetivo introduzir um problema como o *cyberbullying*, que se vai intensificando por via do aumento da utilização das tecnologias, da sua acessibilidade e pelo facto de as pessoas começarem a estar conectadas à internet desde muito cedo. A presença deste tipo de ameaças motivou que o presente trabalho fosse elaborado com o objetivo de propor um modelo para combate e resolução destas situações, de modo a ir ao encontro de uma solução capaz de mitigar ao máximo todas as consequências que possam surgir.

5.1 Problema Atual

A sociedade está cada vez mais dependente do uso de tecnologia, sendo que a população geral mais jovem tem a necessidade de se conectar à internet em curtos espaços de tempo [69]. Com tantas possibilidades que a tecnologia oferece, existe uma quantidade infinita de hipóteses de as pessoas poderem interagir umas com as outras, podendo dar a sua opinião através de um comentário num fórum, disponibilizar na internet uma foto relativa a determinado assunto ou enviar um email ou mensagem para quem quiserem. Contudo, toda esta interação permite que se abram espaços para discórdia, levando as pessoas a reagirem muitas das vezes de forma insultuosa umas para com as outras ou a partilharem conteúdo com o objetivo de atingir negativamente alguém para fazer valer a sua opinião.

Abre-se aqui espaço para o surgimento da prática de *cyberbullying.* Como já referido anteriormente, é um dos principais problemas emergentes da sociedade e que implica a violência psicológica por via do recurso às tecnologias, sendo que com a forte presença que as redes sociais têm, este problema social está cada vez mais presente.

O facto de as tecnologias estarem disponíveis com grande facilidade de acesso, aumenta ainda mais a dimensão deste problema, pois os jovens conseguem ter acesso desde tenra idade à internet e começam a conectar-se com outras pessoas sem ainda terem a real noção dos perigos que podem enfrentar, pese embora as contínuas tentativas que as escolas e autoridades têm feito para promoverem ações de formação para os alertarem de tais possibilidades [70]. Este problema não existe somente numa faixa etária mais baixa, mas é lá que as situações podem tomar proporções diferentes, pois as crianças e adolescentes estão sempre mais sujeitas a enfrentar cenários que para eles não aparentam qualquer perigo e acabam por tomar atitudes muitas das vezes irracionais.

A perseguição contínua e ameaças constantes, mensagens insultuosas, a publicação de fotografias que levem à humilhação, são algumas das principais técnicas de prática de *cyberbullying* a que os agressores recorrem e que se pretendem evitar, tentando que estas ações não transponham a barreira tecnológica e passem a ser ocorrências presenciais que culminem em agressões físicas. Na maioria dos casos, os pais ou outros encarregados de educação nem sequer imaginam que o seu ente está envolvido nestes acontecimentos, quer como vítima, quer como agressor, pelo que estes podem vir a ser agentes importantes nestes cenários para prestarem auxílio na redução destas práticas.

Este tipo de ações não traz nada de bom para a sociedade pelo que o ideal seria ter uma forma de o combater, procurando a sua extinção. Atualmente, não existe nenhum sistema informático que seja capaz de detetar automaticamente esta prática com grande eficácia, pelo que é necessário explorar quais as tecnologias que permitiriam fazer com que um computador consiga entender que se está perante uma situação de insulto mediante a análise do conteúdo disponível na internet. A crítica ou ameaça que está por detrás de uma situação de *cyberbullying* pode ser representada de várias maneiras, quer por via de texto, quer por via de imagem ou até mesmo por sons. Para corresponder a estes casos, será necessário que o computador disponha de alguns recursos tecnológicos, como ferramentas para implementação de algoritmos de *machine learning*, que deverão ser necessários para que este tenha aptidão para identificar se se encontra perante uma situação de *cyberbullying*, com base em métodos de reconhecimento textual, de *computer vision* e de interpretação de sons.

Embora existam já algumas bibliotecas que efetuem este tipo de análises, ainda nada de muito útil foi construído com estas tecnologias para identificar de forma autónoma a presença de insultos, ameaças ou críticas em conteúdo que esteja disponível na internet praticamente para qualquer indivíduo, pelo que este trabalho se propõe a apresentar um modelo que seja capaz de o fazer e que ao longo do tempo se vá tornando cada vez mais eficaz.

5.2 Abordagem a Seguir

O modelo a apresentar terá de se focar nas principais formas a que se recorrem para a prática de *cyberbullying*. Desde logo, está implícito que o insulto será na maioria das vezes colocado sob a forma de texto, pois esta é a maneira mais fácil e tradicional de comunicação no universo tecnológico. Será também neste estado que a maioria do conteúdo para análise estará disponível, pelo que se poderão analisar mensagens de texto, emails, comentários, publicações, notícias, entre outras. Neste caso será necessário conseguir classificar um pedaço textual como sendo ou não insultuoso, pelo que é necessário construir uma lista de palavras e frases que sejam catalogadas como tal e, além disso, é necessário conhecer as principais características das mesmas.

O possível recurso a imagens para a prática de insulto deverá ser outro dos focos principais da solução. Uma fotografia pode ser usada para causar mau estar em alguém, pois esta pode por si só, ser já suficientemente comprometedora por via do seu conteúdo. A presença de uma pessoa numa fotografia expõe desde logo a mesma, mesmo que a imagem não tenha qualquer problema à partida. No entanto, se esta imagem for apresentada juntamente com algum texto, poderá estar-se perante uma situação de gozo ou crítica, que pode ter na imagem uma forma de reforço e de aumento da prospeção do conteúdo, que o poderá tornar mais grave do que à partida poderia parecer. Assim, verificar a existência de pessoas nas fotografias, assim como em vídeos, pode desde logo, dar indicadores de que as imagens estão a ser utilizadas para tentar de algum modo referenciar um indivíduo, pelo que a presença do texto deverá ser essencial para classificar se a situação pode dizer respeito à prática de *cyberbullying* ou não.

O que se pretende criar, é um sistema que seja capaz de identificar estas situações de forma autónoma e que se vá conseguindo adaptar a novos cenários que possam surgir, podendo aprender ao longo do tempo e aumentando, deste modo, a sua eficácia a cada previsão. O sistema deve então ser desenvolvido com recurso a algoritmos e ferramentas de *machine learning* para ter esta capacidade de reconhecimento, autonomia, aprendizagem e classificação. Basicamente, o seu funcionamento deve consistir em receber como *input* um tipo de conteúdo, dos já referidos acima, que possa ser passível de ser considerado como *cyberbullying* e que retorne como *output* a probabilidade de este pertencer à classe que afirma e à classe que nega a existência da situação.

FIGURA 5.2: Abordagem base para a solução

Finda a análise ao conteúdo, algumas medidas poderão ser tomadas no caso de se encontrar uma potencial situação de *cyberbullying*. A principal passa por lançar alertas para os agentes principais, nomeadamente os pais ou encarregados de educação para que possam estar atentos à questão e confirmarem se esta é realmente grave, podendo abordar os seus educandos e prevenir a continuidade de acontecimentos semelhantes. A vítima deve também receber uma notificação que a aconselhe a falar com alguém responsável para evitar enfrentar situações mais negativas e sofrer de depressão. Os potenciais agressores deverão também ser alertados para as más práticas que possam estar a realizar, tentando assim pressioná-los para terminarem com atos idênticos no futuro e apagarem qualquer conteúdo semelhante que tenham disponibilizado no passado. Além desta, outra medida pode passar por um sistema

de bloqueio de contas, mediante a contínua prática destes atos. Estas medidas fazem sentido em aplicações onde os utilizadores tenham perfis pessoais, contudo, se estes não existirem, a medida a seguir poderá ser optar por ocultar o conteúdo em causa, evitando que este seja visto por um maior número de pessoas e tenha um alcance maior.

Este modelo pode ainda oferecer a possibilidade de ser facilmente adaptável para outras situações. Já que o modelo contempla a análise textual e de imagem, certamente conseguirá, mediante a implementação das alterações necessárias, adaptar-se a outros problemas como por exemplo, deteção de *clickbait* ou de notícias falsas.

Assim, faz todo o sentido desenvolver um modelo com estas características para resolver o problema apresentado e com capacidade de se adaptar a novas e diferentes situações. Com esta abordagem, espera-se apresentar um modelo que permita desenvolver um sistema autónomo e suficientemente eficaz para reduzir ao máximo a atividade de *bullying* que se vai realizando com recurso às tecnologias, especialmente na internet. Pretende-se ainda o recurso a mecanismos de *machine learning* de forma a ter uma ferramenta inteligente que consiga aprender com o conteúdo que vai analisando e se vá superando à medida a que aumenta a sua base de informação.

Capítulo 6

Proposta de Solução

Ao longo dos capítulos anteriores foram sendo apresentados os estudos que têm como objetivo combater o problema identificado do *cyberbullying*, assim como os desenvolvimentos nos sistemas implementados com recurso a mecanismos de inteligência artificial e os algoritmos de *machine learning* mais utilizados para essas mesmas implementações. Além disso, foi identificado e apresentado o problema que este estudo se propõe a resolver e que no presente capítulo verá descrita a sua proposta de solução.

6.1 Arquitetura da Solução

Como já foi referido ao longo deste trabalho, pretende-se fazer um sistema que consiga detetar a presença ou não de *cyberbullying* mediante a análise de conteúdo disponível na internet. Pretende-se que o sistema seja autónomo, inteligente e que vá aprendendo e melhorando a eficácia do seu desempenho à medida que prevê a existência de tais situações.

O sistema deve então ser desenvolvido com recurso a mecanismos de *machine learning*, e contemplar a fundamental análise ao conteúdo textual, pois é por esta via que a maioria dos insultos e ameaças são realizados. É ainda importante ter a preocupação em analisar o conteúdo presente em imagem, seja este uma foto ou um vídeo, pois agregado a um texto potencialmente insultuoso, pode estar-se perante uma situação de *bullying*.

A solução a implementar é essencialmente destinada a jovens, pois são eles os elementos normalmente envolvidos neste tipo de situações, e o objetivo da mesma é monitorizar aplicações que permitam interações entre utilizadores, especialmente tendo em consideração o número de redes sociais que existem na internet e o tempo que as pessoas despendem a navegar nas mesmas, de modo a identificar eventuais situações de *cyberbullying* e a alertar as pessoas para a existência de tais práticas.

De uma forma geral, será necessária uma biblioteca de análise de texto, que seja capaz de indicar se uma determinada frase é potencialmente insultuosa e agressiva. Para tal, será necessário recolher um conjunto vasto de dados composto por frases deste género e já classificadas como *bullying* ou não, para poder treinar o sistema à partida e testá-lo mais à frente. Da mesma forma, irá também ser fundamental a posse de uma biblioteca para identificar o conteúdo presente numa imagem, especialmente para verificar se esta contém a presença de algum indivíduo que possa estar a ser alvo de ataque. Adicionalmente, poderá

avançar-se com uma tarefa de reconhecimento facial, se a presença humana for confirmada nas imagens de modo a verificar a correspondência entre o individuo presente na imagem e no alvo do conteúdo textual. A estrutura aqui descrita pode ser analisada com recurso à figura 6.1.

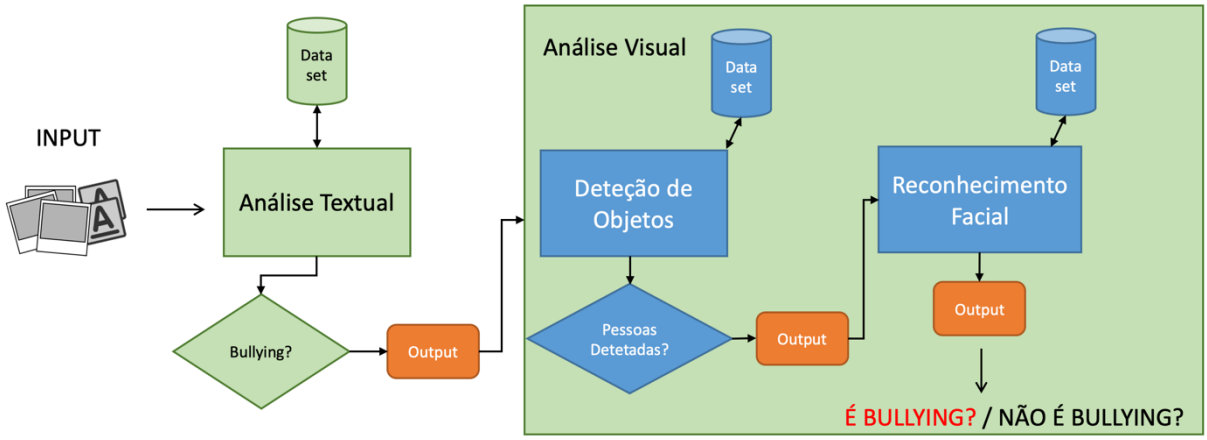

FIGURA 6.1: Arquitetura da Solução

Para o perfeito funcionamento de um sistema que implemente o modelo que aqui está a ser proposto, deverá ser necessária uma configuração prévia, isto no caso de se tratar de uma aplicação onde existam perfis de utilizador, caso contrário apenas a análise textual será possível de forma a ter um resultado com sentido. Essa configuração, deverá prever a inserção de dados relativos aos encarregados de educação, no caso de se pretender seguir crianças de forma a poder ser feita a comunicação de alertas sempre que for suscetível. Informações como email ou telefone serão fundamentais para um acompanhamento daquilo que o sistema possa vir a detetar. Além disto, deve ser possível efetuar o *upload* de várias fotos relativas à pessoa a monitorizar, de forma a que o sistema possa ser treinado para fazer o reconhecimento facial do indivíduo e avaliar a sua presença em imagens que venham a ser analisadas. A estas imagens destinadas para o treino, poderão ainda juntar-se fotos de perfil caso a aplicação onde vai ser feita a análise as disponibilize, para alcançar uma melhor performance e resultado final. Adicionalmente poderá existir uma área para confirmar a possível existência de *bullying* quando o sistema assim o classificar tendo por base o conteúdo analisado, de modo a validar a eficácia do mesmo, para que este depois possa aprender e melhorar. A confirmar-se a classificação de uma publicação como *bullying*, poder-se-á optar por lançar um pedido de suporte/denúncia ao perfil do agressor na aplicação em questão para que a situação seja revista. Contudo, os alertas do sistema que faz a análise devem ser gerados para sensibilizar tanto a vítima como o agressor.

6.2 Principais Casos de Uso

Como referido anteriormente, o conteúdo considerado como essencial para a deteção da prática de *cyberbullying* está disposto sobre a forma de texto ou imagem, como tal, estes são uma parte importante na definição dos principais casos de uso neste problema.

Sempre que é efetuada uma publicação ou comentário, existe a possibilidade de estes estarem a ser utilizados para atacar, ofender, assediar ou ameaçar alguém, pelo que estamos perante uma potencial situação de *bullying* se esta se inserir num dos seguintes cenários:

Casos genéricos:

- o conteúdo publicado é um texto. É composto por características de *bullying* e tem como objetivo atacar, ofender, assediar ou ameaçar alguém;
- o conteúdo publicado é uma fotografia ou um vídeo. Nenhuma pessoa é identificada nas imagens. A descrição presente, caso exista, é composta por características de *bullying* e tem como objetivo atacar, ofender, assediar ou ameaçar alguém;
- em qualquer das situações dos pontos anteriores, se existem comentários por parte dos utilizadores, e o conteúdo destes seja composto por características de *bullying* tendo como objetivo atacar, ofender, assediar ou ameaçar alguém.

Casos em que a vítima seja quem fez a publicação:

- o conteúdo publicado é uma fotografia ou vídeo, e a pessoa que publicou é identificada nas imagens. Existem comentários por parte de outros utilizadores, e o conteúdo destes é composto por características de *bullying* tendo como objetivo atacar, ofender, assediar ou ameaçar alguém.

Casos em que o agressor seja quem fez a publicação:

- o conteúdo publicado é uma fotografia ou vídeo, é identificada uma pessoa nas imagens, mas esta não é a mesma que fez a publicação. A descrição presente, caso exista, é composta por características de *bullying* tendo como objetivo atacar, ofender, assediar ou ameaçar alguém.

Para ter um sistema capaz de corresponder às situações que estes casos referem, será necessário que este consiga analisar e classificar texto, detetar a presença humana e fazer o reconhecimento facial da mesma.

Ao longo dos próximos pontos serão apresentadas as bibliotecas que se consideram ideias para implementação da solução. Tendo por base a utilização do *tensorflow*, será possível desenvolver a solução com recurso aos conceitos e tecnologias apresentadas nos capítulos

anteriores. Tal como referido no seu *website* [71] é rápido e flexível, permitindo que as suas execuções apresentem um desempenho essencial para criar e implementar sistemas de *machine learning.* Além disso, as APIs disponíveis em grande número, praticamente todas *open-source*, facilitam a criação e treino de um modelo para combate ao *cyberbullying.*

6.3 Arquitetura da Solução com Recurso ao Tensorflow

Machine learning é uma temática complexa, mas a implementação de modelos para a resolução dos problemas a que se propõe resolver é muito menos difícil do que costumava ser, graças à existência de várias bibliotecas, como é o caso do *Tensorflow*.

O *Tensorflow* é uma biblioteca de *software open source* para computação numérica que usa gráficos de fluxos de dados. Os nós dos gráficos representam operações matemáticas, enquanto que as arestas representam os vetores de dados multidimensionais, conhecidos como tensores, que comunicam entre eles. A sua arquitetura flexível permite implementar sistemas capazes de serem executados num ou mais *CPUs* ou *GPUs* de um computador pessoal, servidor ou dispositivo móvel apenas com uma única API. É mais rápido, inteligente e flexível que os sistemas anteriores, e por isso pode ser adaptado com mais facilidade a novos produtos e investigações [71].

Foi originalmente desenvolvido por investigadores e engenheiros que trabalhavam na *Google Brain Team,* a parte da empresa americana que está responsável pela investigação direcionada para *machine* e *deep learning*, mas o sistema é genérico o suficiente para ser aplicado numa vasta quantidade de outros domínios. Embora seja principalmente uma ferramenta para *machine learning*, tem uma gama ampla de funcionalidades e é projetado principalmente para modelos de redes neuronais [71].

6.3.1 Arquitetura e Funcionamento do Tensorflow

Esta *framework* oferece uma *programming stack* que consiste em múltiplas camadas de APIs, tal como se pode ver na figura 6.3.1.

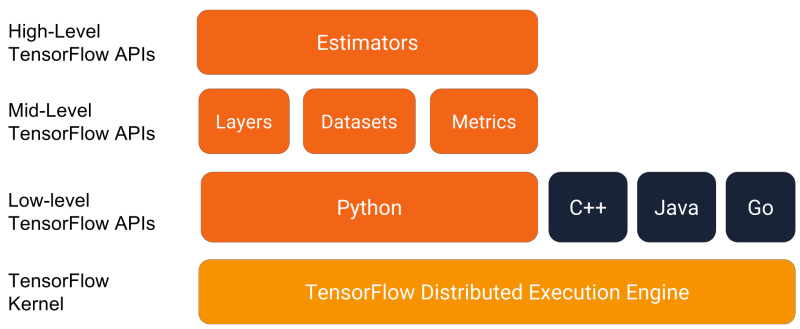

FIGURA 6.3.1: *Tensorflow programming stack*

As APIs de alto e médio nível do *tensorflow* são as mais indicadas para alguém que está a iniciar projetos na área de *machine learning* e que não tem ainda grande experiência com a ferramenta. Compostas essencialmente por *estimators* já pré-preparados, estes níveis permitem especificar arquiteturas predefinidas, como regressão linear ou redes neuronais. Um *estimator* é uma representação de alto nível de um modelo completo, capaz de tratar dos detalhes de inicialização, *logging*, armazenamento e *restore*. Uma instância de um *estimator* encapsula toda a lógica responsável por construir um grafo e executa uma sessão do *tensorflow.* As camadas, *datasets e* métricas que se colocam no nível médio, dizem respeito a bibliotecas reutilizáveis para componentes de modelos comuns. Por sua vez, as APIs de baixo nível permitem construir modelos através da definição de uma série de operações matemáticas, sendo estas maioritariamente programadas em *python* como a maioria dos recursos que se encontram para o *tensorflow*, mas com espaço para outras linguagens como *C++, Java, Go* e *Javascript*. O *kernel* da *framework* pode correr numa ou mais plataformas como *CPUs*, *GPUs* ou *TPUs* [72].

Um dos projetos do *tensorflow*, conhecido como *MobileNet*, consiste em desenvolver um conjunto de modelos de *computer vision* que são concebidos particularmente para trabalhar com a velocidade e eficácia de *trade-offs* que se precisa de ter em conta em dispositivos móveis ou em aplicações embebidas [73].

O contorno da maioria dos programas no *tensorflow* é o seguinte:

- importar e fazer *parse* aos *datasets;*
- criar colunas de características para descrever os dados;
- selecionar o tipo de modelo;
- treinar o modelo;
- avaliar a eficácia do modelo;
- fazer novas previsões a partir do modelo treinado.

Considere-se um sistema que verifica se uma imagem contém um cão ou se contém um gato. Basicamente, o *tensorflow* classifica as camadas de dados, chamadas nós, para aprender se a imagem que está a visualizar diz respeito a um gato. A primeira camada vai pedir ao sistema que olhe para algo tão básico como determinar a forma geral da imagem. O sistema vai avançar para o próximo conjunto de dados, que pode ser por exemplo, procurar patas na foto. O sistema move-se de nó em nó para compilar informação suficiente para dizer o que está presente na imagem.

6.3.2 Tensores e Grafos

Para se conseguir entender melhor o *tensorflow* é necessário perceber os seus dois principais componentes, os tensores e os grafos.

Os tensores são objetos geométricos que descrevem relações lineares entre vetores geométricos, escalares e outros tensores. Exemplos de tais relações são o produto escalar, o produto vetorial e transformação linear. Um tensor é uma generalização de vetores e matrizes para dimensões potencialmente mais altas. Internamente, o *tensorflow* representa os tensores como matrizes de n dimensões de tipos de dados base [74]. Em *deep learning* pretende-se que se pense nos tensores como vetores tridimensionais tal como nos mostra a figura 6.3.2.

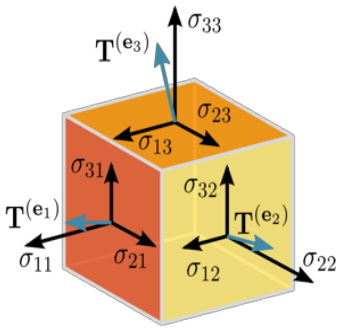

FIGURA 6.3.2: Representação visual de um Tensor

Matematicamente, os tensores são definidos como funções multi-lineares, que consistem em várias variáveis de vetor. Isto significa que os tensores são funções ou *containers* que se precisam de definir, e o cálculo é efetuado quando alimentado com dados. Um tensor consiste num conjunto de valores primitivos moldados numa matriz de um qualquer número de dimensões e tem as seguintes propriedades:

- um tipo de dados (float, int, string, etc.)
- uma forma

Cada elemento no tensor tem o mesmo tipo de dados e estes são sempre conhecidos. A forma, que é o número de dimensões que o tensor tem e o tamanho em cada dimensão, pode ser apenas parcialmente conhecida [75].

Além dos tensores, os grafos são outro componente principal do *tensorflow*. Um grafo é uma série de operações do *tensorflow* dispostas de forma organizada num grafo de fluxo de dados, onde os nós representam unidades de computação e as arestas representam os dados consumidos e produzidos. Cada nó tem zero ou mais tensores como *input* e produz um tensor como *output*, sendo que cada tipo de nó é constante. O *tensorflow* utiliza os grafos de fluxos de dados para representar a computação em termos de dependências entre operações

individuais. Isto leva a um modelo de programação de baixo nível no qual se define primeiro o grafo de fluxo de dados, e depois é criada uma sessão do *tensorflow* para executar partes do grafo num conjunto de dispositivos locais ou remotos [76]. Uma sessão do *tensorflow* encapsula o ambiente no qual os objetos da operação são executados, e os objetos do tipo tensor são avaliados. Estes grafos são então usados para construir modelos para a resolução dos problemas que o *tensorflow* se propõe a resolver.

6.3.3 Aplicações do Tensorflow

Esta ferramenta tem uma vasta quantidade de aplicações práticas e é vista como uma mais valia por parte das empresas para resolver diversos problemas do mundo real. Seguem-se agora alguns dos seus principais *use cases* [77].

6.3.3.1 Reconhecimento de Voz

É já conhecida a possibilidade de pesquisa através da voz, recorrendo a agentes que se ativam pela mesma, com as suas comuns implementações disponibilizadas em *smartphones* e computadores, das quais se destacam *Siri, Google Now* e *Cortana*. Perceção e entendimento de linguagem natural é também um caso comum nesta área, onde muitas das vezes se tenta converter o discurso em texto escrito para depois ser utilizado, por exemplo para legendagem automática.

É possível desenvolver redes neuronais que sejam capazes de efetuar:

- reconhecimento de voz – *Internet of Things* e segurança;
- pesquisa por voz – maioritariamente em telemóveis ou motores de pesquisa;
- análise de sentimento – utilização comum em CRM, para acompanhamento da relação com o cliente;
- deteção de falhas – comum em aviação e automobilística para deteção de avarias através dos barulhos do motor.

6.3.3.2 Reconhecimento de Texto

Semelhante ao ponto anterior, as aplicações baseadas na análise de texto são também casos práticos muito populares do *tensorflow*. A análise sentimental, deteção de ameaças e deteção de fraude podem ser aplicadas em domínios como as redes sociais, organizações governamentais, seguros ou finanças. O *Google* tradutor é capaz de detetar e reconhecer a linguagem e de suportar a sua tradução para mais de 100 idiomas. Além disto, a sumarização de texto é outra das práticas mais comuns, conseguindo reduzir um texto algo longo para um

resumo com a informação essencial ou para construir títulos para notícias mediante o seu corpo. Existem também mecanismos que são capazes de gerar uma resposta automática a um email ou mensagem num *chat*.

6.3.3.3 Reconhecimento de Imagem

O reconhecimento automático de imagens é um dos pontos mais falados na comunidade científica. É maioritariamente usado por redes sociais e fabricantes de dispositivos móveis, para enfrentar situações de reconhecimento facial, pesquisa por imagens semelhantes, deteção de movimentos, *computer vision* e *clustering* de fotos, quer para utilização em questões de segurança, quer para melhorar os seus sistemas de forma a se adaptar mais ao utilizador. Estas situações podem ainda ser aplicadas em áreas automobilística, aviação e indústrias de saúde, para ajudar no desenvolvimento de transportes autónomos e diagnóstico de doenças. O reconhecimento de imagens tem como principal objetivo a identificação de pessoas e objetos nas imagens, assim como entender todo o seu conteúdo e contexto.

Os algoritmos de reconhecimento de objetos que o *tensorflow* executa, classificam e identificam objetos em grandes imagens e são muitas vezes utilizados por redes sociais para identificar as pessoas nas fotografias, como é o caso do *Facebook*.

Esta técnica está a começar a expandir-se para a indústria da saúde, onde também os algoritmos do *tensorflow* podem processar mais informação e detetar mais padrões que os humanos. Os computadores são agora capazes de rever raios X e ajudar a detetar mais doenças do que os próprios médicos.

6.3.3.4 Deteção de Movimento em Vídeo

A análise ao conteúdo do vídeo é outro caso prático desta biblioteca. Recorre-se à mesma para efetuar a deteção de movimento para, por exemplo, verificar ameaças em tempo real em locais como aeroportos e ser um auxílio na segurança dos mesmos, pois pode reconhecer indivíduos que estejam a ser procurados pelas autoridades. Este tipo de implementação é um dos principais motivos para a existência dos carros que são capazes de conduzirem de forma autónoma, pois permite que seja analisado todo o ambiente à volta do veículo detetando todo o movimento existente em seu redor, assim como outras questões como o caso da sinalização rodoviária. Além destes pontos, a *NASA* está também a desenhar um sistema com recurso ao *tensorflow* para classificação da órbita e fazer *clustering* dos objetos relativos a asteroides. O objetivo passa por ser possível classificar e prever objetos que possam passar demasiado próximo do planeta terra e causar perigo para a população.

6.3.3.5 Análise de dados de Time Series

Os algoritmos de *time series* do *tensorflow* são usados para analisar dados das mesmas, de forma a extrair estatísticas com significado. O caso mais comum são os sistemas de recomendação, onde as lojas online poderão analisar os gostos do utilizador, através do tempo que por lá esteve a navegar, para lhe sugerirem produtos pelos quais poderá ter interesse em consultar ou até mesmo comprar. Existem ainda aplicações para gerir e prever os riscos em áreas como finanças, segurança, contabilidade, entre outras, por via da análise dos dados num determinado período.

6.3.4 Contextualização do Tensorflow na Solução

Além dos pontos apresentados, que referem algumas vantagens na utilização do *tensorflow*, a vasta quantidade de APIs disponibilizadas de forma *open-source* na internet são sinais de que propor uma solução que assente no *tensorflow* será algo sustentável. Assim, a solução poderá ser composta por diferentes componentes de análise e facilmente ajustável e adaptável sempre que novos indicadores surjam no que toca ao problema do *cyberbullying* ou para qualquer outro problema idêntico que se identifique e pretenda combater. O célere desenvolvimento de um novo componente ou a sua facilidade e rapidez de execução em qualquer dispositivo, sem necessitar de um hardware muito avançado, permitem que se consiga ter este sistema distribuído por um conjunto de diferentes ambientes. Além disso, o sistema poderá alcançar uma alta percentagem de sucesso na previsão de uma situação de *bullying,* seja qual for a quantidade de dados que terá de processar nas fases de treino, teste ou execução final em produção.

A ideia para a presente solução passa por separar os diferentes tipos de análise em diferentes componentes. Deste modo, será mais fácil proceder a correções e adaptações em qualquer uma das análises assim que novas informações, que possam influenciar o resultado final, forem obtidas. A substituição ou introdução de um novo componente ficará facilitada se não existir um único componente que agregue todas as funcionalidades e reduza a escalabilidade da solução. À partida para a arquitetura da solução, pretende-se ter um conjunto de três componentes principais. Um componente de análise textual para verificação da existência de características de *bullying*, um componente de deteção de objetos para procurar identificar os elementos presentes numa imagem ou vídeo, e uma componente de reconhecimento facial que permita verificar quem são as pessoas que possam estar presentes no conteúdo multimédia que exista durante uma análise.

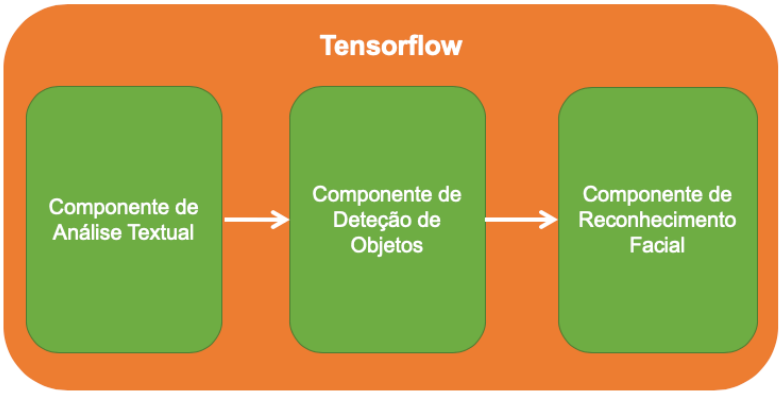

FIGURA 6.3.4: Arquitetura da solução com recurso ao *Tensorflow*

O *tensorflow* permite essencialmente criar programas que façam os seus cálculos com base em redes neuronais, contudo, é também extremamente eficaz quando utilizado para implementar algoritmos de regressão linear ou não linear para efetuar diferentes tarefas de classificação. Como tal, e tendo em conta as tarefas que se pretendem efetuar durante o processo de classificação de uma possível situação de *bullying*, recorrer-se-á a um conjunto de redes neuronais para efetuar as classificações a que se destina cada um dos componentes indicados. O primeiro componente a ser executado deverá receber como *input* os dados de uma nova publicação para análise, e os seguintes componentes deverão receber como *input,* o *output* do seu antecessor, até que o último componente retorne o *output* final da análise. No geral, quase que se poderá equiparar a solução com uma única rede neuronal, onde se pode equiparar cada componente com uma camada de processamento dessa rede, que vai passando o seu *output* para a camada seguinte.

Os próximos pontos apresentam em maior detalhe os componentes a serem executados no *tensorflow*, que se acreditam ser os mais indicados para classificar uma qualquer possível situação de *bullying* com alta performance e num curto espaço de tempo.

6.4 Componente de Análise Textual

Para este modelo, a análise textual terá uma grande influência na identificação de um certo conteúdo dizer respeito a uma situação de *bullying* ou não. O processamento de linguagem natural diz respeito à criação de sistemas que processem ou entendam linguagem de forma a cumprir certas tarefas. As tarefas mais comuns dizem respeito à resposta a questões, como nos casos de tecnologias como a *Siri*, a reconhecimento de discurso falado para fazer a sua transcrição para texto, ou até mesmo traduzir automaticamente um texto entre diferentes idiomas. Além destes casos, é ainda muito comum recorrer a este processamento para fazer uma análise ao sentimento de uma frase ou até mesmo para a deteção de mensagens de *spam.* Ora, identificar se uma frase tem características de *bullying* é um ponto muito

semelhante aos dois anteriores, pelo que se poderá recorrer a técnicas e a mecanismos semelhantes para fazer a implementação no modelo que se está a construir.

As técnicas e os algoritmos existentes para processar linguagem natural são bastantes, assim como dados já pré-processados que facilitam o processo inicial de construção de um *dataset* de frases assinaladas com as identificações para as classes finais pretendidas, que neste caso em concreto serão *bullying* e não *bullying*. Como foi visto em estudos anteriores, uma frase muito provavelmente será considerada *bullying* se contiver palavras ou partes de texto que contenham insultos. Hieróglifos como "*@ss*", entre outros, também poderão ser contemplados desde que existam em relativa frequência no *dataset* que irá ser usado para treino. Desse *dataset* gerado, uma técnica a que habitualmente se recorre para ter alguma consistência nos resultados passa por ter mais ou menos o mesmo número de entradas para cada uma das classes. Além disso, tal como nos casos anteriores, devem utilizar-se frases diferentes no treino e na fase de testes de modo a evitar o *overfitting*, pelo que em última instância se poderá dividir o *dataset* construído para se ficar com 80% dos dados disponíveis para serem utilizados na fase de treino, e os restantes 20% para validarem o classificador na fase de testes [78].

6.4.1 Vetor de Palavras (Word Embeddings)

Neste caso em concreto, onde se pretende classificar texto como sendo ou não relativo a *bullying* imagina-se o seguinte *pipeline* (figura 6.4.1.1).

FIGURA 6.4.1.1: *Pipeline* de análise textual de *bullying*

Contudo, este tipo de *pipeline* é problemático, pois não existe forma de efetuar operações diretamente com estas frases. Assim sendo, como os algoritmos de *machine learning* utilizam números como *input*, é necessário converter cada palavra do texto que é alvo de análise para vetores numéricos (*word embeddings*). Aqui ocorre o processo de *tokenization*, que divide o texto em palavras, e o processo de *vectorization*, que define uma boa medida numérica para caracterizar cada parte. De notar que a pontuação é ignorada neste passo. Pretende-se ainda que estes vetores sejam criados de forma a que se mantenha o contexto, significado e semântica da palavra. Pretende-se que palavras idênticas se posicionem mais ou menos na mesma zona no que diz respeito ao espaço vetorial. Como se vê na figura 6.4.1.2, as palavras "*Love*" e "*Adore*", por terem definições semelhantes, estão muito próximas uma da outra no gráfico. Já a palavra "*Golf*", que nada tem em comum com as anteriores, aparece numa posição bem afastada das outras duas [79].

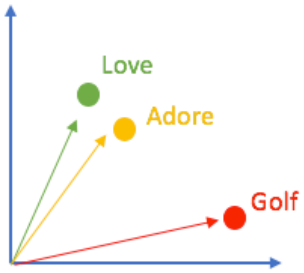

FIGURA 6.4.1.2: Representação das palavras no espaço vetorial (*Word Embeddings*)

Para criar estes *word embeddings* recorre-se à ferramenta *word2vec*, capaz de gerar os vetores tendo em conta o contexto em que a palavra aparece na frase em análise. Tendo em conta as palavras anteriores, consideram-se as seguintes frases onde estas palavras estão presentes:

I love taking long walks on the beach.
My friend told me that they love popcorn.

The relatives adore the baby's cute face.
I adore his sense of humor.

LISTAGEM 6.4.1: Comparação da importância das palavras na frase

Tendo em conta o contexto das frases, é possível verificar que ambas as palavras são geralmente usadas em frases com conotações positivas, que não seriam consideradas *bullying*, e normalmente precedem substantivos ou frases nominais. Estes sinais mostram que estas palavras possuem características em comum, pelo que poderão ser sinónimos. O contexto é também muito importante quando se considera a estrutura gramatical da frase. A maior parte das frases segue a estrutura tradicional de ter verbos após os substantivos, e por esse motivo, é mais provável que estes substantivos apareçam todos em torno da mesma área no espaço vetorial. Ao gerar os vetores para cada palavra distinta na frase, obtém-se uma matriz composta pelos mesmos, que será depois utilizada como *input* para a análise. Esta ferramenta pode ser treinada com um *dataset* genérico que contenha uma vasta quantidade de texto no idioma pretendido, de forma a alcançar o maior número possível de palavras e ter esses valores já definidos no gráfico.

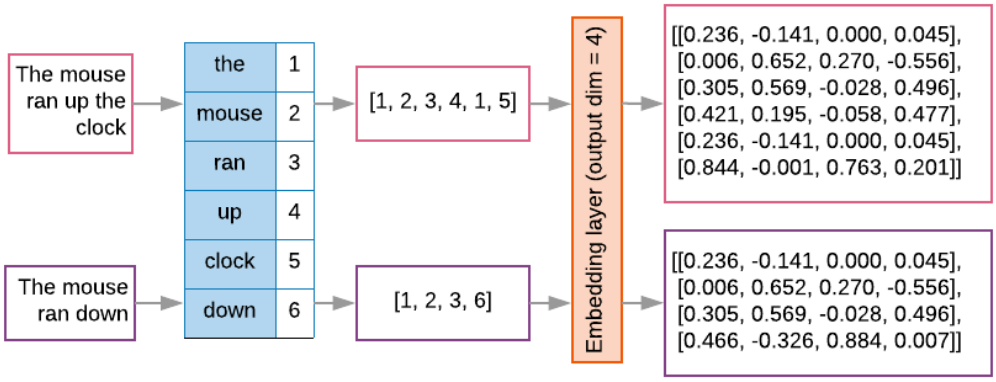

FIGURA 6.4.1.3: Geração de *Embeddings*

Na figura acima é possível ver a sequência gerada para as duas frases de exemplo e os respetivos *embeddings*. Cada palavra tem o seu vetor definido e estes são apresentados pela ordem pela qual a frase que representam aparece na frase. Caso a mesma palavra apareça múltiplas vezes, o respetivo vetor aparecerá, naturalmente, repetido. São estes os resultados dos processos de *tokenization* e *vectorization* referidos [80].

Nem todas as palavras contribuem para a análise e identificação da categoria final. Pode otimizar-se o processo de aprendizagem, descartando do vocabulário palavras raras ou irrelevantes. Normalmente, usar as 20 mil palavras mais frequentes é suficiente. No entanto, como todas as frases têm dimensões diferentes e os *inputs* para a rede devem ter o mesmo tamanho, é necessário definir um tamanho máximo. Para isso, deve-se analisar os tamanhos médios das frases do *dataset*, e preencher as frases mais pequenas até esse máximo, com um *token* que se saiba que será único e que não existirá no *dataset*. Isto irá acontecer na primeira camada da rede, onde os dados serão tratados [81].

Como neste caso se tem apenas duas classes, o output deverá apresentar um valor probabilístico para cada uma delas. Para isso, a função de ativação da última camada deve ser uma função *sigmoid* – mapeia o *output* de regressão logística para um valor de probabilidade entre 0 e 1 (figura 6.4.1.4) [78].

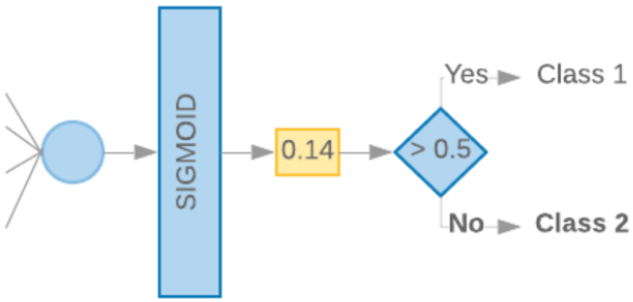

FIGURA 6.4.1.4: Última camada da classificação (função de *sigmoid*)

Com os valores definidos para cada palavra, e com as frases com a identificação atribuída para o treino, a rede consegue aprender quais as partes do texto que são fulcrais para que

este seja considerado como *bullying*, um pouco à semelhança do cérebro humano, pois se numa frase se identificar um insulto, a probabilidade de se estar perante uma situação deste género é elevada. Contudo, a fase de treino é essencial para garantir que os valores dos *embeddings* se ajustam à realidade, podendo ser necessário alterar os pesos de um ou outro para alcançar melhores resultados.

6.4.2 Output do Componente

Este processo de verificação de semelhança entre as frases acaba por ser bastante idêntico àquele que foi apresentado em relação ao reconhecimento facial, muito por força da utilização, novamente, de *embeddings*. O *output* esperado deve ser um valor probabilístico para as classes definidas, podendo depois dar-se seguimento ao *workflow* mediante os valores aqui obtidos para cada frase em análise.

6.5 Componente de Deteção de Objetos em Imagem

Um dos principais focos desta solução prende-se com a análise de imagens, especialmente tendo como objetivo detetar a presença humana nas mesmas. Muitas das vezes, as pessoas confundem os cenários de classificação e de deteção de objetos. No geral, se se pretende classificar uma imagem numa determinada categoria, recorre-se à classificação de imagens. Por outro lado, se se pretende identificar a disposição de objetos numa imagem, e, por exemplo, contar o número de instâncias de cada objeto, opta-se pelo método de deteção de objetos. Isto implica que o modelo consiga prever quer a classe do objeto, quer a sua localização, sendo que o resultado é normalmente apresentado sob a forma de uma caixa à volta do objeto juntando-se a respetiva identificação. Para treinar este tipo de modelo de deteção de objetos é necessário ter dados identificados, ou seja, imagens com as coordenadas dos limites das caixas delimitadoras e os nomes dos objetos [82]. Este tipo de modelo é bastante comum em, por exemplo, problemas de condução autónoma, pois é essencial para efetuar o reconhecimento da envolvência das estradas, ao detetar os sinais rodoviários ou os obstáculos.

A *Google* desenvolve sistemas de *machine learning* para serem utilizados em tarefas de *computer vision* que são usados para melhorar os seus produtos e serviços, e também para impulsionar o progresso da investigação na comunidade. Criar modelos de *machine learning* apurados, capazes de localizar e identificar múltiplos objetos numa única imagem, continua a ser um desafio fulcral no campo e os investigadores estão a investir algum tempo no treino e em experiências com estes sistemas. Em outubro de 2016, o sistema alcançou novos resultados e desde aí gerou vários ganhos para um conjunto de publicações de investigação

tendo sido colocado em produção em aplicações da *Google* como na pesquisa por imagens e na deteção de números e nomes de ruas no *street view* dos seus mapas.

Este mecanismo está agora disponível para todos através da sua implementação no topo do *tensorflow*, o que torna mais fácil a construção, treino e *deploy* de modelos de deteção de objetos [83].

Tipicamente, existem três passos numa *framework* de deteção de objetos [84]:

1) utilização de um modelo ou algoritmo para gerar regiões de interesse. Estas regiões são um conjunto de caixas delimitadoras que abrangem toda a imagem e, assim, delimitam os diferentes objetos detetados;
2) extração de características visuais em cada uma das caixas delimitadoras, que são depois avaliadas para determinar quais os objetos que possam estar presentes nas regiões. Este passo torna-se um pouco num componente de classificação de objetos;
3) pós-processamento, onde as caixas sobrepostas são combinadas numa única caixa delimitadora.

Tendo em conta que se pretende recorrer ao *tensorflow* para definir a proposta de solução para o problema, serão exploradas e apresentadas algumas das possíveis bibliotecas que este pode executar para identificação de conteúdo numa imagem.

A API *object detection* é aquela que torna mais fácil a construção, treino e implementação de modelos de deteção de objetos com as características apresentadas acima. Esta API foi treinada com o *dataset COCO* [85] – *common objects in context* – que contém mais de 300 mil imagens dos 90 objetos mais comuns que se encontram nas mesmas (Figura 6.5).

FIGURA 6.5: Categorias dos objetos do *dataset COCO*

Dos modelos disponibilizados, os *SSD – single shot detector* – que usam a *MobileNet* [86] são mais leves, portanto podem ser confortavelmente executados em tempo real em dispositivos móveis. Alternativamente, pode-se optar pelos modelos *Faster RCNN,* que são computacionalmente mais intensivos, mas significativamente mais eficazes. No total são cinco modelos diferentes que variam entre a velocidade de execução e a precisão na colocação das caixas delimitadoras. O indicador *mAP – mean average precision* – que é apresentado para

cada modelo, representa o produto da *precision and recall* na deteção de caixas de delimitação. É uma boa medida combinada de quão sensível é a rede para os objetos de interesse e quão bem esta evita os falsos positivos. Quanto mais alta a pontuação *mAP*, mais precisa é a rede, contudo isto também acarta o custo da velocidade de execução. No reconhecimento de padrões, obtenção de informações e classificação binária, a precisão (valor preditivo positivo) é a fração de instâncias relevantes entre as instâncias obtidas. Já o *recall* (sensibilidade) é a fração de instâncias relevantes que foram obtidas ao longo da total quantidade de instâncias relevantes [87].

6.5.1 Extração de características

O objetivo da extração de características é reduzir uma imagem de tamanho variável para um conjunto fixo de características visuais de alto nível. Nas *frameworks* de deteção de objetos, usam-se tipicamente modelos de classificação de imagens já pré-treinados para extrair as características visuais, pois estes tendem a generalizar bastante bem. Por exemplo, um modelo treinado no *dataset COCO* é capaz de extrair características bastante genéricas, como o caso do modelo *Inception V3* [88] implementado no *tensorflow* [89].

Existem vários tipos de extrações possíveis sendo que as características mais comuns neste processo dizem respeito a cores, textura e forma. A primeira é uma das mais importantes pois é o efeito visual mais expressivo numa imagem, sendo que o facto de ter um significado semântico mais pequeno torna este tipo de características mais independentes do domínio comparando com outras, juntando-se ainda a sua invariância no que diz respeito ao tamanho da imagem e orientação dos objetos A textura tenta quantificar as qualidades intuitivas descritas por termos como aspereza, suavidade, sedosidade ou algo acidentado como uma função de variação espacial nas intensidades dos pixéis. Por sua vez, a forma é também extremamente importante neste domínio, estando esta geralmente descrita quando a imagem é segmentada em diferentes regiões ou objetos. Para a característica da forma ser boa, esta não deve variar consoante o tamanho ou rotação da imagem [90].

6.5.2 Propostas de Região

Fica então percetível que o principal componente da deteção de objetos numa imagem passa por identificar as regiões da imagem onde estes estarão. Existem várias abordagens diferentes para gerar propostas de regiões. Uma muito comum passa por usar um algoritmo de pesquisa seletiva, que se baseia em tentar agrupar pixéis num *cluster* e depois gerar as propostas com base nesses *clusters*. Isto pode levar a que muitas caixas sejam dispostas na imagem e que haja alguma sobreposição. Para reduzir o número de deteções numa imagem para um número de objetos presentes mais realista, é comum recorrer-se à técnica de *non-*

maximum supression. Isto é algo muito comum se um objeto numa imagem for relativamente grande e se tiverem sido geradas mais de 2000 propostas de região, é bem provável que algumas delas tenham sobreposições entre si e com o objeto. Outras abordagens populares consistem na utilização de recursos visuais mais complexos que são extraídos da imagem ou no recurso a técnicas como *sliding window*. Estas abordagens de *sliding windows* são como que uma janela que desliza pela imagem, sobre várias proporções e escalas para se ajustarem mediante o tamanho da mesma, em consecutivas iterações. Neste caso as regiões são geradas automaticamente, sem ter em conta as características da imagem. Este tipo de abordagem é a mais utilizada atualmente. Na figura 6.5.2 é possível ver a diferença entre as abordagens de pesquisa seletiva e de *sliding window* (nesta, imagine-se a caixa presente a deslizar horizontalmente, baixando um pouco cada vez que atinge o limite lateral da imagem) [84] [91].

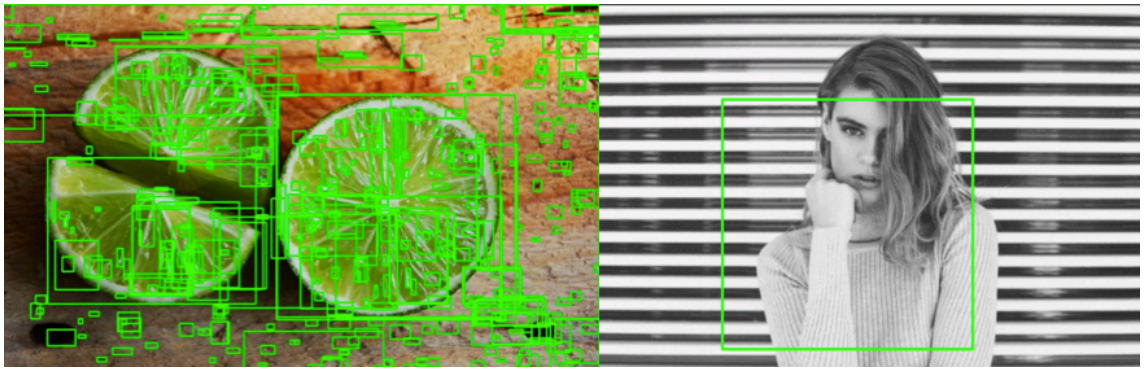

FIGURA 6.5.2: Pesquisa seletiva vs. *Sliding window* para gerar propostas de região

Quanto mais regiões se geram, maior é a probabilidade de se encontrar um objeto. Por outro lado, se se procurar gerar todas as propostas possíveis, não será possível executar o detetor de objetos em tempo real, por exemplo.

6.5.3 Modelos SSD (Single Shot Detector)

Os modelos com arquitetura *SSD* fazem a deteção de objetos prevendo diretamente as classes com recurso a uma única rede neuronal, sem necessitarem de uma segunda fase em cada operação de classificação. É usado um conjunto de caixas com diferentes proporções e escalas por defeito, e é depois aplicado o mapa de características. Como esses mapas de recursos são calculados passando uma imagem através de uma rede de classificação de imagens, a extração de características para as caixas delimitadoras pode ser extraída num único passo. São geradas pontuações para cada categoria de objeto em cada uma das caixas delimitadoras [82].

6.5.4 Modelos Faster RCNN

Os modelos com arquitetura *Faster RCNN* adotam uma abordagem em duas fases para fazer a deteção de objetos. A primeira, conhecida por rede de regiões propostas (*RPN*), funciona com *sliding windows* nas imagens que foram processadas por um extrator de características, e estas são depois usadas para prever as propostas de caixas delimitadoras, em vez de as colocar por defeito como nos modelos *SSD*. Na segunda fase, estas propostas de caixas (tipicamente cerca de 300) são utilizadas para cortar características do mapa de características gerado inicialmente, de modo a prever as classes e a refinar as caixas de cada proposta [82].

6.5.5 Output do Componente

Na figura 6.5.5, é possível ver um *output* de exemplo da API de deteção de objetos executada no *tensorflow*, onde se conseguem encontrar as caixas a delimitarem os objetos relevantes e as suas percentagens probabilísticas de correspondência à classe atribuída. É também possível verificar que mesmo em objetos de reduzidas dimensões a API mostra um desempenho muito bom, indicando valores de percentagem altos e que, como se pode comprovar, correspondem à classe correta. Sabe-se que estes valores poderão variar consoante o modelo escolhido, qualidade dos dados de treino e *hardware* onde a API é executada, contudo é fácil verificar que com esta é possível obter resultados muito positivos e esta será uma escolha acertada para aplicar na resolução do problema.

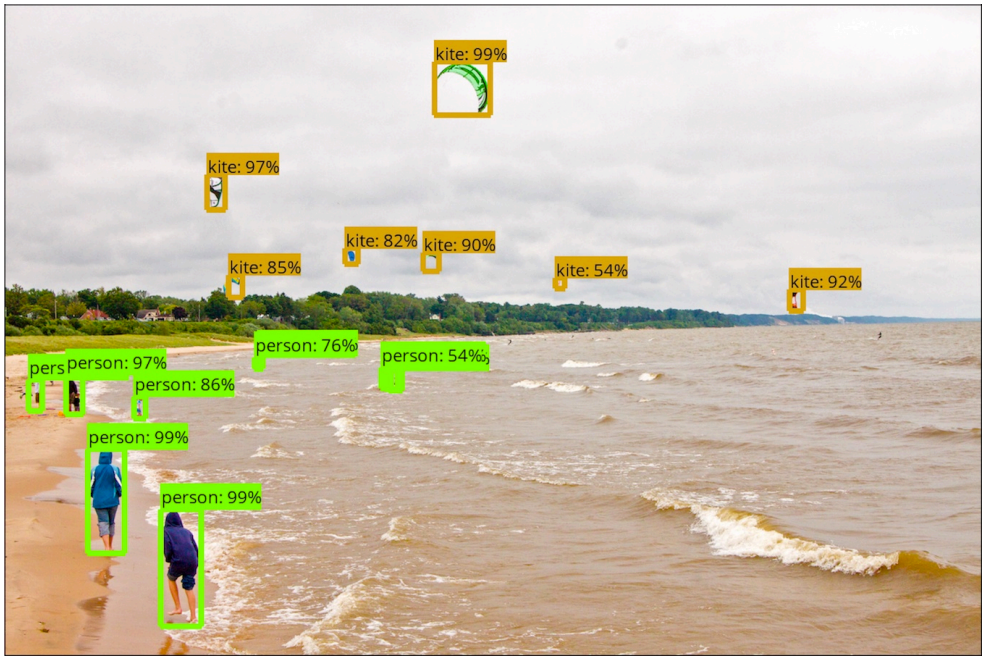

FIGURA 6.5.5: Exemplo de output da API *Object Detection*

Dada uma imagem para análise, espera-se que a API consiga verificar os objetos presentes e detetar a presença humana para avançar com o fluxo da resolução do problema. No futuro, poderão adicionar-se novas funcionalidades para melhorar a resolução do problema tendo em conta a introdução da análise a outros tipos de objetos presentes na imagem. Além de fotografias, a API também pode ser executada em vídeo [92].

Esta API é a selecionada para fazer parte da proposta de solução pois será aquela que melhor poderá responder às necessidades do problema. Contudo, durante o trabalho de pesquisa de APIs foi encontrado o modelo *Show and Tell* [93], também conhecido como *img2txt*, que consegue detetar o conteúdo presente numa imagem e descrevê-lo numa frase, algo que depois implicaria analisar o texto da frase e entender o seu contexto, assim como que para avançar para uma tarefa de reconhecimento facial não acrescentaria muito valor. No entanto, é mais uma opção a ter em conta e poderá vir a ser incluída na solução como melhoria futura.

Assim, detetada a presença humana no novo conteúdo publicado, dever-se-á avançar para tarefas de reconhecimento facial de modo a verificar se a pessoa detetada é alguma das envolvidas numa potencial situação de *bullying*, isto no caso de estarmos perante uma aplicação onde existem perfis pessoais, como foi referido anteriormente.

6.6 Componente de Reconhecimento Facial

Tal como foi referido na explicação inicial da proposta de solução, após se detetar a presença humana numa imagem, avança-se para um processo de reconhecimento facial, de modo a procurar identificar se uma pessoa que possa estar presente é a mesma que faz a publicação, tentando assim entender o contexto de uma potencial situação de *bullying*.

Reconhecimento facial é uma solução biométrica que mede características únicas sobre a cara de alguém. Existem várias aplicações que utilizam esta técnica, em tarefas como efetuar check-ins in voos, identificar automaticamente amigos numa fotografia, abrir uma porta de um edifício ou até para se desbloquear o telemóvel. Esta técnica permite verificar a identidade da pessoa que está a utilizar o sistema onde este reconhecimento está a ser realizado, e é uma forma de tornar mais seguro o acesso a informação privada, não permitindo que se tente o acesso por via de um pin ou uma palavra-passe, que mesmo com um número de combinações bastante elevado, são métodos bem mais vulneráveis.

O início deste tipo de investigação começou há muito tempo, quando em 1960 Bledsoe [94] utilizou uma técnica que consistia em marcar as coordenadas das características proeminentes de um rosto, como o cabelo, olhos, nariz ou boca, pois são aquelas que permitem representar unicamente uma face. O matemático descreveu ainda alguns problemas que, mesmo após 60 anos, ainda se mantêm quando se pretende efetuar esta

tarefa, tais como variações na iluminação, rotação da cabeça, expressão facial e envelhecimento.

Atualmente as coisas são um pouco diferentes, pois pretende-se que os computadores sejam capazes de identificar estas características de forma autónoma, aproveitando ainda a vasta quantidade de imagens disponíveis atualmente. Contudo, essa mesma quantidade pode trazer vantagens no que toca à maior facilidade de encontrar correspondência, mas se a implementação for inadequada, o sistema pode tornar-se extremamente lento. O *Facebook* já implementou um sistema de reconhecimento, o *DeepFace*, para sugerir a identificação das pessoas presentes numa nova fotografia adicionada à sua rede, com uma execução bastante rápida e com uma eficácia de 98% [95].

O reconhecimento facial consiste numa série de problemas relacionados [96]:

- encontrar um rosto numa imagem;
- focar um rosto de cada vez e perceber que mesmo que o rosto esteja inclinado de um modo estranho ou com uma má iluminação, ainda diz respeito à mesma pessoa;
- ser capaz de identificar características únicas no rosto que possam ser depois usadas para diferenciar de outras pessoas, tal como quão grande é a boca, quão longa é a face, e assim por diante;
- comparar as características únicas do rosto em foco com todas as pessoas já conhecidas e determinar o seu nome.

Este tipo de tarefas são todas automaticamente efetuadas pelo nosso cérebro de uma forma instantânea. A capacidade dos humanos identificar rostos é tão boa que por vezes até consegue ver uma espécie de face em alguns objetos, como em frentes de carros ou pedaços de madeira [97]. Por sua vez, os computadores não têm esta capacidade de generalização de alto nível, pelo que é necessário ter um sistema que aprenda a fazer cada tarefa separadamente. Para tal, deve cumprir-se com a seguinte sequência, transportando os resultados de cada ponto para o ponto seguinte:

- encontrar um rosto na imagem (neste caso, após ter sido detetada a presença de uma pessoa com a API de deteção de objetos);
- analisar as características faciais desse rosto;
- comparar com as faces que já se conhecem após o treino do classificador;
- fazer a previsão do nome da pessoa a que corresponde o rosto.

6.6.1 Método HOG

O primeiro foco passa por identificar o rosto na imagem. Esta funcionalidade é muito comum especialmente nos *smartphones*, onde é possível verificar a deteção das faces quando se está prestes a tirar uma fotografia, para focar e obter uma melhor imagem. O método *HOG* (*histogram of oriented gradients*) [98], muito comum para efetuar esta tarefa, começa por converter a foto para tons de preto e branco para depois analisar cada pixel da imagem. Em cada pixel procura-se ver o quão escuro este é, e qual a diferença para os pixéis circundantes, assinalando com uma seta a direção em que a imagem fica mais escura. Ao concluir este processo, as setas apresentarão gradientes que mostram o fluxo da parte mais clara para a mais escura da imagem, e vão-se começando a notar os primeiros pontos de referência. Isto leva a um detalhe em demasia pelo que se deve dividir a imagem em pequenos quadrados de 16x16 pixéis, e depois contar as direções das setas presentes em cada quadrado. Deve-se substituir as setas anteriores com uma única seta com a direção predominante da contagem efetuada, reduzindo no final, o total de setas presentes na imagem o que torna o gradiente mais percetível. O gradiente formado coincide com a estrutura base de um rosto, como se pode ver na figura 6.6.1.

FIGURA 6.6.1: Representação *HOG* do rosto de Barack Obama

6.6.2 Face Landmark Estimation

Após obter representação do ponto anterior, é necessário posicionar o rosto de modo a que fotografias da mesma pessoa não pareçam de pessoas diferentes quando estas se apresentam posicionadas de várias formas ou com diferenças na iluminação. Para fazer isto, ajusta-se a imagem de modo a que os olhos e os lábios estejam sempre no mesmo lugar para todas as fotos (normalmente ao centro). Esta tarefa é realizada com recurso ao algoritmo de *face landmark estimation* [99], que basicamente identifica 68 pontos específicos que existem ao longo do rosto, como a ponta do queixo, as bordas dos olhos, sobrancelhas, nariz, etc. Por

exemplo, é também com esta técnica que são aplicados filtros com objetos ou figuras sobre as caras dos utilizadores em algumas aplicações das redes sociais.

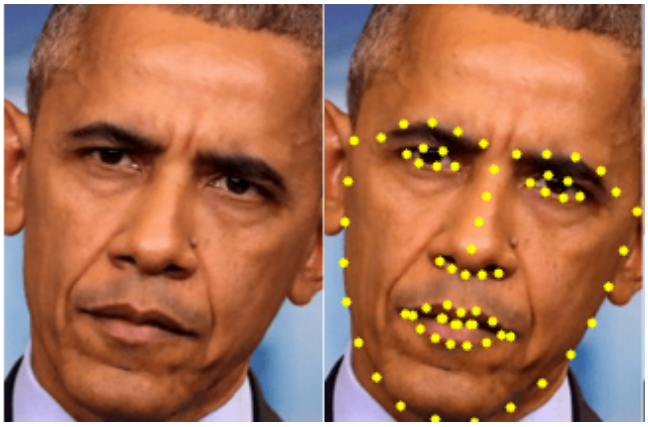

FIGURA 6.6.2: *Face landmark estimation* do rosto de Barack Obama

Tendo agora a referência destes pontos, e sabendo onde se encontram os olhos e a boca, pode-se simplesmente distorcer ou girar a fotografia de modo a que esses pontos fiquem centrados da melhor maneira, e tornar o passo seguinte o mais preciso possível.

6.6.3 FaceNet

Chegando a esta fase fica apenas a faltar identificar a pessoa a que corresponde o rosto detetado. A abordagem baseia-se em comparar o rosto detetado no passo anterior com todos os rostos já previamente identificados, e caso haja correspondência, estar-se-á perante o mesmo indivíduo. Contudo, correr este tipo de análise comparativa numa aplicação com um número bem alargados de utilizadores em tempo útil é praticamente impossível. Para contornar este problema, deve-se deixar o algoritmo identificar as principais características e fazer medições das mesmas para depois iniciar a comparação. Para tal, é importante conhecer o conceito de *embedding*. *Embedding* significa converter dados numa representação de características, normalmente um vetor numérico, onde certas propriedades podem ser representadas por noções de distância. Estes números podem depois ser representados no espaço vetorial, o que permite verificar a distância entre os pontos de dois vetores. É precisamente assim que se consegue solucionar o problema relatado, mapeando os principais pontos do rosto num *embedding* e depois compará-los com os *embeddings* dos rostos já analisados anteriormente. Se os pontos no espaço vetorial forem próximos, deveremos estar perante a mesma face, caso contrário, as faces deverão ser diferentes. Os *embeddings* são gerados com recurso a uma rede neuronal convolucional (*CNN*), onde primariamente devem ser colocadas imagens para treino, gerando *embeddings* semelhantes para fotos da mesma pessoa. Os *embeddings* gerados normalmente têm a dimensão de 128 posições. Assim, a rede irá aprender a gerar as medidas das características para cada pessoa

e vai armazenar esses dados nos *embeddings* gerados, e por exemplo, 10 fotografias diferentes da mesma pessoa, devem gerar 10 *embeddings* de medidas muito próximas. Não se consegue entender ao certo a que parte cada número representado no *embedding* diz respeito, nem mesmo o seu valor, mas o que realmente é importante é que a rede seja consistente a gerá-los para se avançar para as comparações.

Tendo as medidas das principais características das faces disponíveis no espaço euclidiano, basta avançar com a comparação desses valores com todos os valores que já foram gerados pela rede. Recorrendo a um classificador *SVM*, que receba como *input* as medidas presentes no *embedding*, será possível saber a que entidade corresponde a face presente numa fotografia, pois o *output* deste classificador deverá ser o nome da pessoa correspondente, tudo isto numa execução que demorará um reduzido espaço de tempo.

Esta técnica é conhecida por *FaceNet* [72] e foi apresentada pela *Google* em 2015. Os resultados que a empresa apresentou relativamente aos testes efetuados à altura, ultrapassaram os valores da *DeepFace*, alcançando uma percentagem de acerto entre os 98 e os 99, um valor muito próximo do nível humano. Os testes foram realizados com recurso ao *dataset LFW* (*labeled faces in the wild*) [73], que contém mais de 13 mil imagens com o nome da pessoa correspondente.

6.6.4 Output do Componente

Como foi relatado anteriormente, o *output* que se obtém num sistema de reconhecimento facial será constituído pelo nome da pessoa a que se pensa corresponder a face presente numa fotografia em análise e a respetiva probabilidade. Se se pretender, pode-se adicionar uma caixa à volta da face identificada, um pouco à semelhança do que acontece com o modelo de deteção de objetos, com os mesmos valores relativos ao nome e probabilidade, como se pode ver na figura seguinte.

FIGURA 6.6.4: Output do componente de reconhecimento facial

Este tipo de abordagem pode igualmente ser aplicada em vídeo, bastando que se recorra a uma biblioteca capaz de ler o conteúdo de ficheiros destes formatos, e que os divida por *frames*, transformando então os vídeos em conjuntos de imagens que poderão ser analisadas a cada momento, precisamente com as técnicas descritas anteriormente. O estudo da *FaceNet*, mostra no entanto, que a percentagem de acerto nestes casos reduz para um valor máximo de acerto de cerca de 95%, o que não deixa de ser um número bastante interessante.

Contextualizando esta funcionalidade na solução atual, se o *output* apresentar valores altos no que respeita à probabilidade de a pessoa que publicou o conteúdo estar presente no mesmo, pode-se seguir o caminho de análise de comentários para verificar se alguém tenta ofender essa mesma pessoa. Caso contrário, a pessoa que fez a publicação não deverá ser vítima de situação de *bullying*, mas poderá ser o agressor, pelo que a presença de conteúdo textual deve também ser analisado.

No próximo ponto será descrito o *workflow* do modelo e o que sucede em cada passo mediante os *outputs* obtidos para cada um dos componentes utilizados para a construção da solução final.

6.7 Workflow da Solução

Conhecendo-se os pormenores técnicos para implementação da solução, é agora tempo de explicar como estes componentes se interligam e qual a sequência que o sistema vai seguindo mediante os *outputs* que for obtendo em cada etapa.

Este modelo pode ser aplicado de diferentes formas. Contudo, deve existir uma área de preferências da aplicação, onde se ative a sua execução e se defina quais as pessoas que deve seguir para fazer a análise. Imagine-se os encarregados de educação de uma criança, que podem definir que pretendem que este acompanhamento seja feito sempre que esta fizer uma nova partilha, que haja algum tipo interação nos seus conteúdos, ou que seja identificada numa publicação. Aqui poderão também adicionar fotografias de pessoas para auxiliar a tarefa de reconhecimento facial, e consultar o histórico de alertas sempre que uma situação de *bullying* seja identificada.

Tendo em conta os principais casos de uso já apresentados neste capítulo, o principal *workflow* da solução deve ser o seguinte (figura 6.7), assumindo uma aplicação que contenha perfis de utilizador, como o caso das redes sociais:

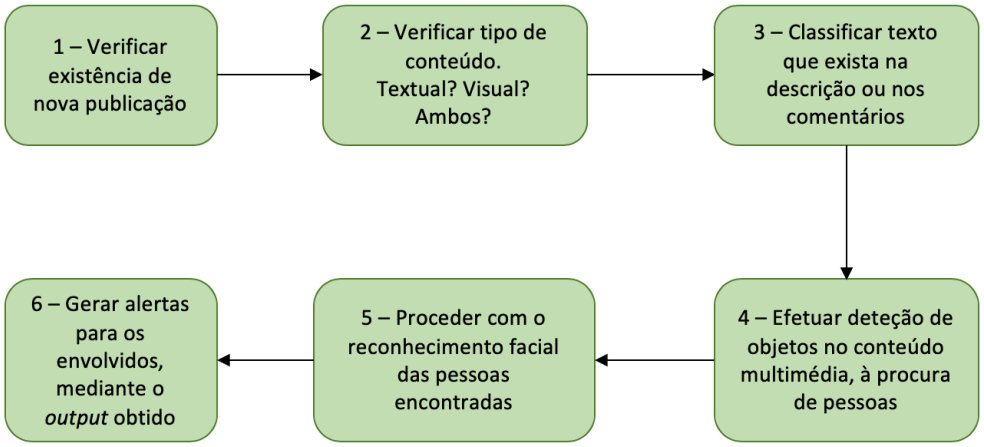

FIGURA 6.7: Principal workflow da solução

1 - Verificar existência de uma nova publicação: quando se pretende ter o sistema a fazer a análise a um determinado perfil de um utilizador, é necessário fazer com frequência uma verificação da existência de uma nova publicação que tenha realizado, ou onde possa ter sido identificado;

2 – Verificar tipo de conteúdo: é depois necessário verificar se a nova publicação contém texto para análise de forma a classificar o mesmo como sendo passível de *bullying*, e ainda complementar a análise verificando se existe algum conteúdo multimédia;

3 – Classificar texto encontrado: seja uma descrição ou um comentário, o conteúdo deve ser analisado e classificado de forma a saber se este texto diz respeito a uma situação de *bullying*;

4 – Procurar presença humana no conteúdo multimédia: caso esteja algum conteúdo multimédia acoplado ao texto, como uma imagem ou um vídeo, deve ser efetuada uma deteção de objetos na tentativa de identificar pessoas;

5 – Reconhecer a pessoa presente no conteúdo: sendo detetada a presença humana no conteúdo multimédia, é necessário efetuar uma tarefa de reconhecimento facial, de modo a verificar se a pessoa presente diz respeito àquela que está a ser seguida (definido nas preferências da aplicação).

6 – Gerar alertas mediante resultados obtidos: caso se considere que o conteúdo detetado diz respeito a uma situação de *bullying*, deve ser enviado um alerta com a indicação do problema para todos os agentes envolvidos. Adicionalmente, poderá confirmar-se a eficácia da análise mediante a confirmação por parte dos encarregados assim que consultarem os alertas e com isto o sistema deverá melhorar a sua performance nas execuções seguintes.

Volta-se agora a olhar para os *use cases* descritos no ponto 6.2, que explicam como identificar o papel dos envolvidos na situação, para uma explicação mais detalhada.

6.8 Caso de Conteúdo Textual

O primeiro passo a completar sempre que se deteta uma nova publicação, deve ser identificar o tipo de conteúdo. Caso esse conteúdo seja composto apenas por texto, este deve ser colocado no classificador textual (ponto 6.6) para verificar qual a probabilidade de pertencer à categoria de *bullying*. Se o valor para a classe *bullying* for igual ao superior a 50%, o texto ficará identificado como insultuoso e um alerta terá de ser gerado. Seguidamente verifica-se a existência de comentários textuais no mesmo conteúdo. O processo de análise deverá repetir-se, e encontrada novamente uma situação classificada como *bullying*, devem ser gerados alertas para os envolvidos.

Se apenas o texto da publicação for *bullying* sabe-se que o agressor é a pessoa que publicou, mas não se consegue identificar a vítima que tem como alvo. Nos comentários pode existir interação entre várias pessoas pelo que poderá ser impossível ter certezas sobre a quem é dirigido cada um dos mesmos.

Este ramo pode ser analisado visualmente com recurso à figura 6.8.

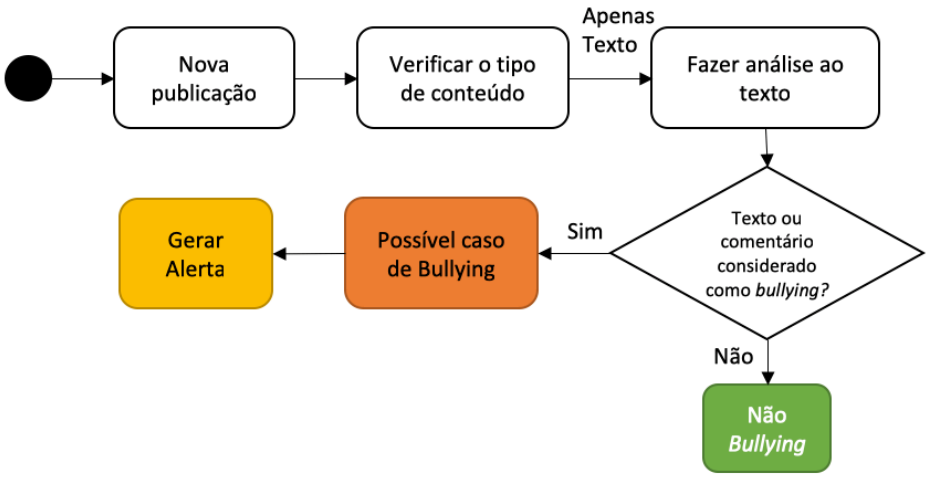

FIGURA 6.8: *Use Case* para conteúdo textual

6.9 Caso de Conteúdo Visual (Imagem ou Vídeo)

Outro caso passa por verificar se o conteúdo publicado diz respeito a uma imagem. Se acompanhada de uma descrição ou existirem comentários pode dizer respeito a uma situação de *bullying*. Se tais não existirem, a publicação não é considerada como preocupante.

Começa-se por fazer a análise ao texto, quer à descrição, quer aos comentários existentes. Se nenhum destes textos forem considerados como *bullying* termina-se a execução. Em cenário contrário avança-se com a deteção de objetos na imagem em busca da presença humana na mesma. Aqui uma pequena referência à diferença que existirá entre uma fotografia

e um vídeo, pois no caso do vídeo será necessário dividi-lo em *frames* para obter as respetivas imagens e as poder utilizar nas APIs. Se nenhuma pessoa for encontrada nas imagens, assumindo que algum do texto analisado foi classificado como *bullying*, nos padrões referidos no ponto anterior, sabe-se que a vítima não estará na imagem, todavia, deve ser gerado um alerta. Se, por ventura, se detetar a presença humana na foto, ou seja, um valor superior a 50%, deve avançar-se com a tarefa de reconhecimento facial. Quando a pessoa detetada não é a mesma que fez a publicação, e a descrição está classificada como *bullying*, considera-se que estamos perante uma possível situação de *bullying* onde a vítima não fez a publicação, mas aparece no conteúdo multimédia apresentado. Assumindo agora o mesmo cenário no que diz respeito à pessoa reconhecida, mas se desta vez não for a descrição a estar classificada como *bullying*, mas sim algum comentário, verifica-se se este pertence à mesma pessoa que fez a publicação. Em caso afirmativo, o resultado da análise será o mesmo do relatado anteriormente, senão considera-se que existem comentários ofensivos por parte de outros e que serão propícios a contemplar uma situação de insulto, pelo que a vítima é quem publicou inicialmente. Quando a pessoa reconhecida na imagem é a mesma que efetuou a publicação, verifica-se se os comentários analisados foram classificados como *bullying*. Estando nessa categoria, procura-se ver se algum comentário desses foi redigido pela mesma pessoa, e se sim, considera-se que a vítima foi quem fez a publicação e deverá estar a procurar atingir outros utilizadores. Esta pessoa será a vítima caso não tenha sido ela a escrever o comentário.

Para cada uma destas possibilidades, sempre que forem detetadas situações consideradas com *bullying,* os alertas devem ser gerados na área da aplicação destinada para o efeito, e se possível, contemplar agressor, vítima e encarregados de educação, se existir ligação.

Estas situações podem ser consultadas visualmente com recurso à figura 6.9.

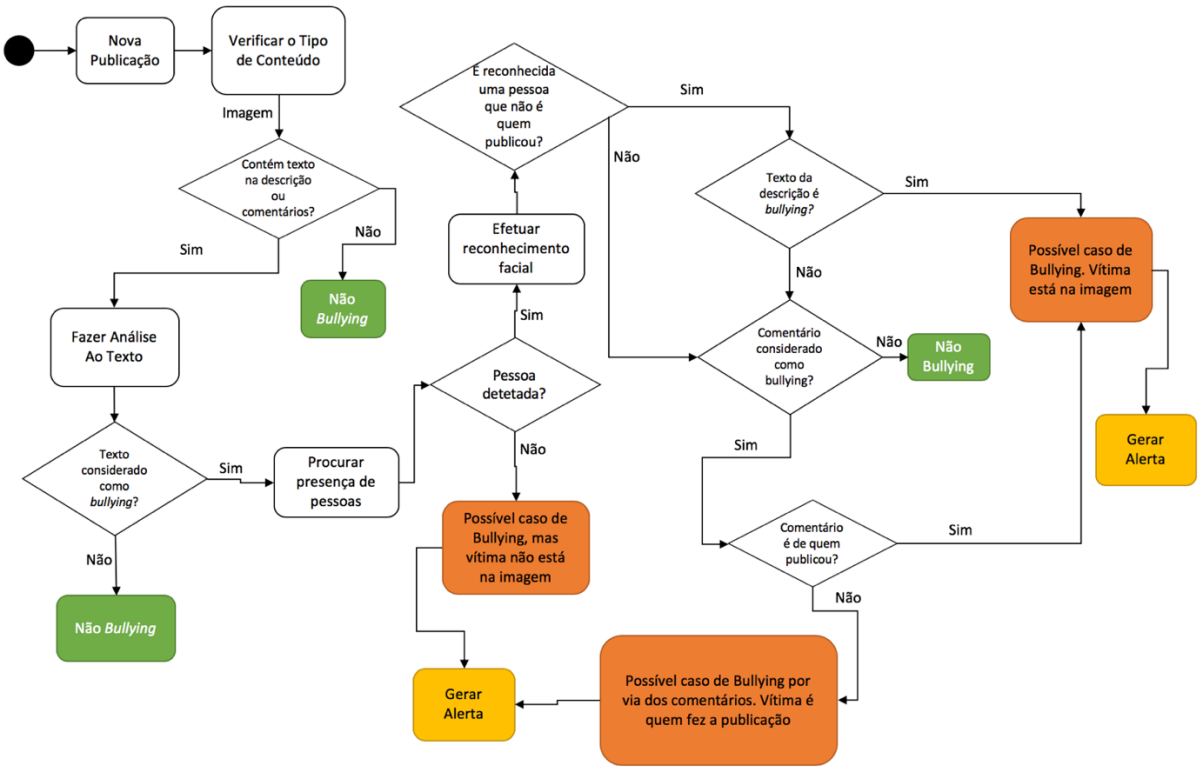

FIGURA 6.9: *Use Case* para conteúdo visual (imagem ou vídeo)

Capítulo 7

Cenários de aplicação prática

Ao longo do presente capítulo serão descritos alguns cenários que pretendem demonstrar o funcionamento da solução no mundo real. Será feita uma cobertura para os diferentes casos de uso apresentados no capítulo anterior e espera-se que assim se consiga entender um pouco melhor a utilidade do modelo.

7.1 Cenário A

Considere-se uma simples aplicação web que é dedicada à partilha de notícias, onde para cada artigo existe a possibilidade de fazer comentários apenas com recurso a texto. Não existem perfis pessoais, de forma a evitar a necessidade de registo na plataforma, e para efetuar um novo comentário apenas é necessário inserir um nome de utilizador para ser apresentado junto do comentário. A figura 7.1 apresenta um cenário onde alguns utilizadores escrevem cometários a um artigo publicado pelo *site*.

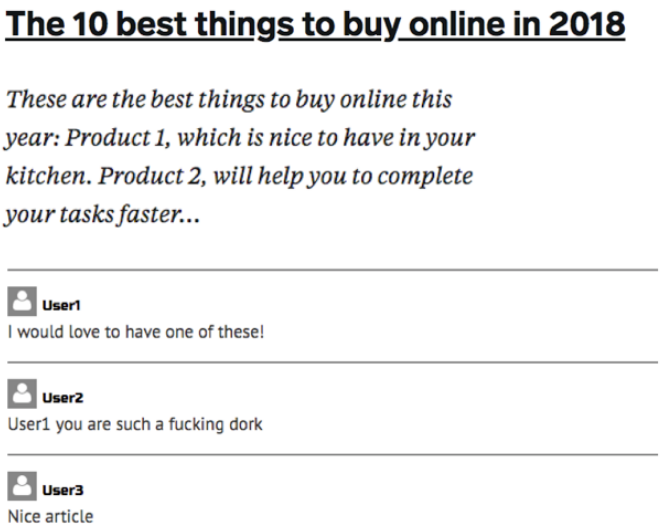

FIGURA 7.1: Cenário A - comentários a um artigo

Como nesta situação se está perante um artigo divulgado pelo próprio *site*, à partida, a descrição do mesmo não terá conteúdo relativo a uma situação de *bullying*, contudo, este deve ser analisado à mesma de modo a poder aumentar a base de conhecimento do sistema. Analisando os comentários, é visível que o primeiro e terceiro, redigidos pelos *User1* e *User3* não apresentam qualquer característica de ameaça, e não contêm qualquer tipo de vocabulário insultuoso. Já o comentário do *User2* tem essas características e até é notório que o seu comentário é direcionado ao *User1* com um insulto. Perante esta situação, o

sistema não irá fazer qualquer tipo análise em imagem, pelo facto de se estar apenas perante conteúdo textual. Analisando o texto, o sistema deverá classificar o comentário do *User2* como *bullying*, e os restantes comentários, juntamente com a descrição, devem ser classificados como *not bullying*. Por via de não existirem perfis pessoais nesta aplicação, o alerta poderá ser gerado apenas para os responsáveis pela aplicação para o poderem ocultar ou até mesmo eliminar de forma manual ou automática. O *output* da classificação deverá ser o seguinte:

"These are the best things to buy online this year: Product 1, which is nice to have in your kitchen. Product 2, will help you to complete your tasks faster..."	**NOT BULLYING**
"I would love to have one of these!"	**NOT BULLYING**
"User1 you are such a fucking dork"	**BULLYING**
"Nice article"	**NOT BULLYING**

Tabela 7.1: *Output* cenário A

7.2 Cenário B

Agora considere-se uma aplicação onde é necessário efetuar um registo e consequentemente iniciar sessão. No processo de registo as pessoas terão de inserir informações pessoais básicas. Esta aplicação será um espelho das redes sociais típicas que conhecemos hoje em dia, onde as pessoas podem partilhar diferentes tipos de conteúdos.

Um jovem, menor de idade, decide partilhar uma fotografia sua nessa aplicação. Não faz qualquer tipo de descrição textual na respetiva publicação, mas surgem alguns comentários por parte de outros utilizadores (figura 7.2.1).

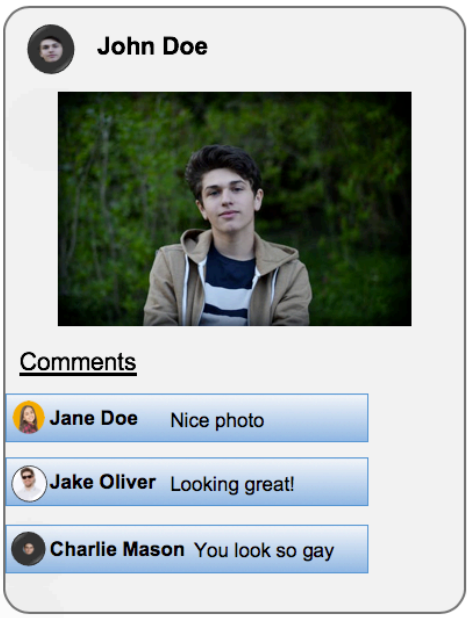

FIGURA 7.2.1: Cenário B – publicação com uma foto

Para início da análise, visto não existir qualquer descrição, o sistema irá primeiro procurar pela presença humana na foto, retornando um *output* como o da figura seguinte.

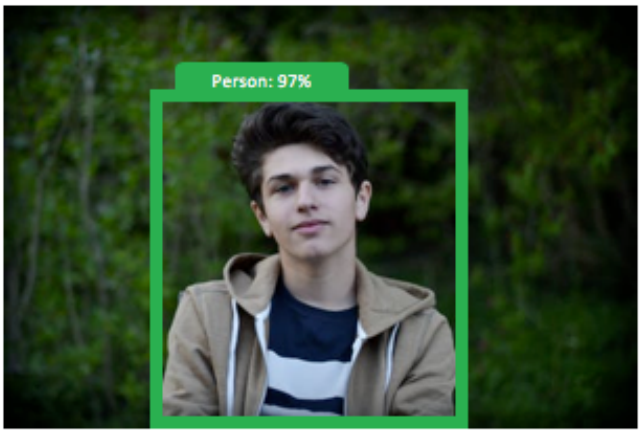

FIGURA 7.2.2: Cenário B – *Output* da deteção de objetos

Como o sistema deteta a presença humana na foto, avança-se para a tarefa de reconhecimento facial. As configurações iniciais para a utilização da ferramenta já deverão ter sido efetuadas neste ponto, tendo adicionado fotos da pessoa para o sistema treinar e conseguir efetuar esse reconhecimento. Adicionalmente poderá recorrer a fotos de perfil para poder treinar e melhorar a sua eficácia. Neste caso, o sistema deverá reconhecer que a pessoa na foto é a mesma que a publicou, ou seja, o utilizador *John Doe*. Assim sendo, parte-se para a análise dos comentários, visto que não existe nenhuma descrição. O *output* da análise textual para os comentários existentes deve ser de que apenas o último comentário, do utilizador *Charlie Mason*, está classificado como *bullying*. Ora, nesta situação deverá ser lançado um alerta para os encarregados de educação da vítima, que aqui está presente na imagem, alertando para a publicação. Ao mesmo tempo a vítima deve ser aconselhada a bloquear o utilizador que fez o comentário classificado como *bullying* e a falar com os seus tutores sobre a situação. O agressor também deverá ser notificado, para o sensibilizar a parar com estas situações e a eliminar o conteúdo que disponibilizou, podendo até ser bloqueado pela aplicação mediante as políticas que estiverem definidas. No caso de este utilizador também ser menor, os seus encarregados devem ser igualmente notificados da situação.

7.3 Cenário C

Assumindo o mesmo contexto aplicativo do cenário anterior, considere-se a seguinte publicação (figura 7.3), onde é publicada uma fotografia acompanhada por uma descrição. Neste caso não foi ainda colocado qualquer tipo de comentário à mesma publicação pelo que será analisada a fotografia e o texto da descrição.

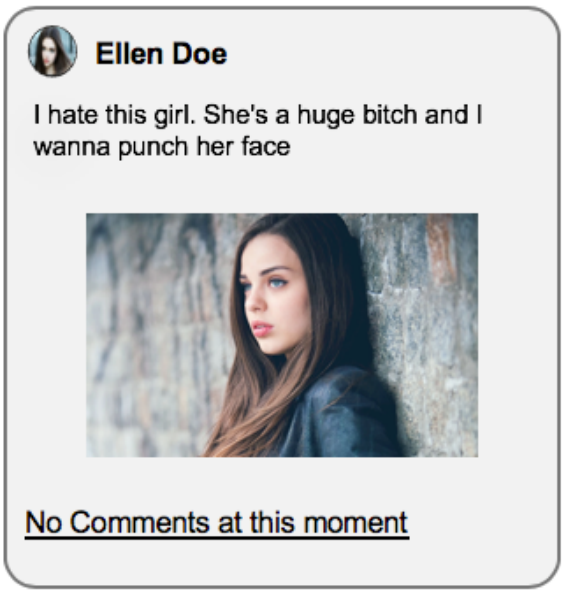

FIGURA 7.3: Cenário C – publicação com foto e descrição

Nesta situação o processo do cenário B deverá repetir-se. Como se pode constatar, a presença humana na foto existe, pelo que se avança novamente para uma tarefa de reconhecimento facial. No atual cenário, o *output* indica que a pessoa que fez a publicação não é aquela que aparece na fotografia. Posto isto, passa-se para a análise ao texto da descrição. O texto é classificado como *bullying*, e está-se perante uma situação onde a vítima estará alegadamente presente na foto, e o agressor é naturalmente quem fez a sua divulgação. O alerta deverá ser gerado igualmente para o agressor para que este seja sensibilizado a apagar a imagem e o texto que a acompanha. Se a vítima for reconhecida como utilizadora da aplicação também poderá receber a notificação, assim como os encarregados de educação de todos os envolvidos.

Capítulo 8

Conclusão

8.1 Síntese

Este trabalho foi desenvolvido no âmbito do Mestrado em Engenharia Informática, da ESTG|PPorto, e tinha como principal objetivo apresentar um modelo que fosse capaz de combater um dos principais problemas presentes na internet, o *cyberbullying*. Sempre existiu a intenção de investigar e desenvolver o trabalho com recurso a mecanismos de *machine learning* por via da sua forte expansão e pela forma como se vai impondo no mundo da informática atualmente. Adicionalmente, o objetivo de propor uma solução que fosse autónoma por via de um sistema capaz de aprender sozinho foi outro dos pontos chave para se seguir esta abordagem.

Identificado o problema a resolver e sabendo da sua forte presença nas redes sociais, o foco inicial passou por fazer uma pesquisa sobre trabalhos relacionados, de forma a aumentar o conhecimento sobre a tecnologia disponível e encontrar pontos a melhorar nessas mesmas soluções. Além disso, foi feito um trabalho de investigação sobre os diversos algoritmos existentes em *machine learning* de forma a conhecer um maior leque de opções pelas quais se podiam optar para a construção do modelo.

A escolha para a proposta de implementação assentou no *tensorflow* visto ser uma biblioteca com provas dadas e pela capacidade que dá para ser executada nas mais distintas plataformas. Além disso, as inúmeras APIs disponíveis e o facto de ser *open source* permitem que se encontrem diversos módulos que permitam melhorar constantemente a solução e ter uma comunidade que auxilie nos possíveis problemas que surjam durante o seu desenvolvimento. De momento, o modelo contempla análise textual, que é fulcral para a decisão final, e análise em imagem, para deteção de objetos e posteriormente reconhecimento facial. A particularidade de recorrer a este tipo de bibliotecas, é que facilmente se poderá adaptar o modelo para situações idênticas, nomeadamente, questão de deteção de *clickbait* e *fake news*, algo que está muito presente na internet à data e que deve ser reduzido.

Com esta proposta de solução espera-se que o combate ao *cyberbullying* seja um pouco mais intensificado assim que esta seja implementada. Com um sistema desenvolvido com as características apresentadas pretende-se que os números da prática de *bullying online* diminuam substancialmente, tendo em conta os alertas gerados para sensibilizar os agressores ou por via de outras medidas que se decidam implementar mediante o *output* final.

Como trabalho futuro, tem-se como objetivo melhorar o sistema adicionando a possibilidade de analisar o conteúdo sonoro de um vídeo ou de *clips* de som por si só. Para tal será sempre necessário converter o som para texto e a partir daí já se poderá utilizar o classificador já existente. O que será também muito interessante de se adicionar é a capacidade de reconhecimento de texto dentro de uma imagem, pois esta pode ser postada sem qualquer descrição e conter ela própria um texto agressivo dentro da mesma. Adicionalmente, ter uma componente de análise cognitiva poderia aumentar a eficácia do sistema e prever quais os utilizadores que mais provavelmente podem partir para a agressão ou ser vítimas destas situações, com base numa análise do seu histórico de utilização da aplicação e das suas preferências.

8.2 Contribuições Científicas

Durante a elaboração e desenvolvimento deste trabalho, foi possível apresentar algumas contribuições científicas, nomeadamente:

- "Detection and Prevention of Bullying on Online Social Networks: The Combination of Textual, Visual, and Cognitive" (Carlos Silva, Ricardo Santos e Ricardo Barbosa - 2018), apresentado na conferência INTETAIN 2018.
- "Usage of Textual and Visual Analysis to Automatically Detect Cyberbullying in Online Social Network" (Carlos Silva, Ricardo Santos e Ricardo Barbosa - 2018), apresentado na conferência HIS 2018.

Referências

[1] C. B. R. Evans and P. R. Smokowski, “Theoretical Explanations for Bullying in School: How Ecological Processes Propagate Perpetration and Victimization,” *Child Adolesc. Soc. Work J.*, vol. 33, no. 4, pp. 365–375, 2016.

[2] K. Dinakar, R. Reichart, and H. Lieberman, “Modeling the detection of Textual Cyberbullying.,” *Soc. Mob. Web*, pp. 11–17, 2011.

[3] G. Lightbody, R. Bond, M. Mulvenna, and Y. Bi, “Investigation into the Automated Detection of Image based Cyber bullying on Social Media Platforms,” *Irish Mach. Vis. Image Process. Conf.*, p. 2 pp., 2014.

[4] R. A. Hardy and J. R. Norgaard, “Reputation in the Internet black market: An empirical and theoretical analysis of the Deep Web,” *J. Institutional Econ.*, vol. 12, no. 3, pp. 515–529, 2016.

[5] F. Rebecca, “Celebgate: Two Methodological Approaches to the 2014 Celebrity Photo Hacks,” Oxford, United Kingdom, 2014.

[6] E. Villar-Rodriguez, J. Del Ser, A. Torre-Bastida, M. Bilbao, and S. Salcedo-Sanz, “A novel machine learning approach to the detection of identity theft in social networks based on emulated attack instances and support vector machines,” *Concurr. Comput. Pract. Exp.*, vol. 28, pp. 1385–1395, 2015.

[7] C. Wang, B. Yang, and J. Luo, “Identity Theft Detection in Mobile Social Networks Using Behavioral Semantics,” *2017 IEEE Int. Conf. Smart Comput. SMARTCOMP 2017*, 2017.

[8] J. Verble, “The NSA and Edward Snowden: surveillance in the 21st century,” *ACM SIGCAS Comput. Soc. - Spec. Issue Whistleblowing*, vol. 44, no. 3, pp. 14–20, 2014.

[9] G. Mascheroni and A. Cuman, “Net Children Go Mobile: Final report,” Milano, Italy, 2014.

[10] “Facebook minimum age requirement.” [Online]. Available: www.facebook.com/help/157793540954833. [Accessed: 06-Nov-2017].

[11] J. Simões, C. Ponte, E. Ferreira, J. Doretto, and C. Azevedo, “Net Children Go Mobile: Crianças e Meios Digitais Móveis em Portugal: Resultados Nacionais do Projeto Net Children Go Mobile,” Lisboa, Portugal, 2014.

[12] T. Guardian, “Google and Facebook to be asked to pay to help UK tackle cyberbullying.” [Online]. Available: www.theguardian.com/technology/2017/oct/11/google-and-facebook-to-be-asked-to-pay-to-help-tackle-cyberbullying. [Accessed: 02-Nov-2017].

[13] R. J. McClowry, M. N. Miller, and G. D. Mills, “What family physicians can do to combat bullying,” *J. Fam. Pract.*, vol. 66, no. 2, pp. 82–89, 2017.

[14] BBC, “Why do people bully others?” [Online]. Available: http://www.bbc.co.uk/newsround/36074395. [Accessed: 02-Nov-2017].

[15] J. W. Patchin and S. Hinduja, “Digital Self-Harm Among Adolescents,” *J. Adolesc. Heal.*, vol. 61, no. 6, pp. 761–766, 2017.

[16] United Nations Children’s Fund, “A Familiar Face: Violence in the lives of children and adolescents.,” New York, USA, 2017.

[17] T. N. Web, “How Dangerous is Cyberbullying?” [Online]. Available: https://thenextweb.com/contributors/2017/10/04/how-dangerous-is-cyberbullying. [Accessed: 31-Oct-2017].

[18] Q. Huang, V. K. Singh, and P. K. Atrey, “Cyber Bullying Detection Using Social and

Textual Analysis," *Proc. 3rd Int. Work. Soc. Multimed. - SAM '14*, pp. 3–6, 2014.

[19] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," *Proc. - 2012 ASE/IEEE Int. Conf. Privacy, Secur. Risk Trust 2012 ASE/IEEE Int. Conf. Soc. Comput. Soc. 2012*, pp. 71–80, 2012.

[20] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," *Proc. - 10th Int. Conf. Mach. Learn. Appl. ICMLA 2011*, vol. 2, pp. 241–244, 2011.

[21] M. Dadvar, F. M. G. de Jong, R. J. F. Ordelman, and R. B. Trieschnigg, "Improved cyberbullying detection using gender information," *12th Dutch-Belgian Inf. Retr. Work. (DIR 2012)*, no. April 2017, pp. 23–25, 2012.

[22] R. Zhao, A. Zhou, and K. Mao, "Automatic detection of cyberbullying on social networks based on bullying features," *Proc. 17th Int. Conf. Distrib. Comput. Netw. - ICDCN '16*, pp. 1–6, 2016.

[23] Y. Bengio, "Deep Learning," Montréal, Canada, 2015.

[24] S. Michalski and Y. Kodratoff, *Machine Learning: An Artificial Intelligence Approach, Volume 3*. 1990.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Montréal, Canada, 2015.

[26] A. Garnham, *Artificial Intelligence: An Introduction*. 1988.

[27] C. Brown, "The Advantages and Disadvantages of Artificial Intelligence," 2017. [Online]. Available: http://findnerd.com/list/view/The-Advantages-and-Disadvantages-of-Artificial-Intelligence/34917/. [Accessed: 17-Jan-2018].

[28] M. Tegmark, "Benefits and Risks of Artificial Intelligence. Future of Life Institute," 2016. [Online]. Available: https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/?cn-reloaded=1. [Accessed: 17-Jan-2018].

[29] N. Bostrom and M. Cirkovic, *Global Catastrophic Risks*. 2011.

[30] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. 2016.

[31] M. Bernard, "What is the difference between artificial intelligence and machine learning?," 2016. [Online]. Available: www.forbes.com/sites/bernardmarr/2016/12/06/what-is-the-difference-between-artificial-intelligence-and-machine-learning/#64dd419d2742. [Accessed: 12-Jan-2018].

[32] P. Lison, "An introduction to machine learning," Oslo, Norway, 2012.

[33] A. Lorena and A. Carvalho, "Uma introdução às Support Vector Machines," São Paulo, Brasil, 2007.

[34] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, Second Edi. 2009.

[35] A. Fernández and A. Salmerón, "BayesChess: A computer chess program based on Bayesian networks," *Pattern Recognit. Lett.*, vol. 29, no. 8, pp. 1154–1159, 2008.

[36] Tesla Motors, "Tesla Motors Autopilot." [Online]. Available: www.tesla.com/autopilot. [Accessed: 23-Jan-2018].

[37] Waymo, "Waymo Technology." [Online]. Available: www.waymo.com/tech. [Accessed: 23-Jan-2018].

[38] NVidia, "NVidia Self-Driving Cars Technology & Solutions." [Online]. Available: www.nvidia.com/en-us/self-driving-cars/. [Accessed: 23-Jan-2018].

[39] Apple, "Apple Siri." [Online]. Available: www.apple.com/ios/siri. [Accessed: 19-Jan-2018].

[40] E. Elwany, “Enhancing Cortana User Experience Using Machine Learning,” pp. 1–5, 2014.

[41] Google, “Google Now.” [Online]. Available: www.google.com/intl/pt-PT/landing/now/. [Accessed: 19-Jan-2018].

[42] Microsoft, “Microsoft Cortana.” [Online]. Available: https://support.microsoft.com/pt-pt/help/17214/windows-10-what-is. [Accessed: 19-Jan-2018].

[43] N. Goksel Canbek and M. E. Mutlu, “On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants,” *Int. J. Hum. Sci.*, vol. 13, no. 1, p. 592, 2016.

[44] Hanson Robotics, “Hanson Robotics. Sophia.” [Online]. Available: http://www.hansonrobotics.com/robot/sophia/. [Accessed: 25-Jan-2018].

[45] J. Retto, “Sophia , First Citizen Robot of the World Sophia , First Citizen Robot of the World,” Lima, Peru, 2017.

[46] A. A. Bahishti, “Humanoid Robots and Human Society,” vol. 1, no. 1, pp. 60–63, 2017.

[47] N. Castle, “Supervised vs Unsupervised Machine Learning,” 2017. [Online]. Available: https://www.datascience.com/blog/supervised-and-unsupervised-machine-learning-algorithms. [Accessed: 13-Mar-2018].

[48] C. Pires, “Correlação e Regressão linear simples.” [Online]. Available: http://home.uevora.pt/~cpires/diagnost/regressao2.pdf. [Accessed: 08-Mar-2018].

[49] V. Maini and S. Sabri, “Machine Learning for Humans.” [Online]. Available: https://medium.com/machine-learning-for-humans/why-machine-learning-matters-6164faf1df12. [Accessed: 11-Mar-2018].

[50] J. Brownlee, “Overfitting and Underfitting With Machine Learning Algorithms.” [Online]. Available: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/. [Accessed: 09-Mar-2018].

[51] A. machine Learning, “Cross-validation.” [Online]. Available: https://docs.aws.amazon.com/machine-learning/latest/dg/cross-validation.html. [Accessed: 14-Mar-2018].

[52] D. Carneiro, “Extração de Conhecimento - Segmentação e Classificação,” Felgueiras, Portugal, 2015.

[53] I. Rish, “An empirical study of the naive Bayes classifier,” pp. 41–46, 1999.

[54] N. Englesson, “Logistic Regression for Spam Filtering,” Stockholm, Sweden, 2016.

[55] J. Béjar, “K-nearest neighbours.” [Online]. Available: http://www.cs.upc.edu/~bejar/apren/docum/trans/03d-algind-knn-eng.pdf. [Accessed: 08-Mar-2018].

[56] T. Mitchell, “Decision Tree Learning,” *Mach. Learn.*, vol. 1, pp. 52–80, 1997.

[57] N. Donges, “The Random Forest Algorithm.” [Online]. Available: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd. [Accessed: 10-Mar-2018].

[58] P. Dayan, “Unsupervised Learning,” *MIT Encycl. Cogn. Sci.*, pp. 1–7, 2004.

[59] K. Alsabti, S. Ranka, and V. Singh, “An Efficient K-Means Clustering Algorithm,” 1997.

[60] A. Roy, “How to calculate cluster similarities between the result of k - means and hierarchical clustering?” [Online]. Available: https://www.researchgate.net/post/How_to_calculate_cluster_similarities_between_the_result_of_k-means_and_hierarchical_clustering. [Accessed: 13-Mar-2018].

[61] E. L. Nylen and P. Wallisch, “Dimensionality Reduction,” *Neural Data Sci.*, pp. 223–248, 2017.

[62] NVidia, “Deep Learning.” [Online]. Available: https://developer.nvidia.com/deep-learning. [Accessed: 17-Mar-2018].

[63] M. Nielson, “Neural Networks and Deep Learning,” 2015. [Online]. Available: http://neuralnetworksanddeeplearning.com. [Accessed: 18-Mar-2018].

[64] A. Karpathy, “Deep Reinforcement Learning: Pong from Pixels,” 2016. [Online]. Available: http://karpathy.github.io/2016/05/31/rl/. [Accessed: 19-Mar-2018].

[65] C. Gershenson, “Artificial Neural Networks for Beginners,” *arXiv*, p. 8, 2003.

[66] DL4j, “What is a convolutional neural network?” [Online]. Available: https://deeplearning4j.org/convolutionalnetwork. [Accessed: 20-Mar-2018].

[67] C. Olah, “Understanding LSTM Networks.” [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 20-Mar-2018].

[68] M. Veloso, “Markov Decision Processes,” Porto, Portugal, 2013.

[69] C. X. Shen, R. De Liu, and D. Wang, “Why are children attracted to the Internet? the role of need satisfaction perceived online and perceived in daily real life,” *Comput. Human Behav.*, vol. 29, no. 1, pp. 185–192, 2013.

[70] P. Eleni, “School Bullying: The Phenomenon, the Prevention and the Intervention,” *Procedia - Soc. Behav. Sci.*, vol. 152, pp. 268–271, 2014.

[71] “Tensorflow.” [Online]. Available: https://www.tensorflow.org. [Accessed: 03-Apr-2018].

[72] Tensorflow, “First Steps with Tensorflow.” [Online]. Available: https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit. [Accessed: 03-Apr-2018].

[73] A. Unruh, “Using TensorFlow on mobile devices,” 2017. [Online]. Available: https://opensource.com/article/17/11/intro-tensorflow. [Accessed: 06-Apr-2018].

[74] Tensorflow, “Tensors.” [Online]. Available: https://www.tensorflow.org/programmers_guide/tensors. [Accessed: 06-Apr-2018].

[75] P. Sarang, “Tensorflow: Understanding Tensors and Graphs to get you started in Deep Learning,” 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/03/tensorflow-understanding-tensors-and-graphs/. [Accessed: 07-Apr-2018].

[76] Tensorflow, “Graphs and Sessions.” [Online]. Available: https://www.tensorflow.org/programmers_guide/graphs. [Accessed: 07-Apr-2018].

[77] “Top five use cases of tensorflow,” 2017. [Online]. Available: https://www.exastax.com/deep-learning/top-five-use-cases-of-tensorflow/. [Accessed: 10-Apr-2018].

[78] Google, “Machine Learning Guides: Text Classification.” [Online]. Available: https://developers.google.com/machine-learning/guides/text-classification/. [Accessed: 17-Jul-2018].

[79] O’reilly, “Perform sentiment analysis with LSTMs , using TensorFlow,” 2018. [Online]. Available: https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow. [Accessed: 16-Jul-2018].

[80] Tensorflow, “Vector representation of words.” [Online]. Available: https://www.tensorflow.org/tutorials/representation/word2vec. [Accessed: 15-Apr-2018].

[81] “Text classification with movie reviews.” [Online]. Available: https://www.tensorflow.org/tutorials/keras/basic_text_classification. [Accessed: 27-Jul-

2018].

[82] J. Huang *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017–Janua, pp. 3296–3305, 2017.

[83] J. Huang, “Supercharge your Computer Vision models with the TensorFlow Object Detection API,” 2017. [Online]. Available: https://ai.googleblog.com/2017/06/supercharge-your-computer-vision-models.html. [Accessed: 12-Apr-2018].

[84] L. Hulstaert, “A Beginner’s Guide to Object Detection,” 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/object-detection-guide. [Accessed: 11-May-2018].

[85] “COCO Dataset.” [Online]. Available: http://cocodataset.org. [Accessed: 23-Apr-2018].

[86] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017.

[87] “Tensorflow Object Detection API.” [Online]. Available: https://github.com/tensorflow/models/tree/master/research/object_detection. [Accessed: 19-Apr-2018].

[88] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” 2015.

[89] Tensorflow, “Image Recognition.” [Online]. Available: https://www.tensorflow.org/tutorials/images/image_recognition. [Accessed: 27-Apr-2018].

[90] A. Tiwari, A. K. Goswami, and M. Saraswat, “Feature Extraction for Object Recognition and Image Classification,” *Int. J. Eng. Res. Technol.*, vol. 2, no. 10, pp. 1238–1246, 2013.

[91] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” 2016.

[92] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, “YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017–Janua, pp. 7464–7473, 2017.

[93] “Show and Tell: A Neural Image Caption Generator.” [Online]. Available: https://github.com/tensorflow/models/tree/master/research/im2txt. [Accessed: 28-May-2018].

[94] J. M. Pandya, D. Rathod, and J. J. Jadav, “A Survey of Face Recognition approach,” *Int. J. Eng. Res. Appl.*, vol. 3, no. 1, pp. 632–635, 2013.

[95] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1701–1708, 2014.

[96] I. Marqués, “Face Recognition Algorithms,” Bilbao, Spain.

[97] S. Kühn, T. R. Brick, B. C. N. Müller, and J. Gallinat, “Is this car looking at you? How anthropomorphism predicts fusiform face area activation when seeing cars,” *PLoS One*, vol. 9, no. 12, pp. 1–14, 2014.

[98] O. Déniz, G. Bueno, J. Salido, and F. De Torre, “Face recognition using Histograms of Oriented Gradients,” vol. 32, pp. 1598–1603, 2011.

[99] X. Zhu and D. Ramanan, “Face Detection, Pose Estimation, and Landmark Localization in the Wild,” 2012.