UNIVERSIDADE AUTÓNOMA DE LISBOA
LUÍS DE CAMÕES

DEPARTAMENTO DE ENGENHARIAS E CIÊNCIAS DA COMPUTAÇÃO
LICENCIATURA EM ENGENHARIA INFORMÁTICA

Sistema de Autenticação (FaceShield)

Autores/as: André Gonçalves, Gonçalo Lemos, Tiago Gonçalves e Ricardo Belo
Orientador/a: Professor Héctor Dave Orrillo Ascama
Número dos/as candidatos/as: 30007039, 30007523, 30007075 e 30007761

Junho de 2023
Lisboa

Agradecimentos

Ao nosso orientador professor Héctor Dave Orrillo Ascama pelos seus concelhos e críticas durante as várias fases do trabalho, que sem elas, as nossas lacunas e dificuldades teriam sido ainda mais óbvias, bem como a sua competência, disponibilidade e determinação para nos ajudar na realização deste projeto.

Ao professor Láercio Cruvinel, pelo tempo despendido e por todo as informações que nos disponibilizou.

Aos colegas do Projeto, pela partilha e apoio que contribuíram e permitiram a realização deste trabalho.

A todos muito obrigado.

Resumo

A ciência de reconhecimento facial é uma área que vem sendo cada vez mais presente em diversas áreas. Isto deve-se essencialmente ao facto de reproduzir uma capacidade natural do ser humano. Nós desenvolvemos uma aplicação de autenticação fácil que utiliza Flutter no front-end, Google ML Kit, TFLite e FaceNet para a deteção e reconhecimento facial. A base de dados da aplicação é gerida pela Firebase. O Flutter é uma framework de desenvolvimento de aplicações móveis que permite criar interfaces para aplicações atraentes e responsivas. O Google ML Kit é um conjunto de ferramentas de machine learning fornecido pela Google, que é essencialmente usado para tarefas como reconhecimento de imagem e deteção de rosto. TFLite é uma biblioteca de machine learning para dispositivos móveis que permite executar modelos de machine learning treinados nos mesmos. O FaceNet é um modelo de machine learning que pode ser usado para reconhecimento facial. Ao combinarmos estas ferramentas, foi possível criar uma aplicação que permitirá aos utilizadores se autenticarem facilmente através de reconhecimento facial. O Firebase é utilizado como uma gestão de base de dados para armazenar informações dos usuários e outras informações que foram consideradas relevantes. Em resumo, o projeto de desenvolvimento de uma aplicação de reconhecimento facial utiliza várias tecnologias que quando usadas em harmonia permite aos utilizadores terem uma experiência simples, com a segurança e a facilidade de uso oferecidas pelo uso de ferramentas como Flutter, Google ML Kit, TFLite e FaceNet, além de ter o suporte e escalabilidade oferecidos pelo Firebase como base de dados.

Palavras-chave: Flutter, Reconhecimento Facial, FaceNet, Machine Learning.

Abstract

Facial recognition science is one that has been increasingly present in many areas. This is essential since it reproduces the natural capacity of a human being. We developed an application of easy authentication that uses Flutter as front-end.
Google ML Kit, TFLite and FaceNet for detection and facial recognition. All the database of the application is stored and managed by Firebase.

Flutter is a framework of development of multiplatform mobile applications that allows you to create attractive and responsive user interfaces. Google ML Kit is a set of machine learning tools provided by Google, that can be used for tasks like image recognition and face detection. TFLite is a machine learning library for mobile devices that allows you to execute machine learning models trained on mobile devices. FaceNet is a machine learning model that can be used for facial recognition.

When you combine all these tools, it is possible to create an application that allows users to authenticate themselves easily using facial recognition. Firebase is used as a database to store information about the users and other relevant information.

In summary, the project of the development of an application with easy authentication uses a variety of cutting-edge technologies to create an innovative solution that allows the users to authenticate themselves using facial, with safety and the easy use of the tools provided by Flutter, Google ML Kit, TFLite e FaceNet, in addition of having the support and scalability provided by firebase as a database.

Keywords: Flutter, Facial Recognition, FaceNet, Machine Learning.

Resumo

La ciencia del reconocimiento facial es una ciencia que ha estado cada vez más presente en una variedad de áreas. Esto es fundamental ya que reproduce la capacidad natural de un ser humano. Desarrollamos una aplicación de fácil autenticación que usa Flutter como front-end.

Google ML Kit, TFLite y FaceNet para detección y reconocimiento facial. Toda la base de datos de la aplicación es almacenada y administrada por Firebase.

Flutter es un marco de desarrollo de aplicaciones móviles multiplataforma que le permite crear interfaces de usuario atractivas y receptivas. Google ML Kit es un conjunto de herramientas de aprendizaje automático proporcionado por Google, que se puede utilizar para tareas como el reconocimiento de imágenes y la detección de rostros. TFLite es una biblioteca de aprendizaje automático para dispositivos móviles que le permite ejecutar modelos de aprendizaje automático entrenados en dispositivos móviles. FaceNet es un modelo de aprendizaje automático que se puede utilizar para el reconocimiento facial.

Cuando combina todas estas herramientas, es posible crear una aplicación que permita a los usuarios autenticarse fácilmente mediante el reconocimiento facial. Firebase se utiliza como base de datos para almacenar información sobre los usuarios y otra información relevante.

En resumen, el proyecto de desarrollo de una aplicación con autenticación fácil utiliza una variedad de tecnologías de punta para crear una solución innovadora que permite a los usuarios autenticarse mediante el reconocimiento facial, con seguridad y el uso fácil de las herramientas proporcionadas por Flutter, Google ML Kit, TFLite y FaceNet, además de contar con el soporte y la escalabilidad que brinda firebase como base de datos.

Palabras clave: Flutter, reconocimiento facial, FaceNet, aprendizaje automático

Índice

1 Introdução.

Neste relatório está elaborado a documentação e informação completa sobre um software para deteção e reconhecimento facial. A aplicação desenvolvida chama-se “Face Shield”.

O tema central do nosso trabalho refere-se à autenticação de pessoas usando Machine Learning.

As tecnologias de informação (TI) têm tido uma enorme evolução ao longo destes últimos anos. Porém com o aumento destas tecnologias, tem havido o aumento de vulnerabilidades que depois podem ser abusadas, para criar problemas a níveis de segurança.

Uma grande preocupação nos dias de hoje é garantir a autenticidade. Atualmente os sistemas mais comuns que garantem a autenticidade, são os sistemas baseados em palavras-passe. O problema é que estes sistemas têm vários problemas, pois ao exigir que o utilizador se lembre de uma password específica, pode correr o risco de a esquecer, dificultando assim o acesso às suas informações pessoais e críticas. Para além disso, as passwords podem ser facilmente capturadas através de ataques de phishing, que consistem em enganar as pessoas, levando-as a partilhar informações confidenciais como passwords e números de cartões de débito/crédito. Existem diversas táticas de phishing, porém a mais comum é quando a vítima recebe um email ou mensagem de texto que imita uma pessoa ou organização fidedigna. Ao abrir o email ou ao ler a mensagem, o atacante vai tentar assustar a pessoa com ameaças de que caso não pague vai sofrer consequências. Depois a mensagem exige que a vítima consulte um dado website e coloque as suas credenciais, que depois são “pescadas” pelo agressor, deixando assim a vítima vulnerável a ataques de autenticidade.

Nós temos como objetivos, desenvolver uma aplicação móvel que funcione tanto em Android como iOS, que permita combater estes problemas através da autenticação a partir do reconhecimento facial.

O nosso trabalho tem a seguinte estrutura: um relatório a explicar todos os pontos importantes sobre a nossa aplicação móvel desde as ferramentas utilizadas até à forma como foi implementada. Depois temos, um ficheiro APK e respetiva instrução de instalação que serve para qualquer pessoa conseguir facilmente testar a nossa aplicação.

1.1 Descrição do problema

A deteção e reconhecimento facial são parte de um campo na área de processo de imagem.

As aplicações utilizadas para estes processos são associadas a uma grande necessidade de poder de processamento, visto que uma imagem contém imensas informações.

O telemóvel tem um poder de processamento muito menor do que computadores convencionais onde a deteção e reconhecimento facial têm sido implementadas com sucesso.

O ponto referido em cima juntamente com o facto da qualidade de uma câmara de telemóvel ter a sua qualidade limitada devido a estar inserida no mesmo, cria um problema na utilização de técnicas como “template-match”(Técnica de localização de imagem),”gabor wavelets” (extração de características) ou ”feature invariante” (Localizar os olhos, nariz, etc).

O método tem então de ser adaptado para poder correr num dispositivo móvel e não depender totalmente da velocidade de processamento de um computador.

1.2 Objetivos

1.2.1 Objetivo geral

O principal objetivo deste projeto passa pela criação e desenvolvimento de uma aplicação de reconhecimento facial, recorrendo a uma câmara de computador ou/e de um dispositivo móvel. Na qual através do uso de inteligência artificial, permita analisar a semelhança entre dois rostos, um num frame de um stream de imagens e no outro informação guardada previamente sobre o rosto de um utilizador.

1.2.2 Objetivos específicos

Para informarmos os objetivos específicos da nossa aplicação “FaceShield”, precisamos aprofundar o conhecimento em diversas áreas fundamentais para o desenvolvimento da mesma.
Posto isto, para além dos mencionados, também temos como objetivos o seguinte:

- Conhecimento aprofundado da área de Processamento de Imagem;
- Conhecimento aprofundado da área de Biometria;
- Conhecimento aprofundado da área de Reconhecimento Facial;
- Conhecimento aprofundado dos Algoritmos existentes na Reconhecimento Facial;

Obviamente, como quase transversalmente a todos os projetos relacionados com a universidade, uma maior e melhor melhoria nas nossas competências como programadores e como colegas de trabalho.
Finalmente após termos um conhecimento aprofundado dos objetivos mais gerais, podemos proceder à menção dos objetivos específicos da aplicação “Face Shield”.

1) **Eficiência de recursos:** Queremos desenvolver uma aplicação que seja leve, para minimizar o seu impacto na memória e no desempenho dos dispositivos móveis em que vai ficar disponível (Android e iOS).

2) **Usabilidade fácil:** Pretende-se que a aplicação seja de fácil compreensão e de utilização.

3) **Estabilidade do sistema:** É de extrema importância, que a aplicação não tenha erros ou bugs que possam comprometer o seu normal funcionamento, e por consequência a experiência/utilização geral do utilizador.

4) **Adaptabilidade às condições de luminosidade:** O nosso sistema de reconhecimento facial deve ser capaz de se adaptar a diferentes condições de luminosidade, para providenciar um desempenho consistente e eficiente.

5) **Proof of Life (Prova de Vida):** Esta aplicação tem de ser capaz de distinguir entre uma pessoa real e uma fotografia, para evitar tentativas de falsificação (Ataque à Autenticação).

6) **Escalabilidade:** A aplicação deve ser desenhada para garantir escalabilidade, tendo como objetivo, permitir a sua adaptação e evolução aos novos desafios e avanços tecnológicos atuais.

1.3 Justificativa

Existem imensas vantagens no desenvolvimento de uma aplicação de reconhecimento facial no campo de autenticação. Reconhecimento facial é uma tarefa fácil para humanos isto porque, a primeira coisa que identificamos quando nos deparamos com uma pessoa é a sua face, e não as suas impressões digitais ou forma das mãos. Este tipo de reconhecimento facial pode ser aplicado nas mais diversas áreas, podendo ir da mais simples autenticação numa aplicação pessoal, a uma utilização mais complexa como um reconhecimento numa análise de um vídeo de vigilância. O desenvolvimento de uma aplicação nesta área é benéfica em muitos sentidos tanto para o melhor estudo em diversas áreas como processamento de imagem, biometria, reconhecimento facial como para a melhoria de competências em programação (backend, frontend) e na utilização de uma base de dados.

1.4 Estrutura do trabalho

O trabalho está organizado em duas partes principais:

1) Um relatório detalhado.
2) A elaboração da aplicação móvel.

O relatório está estruturado da seguinte forma:

1. **Introdução:** A introdução inclui a descrição do problema, os objetivos gerais e específicos do projeto, assim como a justificação para o seu desenvolvimento e a estrutura do próprio trabalho.

2. **Fundamentação teórica:** Na fundamentação teórica, apresentamos uma visão teórica abrangente sobre diversos conceitos, começando com a Inteligência Artificial e Machine Learning, passando pelos tipos de aprendizagem em Inteligência Artificial e Redes Neurais (Neural Networks), focando-nos principalmente em neurónios, perceptrões e Aprendizagem profunda (Deep Learning). De seguida efetuamos uma análise sobre Redes Neurais Convolucionais (CNN), em que destacamos a importância dos Embeddings. E para além disso, também oferecemos uma explicação detalhada sobre Algoritmos de Prova de Vida (Liveness Detection), onde incluímos exemplos e a metodologia específica usada no nosso projeto. Por último, terminamos com uma explicação sobre a Biometria Facial.

3. **Ferramentas utilizadas:** Aqui, detalhamos todas as ferramentas que foram usadas no desenvolvimento da nossa aplicação, para além de explicarmos o papel específico de cada uma.

4. **Especificação e Requisitos do Sistema:** Na especificação e requisitos do sistema, esclarecemos as especificações técnicas e os requisitos mínimos necessários para o bom funcionamento do nosso sistema.

5. **Projeto e Implementação:** Nesta parte, descrevemos o desenho e a implementação da aplicação, onde incluímos cada componente e a forma como foram integrados.

6. **Testes e Avaliações:** Apresentamos nesta seção, os testes efetuados à aplicação e os resultados obtidos, tendo em atenção os aspetos como a usabilidade, a eficiência do algoritmo do reconhecimento facial, etc.

7. **Conclusão:** Na conclusão, como o próprio nome indica, sintetizamos todas as principais descobertas, aprendizagens e resultados do nosso projeto.

8. **Bibliografia:** Por último, apresentamos todas as referências bibliográficas, usadas durante o desenvolvimento deste projeto.

Em conjunto com o relatório, desenvolvemos a aplicação "FaceShield" que fornece ao utilizador um sistema de autenticação baseado em reconhecimento facial. Como complemento, disponibilizámos também um manual de utilizador, para servir de apoio e guiar os utilizadores na utilização correta e eficiente da nossa aplicação.

2 Fundamentação Teórica

2.1 Inteligência Artificial

Não existe uma definição concreta do que é mesmo inteligência artificial, mas todas estão em sintonia, que se trata de uma área que tem como essência a capacidade que as máquinas têm de resolver problemas da mesma maneira que os humanos o fariam, isto é, através das suas capacidades cognitivas (Aprender, Raciocinar e Realizar tarefas).

Muitas vezes considerado o pai da ciência computacional moderna, Alan Turing, ficou famoso pela criação dos primeiros computadores modernos, pela sua descodificação de máquinas enigma alemãs durante a segunda guerra mundial e posteriormente detalhando um procedimento conhecido como o teste de Turing, formando assim a base para a inteligência artificial.

A Inteligência artificial (IA), teve o seu início na década de 1950, porém só com o aumento do poder computacional, e o desenvolvimento de algoritmos mais sofisticados, a partir do final do século XX e início do século XXI, é que evoluiu de forma significativa. Com esta evolução a Inteligência artificial (IA) começou a ser usada em diversas áreas, como finanças, medicina, telecomunicações, indústria, e, obviamente, na segurança digital.

O impacto da Inteligência Artificial (IA) no mundo é por isso muito profundo e multidimensional. A inteligência artificial (IA) tem o potencial de transformar muitos aspetos da vida quotidiana, tais como o diagnóstico de doenças, a otimização de processos em fábricas, até à previsão de tendências de mercado.

As capacidades atuais da Inteligência Artificial (IA) são imensas, e estão em constante evolução. Pois, atualmente, a Inteligência Artificial (IA) já é capaz de reconhecer e interpretar a fala humana, jogar jogos complexos como o xadrez, conduzir veículos autónomos, reconhecer imagens, entre outros. Estas capacidades estão diretamente relacionadas com o avanço da tecnologia.

No campo da segurança digital, a Inteligência Artificial (IA) tem um papel importantíssimo, pois já é capaz de capacitar sistemas para aprender e se adaptar, pois, permite a criação de sistemas de segurança muito mais robustos e versáteis. Esta capacidade é de elevado interesse, especialmente na área de autenticação, onde a Inteligência Artificial (IA) pode criar sistemas de autenticação, baseados em biometria, como o reconhecimento facial (Objeto do nosso trabalho). Estes sistemas mostram-se melhores e mais eficientes do que os métodos de autenticação baseados em palavras-passe, pois são mais difíceis de enganar e não requerem que o utilizador se lembre de um código secreto. Outra mais-valia, consiste na capacidade destes sistemas poderem aprender, adaptar-se e evoluir, tornando-se mais eficazes com o tempo. Realçamos que é particularmente importantíssimo num mundo onde as ameaças á segurança estão constantemente a evoluir.

Inteligência artificial para John McCarthy

Para um dos pais da inteligência artificial, o professor John McCarthy (1927-2011) esta, trata- se, não só da ciência, mas também da engenharia que torna as máquinas inteligentes, dando uma especial ênfase aos programas informáticos inteligentes. O autor sugere alguns ramos da Inteligência Artificial (IA) que, vão desde a lógica, ao reconhecimento de padrões, à inferência de realidades a partir de outras já existentes/observadas, ao planeamento estratégico, à epistemologia, à ontologia, entre outros.

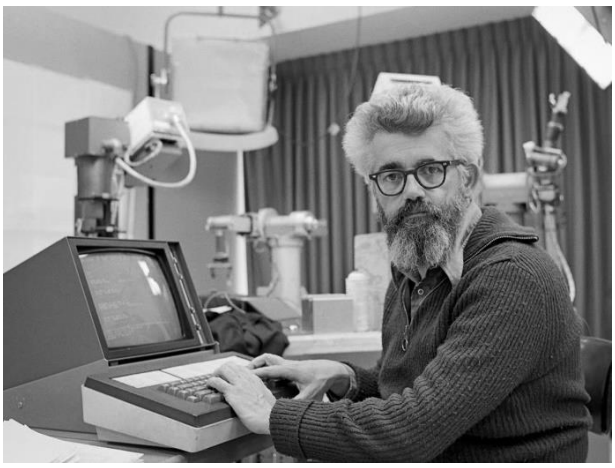


Figura 1 - Pai da inteligência Artificial, John McCarthy

Fonte:Chuck Painter

2.2 Aprendizagem de Máquina (Machine Learning)

A Aprendizagem de Máquina (ou Machine Learning) é um subcampo da Inteligência Artificial. Em termos mais concretos, o Machine Learning, permite que um sistema melhore o seu desempenho numa tarefa específica, através da experiência adquirida, por meio da análise de dados, porque se dedica ao desenvolvimento de algoritmos e técnicas que permitem aos computadores aprender a partir dos dados.

"O aprendizado de máquina, do inglês Machine Learning (ML), consiste na criação de algoritmos, com o fim de permitir a um computador reconhecer padrões e realizar inferências, sem que ele seja diretamente programado para isso. Um sistema baseado em algoritmos de aprendizagem é capaz de, pouco a pouco, melhorar os seus resultados à medida em que vai sendo exposto a dados e consegue acumular experiência a partir deles, algo similar ao que acontece no aprendizado humano. [Alpaydin, 2020]"

"Num texto de 1948, Turing também já descrevia na "máquina inteligente" alguns dos conceitos de Inteligência Artificial modernos como, o de Algoritmo Genético e o de Conexionismo, utilizados em redes neurais para aprendizado de máquina. [Turing and Copeland, 2004]"

Os principais objetivos do Machine Learning, passam pela criação de modelos capazes de aprender a partir de dados, sem que haja uma programação específica para cada tarefa. Podem ser supervisionadas, não supervisionadas, por reforço e semi-supervisionadas, dependendo de como os dados são apresentados ao sistema.

Paralelamente à inteligência Artificial, o Machine Learning também se iniciou na década de 1950, porém com um crescimento mais lento, devido sobretudo às limitações tecnológicas e à falta de dados disponíveis para treinar os modelos. Contudo, a partir da década de 1990, a Aprendizagem de Máquina tem conseguido uma expansão rápida e contínua, devido ao aumento exponencial da capacidade de processamento e do volume de dados disponíveis.

O Impacto do Machine Learning no mundo, é significativo e está em crescente evolução. Tal como a Inteligência Artificial (IA), esta tecnologia também é utilizada numa variedade de campos, como por exemplo a medicina, publicidade, finanças, e-commerce, entre outros, disponibilizando benefícios que ajudam na melhor tomada de decisão, em previsões mais precisas, bem como na personalização de serviços.

A Aprendizagem de Máquina, também é particularmente útil, na área de segurança digital. Pode ser usado para detetar padrões de atividade suspeita, identificar malware, melhorar os sistemas de autenticação, etc. No contexto da autenticação, por exemplo, o Machine Learning pode ser utilizado para criar modelos de reconhecimento facial altamente precisos, que conseguem aprender e adaptar-se às mudanças nas condições de iluminação, ângulos de visão, entre outros. Tornando-os mais convenientes e seguros, do que os sistemas baseados em palavras-passe, pois estes modelos não dependem de informações que o utilizador precise de se lembrar. Em conclusão, a Aprendizagem de Máquina é altamente importante, na revolução da forma como autenticamos os utilizadores, sendo crucial para um futuro digital mais seguro.

CLASSIFICAÇÃO DOS SISTEMAS DE ML				
MODOS DE APRENDIZADO	PARADIGMAS DE APRENDIZADO	LINGUAGEM DE DESCRIÇÃO	FORMAS DE APRENDIZADO	TAREFAS DE APRENDIZADO
Supervisionado	Simbólico	Exemplos	Incremental	Classificação
Não Supervisionado	Estatístico	Hipóteses	Não Incremental	Regressão
Semissupervisionado	Conexionista	Conhecimento de domínio		
	Genético			

Figura 1 – Características gerais dos sistemas de aprendizado por máquinas

Fonte: Renata Stange

2.2.1 Tipos de Aprendizagem

Em Machine Learning, existem 4 tipos principais de aprendizagem: Aprendizagem Supervisionada, Aprendizagem Não Supervisionada, Aprendizagem por Reforço e Aprendizagem Semi-Supervisionada. Cada uma com as suas próprias características, objetivos e aplicações.

- *Aprendizagem Supervisionada*

A aprendizagem supervisionada é a mais comum dos 4 tipos principais de aprendizagem em Machine Learning. Na Aprendizagem Supervisionada, o objetivo é que o algoritmo aprenda a mapear entradas e saídas corretas, de forma a poder fazer previsões precisas para novas entradas, pois nela, os algoritmos são treinados num conjunto de dados rotulados, onde tanto as entradas quanto as saídas desejadas são fornecidas.

As inferências de um modelo de Aprendizagem Supervisionado, são de grande importância para classificações e regressões. Elas tornam possível a previsão de casos futuros não presentes durante o treinamento e a deteção de fraudes a partir da perceção de outliers. [Alpaydin, 2020].

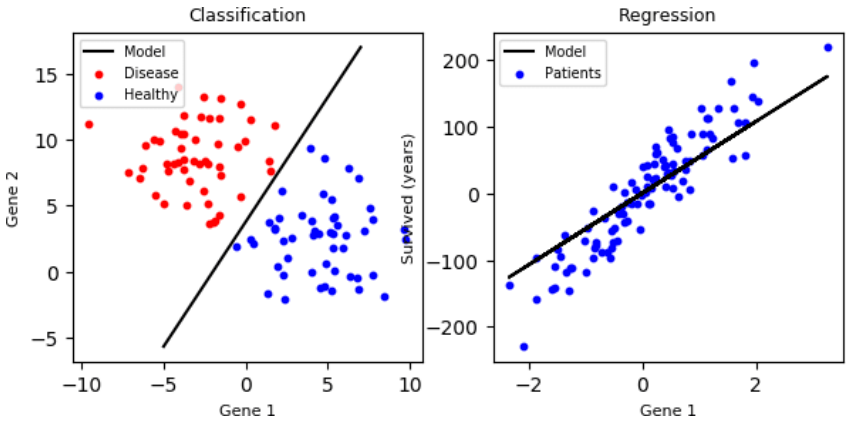


Figura 3- Classificao e Regressao

Fonte:Statplacep

Vantagens: A eficácia, facilidade de entender e medir o seu desempenho.

Desvantagens: A necessidade de um grande volume de dados rotulados, que são difíceis e dispendiosos de obter**.**

- *Aprendizagem Não Supervisionada*

Na aprendizagem não supervisionada, os algoritmos são treinados num conjunto de dados que não são rotulados, ao contrário da anterior. O objetivo traduz-se em encontrar padrões e estruturas implícitas nos dados. A utilização deste tipo de aprendizagem é mais comum na resolução de problemas de agrupamento, deteção de anomalias e redução da dimensionalidade (redução da quantidade de variáveis aleatórias, obtendo um conjunto de variáveis principais).

Vantagens: A capacidade de descobrir padrões não observados nos dados e a não necessidade de dados rotulados.

Desvantagens: Os resultados são mais difíceis de interpretar e a qualidade da aprendizagem é também mais difícil de medir.

- *Aprendizagem por Reforço*

A aprendizagem por reforço é um tipo de aprendizagem em que um agente aprende a tomar decisões interagindo com o ambiente onde se encontra. Para isso, recebe recompensas ou penalizações (reforços) com base nas ações que realiza, com o objetivo de maximizar a recompensa total a longo prazo.

Vantagens: A capacidade de aprender a partir da interação com o ambiente e a capacidade de lidar com problemas onde a solução implica uma sequência de decisões.

Desvantagens: A aprendizagem por reforço pode ser mais complexo e requer um ambiente que permita a interação e experimentação.

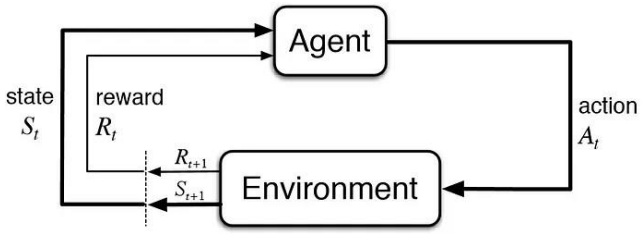


Figura 4 – Esquema de como Funciona Reinforcement Learning

Fonte: Shweta Bhatt

- *Aprendizagem Semi-Supervisionada*

No aprendizado semi-supervisionada, somente alguns dados de uma grande quantidade são supervisionados. Esse tipo de aprendizagem tem como meta, a realização do treinamento de categorias de classificação, em situações que existam um pequeno conjunto de dados rotulados com um conjunto grande de dados não rotulados.

Vantagens: Os algoritmos que usam esta abordagem são capazes de melhorar significativamente o desempenho da aprendizagem supervisionada.

Desvantagens: Este método de aprendizagem ainda está sujeito a muitos erros, devido à presença de ruído ou informações insuficientes.

Em conclusão, estes diferentes tipos de aprendizagem têm papeis importantes no campo de Machine Learning, e permitem-nos abordar uma grande diversidade de problemas de formas distintas. Logo a escolha do tipo de aprendizagem a utilizar, depende sempre do problema específico, dos dados disponíveis e do conhecimento prévio que já possuímos.

2.2.2 Redes Neurais (Neural Networks)

As redes neurais são uma forma de inteligência artificial, que procura replicar/imitar a maneira como os humanos aprendem. São sistemas computacionais com nós interconectados, que se assemelham à estrutura de um cérebro humano, visto que estes sistemas aprendem a realizar tarefas considerando exemplos, que geralmente não são programadas com tarefas específicas.

Vantagens: A principal vantagem desta forma de inteligência artificial inclui a sua capacidade de aprender e também de modelar relações que são não-lineares e complexas.

Desvantagens: A desvantagem traduz-se na necessidade de uma enormíssima quantidade de dados para treino e o risco de sobreajuste (overfiting). Isto significa que o modelo aprende a reconhecer padrões específicos dos dados de treino, incluindo ruídos e outliers, tão perfeitamente que falha ao generalizar para novos dados (dados que não foram usados durante o treino). Em bom rigor, o modelo está tão bem ajustado aos dados de treino que o seu desempenho deixa de ser eficiente quando lhe são apresentados novos dados, não vistos anteriormente. Para evitar o sobreajuste, as técnicas mais comuns são: a validação cruzada, o uso de conjuntos de validação, a regularização (adicionando uma penalidade à complexidade do modelo na função de perda) e a paragem antecipada (parar o treino antes de o modelo começar a sobreajustar).
Logo, evitar o sobreajuste é um grande desafio no design e treino de modelos de machine learning.

- *Neurónios*

O neurónio é a unidade fundamental de uma rede neural. Cada um recebe várias entradas, aplica-lhes pesos (são ajustados durante o processo de aprendizagem), e passa a soma ponderada, através de uma função de ativação para produzir a saída.

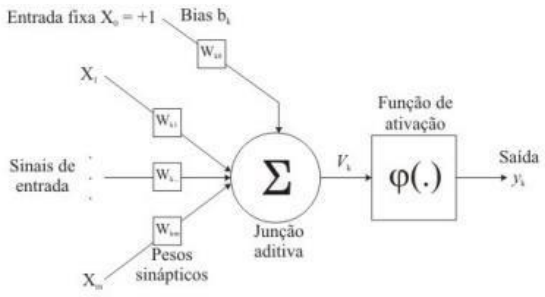


Figura 3 – Neuronio Artifical e o bias como o peso da sinapse (*Wo*)

Fonte: Shweta Bhatt

Os neurónios são capazes de aprender a partir dos erros cometidos através do ajuste dos pesos, possibilitando à rede neural aprender ao longo do tempo.

Um neurónio, é suposto reproduzir a mesma função de um neurónio biológico

Vantagens: O uso de neurônios, tem como vantagem principal a capacidade de aprender e adaptar-se a novos dados.

Desvantagem: Traduz-se na complexidade e dificuldade em interpretar o processo de aprendizagem.

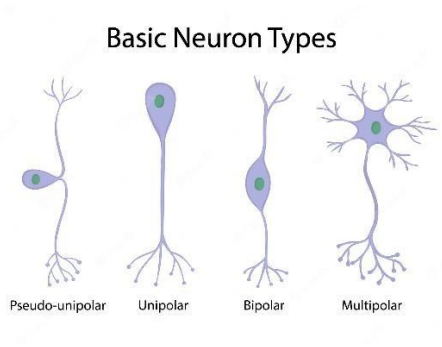


Figura 6– Tipos de Neurónios Biológicos

Fonte: Sanar Medicina

- *Perceptrões*

Um perceptrão é uma das formas mais simples da rede neural, pois traduz-se num único neurónio com entradas ponderadas, uma função de ativação e uma única saída. Sendo capaz de classificar dados linearmente separáveis, sendo normalmente usado em tarefas de classificação binária. (A classificação binária é uma tarefa de aprendizagem de máquina supervisionada que envolve a classificação de dados em um de dois grupos, normalmente representados como 0 e 1.) Por exemplo, uma aplicação comum de classificação binária é a determinação, de se um email é spam (1) ou não-spam (0), ou se uma transação com cartão de crédito é fraudulenta (1) ou não-fraudulenta (0). Outro exemplo seria o diagnóstico médico, onde é preciso classificar se um paciente tem uma determinada doença (1) ou não tem (0).

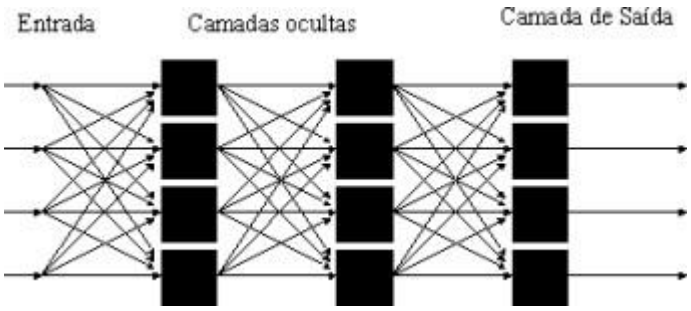


Figura 7 – Esquema de redes de perceptrão

Fonte:Zsolt Kovács

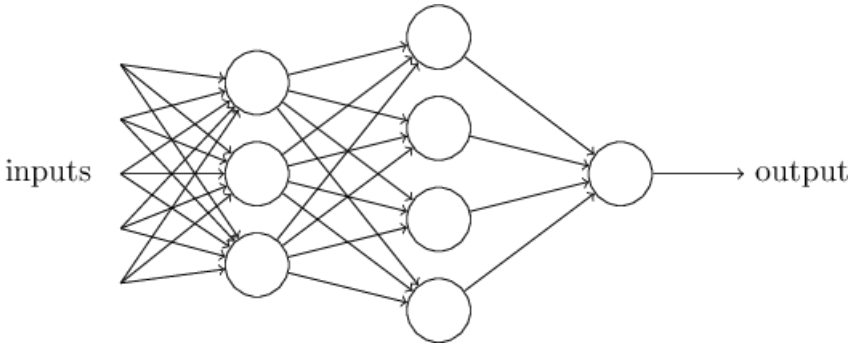


Figura 8 – Modelo de como funciona perceptrão

Fonte:Scesi UMSS

Vantagem: A principal vantagem é a sua simplicidade que possibilita um fácil entendimento e uma fácil implementação.

Desvantagem: Contudo, o perceptrão é incapaz de lidar com problemas de classificação não-linearmente separáveis. Ou seja, situações em que os dados não podem ser separados por uma linha reta (em duas dimensões) ou um hiperplano (três ou mais dimensões), não sendo possível traçar uma linha reta, ou hiperplano para dividir as classes. Para resolver estes problemas, são geralmente usados métodos mais complexos, como redes neurais ou máquinas de vetores de suporte (SVM) com um kernel não linear. Máquinas de Vetores de Suporte (SVMs, do inglês Support Vector Machines) são um tipo de modelo de aprendizagem supervisionado, utilizado para problemas de classificação e regressão. As Máquinas de Vetores de Suporte (SVMs), são especialmente conhecidas pela sua capacidade de lidar com problemas de classificação não-linearmente separáveis. No entanto, na prática, muitos problemas de classificação não são linearmente separáveis. Para lidar com esses casos, as Máquinas de Vetores de Suporte (SVMs), podem usar o que se chama de "truque do kernel" (kernel trick). O truque do kernel, permite mapear os dados para um espaço de dimensões mais altas, onde eles podem ser linearmente separáveis. Um kernel não-linear (como o kernel radial basis function, ou RBF), permite criar limites de decisão complexos e não-lineares no espaço original dos dados.

- *Aprendizagem Profunda (Deep Learning)*

A aprendizagem profunda, é um subcampo da aprendizagem de máquina, que se concentra em algoritmos inspirados pela estrutura e função do cérebro, chamados redes neurais artificiais. O "profundo" na aprendizagem profunda, refere-se ao número de camadas através das quais os dados são transformados, logo, mais camadas permitem modelos mais complexos e mais eficazes.

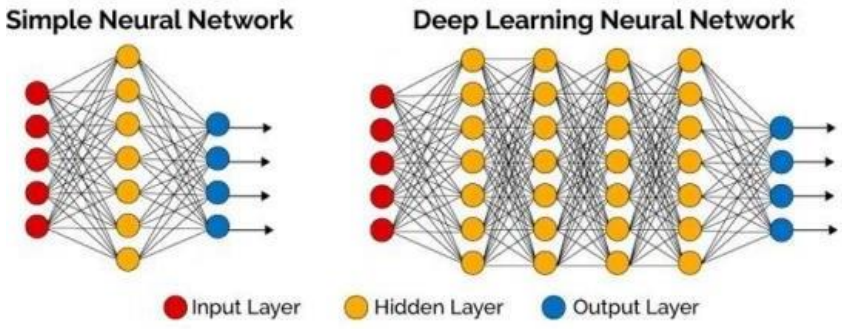


Figura 9 – Diferença entre rede neural comum e de deep learning

Fonte: Dnmetechs, 2017

Vantagens: As vantagens da aprendizagem profunda, traduzem-se na sua capacidade para lidar com grandes conjuntos de dados, e na sua habilidade em aprender características automaticamente a partir dos dados.

Desvantagens: As desvantagens, consistem na necessidade de requererem uma grande quantidade de dados e de poder computacional.

Concluindo, as redes neurais, com os seus neurónios, perceptrões e técnicas de aprendizagem profunda, representam uma ferramenta poderosa para o Machine Learning, pois elas oferecem, um método robusto para aprender a partir de dados complexos e a capacidade de se adaptar a novos dados, apesar dos seus desafios em termos de necessidade de dados e interpretabilidade ou seja, esta característica refere-se à clareza, ou à compreensibilidade das razões pelas quais o modelo fez uma determinada previsão ou classificação. Um modelo é considerado interpretável, se um humano consegue entender facilmente a lógica por trás das decisões do modelo.

2.2.3 Rede neural convolucional (CNN)

As redes neurais convolucionais (CNN) são um tipo de modelo de deep Learning, muito aplicadas em análises visuais. As redes neurais convolucionais (CNN), são criadas com o objetivo de reconhecer padrões visuais, diretamente a partir de imagens com a menor quantidade de pré-processamento possível. As redes neurais convolucionais, conseguem identificar características com uma quantidade mínima de pré-processamento, que faz com que sejam eficientes para classificar e categorizar imagens.

As redes neurais convolucionais, também conhecidas como ConvNet, são estruturas de aprendizagem de máquina, que trabalham com a entrada de maneira matricial. Ao contrário dos algoritmos tradicionais de feedforward, que trabalham com a entrada inteira simultaneamente, as redes neurais convolucionais (CNN) trabalham com apenas uma região por vez. Isso permite um processamento mais eficiente das entradas, analisando informações adicionais, como a posição relativa das entradas. Assim, é largamente utilizado em aplicações que envolvam entradas mais complexas, como processamento de imagens [Heaton, 2017].

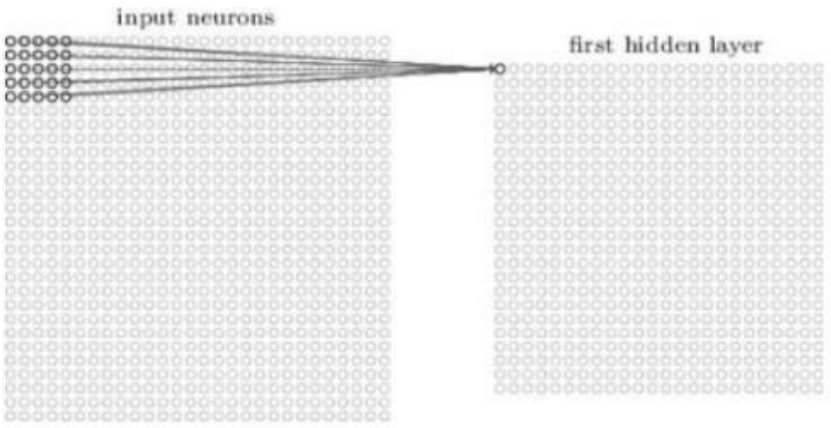
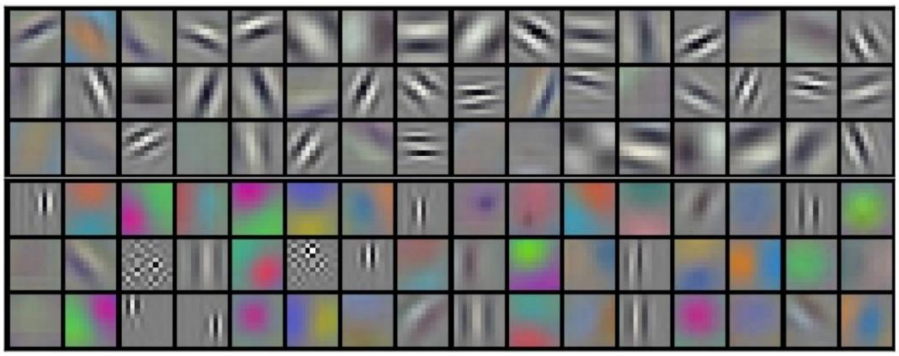


Figura 10 – Ilustração de como funciona uma CNN

Fonte: Zhang et eal.., 2016



Vantagens: As redes neurais convolucionais (CNN), têm um elevado grau de eficiência e precisão no processamento de imagens e possuem também uma capacidade de lidar automaticamente com variações de imagens (luz, ângulo, orientação).

Desvantagens: Para treinar uma rede neural convolucional (CNN), é necessário um grande volume de dados para treino, para além se ter uma complexidade computacional elevada.

- *Embeddings*

Os embeddings, são uma técnica usada para converter categorias com dimensão elevada em vetores de menor dimensão, mantendo a informação pertinente. Os embeddings são úteis para lidar com dados categóricos e são normalmente usados em problemas de linguagem natural. Eles são bastante úteis para obter o significado semântico de palavras ou entidades e para capturar as relações entre elas.

Transformar um dado em embedding, é mapeá-lo numa representação vetorial. Num exemplo de imagem, o embedding do FaceNet é transformado para um espaço Euclidiano d-dimensional, d sendo 128. [Schroff et al., 2015a]

Vantagens: Os embeddings reduzem o tamanho e a extração de relações semânticas complexas.

Desvantagens: Há a dificuldade em escolher qual a dimensão certa para o embedding, pois uma dimensão muito alta, pode levar a um sobreajuste. Já uma dimensão muito baixa, pode fazer com que haja perda de informação importante.

Concluindo, as redes neurais convolucionais e os embeddings são ferramentas poderosas na aprendizagem de máquina. As redes neurais convolucionais (CNN), com a sua elevada capacidade em identificar características visuais, em conjunto com a capacidade dos embeddings de lidar com dados categóricos e extrair relações semânticas, oferece um método robusto para lidar com uma variedade de problemas.

Porém, estes métodos necessitam de um equilíbrio inteligente entre a complexidade e o desempenho, para além de uma consideração cuidadosa das suas necessidades de dados.

2.2.4 Algoritmos de Prova de Vida (Liveness Detection)

Os algoritmos de Prova de Vida, também denominados Liveness Detection, consistem em técnicas usadas, para garantir que a fonte de uma biometria é um indivíduo vivo, em vez de uma imitação ou representação. O principal objetivo, é combater ataques de falsificação, como a utilização de fotografias, vídeos, máscaras ou outro artefacto que possa ser utilizado para tentar enganar o sistema de autenticação.

Existem várias técnicas para a Prova de Vida, sendo exemplos:

1) **Análise de Textura:** Esta técnica, analisa a textura da pele para verificar se o objeto em questão é uma pessoa real. As fotos tendem a possuir uma textura diferente da pele humana real.

2) **Deteção de Batimento Cardíaco ou Respiração:** Estes algoritmos, procuram sinais subtis de vida, como o movimento rítmico do peito de uma pessoa ao respirar, ou o pulsar de um vaso sanguíneo.

3) **Testes de Interação:** Nestes testes, o algoritmo de Prova de Vida pede ao usuário para executar uma determinada ação, como mover a cabeça numa certa direção, piscar, sorrir, entre outros. No nosso caso, o nosso algoritmo de Prova de Vida solicita ao utilizador que olhe para a esquerda, depois para a direita, e que no final sorria.

Vantagens: Estes algoritmos, têm a capacidade de adicionar uma camada adicional de segurança à autenticação biométrica, para além da capacidade de operar em tempo real.

Desvantagens: Apesar de proporcionar uma camada adicional de segurança, há sempre um risco de haver falsos positivos (por exemplo, um teste de interação pode falhar se o usuário não perceber a instrução), para além da necessidade de hardware especializado para certas técnicas, como a deteção de batimento cardíaco e a deteção do pulsar de um vaso sanguíneo.

Concluindo, os algoritmos de Prova de Vida, são uma componente essencial de um sistema de autenticação biométrica eficiente. Ao pedirmos ao usuário para realizar ações específicas, conseguimos melhorar a segurança do sistema e garantir que a fonte da biometria é um ser humano real vivo. No entanto, para se conseguir fazer uma implementação eficaz desses algoritmos, é necessário considerações cuidadosas sobre a usabilidade e robustez.

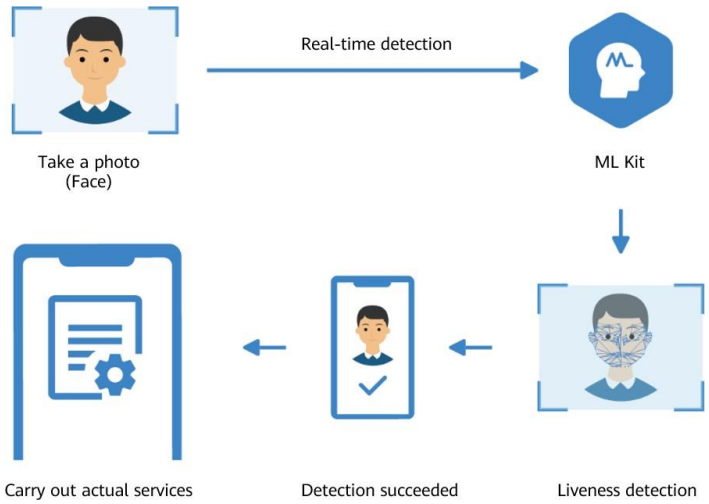


Figura 11- Liveness Detection Fonte: Huawei, 2022

2.2.5 Biometria Facial

A biometria facial é uma tecnologia de reconhecimento, que usa características faciais únicas para identificar um indivíduo. Isto é realizado, através da análise e medição de padrões específicos na estrutura do rosto da pessoa, que posteriormente são convertidos num formato digital, que pode ser comparado com rostos armazenados numa base de dados.

Vantagens: A biometria facial apresenta diversas vantagens, sendo estas:

1) **Não invasiva:** A biometria facial, é uma técnica de identificação passiva, que não necessita da interação direta com o indivíduo.
2) **Flexível e versátil:** Pode ser aplicada numa elevada gama de contexto e com uma diversidade de equipamentos, desde smartphones a câmarasde segurança avançadas.
3) **Rápida e eficiente:** A identificação facial, é geralmente muito rápida, o que a torna ideal, para situações onde elevada velocidade é requerida, como em controlos de acesso ou aplicações móveis.

Desvantagens: A biometria facial, apresenta algumas desvantagens:

1) **Suscetível a alterações físicas:** Com o decorrer do tempo, a aparência de uma pessoa vai mudando, através do envelhecimento, maquilhagem, óculos ou alterações no cabelo, que podem afetar a eficiência do reconhecimento facial.
2) **Sensível à luz e ao ângulo:** A luz ambiente e o ângulo de visão, têm a capacidade de influenciar a precisão do sistema.
3) **Preocupações com a privacidade:** A coleta e o armazenamento de dados biométricos faciais, fazem com que se façam questões sobre a privacidade e a segurança dos dados.

Concluindo, a biometria facial é uma ferramenta útil para a identificação e autenticação. A sua capacidade de operar de forma passiva e com elevada velocidade, torna-a numa solução eficaz para diversas aplicações. Porém, ainda existem desafios significativos a serem superados, especialmente em relação à variação da aparência de um indivíduo e questões de privacidade. Estes desafios são áreas de estudo ativas, e prevê-se que haja melhorias à medida que a tecnologia avança.

3 Ferramentas utilizadas

3.1 Flutter

O Flutter é uma estrutura de desenvolvimento de interface do utilizador, criada pela Google, que nos permite criar aplicações móveis nativas de alta qualidade para Android e iOS a partir de uma única base de código. O Flutter, é famoso pela sua rápida implementação, com a funcionalidade de recarregamento a quente, permitindo assim aos programadores verem as alterações que fazem no código em tempo real, não necessitando de estar sempre a reiniciar a aplicação. Ao escolhermos esta ferramenta, conseguimos desenvolver a nossa aplicação de maneira eficiente e eficaz.

3.2 Google ML Kit

O Google ML Kit fomenta uma API de nível alto para várias tarefas de Machine Learning, incluído reconhecimento facial. Para além da API ML kit, vários algoritmos e técnicas, são utilizados para potenciar as capacidades de reconhecimento facial. Aqui estão os algoritmos mais importantes utilizados no Google ML Kit para reconhecimento facial:

Deteção Facial: Ml Kit utiliza uma combinação de algoritmos de Machine Learning, incluídas modelos de aprendizagem profunda para detetar faces em imagens ou mesmo em vídeos. Os algoritmos subjacentes, empregam redes neurais convolucionais (CNNs) para analisar recursos de imagem e identificar pontos de referência faciais.

Deteção de pontos de referência faciais: O ML Kit usa modelos baseados em técnicas de aprendizagem profunda, para detetar e localizar pontos de referência faciais importantes, como olhos, nariz e boca. Esses modelos, são treinados para reconhecer estruturas faciais específicas,e fornecer coordenadas de referência precisas.

Classificação facial: o ML Kit emprega algoritmos de aprendizagem de máquina para classificar vários atributos ou características faciais. Por exemplo, pode determinar atributos como sorriso, olhos fechados, boca aberta e outras expressões faciais.

É importante observar que, o ML Kit abstrai os algoritmos subjacentes e os detalhes de implementação, fornecendo uma API simplificada e fácil de usar para desenvolvedores. Isto permite que, os desenvolvedores utilizem recursos de deteção facial sem a necessidade de implementar algoritmos complexos diretamente.

3.3 TFLite

É um conjunto de bibliotecas, ferramentas, datasets e recursos, que auxiliam no desenvolvimento de soluções de Machine Learning e treinamento de modelos. É uma ferramenta open-source mantida pela Google e desenvolvida em Python e C++ [Abadi et al., 2015]

3.4 FaceNet

O modelo "MobileFaceNet", normalmente usa uma combinação de redes neurais convolucionais (CNNs) e técnicas de aprendizagem profunda para tarefas de reconhecimento facial. É um modelo leve e eficiente, projetado especificamente para dispositivos móveis e embarcados.

Aqui está uma breve visão geral dos algoritmos e técnicas comumente usadas no modelo MobileFaceNet.

Redes neurais convolucionais (CNNs): MobileFaceNet, emprega uma série de camadas convolucionais, para extrair recursos hierárquicos de imagens de entrada. Essas camadas aplicam filtros para capturar vários padrões e recursos em diferentes escalas.

Convolução separável em profundidade: MobileFaceNet, usa convoluções separáveis em profundidade, que dividem a operação de convolução padrão em convoluções separadas em profundidade e em ponto. Essa técnica, reduz significativamente a complexidade computacional e o tamanho do modelo, mantendo a precisão.

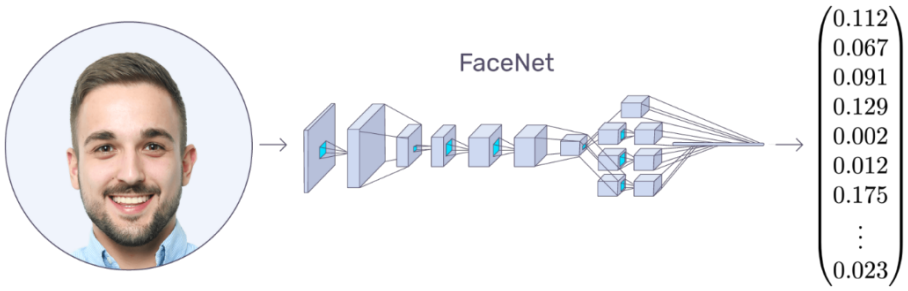


Figura 12- Modelo Facenet

Fonte: arsfutura

Estruturas de gargalo: MobileFaceNet, incorpora estruturas de gargalo para aumentar ainda mais a eficiência do modelo. Os gargalos, comprimem os mapas de recursos de entrada antes de aplicar as convoluções, reduzindo a carga computacional sem sacrificar o desempenho.

ArcFace Loss: MobileFaceNet, geralmente utiliza a função de perda ArcFace durante o treinamento. ArcFace é uma função de perda popular, para tarefas de reconhecimento facial, que aumenta o poder discriminativo das incorporações faciais do modelo. Ele reforça a compacidade intraclasse e a separação interclasse no espaço de incorporação, melhorando a capacidade do modelo de distinguir entre diferentes indivíduos.

Estes são, alguns dos principais algoritmos e técnicas normalmente empregados no modelo MobileFaceNet. A arquitetura exata e os detalhes de implementação, podem variar um pouco, dependendo da versão específica ou variante do MobileFaceNet usada.

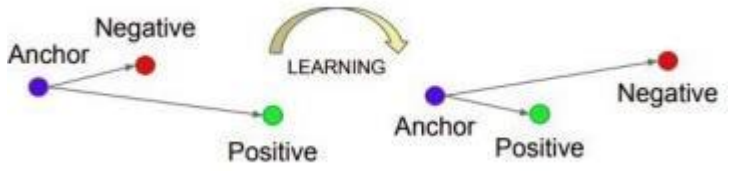


Figura 13- Face net

Fonte: arsfutura

3.5 Firebase

O Firebase, é uma plataforma de desenvolvimento de aplicações da Google, que fornece uma gama de serviços, como autenticação, base de dados em tempo real, armazenamento em nuvem, hospedagem, etc. No nosso projeto, utilizámos o Firebase para gerir a nossa base de dados, fornecendo assim uma solução de confiança e eficiente para o armazenamento e recuperação de dados dos utilizadores.

3.6 Sistemas Operativos

3.6.1 Android e iOS

Após a fase de desenvolvimento, a nossa aplicação “Face Shield” está desenhada para funcionar tanto em dispositivos Android como em dispositivos iOS.

O Android, também desenvolvido pela Google, é o sistema operativo móvel mais amplamente utilizado a nível mundial. Tem como base o kernel do Linux e é projetado principalmente para dispositivos móveis com ecrã táctil, como smartphones e tablets. O Android tem a fama de ser personalizável, o que permite aos fabricantes de dispositivos e programadores adaptarem o software consoante às suas necessidades.

Já o iOS, é o sistema operativo móvel desenvolvido pela Apple Inc. para os seus respetivos dispositivos móveis, como o IPhone e IPad. O iOS é conhecido por possuir uma interface de utilizador suave e intuitiva e graças ao seu ecossistema fechado e controlado, proporciona um nível de segurança mais elevado em relação ao Android.

Embora ambos os sistemas operativos possuam os seus fortes e fracos, o objetivo principal do nosso trabalho foi criar uma aplicação que pudesse ser utilizada de maneira eficiente em ambas as plataformas. Proporcionando assim uma maior flexibilidade aos utilizadores e permitindo que a nossa aplicação alcance uma gama de utilizadores mais diversificada.

Concluindo, no desenvolvimento da aplicação “Face Shield”, recorremos a uma séria de ferramentas e tecnologias, como o Flutter, Facenet, Firebase e Google ML Kit. Estas ferramentas e tecnologias permitiram-nos criar uma aplicação eficaz de deteção e reconhecimento facial para melhorar a segurança digital. Como dito anteriormente, esta aplicação foi desenvolvida no sistema operativo Windows, devido ao facto do mesmo possuir várias funcionalidades e suporte para desenvolvimento de software. Após o desenvolvimento, a “Face Shield” é capaz de funcionar tanto em Android como iOS, garantindo mais acessibilidade a uma grande base de utilizadores. Ao desenvolvermos esta aplicações, mostrámos as capacidades das tecnologias modernas de reconhecimento facial.

3.7 Dart

O Dart é uma linguagem de programação fortemente tipada inicialmente criada pela Google em 2011. A missão inicial do Dart era substituir o JavaScript para desenvolvimento de scripts em páginas web. Porém, com a evolução da linguagem e com o passar dos anos, ela hoje pode ser considerada uma linguagem multi-paradigma, embora a linguagem apresente fortes estruturas típicas de linguagens orientadas a objeto.

4 Especificação de requisitos do sistema.

A especificação de requisitos do sistema é um documento ou conjunto de documentos que descrevem as funcionalidades, a interface, desempenho e as limitações do sistema a ser desenvolvido. Esta especificação de requisitos do sistema é de elevada importância porque ajuda a garantir que todos os intervenientes entendem o que o sistema deve fazer e como deve funcionar, mitigando o máximo possível falhas de comunicação e compreensão e garantindo que não há mal-entendidos e que o produto final satisfaça as necessidades do utilizador.

4.1 Descrição geral

A aplicação “FaceShield” que nos propusemos a desenvolver, consiste num sistema de segurança digital baseado em Android e iOS, que utiliza uma tecnologia de reconhecimento facial para autenticar os utilizadores. A aplicação é desenvolvida usando várias ferramentas e tecnologias, sendo estas Flutter, Google ML Kit, TensorFlowLite, MobileFaceNet e Firebase.

Os requisitos do sistema têm vários aspetos importantes, sendo estes a capacidade de processar e reconhecer imagens faciais facilmente, proteger a privacidade do utilizador para

além de interagir de forma eficiente com a base de dados. Outro aspeto importante é a compatibilidade com os dispositivos Android e iOS, para além da capacidade de funcionar de forma eficaz com diferentes versões do sistema operativo Android e iOS.

O sistema também tem vários objetivos como: ser de fácil uso, com uma interface de utilizador simples e intuitiva e de fácil navegação. Outro objetivo importante traduz-se num bom desempenho em termos de velocidade e eficiência, bem como ser capaz de lidar com potencias e futuros problemas ou erros da melhor maneira possível.

4.2 Requisitos específicos

A nossa aplicação "Face Shield" foi concebida para ter o melhor funcionamento possível, porém possui alguns requisitos específicos para garantir o seu melhor funcionamento e eficiência. Logo, é aconselhado que os dispositivos dos utilizadores possuam os sistemas operativos mais recentes.

O espaço de armazenamento necessário para a instalação e execução da aplicação é de cerca de ~400 MB. Por isso, os utilizadores devem garantir que possuem espaço na memória no dispositivo antes de instalar a nossa aplicação. Em caso de falta de armazenamento, sugerimos que o utilizador limpe o armazenamento do dispositivo, desinstalando aplicações que não utilizada ou transferir os arquivos para um armazenamento externo como cartões de memória, unidades flash USB, discos rígidos externos.

Em relação à conectividade, é crucial que o dispositivo tenha uma conexão estável à Internet, pois a aplicação necessita de acesso á Internet para executar certas operações, de que são exemplo o registo e autenticação dos utilizadores.

Por fim, é imprescindível que o dispositivo possua uma câmara funcional e que o utilizador conceda permissão à nossa aplicação para usar a câmara disponível no telemóvel, de modo a ser possível usar a funcionalidade de reconhecimento.

Todos estes requisitos específicos têm como objetivo assegurar que a aplicação “Face Shield” possa oferecer a melhor experiência de utilização possível, e ao mesmo tempo cumprir as suas funções de forma eficaz e eficiente.

4.2.1 Requisitos de interface externa

Os requisitos de interface externa são essenciais para definir como o sistema se deve comportar quando é solicitado por um utilizador ou por outros sistemas, e também para detalhar o que é necessário em termos de hardware para suportar o sistema. De forma resumida, os requisitos de interface externa descrevem a forma como o sistema interage com o mundo externo, ou seja, com o utilizador, outros sistemas disponíveis e com o hardware.

Os requisitos de interface externa podem ter três categorias principais:

1) Requisitos de interface do utilizador
2) Requisitos de interface de hardware
3) Requisitos de interface de software

Os requisitos de interface do utilizador têm como foco a interação entre o sistema e o utilizador, especificando como o sistema deve responder às ações do utilizador e como deve mostrar a informação ao utilizador, para este a compreender da melhor maneira possível. Incluem por isso elementos que ditam a forma como o sistema é visualizado e navegado, como as informações são apresentadas, a resposta aos comandos efetuados, entre outros.

Os requisitos de interface de hardware descrevem como o sistema interage com o hardware, tal como o nome indica. Exemplos destes requisitos são: Quais os tipos de dispositivos de hardware o sistema deve suportar, quais tipos de conexões precisa, etc.

Por fim, os requisitos de interface de software, definem a maneira com o sistema se comunica e interage com os outros sistemas ou aplicações.

Especificamente na nossa aplicação “Face Shield” os requisitos de interface externa são os seguintes:

- **Requisitos de interface do utilizador:** A nossa aplicação possui uma interface de utilizador intuitiva de fácil utilização. As informações estão organizadas de forma clara e concisa, e a aplicação responde de forma eficaz e rápida às ações executadas pelo utilizador.
- **Requisitos de interface de hardware:** Esta aplicação "Face Shield" requer 400MB de armazenamento disponível, bem como uma câmara funcional para ser possível executar a funcionalidade de reconhecimento facial.
- **Requisitos de interface de software:** A nossa aplicação requer uma conexão à Internet estável para poder interagir com o Google ML Kit e o Firebase para algumas das suas funcionalidades, tais como o reconhecimento facial e a gestão da base de dados respetivamente.

4.2.2 Requisitos funcionais

Os requisitos funcionais são vitais para o desenvolvimento de um sistema, porque fornecem a base para o design, implementação e teste do sistema. Referem-se, portanto, às funcionalidades ou serviços específicos que um sistema deve proporcionar. Descrevendo o que o sistema deve fazer em determinadas condições ou como deve reagir a certos inputs.

Normalmente estes requisitos detalham diversos tipos de situações, por exemplo: A forma como os utilizadores vão interagir com o sistema, os processos que o sistema executa, o controlo das informações de entrada e saída, e respetivos comportamentos do sistema em determinadas situações.

Vantagens: Estes requisitos são fundamentais para a clarificação no entendimento geral das capacidades do sistema, também facilitam na descrição dos objetivos de projeto, bem como a utilidade no teste do sistema para garantir que conseguimos cumprir todos os requisitos.

Desvantagens: Os requisitos funcionais, sendo muito específicos ou complexos, podem levar a uma maior dificuldade de implementação do projeto ou a um sistema que não seja muito flexível a mudanças (pouca escalabilidade).

Na nossa aplicação "Face Shield", os requisitos funcionais incluem:

1) Autenticação do utilizador:
2) Reconhecimento facial
3) Prova de Vida (Liveness Detection)
4) Gestão de dados
5) Comunicação com servidores externos

1) A autenticação do utilizador permite ao utilizador criar uma conta, efetuar login, ver o registo de utilizadores guardados na base de dados, gerida pelo FireBase e fazer logout.
2) O reconhecimento facial deve ser desenhado de forma que o sistema seja capaz de detetar e reconhecer rostos em tempo real.
3) Na prova de vida (Liveness Detection), o sistema deve pedir ao utilizador para fazer ações específicas como olhar para a esquerda, olhar para a direita e sorrir e piscar os olhos.
4) Na gestão de dados, o sistema armazena e gera os dados de forma segura.
5) A comunicação com servidores externos, traduz-se na capacidade do sistema se comunicar com o Google ML Kit e o Firebase que são necessárias para a execução de algumas das suas funcionalidades.

Vantagem dos nossos requisitos funcionais: Os nossos requisitos garantem uma vasta solução para garantir a segurança e a privacidade dos dados do utilizador para alem da verificação da identidade do utilizador para evitar ataques de autenticação.

4.2.3 Requisitos de armazenamento de dados

Os requisitos de armazenamento de dados correspondem ao volume, velocidade, variedade, veracidade e valor dos dados que o sistema deve de ser capaz de processar. O armazenamento de dados é por isso uma parte importantíssima/crucial em qualquer sistema de informação, uma vez que é responsável por guardar e manter as informações produzidas e utilizadas por um sistema.

No nosso projeto, utilizámos o Firebase. O Firebase é uma plataforma de desenvolvimento de aplicações da Google, que foi criado pela empresa Firebase, Inc. em 2011 e mais tarde em 2014 foi adquirida pela Google.

Esta plataforma oferece uma variedade de funcionalidades, onde se inclui a base de dados em tempo real, a autenticação de usuários, o armazenamento de dados, o hosting, entre outros, por isso serve para desenvolver aplicações de alta qualidade, de forma rápida e aumentar o crescimento do respetivo público.

Vantagens: O Firebase é conhecido pela sua fácil utilização, escalabilidade automática, base de dados em tempo real e a integração com outros serviços da Google.

Desvantagens: A dependência de um único fornecedor (Google) e as limitações na consulta de dados complexos tornam o Firebase uma estrutura com custos muito elevados, em caso de muita utilização.

No nosso projeto “Face Shield” utilizámos o Firebase para gerir a base de dados. A nossa opção baseou-se no facto de ser uma solução eficaz que proporciona um armazenamento de dados robusto e escalável do email e dos dados da Face (Face Data). Os requisitos de armazenamento de dados incluem a capacidade de guardar informações de autenticação dos utilizadores, dados biométricos para reconhecimento facial e registos das provas de vida realizadas pelos utilizadores.

4.2.4 Atributos do sistema de software

Os atributos do sistema de software propriedades que têm o potencial de afetar o desempenho, a utilização, a eficiência, o quão fácil é usar o sistema, entre outros. Estes atributos permite obter segurança, desempenho, escalabilidade, confiabilidade, manutenibilidade, não repudiação, entre outros. Com estes atributos conseguimos ter bastantes pontos positivos que irão contribuir para o sucesso de uma aplicação, pois irão controlar a forma como a aplicação lida em diferentes circunstâncias e como os utilizadores que a usam pensam sobre ela.

Vantagens: Se se conseguir um sistema com um desempenho elevado e que seja minimamente seguro, este consequentemente se tornará mais atrativo para os utilizadores, aumentando a possibilidade de fazerem download, para além de conseguirmos cumprir os nossos objetivos de forma mais eficiente, para evitar conflitos.

Desvantagens: Uma possível desvantagem é a dificuldade em manter o equilíbrio entre a segurança e a facilidade de uso. Por um lado, se nos focarmos excessivamente na segurança, podemos tornar a aplicação mais difícil de se usar, por outro, se só focarmos na facilidade de uso, podemos correr vários riscos de segurança.

- *Segurança*

Garantir que o nosso sistema é seguro, é uma das preocupações mais importantes num sistema de software, especialmente em aplicações que lidam com dados pessoais e sensíveis que podem causar elevados estragos se caírem nas mãos erradas. Ao usar um sistema seguro, para além de protegermos os dados e conseguir controlar as funcionalidades do acesso não autorizado, garantimos vários objetivos importantes quando se pretende criar um sistema seguro, como a confidencialidade que toda a gente tem direito, a integridade, pois não queremos que alterem os nossos dados para além de conseguirmos oferecer disponibilidade das informações.

Na nossa aplicação “Face Shield”, a segurança é uma prioridade. A nossa aplicação usa um sistema de registo de utilizadores com password para controlar o acesso e evitar que pessoas não autorizadas tenham acesso à aplicação. Para além disso, todos os nossos dados transmitidos e armazenados são protegidos com medidas de segurança eficientes provisionadas pelo Firebase, para tentarmos garantir a sua confidencialidade e integridade. Isto garante que as informações pessoais dos utilizadores estão protegidas, fazendo com que os utilizadores gostem da nossa aplicação e deixem um feedback positivo e construtivo, sem esquecer de cumprir as obrigações legais e éticas.

4.2.5 Características do ambiente

As características do ambiente dizem respeito à parte física e lógica do ambiente onde um determinado sistema de software será implementado e operado. Estas tais características do ambiente incluem o hardware, o sistema operativo utilizado, rede e outras infraestruturas de TIAs infraestruturas de Tecnologias de Informação (TI) consistem em usar computadores para fazer o processamento, o armazenamento, a recuperação e a troca de todos os tipos de dados e informações. A Tecnologia de Informação (TI) pertence ao ramo das tecnologias da informação e comunicação (TIC)

As características do ambiente podem ter um impacto positivo ou negativo no desempenho, segurança e na utilização geral do nosso sistema. Isto tudo deve ser previsto durante o desenvolvimento e implementação do sistema, para evitar problemas futuros.

- *Hardware*

O Hardware corresponde à infraestrutura física em si que suporta o sistema de software. Exemplos de hardware são: servidores, computadores pessoais, dispositivos móveis, entre outros. Como foi dito anteriormente o hardware pode afetar tanto de forma positiva como negativa o desempenho e a capacidade geral do sistema, além da sua segurança e confiabilidade.

Um aspeto importante a ter em conta é que diferentes dispositivos são mais adequados para certas funções. Por exemplo, um servidor de armazenamento é mais adequado para suportar e manter um elevado número de transações. Já um dispositivo móvel com os recursos muito mais limitados é mais adequado para tarefas mais leves, pois tem menos capacidade de processamento, porém em contrapartida tem melhor desempenho que um servidor de armazenamento.

A nossa aplicação "Face Shield" foi cuidadosamente testada e otimizada para funcionar no maior número possível de dispositivos móveis, tanto em Android como iOS.

Durante o desenvolvimento da nossa aplicação, fomos testando a mesma com os nossos telemóveis pessoais para garantir que estava tudo a correr como planeado. Graças ao nosso trabalho árduo e também graças ao nosso uso de testes, estamos confiantes que a nossa aplicação é eficiente na maioria dos dispositivos móveis modernos para além de cumprir com as normas éticas de privacidade e segurança.

5 Problema

5.1 Problema a ser resolvido

O nosso projeto tem como tema central a "Autenticação de pessoas usando Machine Learning". Para conseguirmos atingir esse objetivo usamos uma aplicação de telemóvel chamada "Face Shield".

Devido às enormes dificuldades encontradas nos sistemas de autenticação por password, o uso de sistemas que usam o rosto para autenticar as pessoas tem crescido exponencialmente. Visto isto ser do nosso interesse, nós usámos ferramentas e bibliotecas específicas para termos um melhor desempenho.

No front-end usámos o flutter para a criação da interface do Utilizador. Uma das principais razões que nos levou a usar o flutter foi o facto de este conseguir correr tando em sistemas Android como iOS.

Já em relação à autenticação de pessoas através do reconhecimento facial, usámos o Google ML Kit, TFLite e MobileFaceNet. Optámos pela versão Lite do TF para ser mais leve e cómodo para o utilizador se autenticar de forma mais rápida.

Agora, para gerenciar a base de dados, usámos o Firebase, porque consideramos uma aplicação bastante eficiente e segura para guardar os dados do utilizador, respeitando assim as

éticas de privacidade e segurança. Para além disso, a aplicação dispõe de várias funcionalidades como base de dados em tempo real, extensões, funções para ser utilizado no back-end.

Em resumo, o projeto de desenvolvimento de uma aplicação de autenticação fácil utiliza várias tecnologias de ponta para criar uma solução inovadora que permite aos usuários se autenticarem facilmente usando reconhecimento facial, com a segurança e a facilidade de uso oferecidas pelo uso de ferramentas como Flutter, Google ML Kit, TFLite e FaceNet, além de ter o suporte e escalabilidade oferecidos pelo Firebase como base de dados.

5.2 Análise dos modelos

O modelo utilizado no nosso projeto foi o MobileFaceNet. O MobileFaceNet é uma rede neural profunda que é utilizada para se conseguir extrair recursos de uma imagem do rosto de uma pessoa. Foi disponibilizada em 2015 pelos pesquisadores do Google Schroff.

O MobileFaceNet, pega numa imagem do rosto da pessoa como entrada e gera um vetor de 192 números que correspondem às características mais importantes de um rosto, para evitar ambiguidade. Basicamente, o MobileFaceNet pega o rosto de uma pessoa e diminui para um vetor de 192 números.

Tem-se tornado uma prática bastante comum na aprendizagem de máquina atualmente mapear dados de alta dimensão (ex: imagens) em representações de baixa dimensão (embeddings).

Como o modelo MobileFaceNet (mobilefacenet.tflite) é um modelo pre-trainado podemos simplesmente utizar-lo em conjunto com um interpretador do TensorFlowLite para realizar os embbedings do rosto que iremos guardar na base de dados com o resto da informação do utilizador.

Simplificando o máximo possível, a MobileFaceNet é uma função que recebe a informação de um rosto como entrada e gera um vetor das características faciais mais importantes.

5.2.1 Processamento de imagens

Sistema Cliente-Servidor

A solução foi projetada para funcionar com a arquitetura descrita na Figura 14. O

sistema funciona com o back-end dentro da aplicação com uma base de dados exterior e um front-end a correr numa aplicação dum dispositivo móvel.

BackEnd

O back-end do projeto engloba todos os processos de manipulação de imagem para extração da face, criação do embedding e a API para a base de dados. O código foi feito em Dart, utilizando as bibliotecas Google Ml Kit (Face Detection) e TensorFlowLite em conjunção com MobileFaceNet para o tratamento do embedding. E para processamento de imagens (crop inicial da face do utilizador) utilizamos também uma biblioteca padrão dart. Para concluir utilizamos cloud_firestore.dart e firebase_auth.dartpara gerir a nossa base de dados Firebase.

Para o tratamento de falso positivos no login de utilizadores, é perguntado ao utilizador se o resultado corresponde com o resultado esperado, caso seja certo o utilizador e pedido a password e caso seja incorreto o utilizador e pedido para tentar novamente.

O nosso projeto abrange uma variedade de funções essenciais projetadas para obter testes confiáveis de prova de vida e comparação facial precisa. Essas funções incluem:

- checkLiveness: Esta função utiliza vários testes de prova de vida, como verificação de piscar de olhos, movimento do rosto, sorriso e rotações da cabeça para a esquerda e para a direita. Ao avaliar esses fatores, podemos verificar se o usuário está fisicamente presente e participando ativamente do processo de autenticação.

- checkEyeBlink: Esta função se concentra especificamente na detecção de piscar de olhos como um indicador de vivacidade. Ao analisar os movimentos dos olhos do usuário, podemos garantir que o processo de autenticação não esteja sendo manipulado por meio de imagens estáticas ou não humanas.

- checkFaceMovement: Com esta função, avaliamos o movimento do rosto do usuário para determinar se ele se alinha com o comportamento humano esperado. Ao detectar movimentos faciais, podemos verificar a presença de um usuário ao vivo, mitigando possíveis tentativas de falsificação.

- checkSmiling: Ao detectar e analisar o sorriso do usuário, esta função aprimora ainda mais o teste de prova de vida. Sorrisos genuínos podem ser indicadores de um usuário ao vivo, distinguindo-os de imagens estáticas ou representações não humanas.

- checkLookLeft e checkLookRight: essas funções avaliam a capacidade do usuário de girar a cabeça nas direções esquerda e direita. Essa avaliação garante que o usuário esteja ativamente envolvido no processo de autenticação e fortalece ainda mais o teste de prova de vida.

Além das funções de teste de prova de vida, nosso projeto também incorpora funcionalidades importantes para incorporação e comparação facial:

- imageToByteListFloat32: esta função converte uma imagem em uma lista de bytes de valores de ponto flutuante, uma etapa de pré-processamento necessária para incorporação de faces.
- imageToFaceData: Ao utilizar esta função, extraímos dados faciais relevantes da imagem de entrada, preparando-os para processamento posterior, como incorporação de rosto.
- cosineSim: Esta função calcula a semelhança de cosseno entre duas incorporações faciais, medindo a semelhança entre suas características faciais.
- findBestMatchingUserCosine: Esta função identifica o melhor usuário correspondente comparando as semelhanças de cosseno de suas incorporações de face armazenadas com a incorporação de face de entrada. Ele determina a correspondência mais próxima, permitindo uma comparação facial precisa e confiável.

Ao incorporar essas diversas funções em nosso projeto, desenvolvemos um sistema abrangente de reconhecimento facial capaz de realizar testes eficientes de prova de vida e comparação facial precisa. Essas funcionalidades contribuem coletivamente para aumentar a segurança e garantir a autenticidade dos usuários durante o processo de autenticação.

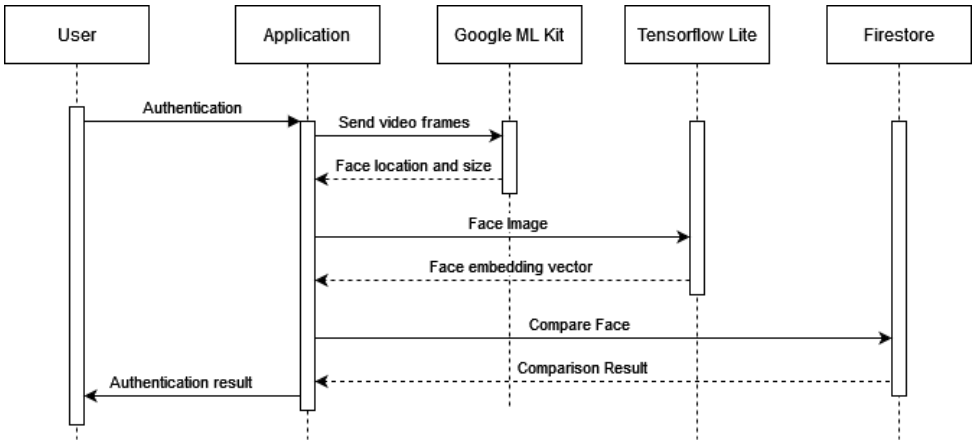


Figura 14 - Esquema do projeto

Fonte: Autores

Front-End

O front-end do nosso projeto, desenvolvido em Flutter, fornece uma interface de usuário perfeita com várias rotas para facilitar diferentes funcionalidades. Aqui está um resumo das rotas e principais componentes do aplicativo Flutter:

- FeedPage.dart: Esta rota corresponde à página de feed, exibindo conteúdo ou informações relevantes ao usuário após autenticação bem-sucedida.
- ListUsersPage.dart: Esta rota permite que os usuários visualizem uma lista de usuários registados no aplicativo.
- SignUpDetectionRoute.dart: essa rota permite que os usuários se inscrevam no aplicativo capturando seus dados faciais por meio do sistema de reconhecimento facial.
- SignUpPage.dart: esta rota exibe a página de inscrição onde os usuários podem fornecer seus detalhes e registrar-se no aplicativo.
- UserDetailPage.dart: esta rota fornece informações detalhadas sobre um usuário específico, incluindo seu ID, e-mail e dados faciais.

- LogInDetectionRoute.dart: Esta rota permite que os usuários façam login no aplicativo usando seus dados faciais por meio do sistema de reconhecimento facial.
- failedLogin.dart: Esta rota representa a página mostrada aos usuários em caso de falha na tentativa de login.
- esqueceuPassword.dart: Esta rota facilita o processo de recuperação de uma senha esquecida.
- home.dart: Esta rota serve como página inicial do aplicativo.
- login.dart: Esta rota exibe a página de login onde os usuários podem inserir suas credenciais para acessar o aplicativo.
- successLogin.dart: Esta rota representa a página exibida aos usuários após um login bem-sucedido.

A função principal do aplicativo inicializa o Firebase e configura o aplicativo Flutter. Ele garante que os serviços do Firebase sejam inicializados corretamente antes de executar o aplicativo.

A classe MainApp, como widget raiz da aplicação, configura o tema, as rotas e a tela inicial. Ele configura o MaterialApp com as configurações desejadas, incluindo o título do aplicativo, o modo de tema e o modo de depuração. Ele também define as rotas e seus construtores de widgets correspondentes para facilitar a navegação dentro do aplicativo.

Além disso, a função onGenerateRoute manipula o roteamento dinâmico especificando o comportamento quando uma determinada rota, como '/userdetail', é acessada. Nesse caso, ele cria o UserDetailPage e passa os argumentos necessários, como ID do usuário, email e dados faciais.

No geral, o front-end do nosso projeto em Flutter fornece uma interface amigável com rotas bem definidas, permitindo que os usuários naveguem pelo aplicativo e acessem diversas funcionalidades de forma integrada.

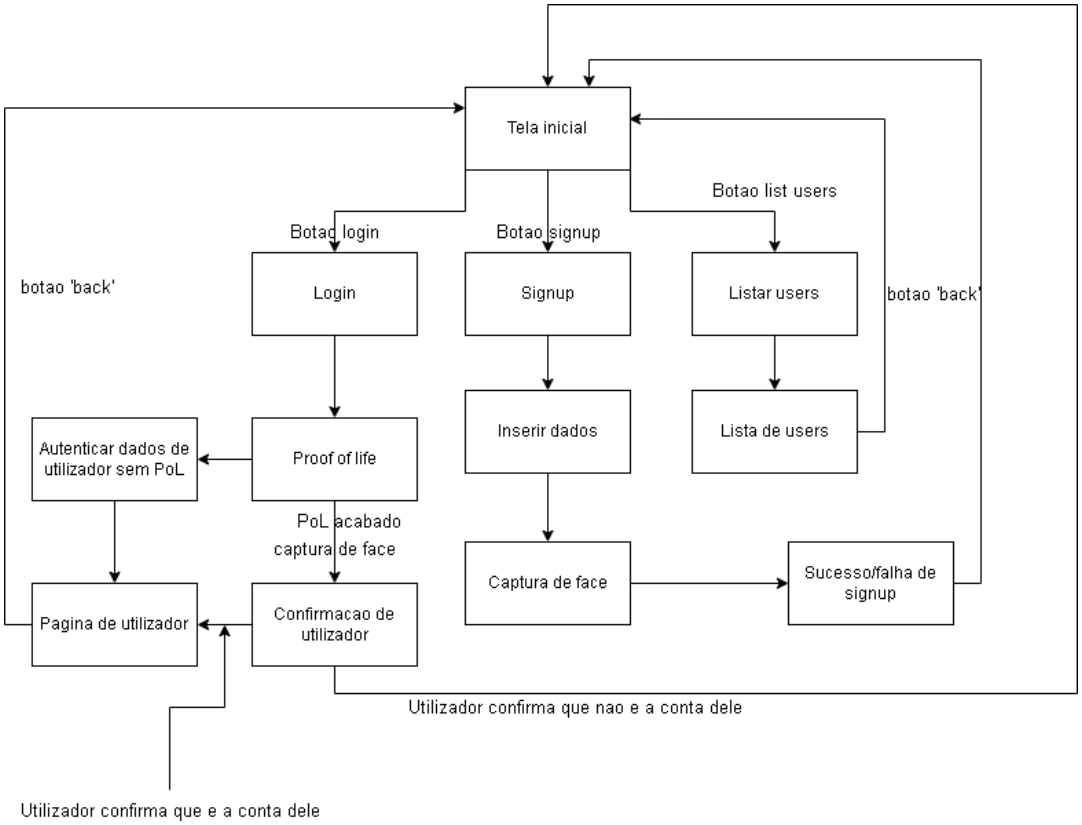


Figura 15- Diagrama de estados

Fonte: Autores

O Diagrama da figura 15, é um diagrama de estados que ilustra a aplicação FaceShield de uma maneira simples

5.2.2 Design

Nesta seção se mostram todas as telas que compões a aplicação e as suas funcionalidades.

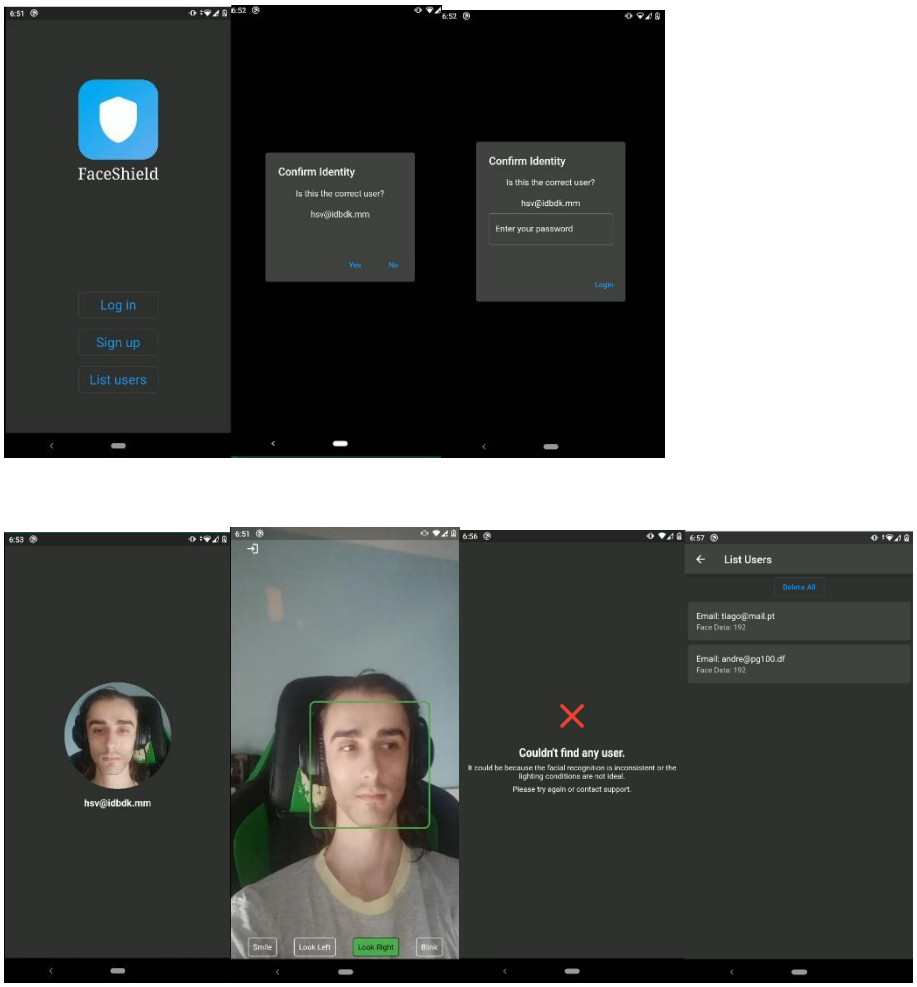


Figura 16 - Fotos do estado da aplicação

Fonte: Autores

Tela inicial: Esta Tela tem o símbolo da aplicação com o nome da mesma e por baixo apresenta 3 botões cada um com uma funcionalidade diferente.

Tela Sign up: Esta Tela apresenta 3 campos para preencher com as credenciais necessárias para criar conta; email, password e confirmação da password.

Tela Log in: Vai ser utilizado a câmera do dispositivo móvel, nesta tela o utilizador

vai ter a opção de se validar através do *proof of life* e posteriormente a captura do rosto ou também terá a opção de se validar através das credencias previamente registadas.

Tela List Users: Nesta tela vai ser disponibilizado os utilizadores tanto com as suas informações (e-mail e Facedata), e uma opção para apagar todos.

Tela de Confirmação de identidade: Mesmo após a captura para ter mais uma medida de segurança, é perguntado se essa corresponde a conta que quer utilizar.

Na figura 17, mostra-se o fluxograma, onde se descreve as etapas que um utilizador precisa seguir para realizar uma tarefa específica na aplicação.

6 Testes e Avaliação

6.1 Teste do módulo

6.1.1 Back-End

No back-end os testes utilizados foram uma maneira de manter a integridade e bom funcionamento da aplicação e da informação que esta utiliza.

Os testes realizados foram os seguintes:

Testes:

- Verificar que o login, sign up e list users aparecem na página.
- Clicar em cada um dos botões e eles forem para a página correta (ideal).

Sign Up:

- Verificar se o email, password, repeat password e create account aparecem na página.
- Clicar em create account e deve aparecer as mensagens de erro.
- Escrever um email válido.
- Escrever uma password e repeat password válidas.
- Fazer toggle da visibilidade da password.

List Users:

- Ver listagem dos utilizadores previamente criados.
- Ver a lista vazia quando não tem utilizadores.
- Aparecer delete all users.
- Clicar no utilizador e ir para a página detalhes.

User Detail:

- Deve aparecer o email e o respetivo email.
- Deve aparecer dois botões a dizer edit e delete.
- Quando clico delete, deve abrir um dialog.

Delete Dialog:

- Deve aparecer "Do you want to delete this user?"
- Aparecer botão delete.
- Aparecer botão cancel.
- Quando clicamos em cancel deve fechar o dialog.
- Quando clica em delete, deve eliminar o utilizador da lista.

Edit user:

- Deve aparecer Edit user, email e o email atual.
- Deve aparecer o botão edit e cancel.
- Quando clicamos em cancel deve fechar o dialog.
- Quando clicamos em edit, deve editar o email
- Fazer toggle da visibilidade da password.

6.1.2 Front-end

No Front-end a expiação do mesmo foi feita através do navegador. Para assegurar uma compatibilidade de hardware fora, utilizados vários tipos de dispositivos moveis diferentes, no qual no início houve problemas que rapidamente foram observados e corrigidos.

6.2 Teste de Integração

Os testes de integração foram feitos em simultâneo com o desenvolvimento parcial da aplicação. Visto que o ponto fulcral dos testes de integração toca muito no que é a interação sistemática entre o back-end e front-end, foram implementadas medidas de erro caso requisições do front-end não fossem enviadas de uma forma íntegra.

6.3 Teste de Validação

Os testes de validação são essencialmente formas de validar os vários comportamentos e reações da arquitetura no seu estado mais completo. Para os testes de validação foram então postas á prova várias situações para diferentes fotografias testadas.

A ambiência de uma fotografia pode ser um grande determinante de como esta se vai reproduzir na aplicação, isto é, a pouca ou muita luz, as diferentes distâncias entre o aparelho e a pessoa, acessórios faciais e os olhos abertos ou fechados durante a captura de rosto.

Foram então realizados os seguintes testes que tem em conta todos estes fatores referidos em cima:

- Capturar uma pessoa a piscar os olhos.
- Capturar uma pessoa a piscar os olhos, com óculos.
- Capturar uma pessoa a uma distância substancial.
- Capturar uma pessoa com pouca distancia da câmera.
- Capturar uma pessoa com óculos.
- Capturar uma pessoa a piscar os olhos a uma distância substancial.
- Capturar uma pessoa com olhos fechados.
- Capturar uma pessoa a uma distância substancial, com óculos.
- Capturar uma pessoa com pouca distancia da câmera, com óculos.

Analisou-se então o feedback que cada destes testes retornava em cada caso e chegou-se á conclusão de que existiam fatores que afetavam consideravelmente os resultados, estes foram:

Óculos:

Nos testes notou-se que os óculos vistos que a prova de vida é a mais afetada neste caso, eram um elemento completamente disruptivo para todo o cálculo da aplicação. Estes para além de afetarem o cálculo também quase exclusivamente ponha em causa a comprovação de vida.

Luminosidade:

A luz também é fulcral na precisão de validação, especialmente a falta dela visto que quase sempre retorna uma negação por nem sequer reconhecer um rosto.

Distancia:

A distância apresentou-se critica apenas em casos variação extrema, tudo dentro da distância de o braço a segurar o telemóvel entrou na conformidade da aplicação.

7 Conclusões

Em conclusão, nosso projeto de autenticação de reconhecimento facial utilizando Flutter, Google ML Kit, TensorFlow Lite e o modelo MobileFaceNet foi um esforço significativo para melhorar a segurança e a experiência do usuário. Ao longo do processo de desenvolvimento, focamo-nos em dois aspetos cruciais: o sistema de reconhecimento facial e a implementação de uma solução proof of life.

Primeiramente, empregamos o TensorFlow Lite, um framework de deep learning amplamente utilizado, e especificamente o modelo MobileFaceNet, conhecido por seu tamanho compacto e desempenho eficiente em dispositivos móveis. Inicialmente, utilizamos o algoritmo

de distâncias euclidianas para comparação de faces. No entanto, devido ao seu desempenho inferior, decidimos mudar para o algoritmo de similaridade de cosseno. Essa decisão melhorou significativamente a precisão e a confiabilidade do nosso sistema de reconhecimento facial, permitindo uma identificação mais precisa e eficiente dos indivíduos, apesar de o reconhecimento facial por vezes ainda falhar, algo que pretendemos melhorar.

Além disso, integramos o Google ML Kit, uma ferramenta poderosa para detecção facial, para implementar uma solução de prova de vida. Aproveitando os recursos do ML Kit, desenvolvemos um conjunto abrangente de verificações para garantir a autenticidade do usuário. Essas verificações incluíram o monitoramento de piscadas (*blinking)*, sorrisos e rotação da cabeça usando os ângulos de Euler fornecidos pelo módulo de detecção facial do Google ML Kit.

É importante observar que nossa solução de prova de vida opera em um fluxo de imagem contínuo em vez de imagens estáticas. Esse monitoramento em tempo real garante que o usuário esteja fisicamente presente durante o processo de autenticação, mitigando efetivamente possíveis tentativas de falsificação usando imagens estáticas.

Ao combinar o sistema de reconhecimento facial baseado no TensorFlow Lite e o modelo MobileFaceNet com a solução de prova de vida usando o Google ML Kit, desenvolvemos um sistema de autenticação robusto e seguro. Este sistema não apenas verifica com precisão a identidade dos usuários, mas também garante que eles estejam fisicamente presentes e participando ativamente do processo de autenticação.

Avançando, estamos comprometidos em refinar e otimizar ainda mais nosso projeto de autenticação de reconhecimento facial, explorando avanços em algoritmos de aprendizado de máquina e incorporando recursos adicionais para aumentar a segurança e a usabilidade.

Por último foi também desenvolvido um pequeno manual de instruções para utilizadores que vai em conjunto com este relatório.

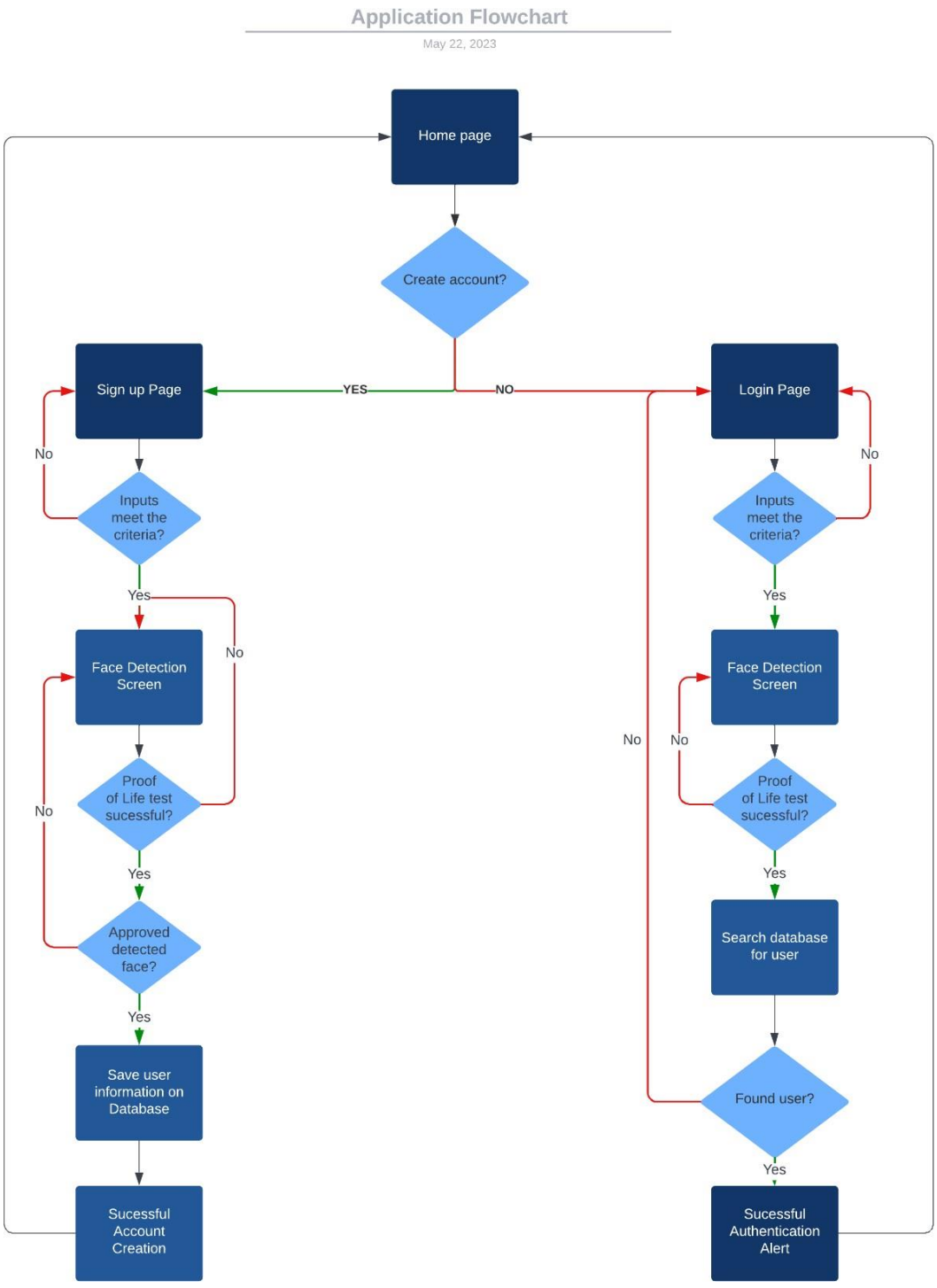


Figura 17 – Fluxograma do projeto

Fonte: Autores

Referências

[Alpaydin, 2020] Alpaydin, E. (2020). Introduction to Machine Learning. Adaptive Computation and Machine Learning series. MIT Press.

[Schroff et al., 2015a] Schroff, F., Kalenichenko, D., and Philbin, J. (2015a). Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2015.

[Heaton, 2017] Heaton, J. (2017). Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. Genetic Programming and Evolvable Machines, 19.

[Abadi et al., 2015] Abadi, M., et al (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Turing and Copeland, 2004] Turing, A. and Copeland, B. (2004). The Essential Turing. Clarendon Press.

[Gates, 2011] Gates, K. (2011). Our Biometric Future: Facial Recognition Technology and the Culture of Surveillance. Critical Cultural Communication. NYU Press.

[Frantz, 2003] Frantz, R. (2003). Herbert simon. artificial intelligence as a framework for understanding intuition. Journal of Economic Psychology, 24(2):265 – 277. The Economic Psychology of Herbert A. Simon.

[Asawa et al., 2020] Asawa, C., Zakka, K., Pan, B., Zhou, L., Zhang, Y., and Liu, X. (2020). Stanford cs class cs231n: Convolutional neural networks for visual recognition.Acesso em: 20 set. 2022.

[Amazon Web Services, 2020b] Amazon Web Services, I. (2020b). Instâncias amazon south america. Acesso em: 5 dez. 2022.

ANEXOS

Manual de instalação/utilização - Sistema de Autenticação (FaceShield)

Nota: Por questões de segurança, os telemóveis geralmente não permitem a instalação de ficheiros APK, através de fontes desconhecidas, por motivos de segurança. Visto isto, o utilizador deverá seguir estes passos:

1. Habilitar instalação de aplicações de fontes desconhecidas

a. No seu telemóvel, navegue até as definições.
b. Dentro das definições, abra a seção de privacidade e segurança.
c. Permita a instalação de fontes desconhecidas, ativando a opção que fala sobre isso.
Nota: Diferentes modelos de telemóvel têm diferentes interfaces, por isso estes passos podem mudar ligeiramente consoante o modelo.

2. Fazer download do ficheiro APK (Do computador para o telemóvel)

a. Conecte o seu telemóvel ao computador através de um cabo USB.
b. Dirige-se à localização do ficheiro APK no seu computador.
c. Copie o ficheiro para o seu telemóvel na localização que pretender.
2. Fazer download do ficheiro APK (Apenas do telemóvel)
a. Faça download do ficheiro APK que recebeu no seu telemóvel
b. Guarde o ficheiro na localização que pretender.

3. Instalar o ficheiro APK no telemóvel

a. Vá à localização do ficheiro APK no seu telemóvel.
b. Clique nele.
c. Vai aparecer uma solicitação a perguntar se quer efetuar a instalação.
d. Aceite essa opção.
e. Após o término da instalação, a aplicação está pronta para ser executada.
Nota: Tenha cuidado com as fontes que faz download, pois podem conter software malicioso.

4. Conceber as permissões necessárias

a. Abra a aplicação já instalada no seu telemóvel.
b. Conceda permissão para a aplicação usar a sua câmara, para além do seu armazenamento.
c. Clique no botão “Permitir” para conceder esta permissão.

Se encontrar problemas ao utilizar a aplicação, tente:

- Garantir que a iluminação é adequada e que não há obstruções no rosto.
- Remover óculos, chapéus ou outros objetos que possam obstruir o rosto.
- Verificar se a câmara está a funcionar corretamente.
- Atualizar a aplicação e o sistema operativo.
- Reiniciar o dispositivo.

Repositório do Projeto.
https://github.com/AndrePG98/face_shield