INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Tradução automática de língua gestual a partir de vídeo: hand tracking

JORGE MIGUEL DA SILVA VIEIRA
Outubro de 2020

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Tradução automática de língua gestual a partir de vídeo: hand tracking

Jorge Miguel Silva Vieira

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas Gráficos e Multimédia

Orientador: Professor Nuno Escudeiro

Porto, outubro 2020

Dedicatória

"Este trabalho é dedicado a toda a minha família e amigos
que me ajudaram durante os momentos mais difíceis"

Resumo

Atualmente existem bastante dificuldade na comunicação entre a comunidade surda com a restante sociedade. Se esta comunidade tivesse, através de uma aplicação, a possibilidade de comunicação sem que seja necessário aprender língua gestual, um leque de novas oportunidades iria surgir, tanto a nível do mundo profissional, como a nível social.
Apesar de já existirem formas de comunicação entre a comunidade surda e ouvinte, estas são, no entanto, bastante complicadas e dispendiosas. É necessário o uso de luvas ou outro tipo de ferramenta externa o que dificulta o acesso e a utilização de tal tecnologia.

Esta dissertação tem como objetivo melhorar a inclusão social e a comunicação das pessoas surdas com recurso à tradução de língua gestual para língua oral através de um vídeo, com o mínimo recurso a ferramentas externas. É necessário que o acesso à aplicação seja fácil e acessível em vários dispositivos, para isso a aplicação foi hospedada num website, para que possa ser acedida tanto no computador, como num dispositivo móvel, de forma a prever os movimentos realizados pelo utilizador. No entanto, é necessário primeiramente captar os movimentos e treiná-los com ajuda de modelos de classificação.
Finalmente, foram realizados testes de forma a concluir se certas características favoreciam os modelos de classificação utilizados e quais destes contêm uma com maior precisão no acerto das palavras ou frases.
Os resultados obtidos foram satisfatórios, o uso da previsão da configuração da mão provou ser eficaz e aumentar a precisão.
Este projeto contribui para identificar que o uso de características como a configuração de mão durante a gravação, podem ser significativos, aumentando a precisão, bem como quais os modelos de classificação mais indicados para a previsão de palavras/gestos, ampliando assim a possibilidade e a probabilidade da existência de uma aplicação de fácil acesso sem recurso a ferramentas externas para a integração da comunidade surda.

Abstract

Currently, there is a lot of difficulty in communication between the deaf community and the rest of society. If this community had, through an application, the possibility of communication without the need to learn sign language, a range of new opportunities would arise, both in the professional world and at the social level.
Although there are already forms of communication between the deaf and the hearing community, these are, however, quite complicated and expensive. It is necessary to use gloves or other types of external tools, which makes it difficult to access and use this technology.

This dissertation aims to improve the social inclusion and communication of deaf people with the use of sign language to oral language through a video, with minimal use of external tools. The access to the application must be easy and accessible on several devices, for this the application was hosted on a website, which can be accessed both on the computer and on a mobile device, to predict the movements performed by the user. However, it is necessary to first capture the movements and train them with the help of classification models.
Finally, tests were carried out to conclude whether certain characteristics favored the classification models used and which of them contains the highest precision.
The results obtained were satisfactory, the use of the prediction of the hand configuration proved to be effective and increase the precision.
This project helps to identify that the use of characteristics such as the hand configuration during recording, can be significant, increasing the accuracy, as well as which classification models are most suitable for predicting words/gestures, thus expanding the possibility and the probability of the existence of an easily accessible application without recourse to external tools for the integration of the deaf community.

Agradecimentos

Ao professor Nuno Escudeiro, pela disponibilidade, confiança e zelosa orientação durante o trabalho desenvolvido.
Ao Nuno Pereira, pelo apoio, colaboração e ajuda durante a elaboração do projeto.
Aos meus amigos dos quais me orgulho e com os quais sei que poderei sempre contar.
Aos meus pais, pela paciência, confiança e por todo o conforto que me ofereceram durante todo o percurso, motivando-me cada dia para trabalhar mais e melhor.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Bounding Box	Caixa que delimita o objeto, bastante usado para se referir os limites do objeto em termos de colisão
Frames	Uma de muitas imagens paradas que compõe uma imagem em movimento
CNN	Convolutional Neural Network
SVM	Support Vector Machine
LSTM	Long Short-Term Memory
FCM	Fully Connected Model

1 Introdução

Este documento foi realizado no âmbito da unidade curricular de Tese/Dissertação/Estágio e tenciona encontrar uma possível resposta ao problema da tradução de língua gestual para língua oral a partir de um vídeo e, posteriormente, em tempo-real.

1.1 Contexto e Problema

A comunicação existente atualmente entre surdos e ouvintes é complicada e pouco eficaz, visto que, é necessária a aprendizagem da língua gestual. Isto resulta com que a integração dos surdos na sociedade atual e no mundo do trabalho é, de uma forma geral, complicada. É necessária uma forma de tradução automática de língua gestual para uma língua oral (como o Português ou Inglês), através de vídeos ou imagens, sem a utilização a ferramentas externas (ou mínimo uso possível destas, como o exemplo de luvas com LEDs relativamente acessíveis monetariamente a todos os surdos). Seria também uma mais valia, a possibilidade da utilização desta aplicação num telemóvel, dado que seria bastante mais acessível e relevante, facilitando a inclusão dos surdos no mundo do trabalho e na educação levando a um aumento do nível de oportunidades face à atualidade.

1.2 Objetivos

O objetivo deste projeto é possibilitar uma melhor comunicação entre surdos e ouvintes minimizando a necessidade de recorrer a materiais auxiliares, tais como luvas especializadas ou outro tipo similar, para isto é necessária a elaboração de uma aplicação capaz de capturar o movimento das mãos do utilizador, através de um vídeo ou em tempo real, guardar numa linha de tempo, para que posteriormente seja possível compreender o movimento e traduzir, com a ajuda de um dicionário, para língua comum, com uma taxa de erro aceitável, bem como a possibilidade de troca de algoritmos de classificação.

1.3 Fase de Planeamento

Numa primeira fase, foram levantadas todas as questões relacionadas com o tema desta dissertação, foi elaborada uma pesquisa intensiva de modo a familiarizar e aumentar o conhecimento relacionado com o assunto em questão. É feita uma descrição dos principais algoritmos, bem como uma comparação de 'tracking' e deteção. Por fim, são apresentadas todas as ferramentas usadas na realização do projeto.
De seguida, foi necessário estudar o trabalho previamente elaborado pela aluna de ERASMUS, Despina, que serviu de ponto de partida para o restante projeto. O programa foi colocado online e foram realizados vários testes de forma a melhor perceber os resultados obtidos, assim como uma análise do estudo realizado pela aluna.
Posteriormente, foram melhorados os algoritmos e foram introduzidas novas características a ter em conta como a posição e o gesto das mãos a cada frame.

Por fim, foi gravado um novo conjunto de dados com diversas palavras e foi feita uma análise comparativa de forma a concluir se a implementação dos gestos e da posição das mãos foram ou não favoráveis aquando o uso dos classificadores no treino dos modelos.

1.4 Estrutura do documento

O presente documento encontra-se dividido em 6 capítulos principais: Introdução, Estado da Arte, Análise de Valor, Desenho e Implementação, Avaliação e Conclusão.

O primeiro capítulo tem como objetivo introduzir o problema para qual este projeto se refere, quais os objetivos, um planeamento com todas as fases do projeto e por fim um resumo da estrutura do relatório.

No segundo capítulo é realizada uma pesquisa referente ao tema do projeto, de modo a que seja possível uma melhor compreensão e introdução aos conceitos relacionados com o tema do projeto. É distinguida a diferença entre rastreamento e deteção de um objeto, quais os algoritmos mais utilizados, como é captado os frames dos vídeos para que posteriormente sejam trabalhados, assim como as principais tecnologias utilizadas.

No terceiro capítulo é retratada a análise de valor, onde se refere essencialmente ao valor que este projeto vai trazer para os seus clientes, assim como o valor geral da utilidade do produto, baseado na perceção do que é recebido e do que é oferecido. De seguida, foi utilizado o método AHP (Analytic Hierarchy Process), de modo a descobrir qual a principal ideia em que o projeto se deve focar, e qual a sua prioridade em relação aos restantes. Por fim, foi elaborado um modelo CANVAS para ser possível ter uma ideia geral sobre todo o projeto, desde qual é o seu público alvo até às atividades ou parceiros chave para o desenvolvimento deste projeto.

No quarto capítulo é apresentado o desenho e implementação da solução. Começa por apresentar os requisitos funcionais e não funcionais. Nos subcapítulos seguintes foram desenhados vários diagramas de forma a compreender melhor a solução e os processos para a sua conclusão. No final do capítulo, tem toda a implementação que foi feita, e um storyboard de toda a aplicação.

No quinto capítulo são apresentados os métodos como a aplicação vai ser avaliada de forma a verificar se cumpre os objetivos pré-estabelecidos. São também explicados os passos que foram seguidos para fazer essa avaliação, bem como algumas descobertas finais depois dos resultados obtidos.

No sexto e último capítulo é apresentada a conclusão do projeto, com uma representação dos resultados obtidos, assim como, quais as melhorias ou objetivos para uma melhor aplicação no futuro.

2 Estado da Arte

Neste capítulo vão ser relatadas todas as ferramentas necessárias e utilizadas para a execução do projeto. Foram investigados vários trabalhos científicos bem como projetos desenvolvidos em prol da comunidade dos surdos de modo a melhorar substancialmente o seu direito à integração social, assim como uma mais fácil comunicação em estados de emergência.
Irão também ser descritos algumas das técnicas que foram abordadas e estudadas para melhor compreensão do problema, podendo assim optar por seguir aquela que tenha mais valia para o projeto em questão.

2.1 Tecnologias utilizadas

Neste capítulo irão ser descritas todas as tecnologias que foram utilizadas na realização deste trabalho. O OpenCV foi uma tecnologia imprescindível, visto que contribuiu para a captura dos frames e uma posterior análise.

2.1.1 OpenCV

OpenCV (Open Source Computer Vision Library) é uma biblioteca de software "open source" de foco computacional e 'machine learning'. Foi construído com base em promover uma infraestrutura comum e para acelerar a perceção de objetos, ou seja, 'machine perception', utilizando uma máquina. Utilizando uma linguagem mais simples, é uma biblioteca usada para processamento de imagem. É utilizada para fazer todas as operações relativas às imagens.
É uma biblioteca que conta com mais de 2500 algoritmos otimizados. Muitos destes algoritmos são utilizados para detetar e reconhecer objetos, caras, classificar ações em vídeos, captar movimentos da câmara, captar o movimento de certos objetos, encontrar e comparar determinadas imagens parecidas com as que se encontram na base de dados, entre outras aplicações.
Conta com o suporte para quatro línguagens de programação (C++, Python, Java e MATLAB) e suporta quatro sistemas operativos (Windows, Linux, Android e MAC OS). OpenCV é focado essencialmente em visão computacional em tempo real [1].

2.1.2 Git

O Git é uma ferramenta open-source de controlo de versões que permite que um grupo de pessoas consiga trabalhar, de forma segura, com os mesmos ficheiros. Por outras palavras, o Git permite a existência de versões nos projetos nos quais é utilizado, criando um histórico das mudanças efetuadas, para além de possibilitar a reversão para versões antigas. De modo a que o Git possa operar, os projetos são colocados num servidor central naquilo a que se chama repositório, sendo este o local propriamente dito onde o trabalho será guardado. Existindo, por tanto, um repositório distribuído de forma central, permite que qualquer elemento da equipa consiga efetuar mudanças e enviá-las para o repositório para que todo o grupo possa fazer pull (ato de transferência das mudanças mais recentes do repositório para o projeto local). Por outro lado, o comando push envia para o repositório central todas as

mudanças efetuadas, ao nível local, que foram mapeadas durante o processo de commit. O processo de commit consiste, de forma sucinta, na indicação dos ficheiros, no projeto local, que foram alterados e que se pretende que sejam enviados para o repositório central. Salienta-se também que, apesar de o Git permitir efetuar um conjunto bastante alargado de operações, a sua utilização limitou-se às funcionalidades mais básicas por não ter havido necessidade de utilização das outras capacidades da ferramenta.
Como a utilização do projeto foi unicamente do aluno, não houve a imposição de se criar um novo ramo, no entanto, o projeto foi todo ele partilhado, para que exista a possibilidade de outras pessoas cederem uma nova visão ou opinião para melhoria do projeto. [2]

2.1.3 Bitbucket

Bitbucket é o repositório Git de controlo de soluções desenhadas por equipas profissionais. Oferece um local central para gerir os repositórios Git, colabora com o código fonte e guiar através de todo o fluxo de desenvolvimento. [3]
Os projetos são carregados para o repositório do Bitbucket utilizando o Git, conseguindo alterar para/consultar versões mais antigas.
Foi também usufruído a parte de criação de *issues*, ou seja, tarefas ou bugs que ficam atribuídos aos colaboradores do projeto, com a data de início e término, desta forma existe um maior controlo sobre todo o projeto e do tempo usada para cada um destes *issues*.

2.1.4 Open Pose

OpenPose representa o primeiro sistema de deteção de múltiplas pessoas em tempo real que deteta várias partes do corpo humano, como o corpo, as mãos, a cara, os pés entre outros pontos em diferentes imagens [4].
Este software conta com o facto de não só detetar, mas também estimar pontos chaves de partes do corpo dos indivíduos. Existem alguns desafios que o OpenPose teve de resolver, como o caso de existir um número desconhecido de pessoas na imagem ou vídeo que podem aparecer em qualquer posição ou escala, assim como o facto que a interação entre pessoas pode induzir interferências devido ao contacto, como oclusões ou articulação de membros. Por último, também existe o facto de manter a performance independentemente do número de pessoas que possam aparecer na imagem, o que pode trazer um número bastante significativo de pontos chave e pode ter um impacto significante na performance a tempo real [5].
OpenPose tenta resolver todos estes problemas contando sempre com uma fácil instalação e utilização [6].

2.1.5 Heroku

“Heroku é uma plataforma nuvem que deixa todas as empresas construírem, entregarem, monitorizarem e escalarem as aplicações.” [7] [8]
Heroku é um serviço de nuvem bastante adotado por bastantes aplicações focadas em clientes, devido à facilidade de desenvolvimento e entrega de aplicações. Conta com um modelo de serviço grátis para projetos pequenos, no entanto também contém pacotes de serviços para casos de negócios de maior complexidade [8].

A plataforma de serviço de nuvem do Heroku é baseada num sistema de contentores monitorizados com serviços de informação integrados e um ecossistema poderoso de distribuição e execução de aplicações modernas. [9]

2.1.6 SVM

SVM (Support Vector Machine), ou máquina de vetor suporte, é um classificador usado para tarefas de regressão e classificação.

O objetivo do algoritmo usado pelo SVM é encontrar um hiperplano numa dimensão N, em que N é o número de características, que classifica distintamente os pontos de informação. Ou seja, encontrar uma linha de separação, denominada de hiperplano entre duas classes na qual é necessário maximizar a distância entre os pontos mais próximos em relação a cada uma das classes como é possível verificar na Figura 1 e na Figura 2.

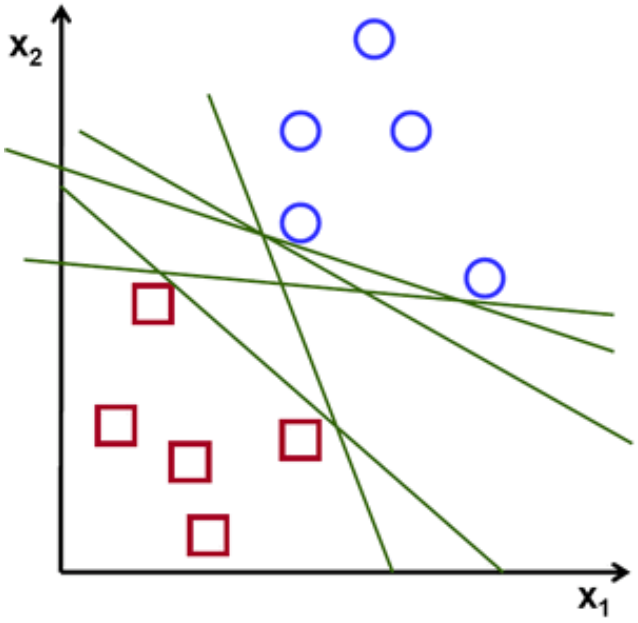

Figura 1 – Hiperplano possível

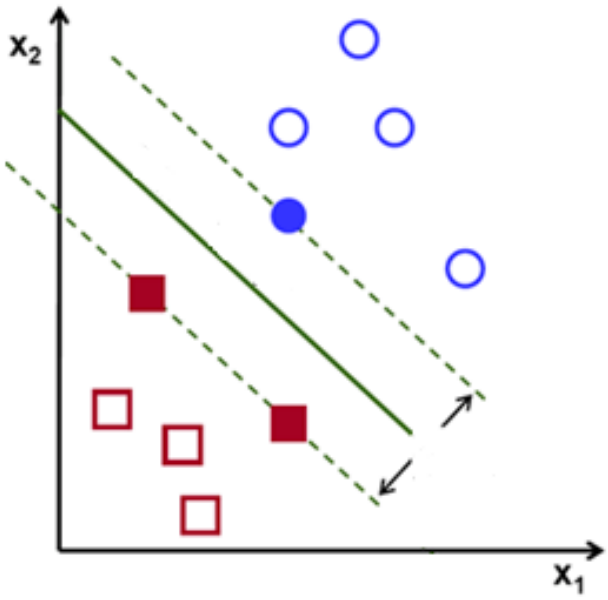

Figura 2 - Hiperplano Ideal

De maneira a separar as duas classes existem bastantes hiperplano possíveis que poderiam ser escolhidos, contudo o hiperplano a ser escolhido é o que tem a maior margem (exemplo indicado pelas duas setas que apontam para as linhas a tracejado na Figura 2) [10] [11].

2.1.7 LSTM

LSTM (Long Short-Term Memory) é uma *network* do tipo de redes neuronais recorrentes capazes de aprender dependências de ordem em problemas de previsão sequencial.
"Redes recorrentes têm um estado interno que representa a informação do contexto. Essas redes mantêm informação acerca dos registos passados por uma quantidade de tempo que não é fixo à priori, mas depende dos seus pesos e da informação inserida." [12]
As redes neuronais recorrentes têm de usar o contexto quando fazem as suas previsões, no entanto, o contexto requerido também necessita de ser aprendido.
LSTM ao contrário de outras arquiteturas utiliza conexões com *feedback,* processando assim não só pontos de data singulares (como uma imagem), mas também sequencias de informação (como um vídeo) [13] [14].

2.2 Processamento de imagem através de tracking

É necessário que, através de um vídeo, seja possível captar e processar determinados objetos. Qual foi a sua posição inicial e final numa determinada linha de tempo, para que mais tarde seja processado e traduzido.

2.2.1 Rastreamento de um objeto

Rastreamento (ou *Tracking*), significa seguir determinado objeto que está em movimento. Em termos mais simples, tracking é o "problema de estimar a trajetória de um objeto numa imagem plana enquanto este se move pela cena. Alternativamente, um 'tracker' determina

etiquetas consistentes aos objetos que estão a ser captados em diferentes 'frames' do vídeo. Dependendo também do algoritmo, do domínio do 'tracker' ou mesmo do método, este pode também devolver outra informação importante sobre um objeto, tal como a sua orientação, tamanho e área." [15] [16]

Uma vez que que os objetos foram detetados, a próxima tarefa será captar os objetos de uma 'frame' para outra. O registo dos objetos pode ser complicado devido à forma do objeto, movimento do objeto, o facto do objeto não ser rígido ou mesmo o facto do objeto ter oclusões parciais, entre outros problemas [15] [16]. Alguns destes inconvenientes podem ser resolvidos com uma simples constante, como o exemplo do movimento do objeto ser suave, sem alterações abruptas, conhecimento prévio do número e tamanho dos objetos, a aparência dos objetos e forma. Existem já várias ferramentas disponíveis para 'tracking' de um objeto.

2.2.2 Problemas com o tracking

Aquando da realização do tracking de um objeto, existem alguns problemas que podem surgir. Nem sempre os objetos são nítidos e existem vários detalhes a ter em conta. Nos seguintes subcapítulos irão ser relatados alguns dos exemplos de ruído.

2.2.2.1 Oclusão

Oclusão de objetos em vídeos é um obstáculo bastante comum e conhecido no campo do 'tracking' de objetos. É comum quando existe um vídeo em que objetos se coloquem à frente de outros, dificultando assim a sua captação.

É necessário continuar a seguir o objeto que foi pré-definido mesmo que outros se posicionem à frente e obstruam a sua visão, podendo ser relevante a detenção de novos objetos nos próximos frames. Contudo, é imprescindível o facto de conseguir captar e guardar toda a trajetória de todos objetos que forem relevantes [16] [17].

2.2.2.2 Variações nos pontos de vista

Devido ao facto de ser uma análise no tracking de um objeto num vídeo, podem existir várias direções que a câmara pode tomar, fazendo com que exista diferentes pontos de vista, levando a que existam alguns frames em que o objeto esteja oculto, reaparecendo num frame posterior.

É necessário conseguir detetar o objeto em questão, em todos os momentos do vídeo, para que a devida análise possa ser realizada, independentemente se este está sempre presente ou não [17].

2.2.2.3 Câmara não estacionária

O facto da câmara se encontrar em movimento pode provocar dificuldades na deteção do objeto por parte do tracker. Como os tracker se baseiam em características dos objetos para os identificar, diferentes cenários podem complicar o processamento, assim como se o objeto se encontrar em movimento, levando à falha ao tentar localizar os objetos [17].

2.2.2.4 Iluminação e alteração na escala

Alterações da iluminação enquanto é feito o tracking tem um grande impacto. A luz pode alterar significativamente a aparência do objeto.

O mesmo acontece com a escala, originado por exemplo, aquando do uso do zoom da câmara causando com que o objeto se pareça diferente ao original levando a que não seja identificado em frames futuros [17].

2.2.3 Rastreamento de um objeto vs. deteção de um objeto

Existe uma grande diferença entre estes dois grandes tópicos, rastreamento e deteção de um objeto.

Enquanto que no processo de rastreamento de um objeto, este já é conhecido previamente, não sendo preciso detetar novos objetos ao longo do vídeo, apenas seguir todos os movimentos desse(s) objeto(s) pré-definido(s) com a ajuda da caixa delimitadora (bounding box).

Por outro lado, a detenção de um objeto passa por mais que isso. É necessário conseguir captar um determinado objeto a qualquer momento do vídeo, e no caso de existir mais que um elemento deste objeto ou mesmo mais que um objeto, é necessário que estes sejam captados e delimitados tal como aconteceria no método de rastreamento [17].

2.3 Técnicas para processamento de vídeo e tracking de objetos

Nos seguintes subcapítulos irão ser relatados algumas das técnicas existentes para o 'tracking' de um objeto. Nenhuma das técnicas é perfeita, no entanto todas se sobressaem em determinadas características.

2.3.1 Kalman Filters

A ideia geral deste algoritmo é basicamente basear-se em antigas deteções e previsões para poder concluir e tentar supor de forma mais assertiva possível onde é que o modelo se pode encontrar no frame seguinte, no entanto pode existir a probabilidade de ocorrer erros nesse processo.

Utilizando um exemplo similar ao [18], assumindo que é contido um bom detetor de objetos que consiga detetar pessoas. No entanto, não é 100% assertivo, falhando ocasionalmente, 1 em cada 8 frames. Para conseguir captar efetivamente a pessoa, é usado a "velocidade constante do modelo", neste caso, a pessoa.

Numa situação perfeita seria possível captar e prever um objeto no próximo frame, no entanto como foi referido no capítulo anterior, pode existir a presença de ruído.

Relativamente ao ruído existem 2 tipos de factos que vão influenciar:

- O facto que é assumido que a velocidade vai ser sempre constante, o que na realidade é pouco provável que aconteça;
- O resultado do detetor é baseado em previsões, portanto também contém ruído associado, não sendo totalmente confiável.

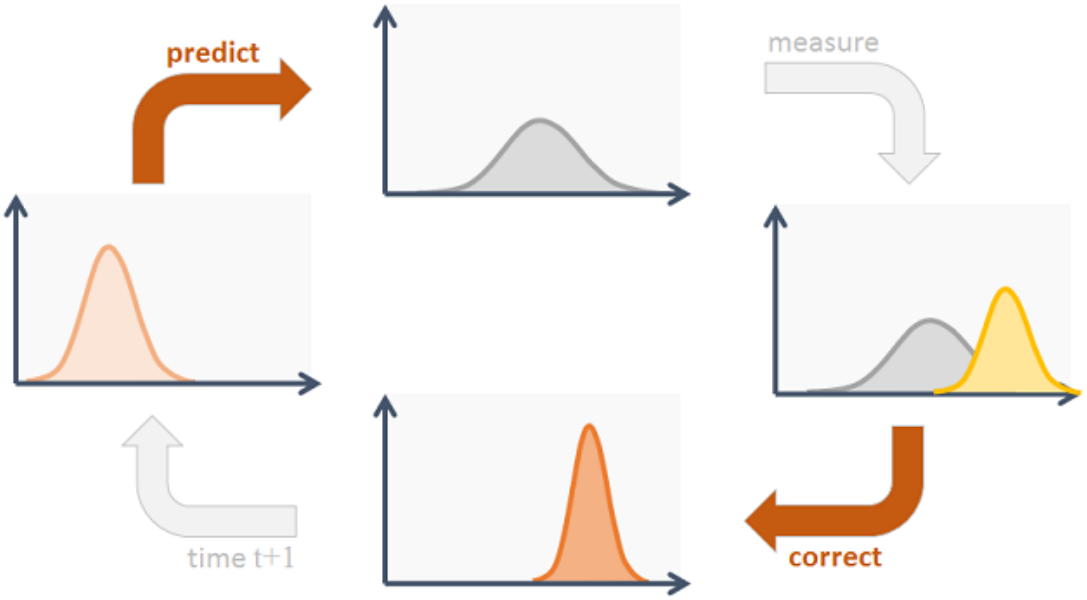

Figura 3 – Kalman Filter [19]

Como é possível verificar na Figura 3, este é um algoritmo recursivo que à medida que prevê e vai medindo, altera e corrige para as próximas iterações, ficando cada vez mais preciso e robusto.

2.3.2 SORT

SORT é um dos vários algoritmos de tracking que se foca basicamente num mecanismo de deteção de objetos adjacentes. É possível introduzir este algoritmo noutro tipo de algoritmo de deteção de objetos.

O algoritmo é capaz de captar vários objetos em tempo real, associando os objetos em cada frame com aqueles que foram detetados os frames anteriores com heurísticas simples. [20]

Este algoritmo focasse em associações frame a frame fazendo com que o número de vezes que o modelo é perdido seja relativamente baixo. [21] [22]

2.3.3 Deep SORT

Deep SORT é uma extensão do algoritmo do subcapítulo anterior, SORT, no entanto, adiciona uma nova característica a este algoritmo que permite perceber melhor a 'bounding box'. O tracking não é só baseado na velocidade e na distância, mas também no que o objeto se parece. O Deep SORT permite adicionar esta característica através da computação para cada 'bounding box', e utilizar semelhanças como um fator para a lógica do tracking [17] [23] [24].

Este algoritmo contém alguns inconvenientes, como o facto de se a 'bounding box' for demasiado grande apenas para capturar uma parte do objeto e contiver bastante fundo, que vai ser interpretado como ruído, pode provocar com que o algoritmo seja menos eficiente e eficaz. Existe também a possibilidade dos objetos que contenham características muito idênticas como a mesma cor, possam ser trocados identificados incorretamente [23].

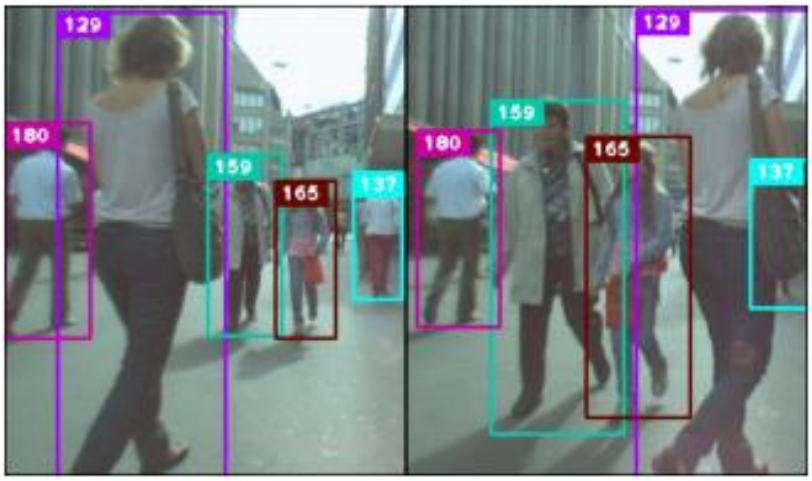

Figura 4 - Exemplo do rastreio do algoritmo Deep SORT [23]

Na Figura 4 é possível verificar o algoritmo em funcionamento. Este consegue detetar e seguir o objeto ao longo de todos os frames. Cada objeto contém um número de identificação e uma 'bounding box' associada.

2.3.4 Algoritmos baseados em Convolutional Neural Network (CNN)

Nos próximos capítulos vão ser referidos alguns algoritmos baseados em CNN, devido ao facto que este algoritmo é bastante eficaz, através de alguns detalhes da imagem/vídeo que está a analisar, é capaz de recolher detalhes importante sem necessidade de intervenção alguma. É também computacionalmente eficiente, tornando-se assim possível correr este algoritmo em diversos dispositivos.

2.3.4.1 O que é CNN?

"Convolutional Neural Network (CNN) é um dos Deep Neural Network mais populares. Tem uma excelente performance em problemas de 'machine learning'" [25] e é englobado na categoria do 'Neural Network', contudo bastante efetivo no campo de reconhecimento de imagem e classificação. CNN ajuda também no reconhecimento facial e de objetos [26].

2.3.4.2 Generic Object Tracking Using Regression Network (GOTURN)

Este método usa vários frames de diferentes dimensões referentes a diversos vídeos. No primeiro frame de todos, a localização do objeto a ser estudado já é conhecido, sendo que esse objeto é recortado pelo dobro do tamanho que a sua 'bounding box', mantendo sempre o objeto centrado.

Através dos limites predefinidos pela bounding box, o algoritmo tenta prever a localização do objeto no segundo frame. Essa bounding box utilizada no primeiro frame é também usada para recortar o segundo frame. Devido ao facto que o objeto se pode ter movido, o objeto pode não se encontrar centrado.

Para isto é usado o algoritmo CNN, de forma a conseguir prever a localização do objeto no segundo frame.

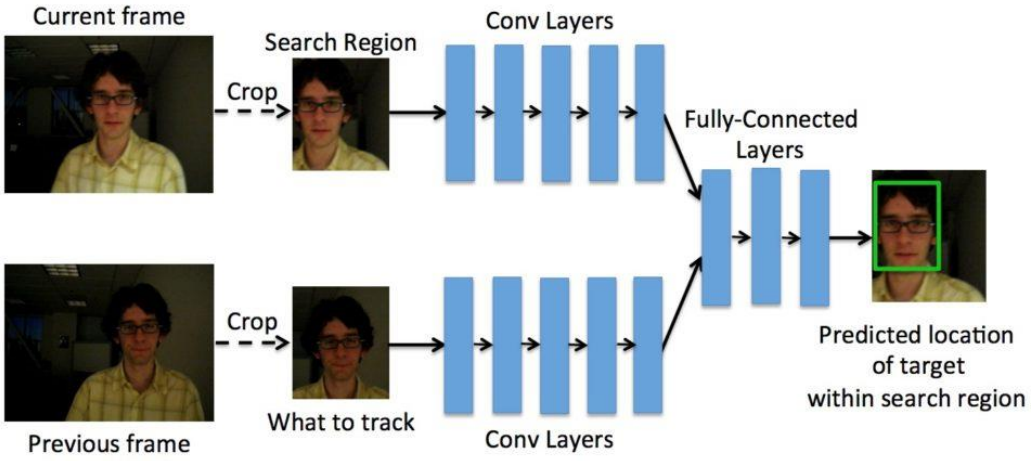

Figura 5 – Arquitetura do GOTURN [18]

Como é possível verificar na Figura 5, GOTURN capta 2 frames recortados. O frame anterior encontra-se centrado no objeto que se pretende, sendo que o objetivo seja encontrar a 'bounding box' para o frame atual.

Ambos os frames passam por uma camada convolucional, sendo que o resultado desta é concatenado tornando-se único vetor com o tamanho de 4096 (2^{12} = 4096). Este vetor vai ser usado para 3 camadas totalmente conectadas. Esta última camada é conectada ao resultado contendo 4 pontos que correspondem às 4 laterais que formal o retângulo da 'bounding box' [18].

2.3.5 Multi-Domain Network (MDNet)

O objetivo desta arquitetura é basicamente aumentar a velocidade da aprendizagem de forma a obter resultados em tempo real. A estratégia utilizada é dividir a rede de informação em duas partes.

A primeira fase foca-se essencialmente como uma função genérica de extração que treina com vários fundos e diferentes formas para que seja mais fácil distinguir os objetos independentemente do que os rodeia.

A segunda parte é treinada num conjunto específico de treinamento e aprende a identificar os objetos que se encontram nos frames dos vídeos.

Este algoritmo torna possível alterar o peso apenas das últimas camadas CNN durante a sua aprendizagem, reduzindo o tempo de computação significativamente [20].

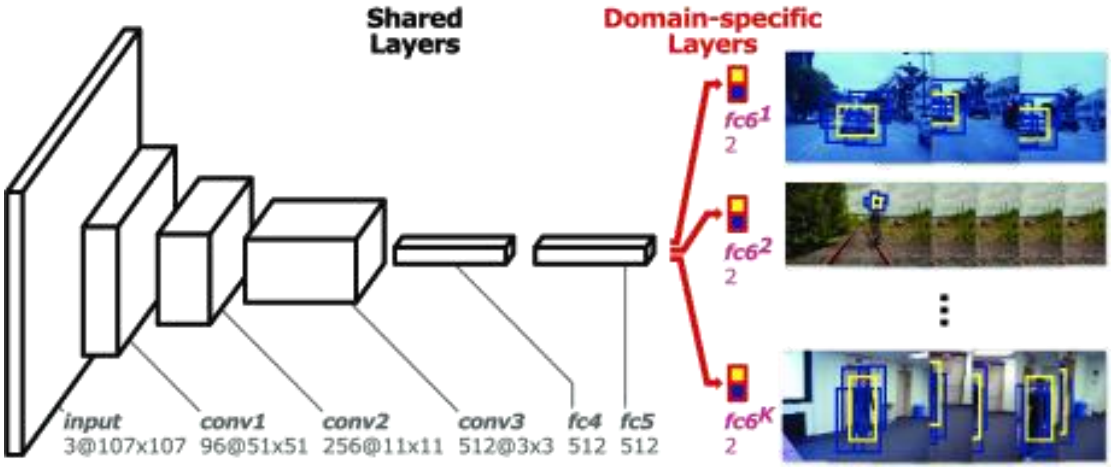

Figura 6 – Arquitetura de uma rede como múltiplos domínios [27]

Na Figura 6 está "representada uma ilustração da rede, onde é possível verificar que recebe como valores de entrada, 107x107 RGB e contém 5 camadas escondidas, dentro delas 3 são convolucionais e as restantes 2 camadas totalmente conectadas. Adicionalmente, a rede contém K ramos para as últimas camadas conectadas, correspondente a K domínios, em outras palavras, sequências de aprendizagem. " [27]

2.3.6 Recurrent Neural Network (RNN)

Recurrent Neural Networks é um algoritmo que se lembra e baseia nas ações passadas para as suas decisões. RNN pode receber mais que um vetor de *input* e produzir mais que um vetor de *output*, onde este serão influenciados não só pelo peso do *input* (que seria com um algoritmo de Neural Network normal funcionaria), mas também por um vetor 'escondido' que contém informação do contexto dos *inputs/outputs* dos algoritmos anteriores. O mesmo *input* poderá resultar em diferentes resultados dependendo dos *inputs* da série anterior.

Na Figura 7 é possível verificar o funcionamento deste algoritmo. Começa em h_0, sem informação de nenhuma ação passada, e passa o seu peso W_h para o próximo vetor. Este vetor recebe então um input x_1, utiliza o normal Neural Network, influenciado pelo h_0 gerando o h_1 e como resultado obtém o y_1. O h_1, que foi o resultado desse vetor é passado para o vetor seguinte, tal como aconteceu com o h_0, e assim sucessivamente até ao último vetor [28].

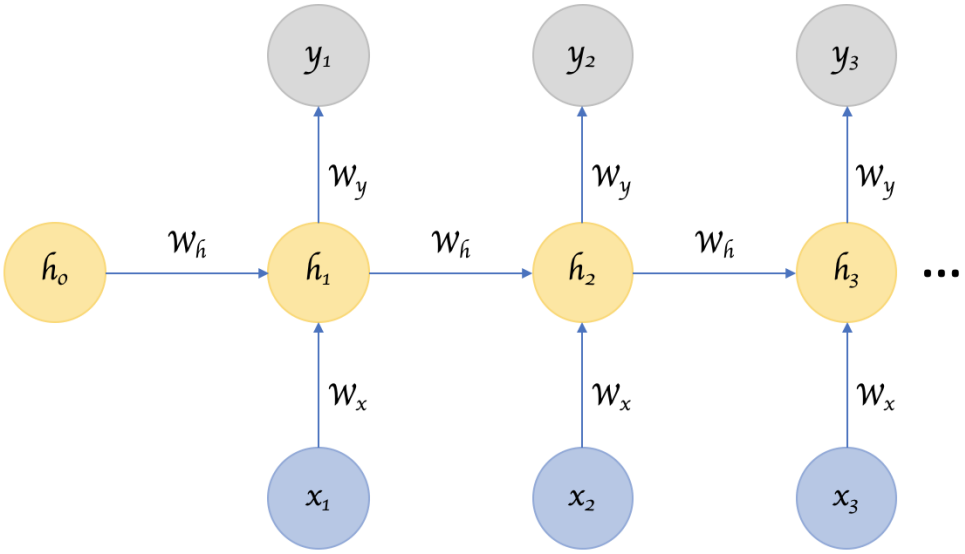

Figura 7 - Representação do funcionamento do algoritmo RNN [28]

Para melhor explicar irá ser usado o seguinte exemplo [29]. Imaginando-se que existe uma bola e o objetivo é descobrir qual a direção que tomou para a chegar à posição atual. Se a única informação que se tem é o ponto final da bola, ou seja, o último frame, é impossível ter certeza do local inicial. Para que seja possível dizer com certeza é necessária mais informação, que provêm dos frames anteriores. Na Figura 8 é possível ter uma ideia de um padrão possível do caminho da bola. Note-se que neste caso os frames anteriores seriam as bolas de cor azul mais clara, ou seja, as primeiras quatro, e a mais à esquerda, com cor azul mais escuro seria o frame atual.

Figura 8 - Representação de RNN através de um exemplo

2.4 Representação dos dados para machine learning

Existem diferentes formas de aplicar o termo 'machine learning', de forma a que os diferentes padrões de imagens sejam aprendidos para que exista uma comparação e uma percentagem desse valor estar correto [30].

Seria correto usar *Convolutional Neural Networks* (CNN) para vetores fixos, como uma imagem de algo, no entanto, para vídeos, ou seja, conjuntos de imagens, é necessário ter em conta, não só o presente frame, mas também o histórico dos frames passados. Para isso é usado o Recurrent Neural Network (secção 2.3.6), em que o presente frame analisa o resultado do frame anterior, para os frames seguintes. É necessário ainda conseguir identificar o conjunto de imagens que sejam repetidos e consequentemente retirados, para prevenir duplicação de imagens, como é possível verificar na figura seguinte [31].

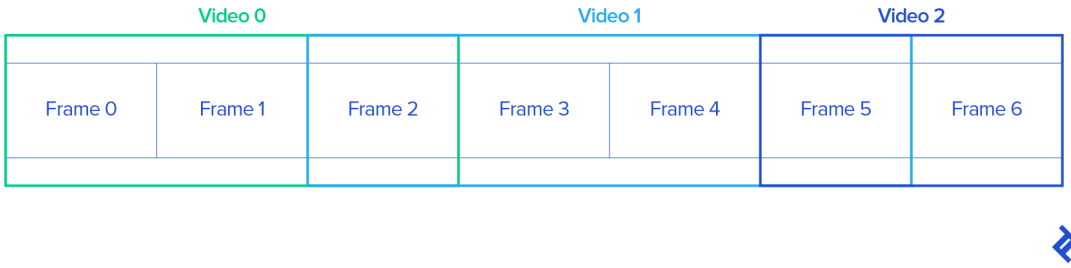

Figura 9 - Duplicação de frames [31]

Na Figura 9 é possível verificar a existência de três vídeos (0,1,2). O vídeo 0 contém 3 frames (Frame 0,1,2), contudo o vídeo 1 também contém 3 frames (Frame 2,3,4). Com isto é possível verificar que ambos os vídeos contêm o frame 2, tornando-o assim um frame duplicado com a necessidade de ser removido.

De forma a fazer a melhor análise possível de um vídeo ou uma imagem, é usado o algoritmo CNN (*Convolutional Neural Networks*) complementado com o RNN (*Recurrent Neural Network*).

No começo, é necessário treinar o CNN para a(s) imagem(ns) que vamos analisar de maneira a gravar o resultado em diferentes camadas do disco. Para identificar os objetos, existem diferentes algoritmos (InceptionV3, Xception, entre outros) que usam *ImageNet dataset* como base de comparação de imagens [30] [31].

Inception e Xpection são algoritmos que procuram imagem similares numa base de dados de grandes dimensões, onde estão armazenadas grandes quantidades de imagens. Xception foi o algoritmo dentro dos existentes com o mesmo objetivo com maior precisão na busca de imagens similares [32].

Esta base de dados já contém uma grande quantidade de imagens o que torna mais fácil a existência e comparação com outras imagens.

De seguida, foca-se na conversão de frames individuais para sequencia de frames, treinando então o RNN. Por fim, treinar o RNN para que este atinja valores de precisão superiores aos que foram anteriormente definidos [30].

A seguinte figura ajuda a sumarizar e a perceber o esquema de acontecimentos, não incluindo, no entanto, a utilização do algoritmo RNN.

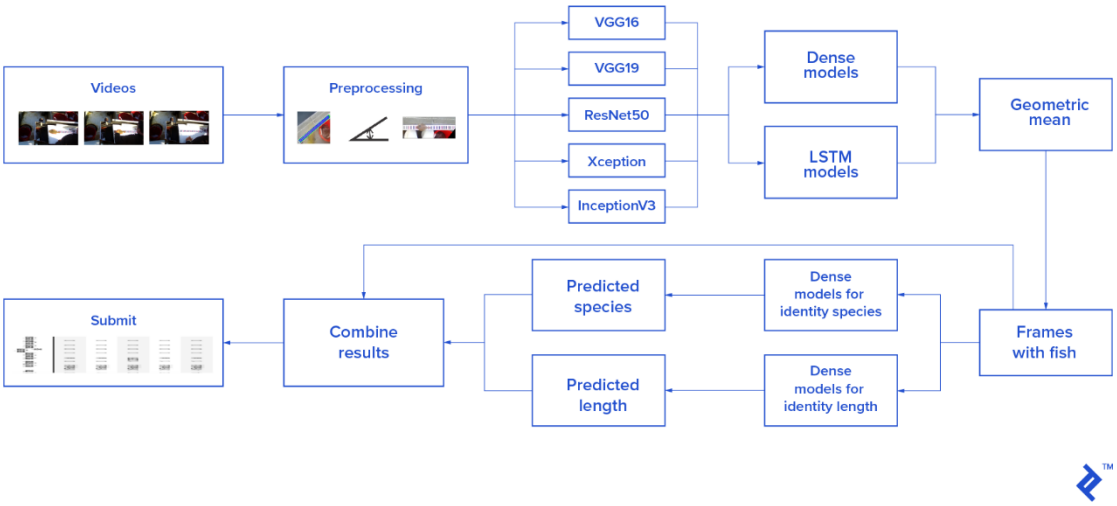

Figura 10 - Esquema de acontecimentos para análise de um conjunto de imagens [31]

A Figura 10 é um exemplo da sequência de acontecimentos usado para análise da espécie de um peixe. Note-se que este exemplo não conta com o algoritmo RNN, o que ajudaria na compreensão de frames anteriores para análise, em caso da existência de frames duplicados. É possível, no entanto, ficar com uma ideia geral de como o 'machine learning' é aplicado, e como é que as futuras frames podem usar as frames anteriores para 'aprender' e melhorar a probabilidade de forma a atingir a taxa de aceitação definida.

Linha de Tempo

Através da ajuda da tecnologia OpenCV, é possível ler e processar cada frame de um vídeo. Com a utilização deste método é possível capturar todos principais frames para que, posteriormente, seja construída uma linha de tempo com todos os objetos que sejam cruciais para a aplicação em questão.

No seguinte excerto de código, através da ajuda da linguagem Python, é possível verificar como a extração dos frames é realizável.

```
#Import libraries
import cv2
import os
#Function to extract frames
def extractFrames(pathIn, pathOut):
   #directory path, where my video images will be stored
   os.mkdir('c:/users/User/Desktop/data')
   #Capture video from video file
   cap = cv2.VideoCapture(pathIn)

while (cap.isOpened()):
   # Capture frame-by-frame
   ret, frame = cap.read()
   if ret == True:
      print('Read %d frame: ' % count, ret)
      # save frame as JPEG file
      cv2.imwrite(os.path.join(pathOut, "frame{:d}.jpg".format(count)), frame)
      count += 1
```

Listagem de Código 1 – Excerto de código usado na extração de frames [33]

Visto que todos os frames são capturados, cabe aos algoritmos de deteção/rastreamento identificar quais são os objetos das imagens que compõe a linha de tempo criada, que irão ser importantes para a aplicação de modo a serem analisados. É possível também capturar o vídeo de uma Webcam imagem a imagem com ajuda do código apresentado anteriormente. [34]

Verifique-se que foi conseguido, não só a leitura da webcam, mas também a escrita de um vídeo final. Esta parte pode ser relativamente importante quando é necessário gravar um ficheiro com apenas os frames que são importantes para realizar uma futura análise.

2.5 Validação cruzada (k-fold)

Validação cruzada é uma estratégia de 'machine learning' para avaliar se o classificador usado tem uma grande precisão com os dados treinados [35] que se dividem em categorias conhecidas.

É geralmente usado em 'machine learning' para comparar e selecionar um modelo para um dado problema de modelagem preditiva, pois é fácil de compreender, fácil de implementar e os resulta em estimativas da capacidade dos métodos sem qualquer tendência para qualquer método. [36]

O procedimento tem um único parâmetro, k, que se refere ao número de grupos que um grupo de mostra pode ser dividido. Quando um valor específico é usado para k, esse será então o valor usado de referência ao modelo, como por exemplo, se k for igual a 5, tornar-se-á numa validação cruzada de 5.

Validação cruzada é principalmente usado para estimar a habilidade de um modelo de 'machine learning' aquando o uso numa mostra que não vou vista previamente [36], isto é, usar uma amostra limitada de forma a estimar como o modelo iria responder de ima forma geral quando usado para fazer predições nos dados que não foram usados durante o seu treino.

3 Análise de valor

Neste capítulo é feita uma análise de valor ao tema e objetivo deste projeto, bem como as decisões que levaram à sua escolha.

3.1 Proposta de valor

A aplicação desenvolvida neste projeto, tem como objetivo facilitar a inclusão e a comunicação da comunidade surda com o resto da sociedade através da implementação de um software capaz de traduzir língua gestual para língua oral, tal como o Português ou Inglês, através de um vídeo, com mínimo uso de ferramentas exteriores possível (como luvas especializadas). Este é um tema a ser estudado há alguns anos, o que me interessou, de forma a poder completar ou contribuir para a resolução de tal problema que pode vir a facilitar a vida de várias pessoas, assim como explorar uma área, reconhecimento de objetos específicos.

O valor deste projeto recai na contribuição numa melhoria do acesso destas pessoas com condições limitadas na realização de uma vida quotidiana o mais normal e fácil possível, tanto a nível de educação como a nível de emprego.

3.2 Valor para o cliente

O valor para o cliente, é aquele que o cliente acha que o produto ou serviço vale de acordo com a sua perspetiva pessoal. Existe a possibilidade do mesmo produto ou serviço ter diferente valor para clientes diferentes, dependendo do seu uso. Este cliente, se tiver uma boa experiência com o produto ou serviço que adquiriu, tende a crer voltar a repetir a compra ou mesmo a partilhar a sua experiência com outras pessoas, criando possíveis novos clientes, aumentando assim o valor do produto ou serviço, à medida que a sua procura também aumenta.

Ao aplicar o conceito acima apresentado ao presente projeto, o valor para o cliente foca-se essencialmente no facto de este poder traduzir os seus vídeos de língua gestual para língua oral, recorrendo a diferentes algoritmos selecionados pelo utilizador. A eficácia irá ser aumentada exponencialmente com o seu uso, através da existência de 'machine learning'.

3.3 Valor Percebido

O valor percebido é "a avaliação geral da utilidade do produto baseados na perceção do que é recebido e do que é oferecido" (Zeithaml, 1988). Isto significa que o valor percebido é a diferença do resultado da diferença entre os benefícios e os sacrifícios do cliente.

3.3.1 Benefícios e Sacrifícios

Tendo em conta o projeto atual, os benefícios proporcionados com a introdução desta aplicação seriam: facilidade de comunicação entre pessoas surdas e o resto da sociedade, devido à tradução de vídeos de ficheiros ou em tempo-real, conveniência no uso de tal aplicação no caso de ter a aplicação num portátil ou mesmo uma aplicação móvel, contém também um bom nível de segurança através da utilização de login e perfil de utilizador

guardando os dados sensíveis. Contudo também existem algumas desvantagens como o esforço e o tempo para aprender a usar a aplicação e o conhecimento de qual o melhor algoritmo a ser utilizado, dependendo do seu objetivo.
O peso da diferença entre vantagens e desvantagens varia de acordo com o gosto, as experiências pessoais de acordo com a experiência e conhecimento entre outros fatores.

Tabela 1 – Benefícios e Sacrifícios

Benefícios	Sacrifícios
Facilidade de comunicação	Escorço de aprendizagem
Conveniência	Tempo
Segurança	

3.4 New Concept Development

O processo de inovação está dividido em três áreas: Fuzzy Front-end (FFE), o processo de New Product Development (NPD) e a comercialização. A primeira parte, o FFE é relatado como sendo uma das maiores oportunidades de desenvolvimento de todo o processo de inovação.

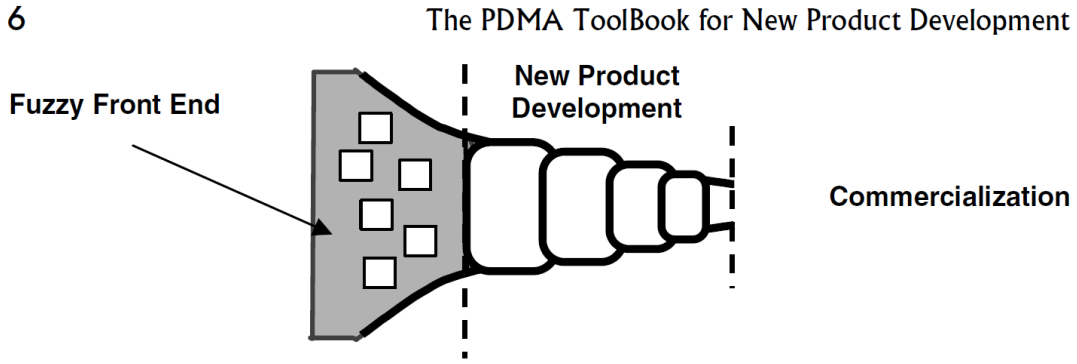

Figura 11 – Processos para inovação de um produto

O processo *Fuzzy Front End* contém processos menos otimizados que o NPD, não é organizado e é movimentado por ideais espontâneas, enquanto que o NPD tem um processo mais organizado e orientado ao objetivo com um plano do projeto. FFE é incerto, devido à falta de planeamento, o que torna difícil especular acontecimentos futuros relativos aos ganhos e ao sucesso do projeto, no entanto, o NPD pelo contrário, é possível calcular o futuro, com uma maior certeza. Estas são algumas das diferenças entre os dois processos.
De maneira a criar um modelo mais otimizado que o *Front-End of Innovation* desenvolveu-se um novo modelo mais moderno e de mais fácil interpretação, denominado de New *Concept Development*. O objetivo é ser mais objetivo, oferecendo uma linguagem e termos mais práticos.

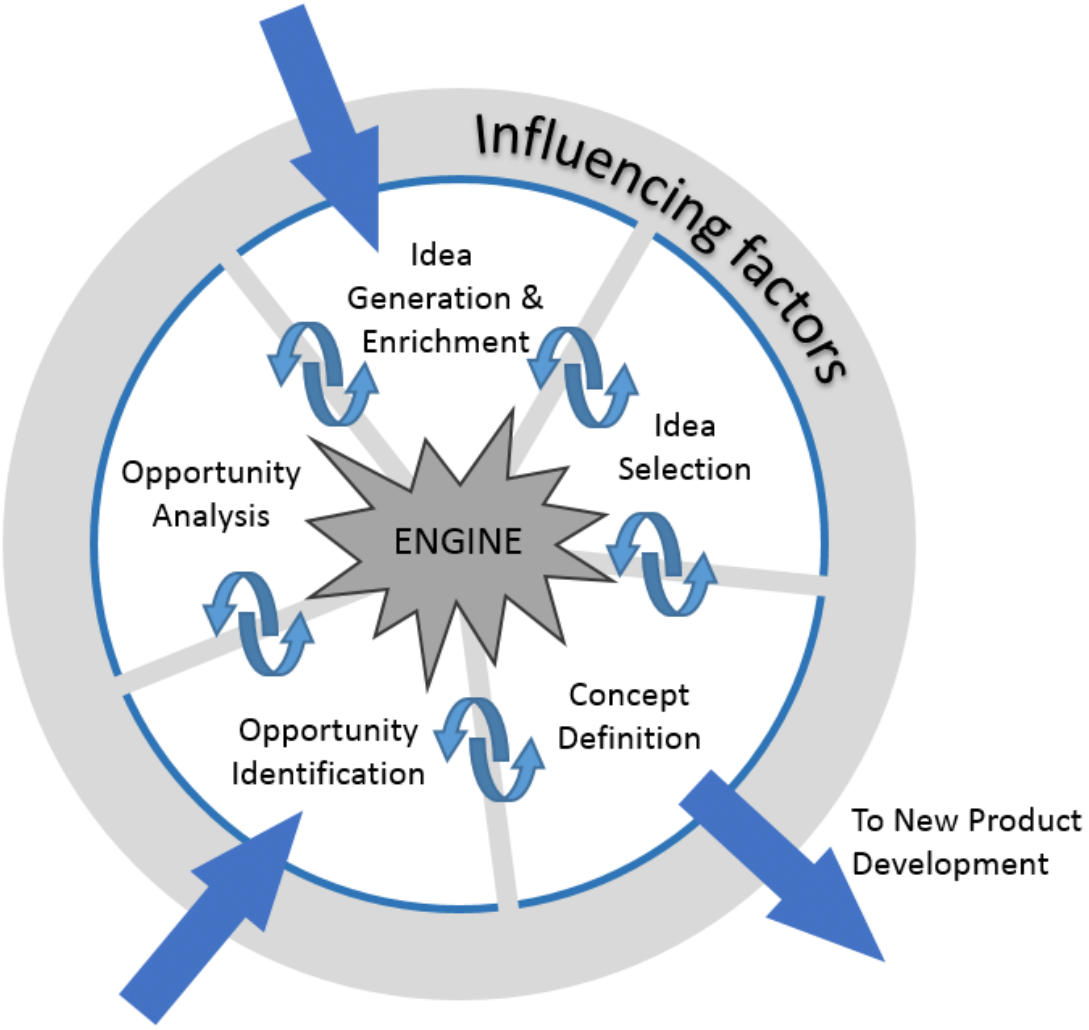

Figura 12 – Modelo New Concept Development

Este modelo (Figura 12) é constituído por três partes principais:

- A parte central, denominada por 'motor', que simboliza a liderança, escolha e as estratégias optadas por uma organização;
- A área interna que contém os principais elementos do modelo representado na figura acima;
- A área externa que representa os fatores externos que influenciam o funcionamento dos elementos do modelo.

O modelo *New Concept Development* foca-se essencialmente numa visão geral da empresa/organização tornando mais fácil detetar os fatores positivos e negativos. Com a utilização deste modelo é possível saber quais os fatores e constantes que influenciam as equipas de desenvolvimento, é possível saber as capacidades organizacionais, as ameaças de possíveis competidores, quais os clientes em questão, bem como uma melhor aprendizagem das tecnologias e ciências usadas no produto em causa.

New Concept Development é constituído por cinco elementos, sendo eles:

1. **Identificação de Oportunidades** – A existência de pouca ou nenhuma aplicação de auxílio à comunidade surda chamou a atenção do GILT. Esta ideia/projeto já é estudada há alguns anos, mas ainda não foi concluída com todas as vertentes.

2. **Análise de Oportunidades** – Após analisada e verificado o mercado de aplicações do mesmo tipo, chegou-se à conclusão que seria uma mais valia para esta comunidade.

3. **Geração de Ideias e Enriquecimento** – Várias ideias foram debatidas durante as reuniões com o professor e os restantes membros envolvidos neste projeto, inclusive ideais de anos anteriores. Dentro delas, foi discutido qual a melhor tecnologia para este projeto, quais as partes principais que vão ser trabalhadas, como é que vai ser possível testar a aplicação, como é que vai ser dividido o trabalho de entre os membros envolvidos no projeto, entre outras.

4. **Seleção de Ideias** – Depois de uma reunião onde foram discutidas todas as ideias plausíveis, foi escolhida uma ideia, tendo em conta todas as variáveis de tempo, monetárias e necessidades. De modo a apoiar a decisão da ideia, foi usado o método *Analytic Hierarchy Process*. Os critérios usados foram: Conveniência, Custo monetário e tempo de desenvolvimento. Todos estes fatores tiveram suporte na escolha entre as várias ideias: utilização de luvas de dados, através de um vídeo, mas usando luvas com LEDs e apenas usando o vídeo.

5. **Definição de Conceito** – De modo a que o projeto tenha sucesso, foi criado e planeado um plano de negócios. Houve um estudo de todas as tecnologias que podiam auxiliar, bem como um estudo dos projetos similares já existentes no mercado, de forma a auxiliar na toma de decisões aquando a implementação ou inovação.

3.5 Analytic Hierarchy Process (AHP)

Um dos principais métodos desenvolvidos no ambiente das Decisões Multicritério Discretas é o Método de Análise Hierárquica (AHP-*Analytic Hierarchy Process*), criado pelo professor Thomas L. Saaty em 1980. Como reduz as decisões complexas numa série de comparações emparelhadas, AHP facilita na compreensão e avaliação do problema. Para além de mais, AHP contém algumas técnicas úteis para verificar a consistência na avaliação das decisões. (Thomas Saaty, 1908)
Saaty apresentou algumas etapas aquando da construção da estrutura hierárquica (Figura 13), sendo elas:

- Definição do enunciado do problema, incluindo possíveis alternativas, objetivos e critérios relevantes para a solução deste;

- Avaliação de cada critério e subcritério e a sua importância relativa comparativamente à avaliação das alternativas;

- Execução da avaliação global para todas as alternativas, contendo todos os elementos estruturados de forma hierárquica.

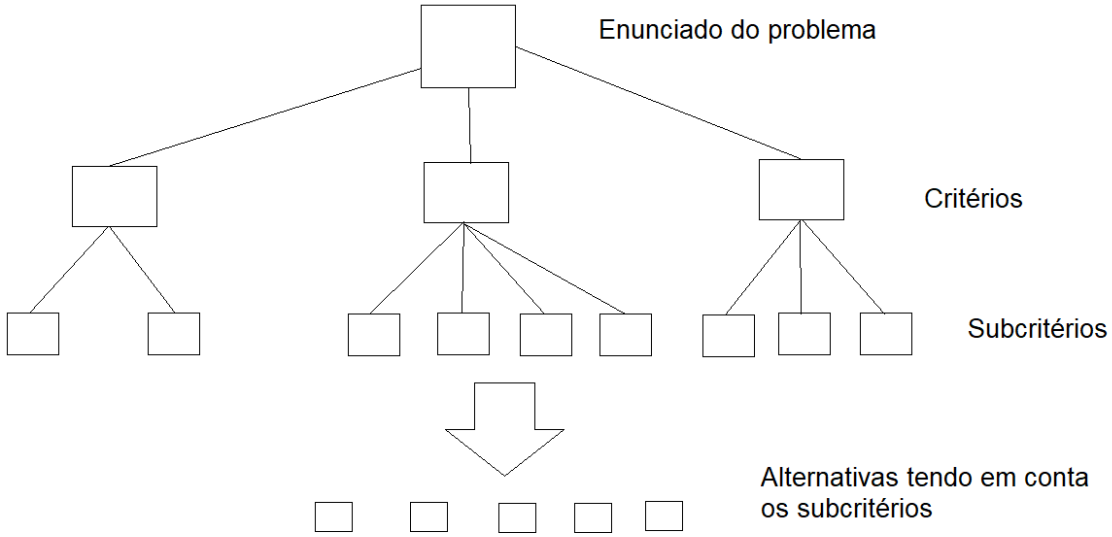

Figura 13 – Divisão Hierárquica (Saaty, 1908)

No método AHP é estabelecida uma relação entre todos os elementos de todos os níveis da árvore de hierarquia com a comparação entre os critérios ou subcritérios, isto é, a comparação dos atributos tem de ocorrer no mesmo nível da estrutura de decisão. Os valores estabelecidos por Saaty vão desde 1 a 9, sendo que 1 representa os elementos que são igualmente importantes, enquanto que 5 indica uma importância maior e 9 indica uma importância gigante e absoluta de um determinado elemento comparando com o restante. (Iañez & Cunha, 2006)

3.5.1 Fase 1 – Construção da árvore hierárquica de decisão

Definir o problema e estruturar em diagrama hierárquico. Esta fase consiste na decomposição do problema/decisão em uma hierarquia, composta, no mínimo, de um objetivo, critérios e alternativas. De acordo com a Figura 14, é possível visualizar a árvore hierárquica de maneira a selecionar a melhor ideia possível.

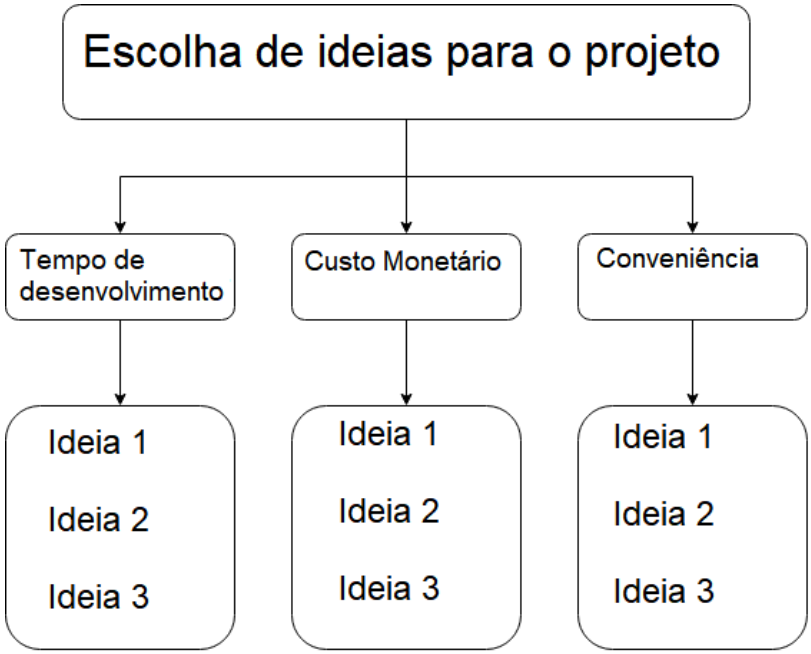

Figura 14 - Divisão Hierárquica

Os critérios que foram usados na divisão hierárquica foram tempo de desenvolvimento, custo monetário e conveniência no uso. Estas foram as ideias iniciais para o desenvolvimento deste projeto.

1. **Tempo de desenvolvimento** – Como todos os projetos é necessário planear o tempo de implementação. Projetos muito longos deixam a desejar, e se tal projeto for elaborado num curto período, algumas das funcionalidades cruciais podem não ficar implementadas.
2. **Custo monetário** – Se é necessário um grande peso monetário, não só para a elaboração do projeto em si, mas também para o seu uso, como por exemplo a necessidade de usar equipamentos externos.
3. **Conveniência** – Conveniência é um fator bastante importante, pois o utilizador vai necessitar desta aplicação em qualquer situação, garantindo sempre que seja o mais conveniente possível, independente de certos fatores como a localização.

3.5.2 Fase 2 – Comparação das alternativas e critérios

A segunda fase consiste em estabelecer prioridades entre os elementos para cada nível da hierarquia, por meio de uma matriz de comparação.

O primeiro ponto a ser considerado é a determinação de uma escala de valores para comparação, que não deve exceder um total de nove fatores, a fim de se manter a matriz consistente. Assim, Saaty definiu uma Escala Fundamental.

Tabela 2 – Escala Fundamental, níveis de importância de comparações

Nível de Importância	Definição	Explicação
1	Igual Importância	As duas atividades contribuem igualmente para o objetivo

3	Fraca Importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte Importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito Forte Importância	Uma atividade é muito fortemente favorecida em relação à outra
9	Importância Absoluta	A evidencia favorece uma atividade em relação a outra com o mais alto grau de certeza
2,4,6,8U	Valores Intermediários	Quando se procura uma condição de compromisso entre duas definições

Para poder comprar as três ideias utilizando os critérios definidos na divisão hierárquica, foi desenvolvida a matriz de comparação (Tabela 3).

Tabela 3 - Matriz de comparação de critérios

Critérios	**Tempo de desenvolvimento**	**Custo monetário**	**Conveniência**
Tempo de desenvolvimento	1	1/2	1/3
Custo monetário	2	1	1/3
Conveniência	3	3	1

3.5.3 Fase 3 – Prioridade relativa de cada critério

Esta fase tem por objetivo igualar todos os critérios a uma mesma unidade, para isto cada valor da matriz é dividido pelo total da sua respetiva coluna.

Tabela 4 - Matiz de comparação dos critérios do Segundo Nível

	Tempo de desenvolvimento	**Custo monetário**	**Conveniência**
Tempo de desenvolvimento	1	1/2	1/3
Custo monetário	2	1	1/3
Conveniência	3	3	1
Soma	6	9/2	5/3

Tabela 5 - Matriz normalizada dos critérios do Segundo Nível

	Tempo de desenvolvimento	**Custo monetário**	**Conveniência**
Tempo de desenvolvimento	1/6	1/9	1/5
Custo monetário	1/3	2/9	1/5
Conveniência	1/2	2/3	3/5

Depois da matriz normalizada dos critérios do Segundo Nível, é necessário identificar a ordem de importância de cada critério estabelecido, para isto é calculado a média aritmética dos valores de cada linha da matriz que for normalizada na tabela anterior.

Tabela 6 - Prioridade Relativa

Critérios	**Prioridade Relativa**
Tempo de desenvolvimento	0.16
Custo monetário	0.25
Conveniência	0.59

Com a Tabela 6 é possível concluir que o critério com mais prioridade relativa e importância é o 'Conveniência, seguido pelo 'Custo monetário' e por fim 'Conveniência.

3.5.4 Fase 4 – Avaliar a consistência das prioridades relativas

A próxima etapa é calcular a Razão de Consistência (RC) para medir o quanto os julgamentos foram consistentes em relação a grandes amostras de juízos completamente aleatórios.

Para calcular RC, necessário calcular o índice de consistência (IC) e o índice aleatório (IR). Irá ser usada a seguinte fórmula:

$$RC = \frac{IC}{IR}$$

Para calcular IC usa-se a seguinte fórmula:

$$IC = \frac{\lambda_{max} - n}{n - 1}$$

O n é o número de critérios e o pode ser obtido através dos cálculos em baixo demonstrados.

$$\begin{bmatrix} 1 & 1/2 & 1/3 \\ 2 & 1 & 1/3 \\ 3 & 3 & 1 \end{bmatrix} * \begin{bmatrix} 0.16 \\ 0.25 \\ 0.59 \end{bmatrix} = \begin{bmatrix} 3.02 \\ 3.04 \\ 3.10 \end{bmatrix}$$

$$\lambda_{max}\ média \left\{\frac{3.02}{0.16}, \frac{3.04}{0.25}, \frac{3.10}{0.59}\right\} = 3.05$$

Depois de determinar o valor de λ_{max}, o valor de IC já pode ser calculado:

$$IC = \frac{3.05 - 3}{3 - 1} = 0.025$$

De seguida e finalizando, é então calculado o rácio de consistência, conseguindo assim descobrir os valores de IR.

Tabela 7 – Tabela de Índices

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
IR	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

De forma a provar a consistência que foi anteriormente descrita, irá ser o utilizado o valor de índice aleatório correspondente a n = 3.
Tendo os valores de IC e IR, já possível calcular o valor de RC:

$$RC = \frac{0.025}{0.58} \cong 0.04$$

De forma a concluir toda esta fase, é possível verificar que o valor obtido ($\cong 0.04$) é menor que 0.1, podendo assim afirmar-se que os valores das prioridades relativas são consistentes.

3.5.5 Fase 5 – Construção da matriz de comparação paritária para cada critério, considerando cada uma das alternativas selecionadas

Todos os procedimentos para a construção da matriz de comparação e para a determinação da prioridade relativa de cada critério são feitos novamente, observando agora a importância relativa de cada uma das alternativas que compõem a estrutura hierárquica do problema em questão.
As ideias estabelecidas na secção 3.4, no ponto 4 de 'Seleção de ideias', são as seguintes:

1) Utilização de luvas de dados
2) Apenas vídeo
3) Utilização de luvas com LEDs

Na tabela é possível verificar uma matriz para o critério que foi definido anteriormente, tempo de desenvolvimento. A primeira e a segunda ideia têm possivelmente menos tempo de desenvolvimento devido ao facto da necessidade de implementar uma leitura e compreensão mais especifica e complexa na ideia 3 comparando com as restantes duas ideias.

Tabela 8 - Matriz de comparação do critério Tempo de Desenvolvimento

	Ideia 1	**Ideia 2**	**Ideia 3**
Ideia 1	1	4	3
Ideia 2	1/4	1	1/2
Ideia 3	1/3	2	1
Soma	19/12	7	9/2

Tabela 9 - Matriz de comparação do critério Tempo de desenvolvimento normalizada

	Ideia 1	**Ideia 2**	**Ideia 3**	**Prioridade relativa**
Ideia 1	5/8	4/7	2/3	0.62
Ideia 2	1/6	1/7	1/9	0.14
Ideia 3	1/5	2/7	2/9	0.24

Relativamente aos custos monetários, a Ideia 1 é a que contém um maior custo, devido ao facto do valor das luvas de dados, seguido pelas luvas de LEDs, e por fim a Ideia 2 com o uso de apenas vídeos, sem qualquer tipo de luvas ou custos em equipamentos externos.

Tabela 10 - Matriz de comparação do critério Custo Monetário

	Ideia 1	**Ideia 2**	**Ideia 3**
Ideia 1	1	1/3	1/2
Ideia 2	3	1	4
Ideia 3	2	1/4	1
Soma	6	19/12	11/2

Tabela 11 - Matriz de comparação do critério Custo Monetário normalizada

	Ideia 1	**Ideia 2**	**Ideia 3**	**Prioridade relativa**
Ideia 1	1/6	1/5	1/11	0.16
Ideia 2	1/2	12/19	8/11	0.62
Ideia 3	1/3	1/6	1/5	0.22

Por fim, em relação ao critério conveniência, a ideia 2 é a melhor de entre as três, devido ao facto que não precisa de nenhum equipamento exterior, podendo ser utilizado em qualquer lugar com menos restrições comparativamente às restantes duas. No entanto, a Ideia 3 é melhor que a 2, devido ao facto de ser relativamente mais fácil transportar e adquirir as luvas de LEDs.

Tabela 12 - Matriz de comparação do critério Conveniência

	Ideia 1	**Ideia 2**	**Ideia 3**
Ideia 1	1	1/3	1/2
Ideia 2	3	1	4
Ideia 3	2	1/4	1
Soma	6	19/12	11/2

Tabela 13 - Matriz de comparação do critério Conveniência normalizada

	Ideia 1	**Ideia 2**	**Ideia 3**	**Prioridade relativa**
Ideia 1	1/6	1/5	1/11	0.16
Ideia 2	1/2	12/19	8/11	0.62
Ideia 3	1/3	1/6	1/5	0.22

Para concluir, foi construído um diagrama atualizado, como é possível visualizar na Figura 15. Tal diagrama sumariza toda a informação que foi obtida nesta secção.

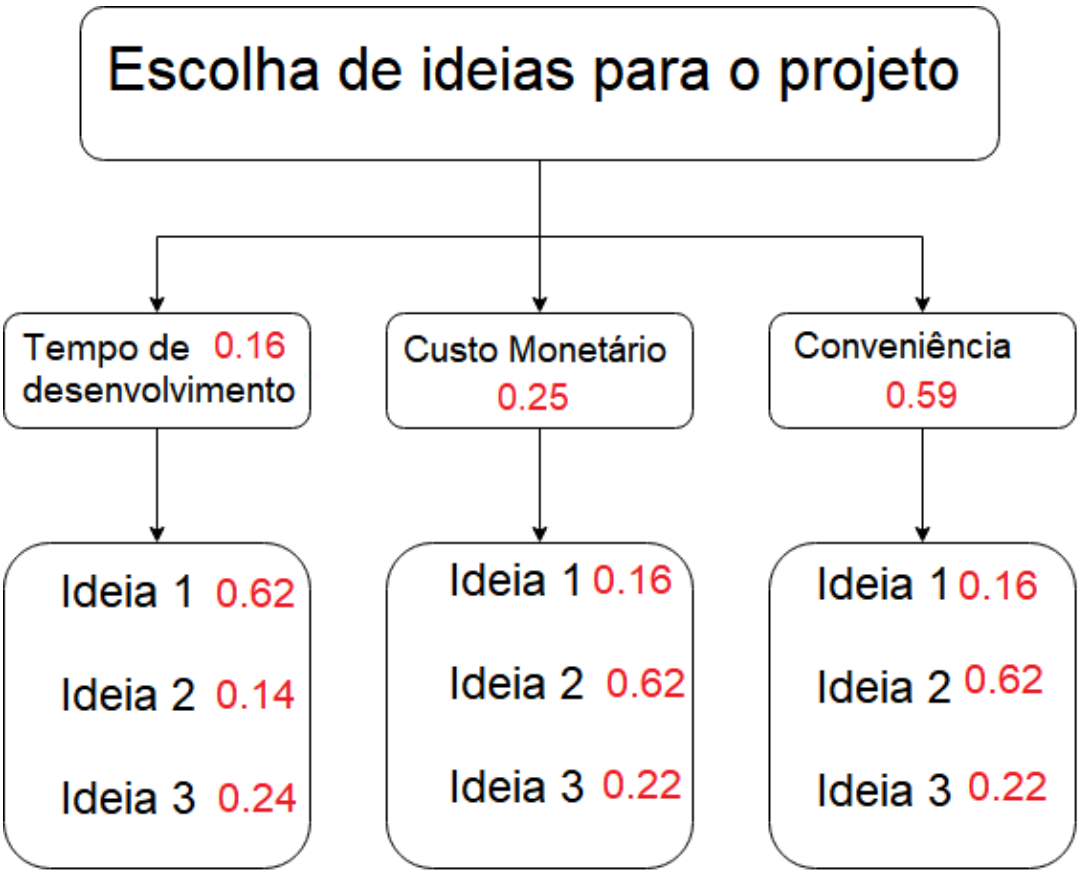

Figura 15 – Diagrama atualizado de comparação de prioridades dividido em três critérios

3.5.6 Fase 6 – Obter a prioridade composta para as alternativas

Esta é a última secção que contém cálculos. Os valores obtidos da matriz de prioridades da fase 5 irão ser multiplicados com o vetor de prioridade, resultando nos valores finais de cada alternativa, sendo que, o que obter um valor mais elevado, será a melhor alternativa.

$$\begin{bmatrix} 0.62 & 0.16 & 0.16 \\ 0.14 & 0.62 & 0.62 \\ 0.24 & 0.22 & 0.22 \end{bmatrix} * \begin{bmatrix} 0.16 \\ 0.25 \\ 0.59 \end{bmatrix} = \begin{bmatrix} 0.23 \\ 0.54 \\ 0.22 \end{bmatrix}$$

3.5.7 Fase 7 – Escolha da alternativa

Para concluir, observando os valores obtidos na fase anterior e o peso de cada critério definido anteriormente, podemos dizer que a Ideia 2 é a melhor diante as três alternativas, com um peso final de 0.54. Usando este método analítico, conclui-se assim que a melhor ideia é a construção de um projeto que não recorra a dispositivos externos.

3.6 Modelo de negócio CANVAS

Um modelo de negócio descreve como uma organização é criada e a definição do seu valor. O modelo de negócio CANVAS é um simples modelo gráfico que ajuda a centralizar os aspetos mais cruciais de um modelo de negócio de uma organização. Normalmente é dividido em quatro secções principais: a parte da esquerda refere-se à infraestrutura, a parte central demonstra o que pode ser oferecido ao cliente, a parte da direita é relativa ao cliente e, por fim, a parte inferior é referente às finanças. No entanto, todos os nove blocos do modelo estão relacionados e comunicam entre si. (Osterwalder et al.,2010)

- **Proposta de Valor (Value Propositions)** – Quais os benefícios trazidos para o cliente que o façam adquirir certo produto ou serviço. Este tema já foi explicado na secção anterior 3.1 com mais detalhe.

- **Parceiros Chave (Key Partners)** – Todas as instituições que participaram como parceiros ou fornecedores do projeto.

- **Atividades Chave (Key Activities)** – As atividades mais relevantes de forma a serem desenvolvidas pela instituição de maneira a que o modelo de negócio seja cumprido de uma maneira eficaz.

- **Recursos Chave (Key Resources)** – Recursos necessários para a criação de valor. Para a criação deste projeto é necessário software e hardware para o desenvolvimento do mesmo. Eventualmente também será necessário a criação de uma base de dados para guardar os perfis de utilizador.

- **Canais (Channels)** – A forma pelo qual o produto final vai ser distribuído para chegar ao cliente. Essencialmente, os canais é a forma de comunicação com o cliente, tendo em atenção a sua distribuição e venda.

- **Fluxo de Receita (Revenue Streams)** – A forma de ganhar dinheiro com a venda do serviço ou produto.

- **Estrutura dos Custos (Cost Structure)** – Todos os custos necessários para o modelo de negócio funcionar corretamente. Principalmente os cursos de manutenção e desenvolvimento do projeto e dos servidores.

- **Relacionamento com Clientes (Customer Relationship)** – Como é que os utilizadores da aplicação podem comunicar com os desenvolvedores da aplicação. Inclui casos de *helpdesk*[1], para responder a dúvidas dos utilizadores, assim como *bugreports*[2] para melhoria da aplicação em caso da existência destes.

[1] Segmento de pessoas destinado a ajudar os utilizadores com dúvidas da aplicação
[2] Comunicação de erros da aplicação aos desenvolvedores

- **Segmento dos Clientes (Customer Segments)** – Esta componente contém o segmento de utilizadores ou grupo de utilizadores para qual a aplicação ou serviço é destinado a servir.

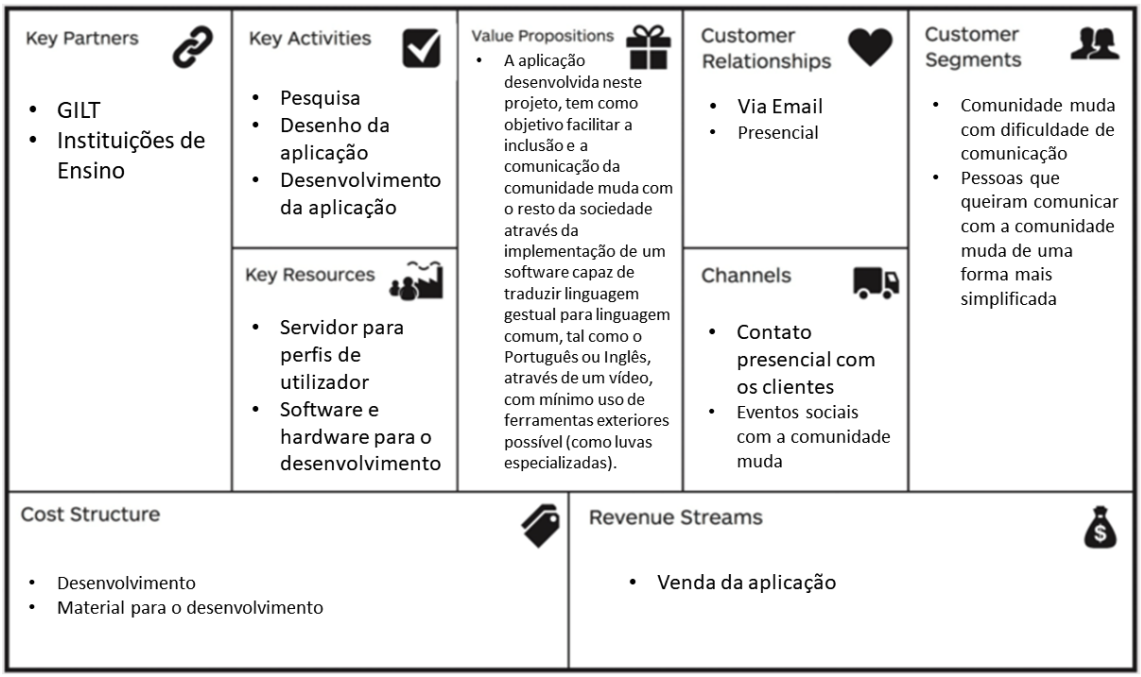

Figura 16 – Modelo de negócio CANVAS

4 Desenho e Implementação da solução

Neste capítulo vão ser retratados as decisões para construção do desenho da solução, tendo sempre em conta os padrões e boas práticas lecionados em Engenharia Informática.

4.1 Requisitos funcionais e não funcionais

A principal audiência desta aplicação são as pessoas com deficiências auditivas e como tal, os requisitos funcionais encontrados para esta aplicação são:

- Permitir que a pessoa em frente a uma câmara, grave o gesto correspondente um movimento (palavra/frase);
- Construir um modelo de classificação a partir de um conjunto de vídeos;
- Usar um modelo de classificação para inferir palavras que estejam a ser executados na câmara;
- Permitir alterar o algoritmo de classificação a utilizar em *runtime* ou através de um ficheiro de configuração exterior;
- Tendo um conjunto de vídeos(*dataset*), é possível gravar um modelo;
- Utilização do modelo criado anteriormente para classificar novos vídeos;
- Ter a possibilidade de selecionar o modelo de classificação a utilizar para fazer as previsões.

4.1.1 Requisitos não funcionais

Na Tabela 14 encontram-se todos os requisitos não funcionais que fazem parte do FURPS+.

Tabela 14 – Requisitos não funcionais

Usabilidade	• Aplicação de fácil utilização de forma a não necessitar de nenhuma formação previa; • Conteúdo fácil de entender;
Reliabilidade (Confiabilidade)	• Baixa frequência de falhas; • Em caso de erro durante a aplicação, esta irá mostrar uma mensagem clara

	• Se o servidor estiver em baixo, a aplicação pode continuar a ser usada;
Performance	• Eficiente para permitir a tradução de língua gestual em tempo real sem atrasos significativos; • Resposta rápida às ações do utilizador;
Suportabilidade	• Escalável; • Deve ser compatível com diferentes dispositivos moveis e sistemas operativos;

4.2 Base de Dados

A base de dados neste projeto está dividida entre utilizadores que se registaram e utilizadores que não se registaram. Utilizadores convidados (que não se registaram), podem apenas utilizar o dataset geral (Dataset que contém a junção dos datasets de todos os utilizadores registados), no entanto, não podem ter o seu próprio enquanto usarem a conta de utilizador convidado. Para os utilizadores registados, existem um dataset próprio desse utilizador, mas é possível também o uso do dataset geral. Estes utilizadores também podem usufruir da escolha do modelo de classificação que querem utilizar. Existem vários modelos de classificação e cada um pode ser escolhido por vários utilizadores. Cada conjunto de Modelo de Classificação-Utilizador contém o número de vezes que este foi usado.
Cada dataset contém todas as palavras que foram aprendidas e a probabilidade de acerto de cada uma delas (percentagem de acerto baseado no número de vezes que foram treinadas).
A representação da base de dados pode ser vista na Figura 17.

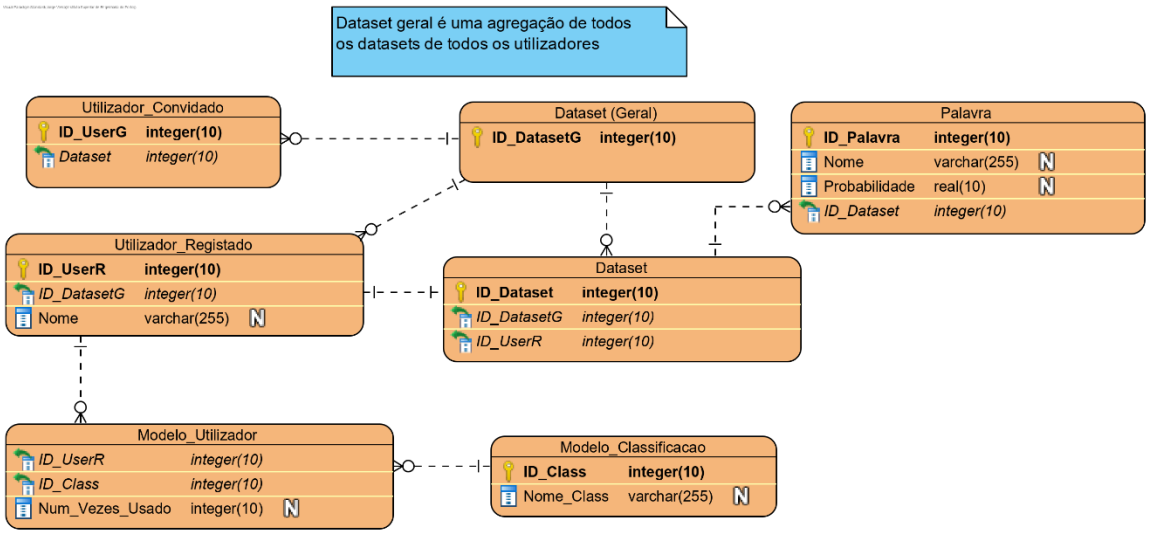

Figura 17 – Base de Dados

4.3 Diagrama de componentes

De modo a encontrar uma solução para o problema proposto, é necessário contextualizar e demonstrar através de uma arquitetura de alto nível capaz de resolver a comunicação entre todos os componentes que fazem parte da solução.

Com isto, é possível verificar a arquitetura que foi utilizada e implementada com ajuda da Figura 18.

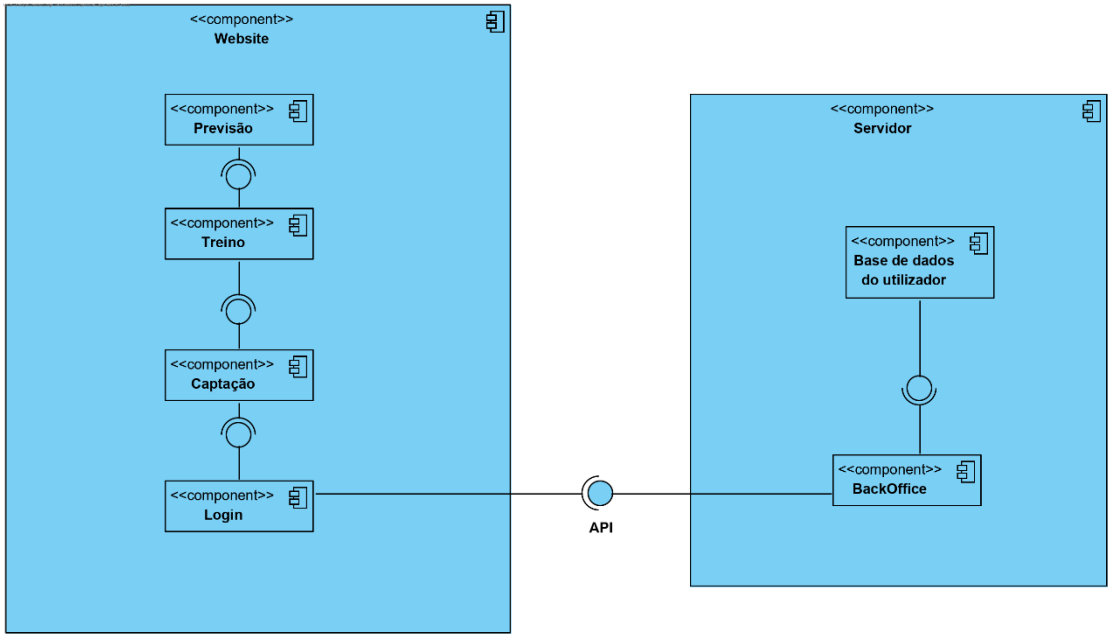

Figura 18 - Diagrama de Componentes

Como é possível verificar no diagrama de componentes, o sistema encontra-se dividido em duas componentes principais, o website e o servidor.

O projeto encontra-se maioritariamente hospedado num website, utilizando o serviço Heroku (cf. secção 2.1.5), onde conta com as três principais funcionalidades: captação, treino e

previsão. Todas elas comunicação com pelo menos uma das restantes três, isto é, depois de captado o vídeo e guardado, este vai ser consumido e usado para treinar o modelo. Tal modelo, vai ser usado como referência quando a previsão dos movimentos está a ser realizada.
Por outro lado, a parte do servidor, embora planeada, mas não implementada, esteve sempre prevista, comunicando com entre o servidor e o website usando um sistema de login.

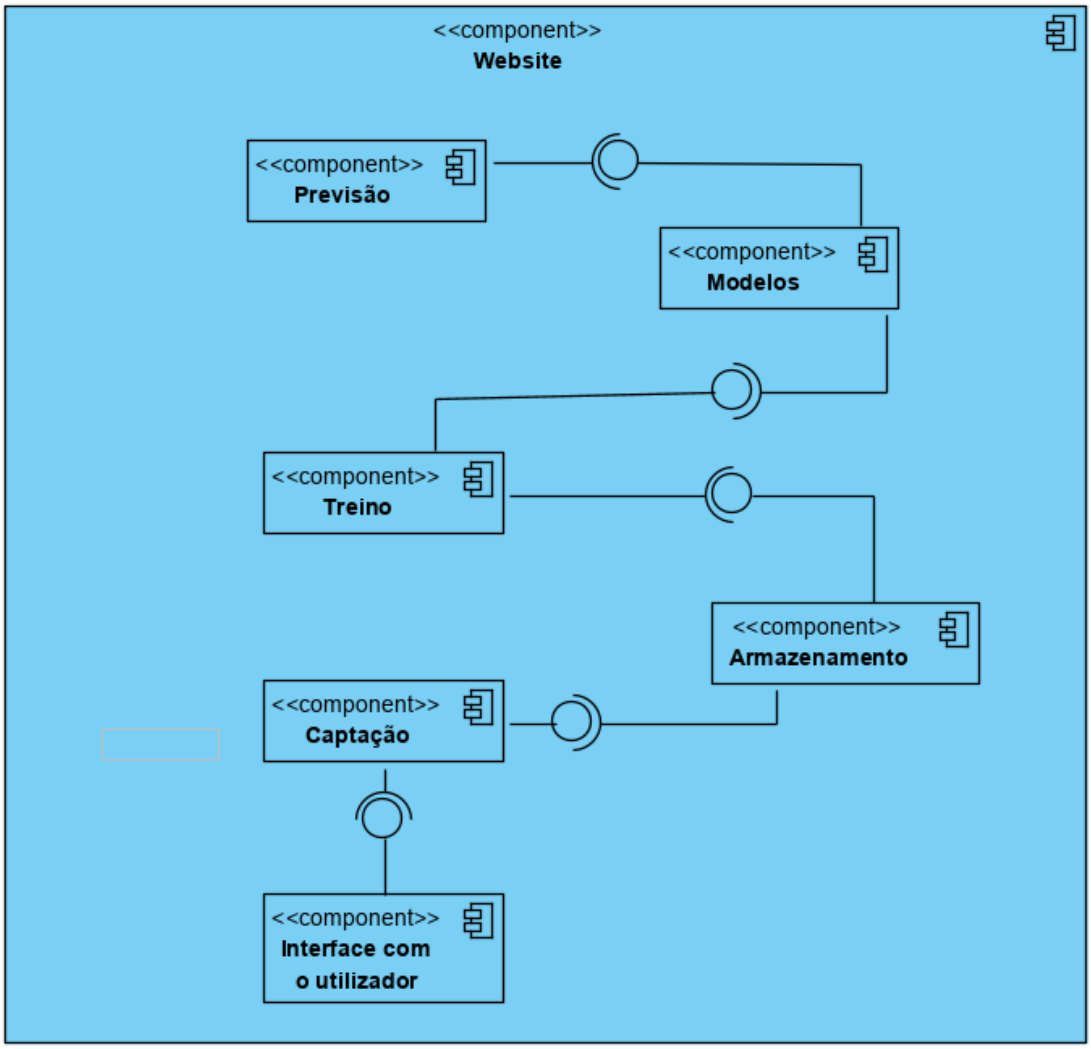

Figura 19 – Novo diagrama de componentes

Visto que a base de dados não iria sem implementada, procedeu-se ao à elaboração de um novo diagrama de componentes, Figura 19.
Com este novo desenho, os utilizadores não precisam de criar uma conta para poderem aceder ao à captação ou treino dos seus movimentos, bem como a dos outros utilizadores, no entanto, a privacidade e proteção dos seus dados foi perdida.

4.4 Diagrama de Atividades

Nas figuras seguintes serão apresentados os diagramas de atividades da solução que irá ser implementada. Com a ajuda dos diagramas é possível ter uma ideia do fluxo que das atividades e das suas respetivas interações.

Nestes diagramas parte-se do princípio de que o utilizador já se encontra dentro da sua sessão, com todas as suas definições associadas.

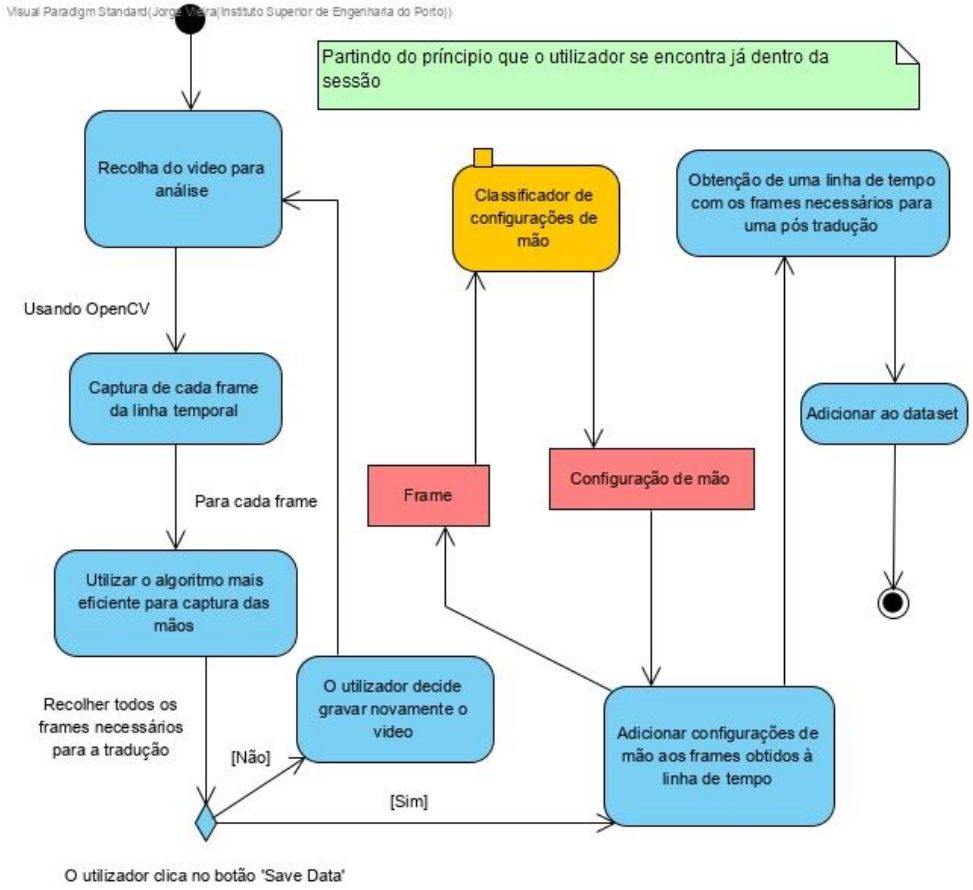

Figura 20 - Diagrama de atividades principal

Como é demonstrado na Figura 20, a aplicação inicia-se com a recolha de um vídeo para análise. O utilizador seleciona o vídeo que pretende traduzir e, com a ajuda do software 'OpenCV', é possível capturar cada frame do vídeo que foi selecionado e representá-los numa linha de tempo.

Entretanto, para cada frame, é necessário localizar as mãos no vídeo, bem como analisar e verificar se existem algum tipo de problema que possa ser resolvido, como o caso da oclusão das mãos.

Posto tudo isto, e tendo o frame com a mão a representar uma palavra/letra, esse frame irá ser enviado para o classificador de classificações de mão (explicado com mais detalhe na Figura 21) que, por sua vez, retornará a própria configuração de mão para a linha de tempo.

Por fim, obtém-se uma linha tempo, apenas com os frames que representam palavras, sendo que estes são adicionados ao dataset (explicado com mais detalhe no diagrama da Figura 22), para que, no futuro caso exista uma palavra similar, esta seja detetada, com a ajuda do machine learning.

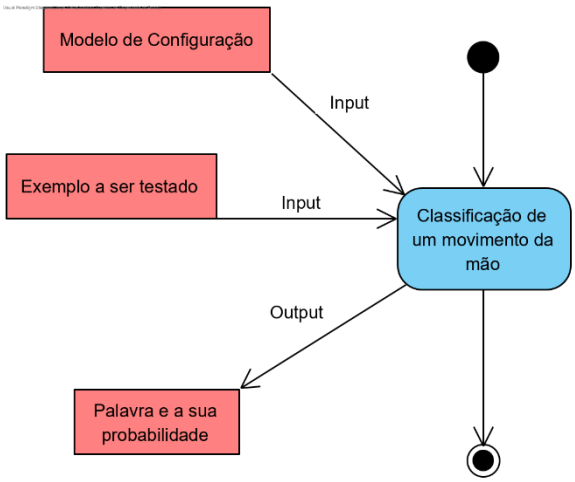

Figura 21 – Classificação de um movimento

Na Figura 21 possível verificar que o classificador recebe o exemplo que vai ser testado junto com o modelo de configuração até agora implementado. Após a classificação desse movimento ser executada por um software exterior, irá retornar um conjunto de palavras/letras e a probabilidade de estar correto.

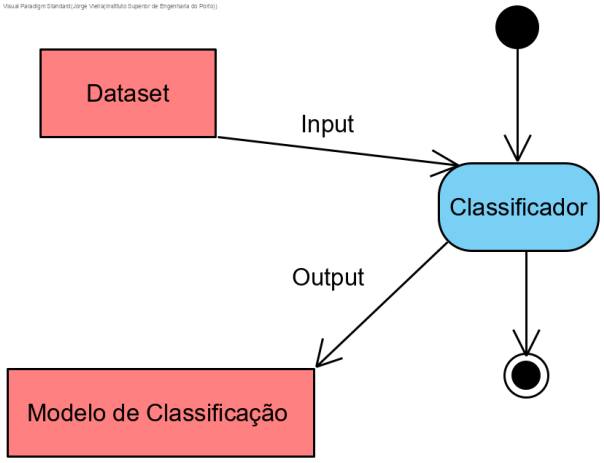

Figura 22 – Classificador

Através da Figura 22 verifica-se que o classificador irá receber o dataset que foi construído ao longo dos frames, resultando na inserção desse dataset no modelo de classificação, contribuindo para a melhoria do machine learning. São testados vários modelos de datasets, com diferentes configurações de mão contra o classificador para verificar se existe algum similar, assim como diferentes classificadores para verificar se a palavra/letra é existente em algum deles.

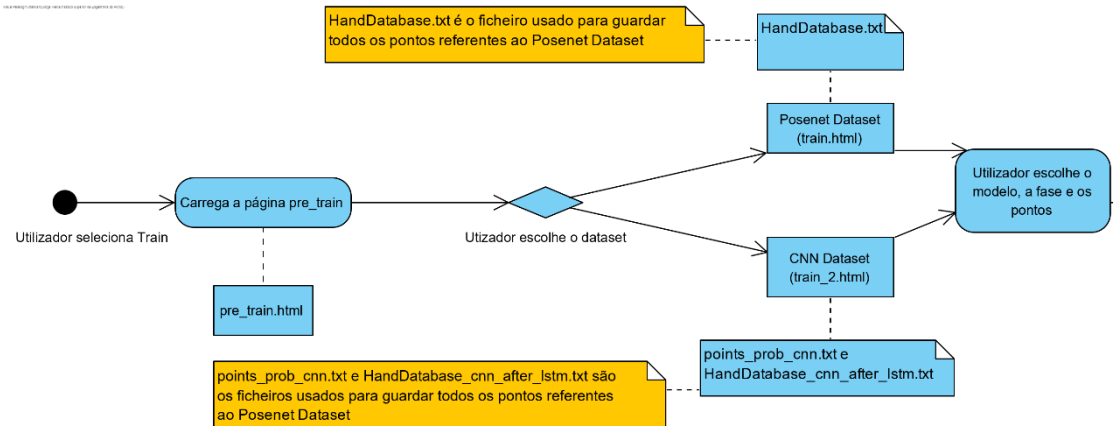

Figura 23 – Diagrama de fluxo do treino parte 1

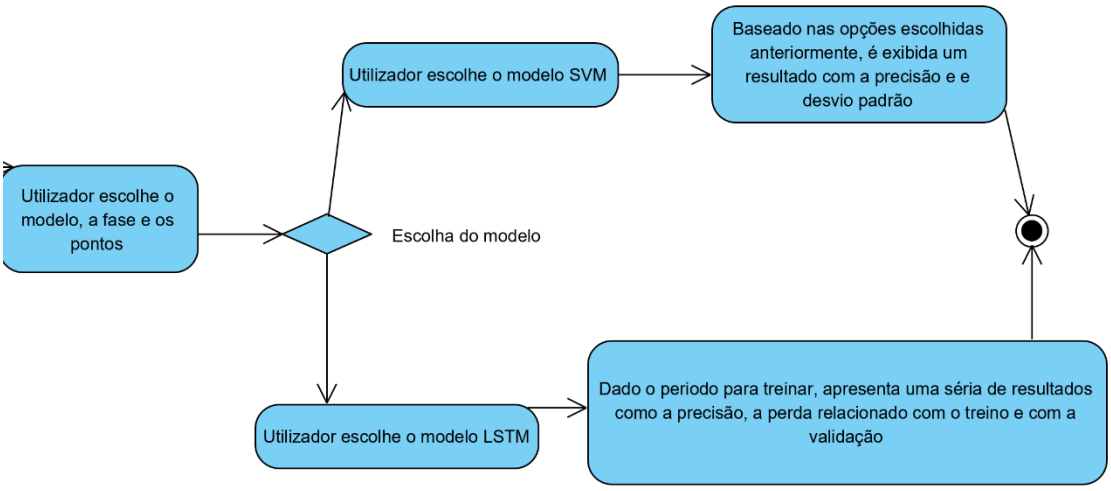

Figura 24 – Diagrama de fluxo do treino parte 2

Com a Figura 23 e Figura 24 é possível perceber o funcionamento do fluxo da informação aquando a realização do treino do modelo.

Primeiro o utilizador escolhe a função de treino, onde é dado a opção de escolha de um dos dois conjuntos de informação, ou datasets, onde foram gravados os movimentos. Cada um deles contém um ficheiro separado, com formas de gravação diferentes, modificando assim o resultado.

Após essa decisão, o utilizador é confrontado com a escolha de um dos dois modelos implementados, LSTM ou SVM.

SVM é um algoritmo mais demorado que têm em conta mais características, mas que no final apenas demonstra o resultado com a precisão e o desvio padrão. Por outro lado, o LSTM é um algoritmo mais rápido, onde o utilizador pode estar a observar um gráfico com o desvio a cada segundo para uma melhor observação, contudo também obtém a precisão e o devido desvio padrão.

4.5 Implementação

Nesta secção são discutidos todos os passos e decisões tomados ao longo de todo o desenvolvimento do projeto. Toda a solução foi desenvolvida utilizando as ferramentas apresentadas no estado da arte, secção 2.1. Todas as sugestões obtidas durante a fase de testes foram obtidas depois de reuniões com o orientador, assim como uma análise durante todo o processo.

4.5.1 Acessibilidade

Para que seja possível fazer a previsão das palavras através de vídeo é necessário primeiramente adicionar as palavras a um dataset para que sejam treinadas posteriormente. Para isto os utilizadores dirigem-se à página de captação como é demonstrado na Figura 25. É necessária uma câmara para a recolha dos movimentos, independentemente se a aplicação se encontra no telemóvel ou no computador, contudo para iniciar a captação do vídeo é essencial clicar no "Toggle video" depois da câmara pretendida aparecer no menu dropdown que se encontra à direita do botão referido.

4.5.2 Storyboard

Neste tópico será abordado as partes principais do projeto, explicando brevemente o objetivo de cada uma assim como a sua interface. O website encontra-se dividido em três partes principais: captação, treino e previsão.

4.5.2.1 Captação

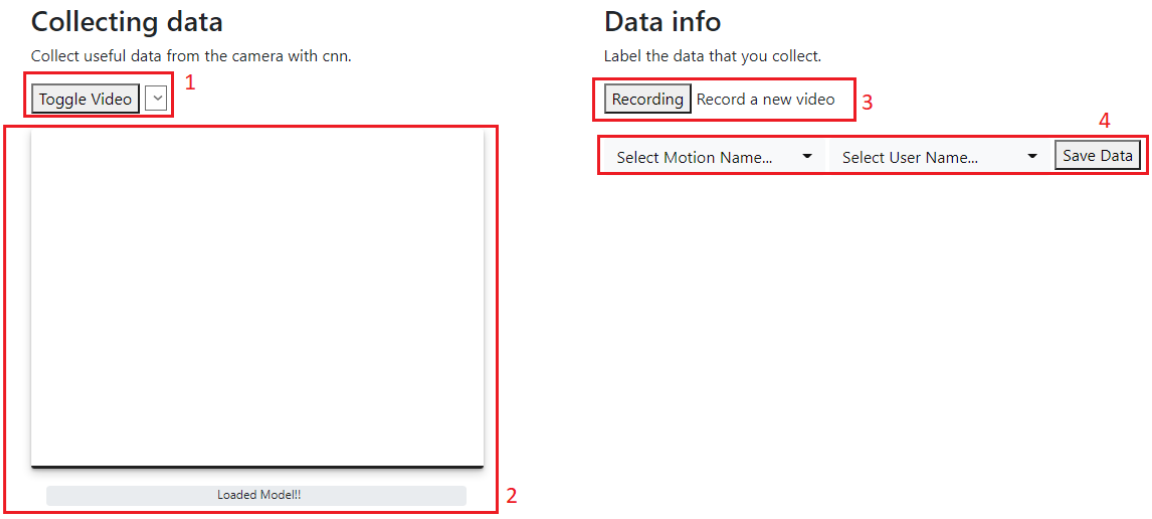

Figura 25 - Storyboard do menu de captação

A Figura 25 resume o esquema de treino de dados para a formação de um conjunto de dados (*dataset)*. É necessário a escolha de como a informação vai ser captada, com o auxílio de alguns algoritmos como *Posenet, CNN (online)* ou *CNN.* Depois de uma análise dos dados obtidos, o algoritmo que obtém melhores resultados é o *CNN*, sendo então a opção normalmente adotada aquando a gravação dos movimentos.

Indicado com o número 1 é a zona associada à câmara. Todas os dispositivos de gravação disponíveis encontrar-se-ão no menu de *dropdown*. Depois de escolhido o dispositivo, o utilizador necessita de carregar no botão "Toggle Video" para a câmara selecionada estar de facto a ser utilizada.
A secção com o número 2 destina-se à visualização do que é captado pela câmara do dispositivo. A barra inferior tem como efeito avisar o utilizador da gravação que irá ser feita, para que este possa ter tempo para se preparar para o movimento a realizar e de seguida para saber a duração da gravação. Neste momento tanto a duração da pré-gravação como a da gravação estão estabelecidos para 5 segundos cada. Para iniciar a gravação é necessário clicar no botão indicado com o número 3, note-se que a barra falada anteriormente irá começar a contagem decrescente.
Por fim, com o número 4, encontra-se o painel onde é possível escolher o nome com o qual o movimento vai ser gravado, bem como o nome do utilizador que o efetuou. De modo a guardar o conteúdo que gravado depois de toda a contagem é necessário clicar no botão "Save Data" e esperar que aparece um pop-up com o nome do movimento a avisar o utilizador que o movimento foi gravado com sucesso.

4.5.2.2 Treino

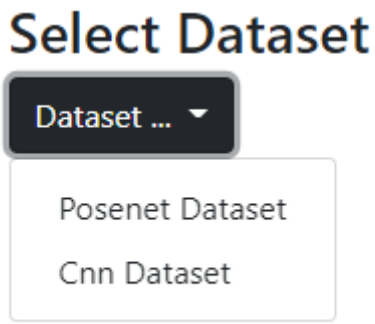

Figura 26 - Storyboard do pré-treino

No painel da Figura 26 o utilizador pode escolher o dataset que usou para gravar os movimentos referidos na secção anterior de forma a poder treiná-los.

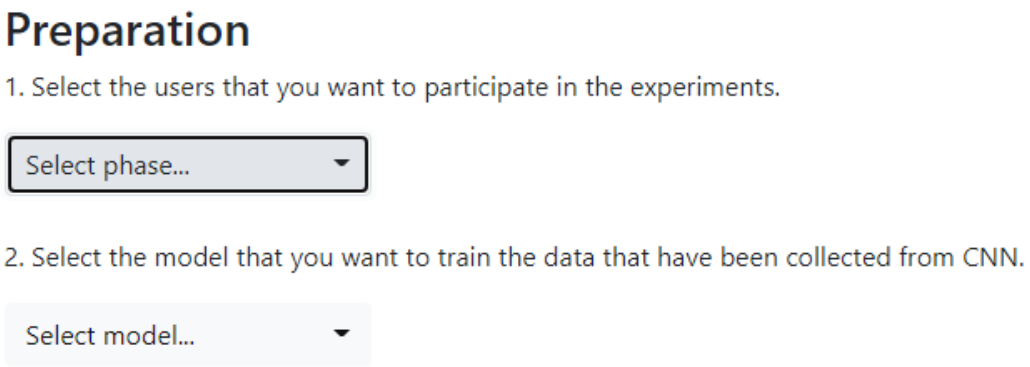

Figura 27 - Storyboard do treino usando CNN

Se o dataset escolhido for o CNN, aparecerá as opções iguais às da Figura 27, onde o utilizador pode escolher quais utilizadores, previamente criados, que irão realizar o treino, assim como o algoritmo com o modelo de treino irá ser usado para o treino. Este componente é explicado com mais detalhe na secção 4.5.6.

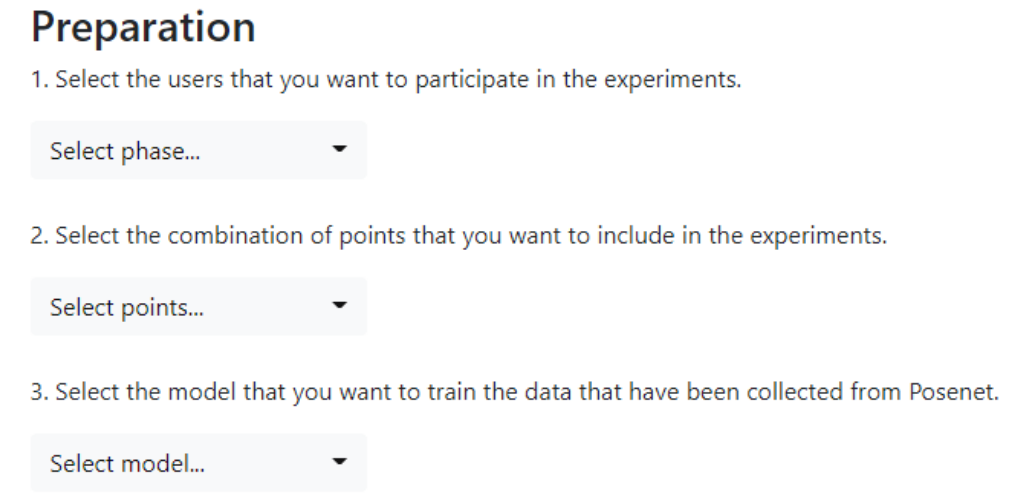

Figura 28 - Storyboard do treino usando Posenet

A Figura 28 representa as opções aquando a escolha do Posenet dataset. A primeira e a última opção são similares às do treino usando CNN, conforme a Figura 27.
No entanto, este dataset contém outra preferência, a seleção da combinação dos pontos a utilizar quando for realizado o treino. Esta opção dá a oportunidade ao utilizador de escolher quais as coordenadas a ter em conta, como por exemplo, usar apenas as mãos e os ombros, ou apenas a mão direita, ou apenas os cotovelos e assim sucessivamente. Apesar deste dataset não ser usado (Posenet), devido a ter uma pior precisão nos testes realizados em relação ao CNN, foi analisado pela Despoina (cf. Anexos) que a combinação com mais precisão é a que usa apenas a mão direita

4.5.2.3 Previsão

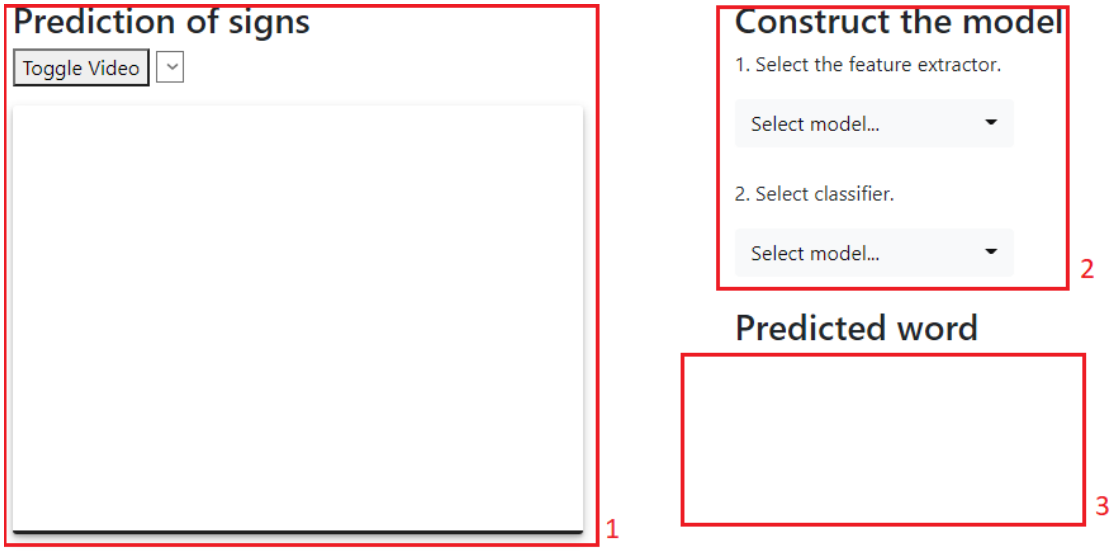

Figura 29 - Storyboard da previsão

A Figura 29 apresenta a última opção do menu que contém a previsão utilizando o modelo treinado anteriormente.
Esta página divide-se essencialmente em três partes referidas na figura apresentada anteriormente. Indicado com o número 1 e similar à parte de treino, encontra-se um painel com o menu *dropdown* onde o utilizador vai escolher o dispositivo de gravação. Consequentemente a imagem irá aparecer por baixo do botão, imediatamente debaixo do botão.
Assinalado com o número 2 encontram-se as escolhas para o modelo que irá realizar a previsão. Primeiramente o modelo que irá extrair o modelo (similar à captação) e qual será o classificador escolhido (tal como os que foram escolhidos na parte do treino).
Por fim, e com o número 3, é apresentado o painel onde a palavra com mais precisão é apresentada, juntamente com a sua percentagem de acerto, de acordo com o modelo treinado pelo utilizador. Esta palavra e a sua percentagem estão em constante alteração de acordo com os movimentos realizados pelo utilizador e captados pela câmara.

4.5.3 Condições de gravação

É necessário ter em conta certos fatores que podem influenciar a qualidade da gravação e o número de características detetadas, tendo um direto impacto no resultado final e na precisão.
É fundamental que o utilizador não se encontre demasiado próximo da câmara, de maneira a que os seus movimentos se encaixem dentro do raio de visão da gravação.
Verificar a luminosidade do local de gravação. É preferível gravar num local brilho proveniente do sol ou utilizar luzes específicas. Se o ambiente de gravação tiver demasiado brilho ou for excessivamente escuro, podem existir pontos ou imagens da mão que não são captados.
Aquando a gravação, o utilizador necessita de realizar os movimentos pausadamente, dependendo da capacidade da câmara. Se os movimentos forem demasiado rápidos, e a câmara não tiver capacidade de captar os frames nitidamente, existe a possibilidade de bastantes pontos vão ser perdidos.

4.5.4 Alocação de website

Depois de uma pesquisa a várias plataformas que oferecem serviços de nuvem para alocar um projeto elaborado em Python, o mais acessível e menos dispendioso encontrado foi a plataforma denominada de Heroku (cf. Secção 2.1.5).
Para poder alocar o projeto é necessário uma série de passos a cumprir. Para poder começar é essencial a instalação do serviço Git, bem como o toolkit proveniente do Heroku.
É necessário criar um ambiente virtual dentro da pasta do projeto, com ajuda do comando *virtualenv <nome da pasta>* e de seguida ativar os scripts dentro da pasta Scripts no diretório do ambiente virtual.
De seguida, instalar todas as dependências do projeto com a ajuda de um instalador de pacotes. Uma das dependências a dar destaque é o *gunicorn*, com a ajuda deste pacote e depois da criação do ficheiro *Procfile*, é possível alterar a porta de entrada, ou seja, o primeiro ficheiro a ser lido quando for feita a implementação do projeto.

Outro ficheiro essencial é a criação de um ficheiro designado de *requirements*, ou seja, que que contém todos os requerimentos de librarias e dependências que o projeto tem, de forma a poderem ser instaladas de forma automática.
Por fim, é necessário a criação de um website com toda a informação que foi gravada, usando o comando *Heroku create* para a criação da aplicação Heroku, e de seguida, com ajuda do Git, enviar todas as pastas e ficheiros criados para o ramo *master*.
Para saber qual é o URL do site, existe um comando designado de *Heroku open* que abre uma página web com o website em funcionamento.

4.5.5 Previsão da configuração da mão em cada imagem capturada

De maneira a testar se o uso de informação extraordinária em relação aos pontos do movimento da mão, como o exemplo da previsão da letra da mão em cada frame capturada, iria ajudar a obter melhores resultados aquando o treino dos modelos capturados, foi implementado um algoritmo que tenta capturar imagens de cada frame, se estas forem visíveis e percetíveis, e no momento de gravação de informação, vai a cada imagem que foi gravada e devolve as três letras com maior precisão e a probabilidade de acerto de cada uma destas.

```
if(flag_recording == true) {
  var dataURL = c1.toDataURL();
  dataURL = dataURL.replace('', '');
  $.post("/frame", {
    javascript_data: dataURL
  });
}
```

Figura 30 - Envio da imagem em base64 para Python

Como é possível verificar na Figura 30, para cada imagem (frame) do vídeo a ser capturado, estas são guardadas e passadas para base64 de forma a poder enviar como *post* com o intuito de ser tratada em Python.

```python
@app.route("/frame", methods=['POST'])
@cross_origin()
def frame():
    global detection_graph
    if request.method == 'POST':
        tf.print(detection_graph)
        # Receive the base64 frame from js
        frame = request.form['javascript_data']
        # Base64 decoder to transform in an image
        data = base64.b64decode(frame)
        with detection_graph.as_default():
            with tf1.Session() as sess:
                nparr = np.frombuffer(data, np.uint8)
                img = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
                if (img is None) == False:
                    img = detect_image(img, "Both_Hand_Images", detection_graph, sess)
    return Response(status=200)
```

Figura 31 – Descodificação de base64 e gravação da imagem

Depois de recebido em Python (Figura 31), a imagem é descodificada de base64, criada uma sessão e enviada para o detect_image que, após receber uma imagem, procura a presença de mãos e recorta o mais curto possível de forma a conter o menos ruido possível. Por útlimo, guarda essa imagem na pasta 'Both_Hand_Images'.

```python
for timeSteps,_ in enumerate(new_obj['poses']):
    f.write(new_obj['name'] + ",")
    f.write(new_obj['user'])
    f.write("," + str(count_points))
    count_points = count_points + 1
    for value in new_obj['poses'][timeSteps]:
        f.write( "," + str(new_obj['poses'][timeSteps][value]))

    dataset[count_txt] = []
    if (count < len(filename_list) and os.path.exists(os.path.join('Both_Hand_Images',filename_list[count]))):
        dataset.append(give_letter_and_probability(os.path.join('Both_Hand_Images',filename_list[count])))
```

Figura 32 – Gravação dos movimentos em ficheiro

No final, quando o utilizador clica em 'Save Data' para gravar toda a informação que foi capturada, é onde existe a utilização do algoritmo, depois de enviar uma imagem de uma mão com um gesto, devolve as três letras com maior probabilidade e as suas probabilidades num conjunto de dados, como na Figura 32. O ficheiro com a informação da gravação que vai ser gravado, contém o nome do movimento, o nome do utilizador, os pontos da mão durante o frame e, no final, as três letras e as suas probabilidades.

O algoritmo passa por todas as imagens da pasta onde foram gravadas todas as imagens anteriormente, devolve o conjunto de informação e apaga todas as imagens.

Todo este processo é feito sempre que existe uma gravação e salvação do vídeo de um movimento.

4.5.6 SVM vs. LSTM

Existe uma grande diferença, tanto na gravação como no treino destes dois algoritmos.

Aquando é feita a gravação dos dados para os ficheiros de texto, ambos os algoritmos precisam de ser gravados de maneira diferente. Enquanto que o SVM necessita de uma linha com todos os pontos e a previsão do gesto da mão em linhas separadas, o LSTM necessita que

os frames todos sejam comprimidos em apenas uma linha, assim como o terceiro parâmetro de cada linha necessita de ser o número de frames que foram guardados durante toda a gravação para esse mesmo movimento. O SVM não necessita dessa informação, pois sabe que estão todos separados por linhas, verificando apenas o utilizador e o nome do movimento.
Relativamente ao treino, é onde as diferenças entre cada um deles se destaca mais.

```
model = SVC(C=0.1, kernel='linear', gamma=1, probability=True,verbose=True)
new_X = preprocessing.normalize(new_X, axis=1) # normalize data: Scale input
if phase == "1":
    accuracy, std = run_model_3(model, new_X, y, new_users)
elif phase == "2":
    accuracy, std = run_model_2(model, new_X, y, new_users)
elif phase == "3":
    accuracy, std = run_model_3(model, new_X, y, new_users)
elif phase == "4":
    accuracy, std = run_model_4(model, new_X, y, new_users)
combine = accuracy
result = {
"output": combine, "std": std
}
```

Figura 33 – Treino do modelo usando SVM

Na Figura 33 temos o início do treino usando SVC da libraria sklearn[3] de forma a criar um modelo vazio. De seguida, depois de removido o excesso de informação, 3 parâmetros fixos, 1000 que identificam os pontos todos no frame e por fim os últimos 6 relativos ao conjunto de dados da previsão da mão e das suas probabilidades.
Existem diferentes algoritmos, como é possível verificar na Figura 33, dependendo do número de utilizadores que foram escolhidos.
Foi usada uma validação cruzada com 5 cortes, onde para cada um destes cortes é comparado a informação de teste com a informação de treino e feito o treino deste modelo pelo número de iterações de um conjunto de dados de forma a encontrar o melhor modelo possível.
Finalmente, é demonstrado o grau de precisão de cada uma das 5 iterações.
Todo o processo é completo quando todas as iterações estiverem completas e é feito a média da precisão de todas, assim como o desvio padrão.

```
######## Cross validation
kf = StratifiedKFold(n_splits=5, random_state=None, shuffle=True)
accuracy = []
i = 0
train_index_all = []
test_index_all = []
for train_index, test_index in kf.split(X1,y22):
    train_index_all.append(train_index.tolist())
    test_index_all.append(test_index.tolist())
```

Figura 34 - Validação cruzada do LSTM

[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Ao contrário do SVM, o LSTM é mais simples e utiliza menos características, fazendo com que o treino seja consideravelmente mais rápido.
Apesar de utilizar a validação cruzada, KFold que o SVM usa, LSTM utiliza a StratifiedKFold, verificar na Figura 34, que é uma variação da validação usada pelo SVM.
Usando StratifiedKFold, LSTM, apenas mistura a informação, divide-a em 5 e utiliza cada uma dessas partes para um conjunto de teste.
Pelo contrário o SVM, com o KFold mistura o conjunto de dados de forma aleatória, divide esse conjunto em 5 grupos. Para cada um destes grupos, utiliza um desses grupos como teste e os restantes como um conjunto de dados de treino, tenta ajustar o modelo no conjunto de treino e avaliar o conjunto de teste, por fim, avalia esse conjunto, guarda a informação da sua classificação e descarta.

4.5.7 Possíveis melhorias

Embora tivesse sido planeado, a implementação de um sistema de login com persistência de dados não foi elaborado. Todos os utilizadores deveriam ter um sistema de acesso com a sua própria conta para que os seus dados pudessem ser gravados de forma privativa e com segurança.
Outra ideia seria o uso dos dados de treino de todos os utilizadores de forma anónima, que tivessem aceitado a partilha dos mesmos, de forma a criar um modelo de treino com um maior espectro de palavras e pontos nos diferentes movimentos.

5 Avaliação

5.1 Hipótese

De modo a verificar se o desenvolvimento deste projeto contribui para melhorar a inclusão social e o acesso a novas oportunidades das pessoas mudas, a hipótese do trabalho consiste na convicção de que é possível traduzir língua gestual para língua oral com recurso à visão computacional, contribuindo assim para a inclusão da comunidade surda na sociedade.

Hipótese Nula (H_0) - A utilização da aplicação desenvolvida não contribui para a inclusão da comunidade surdas nem contribui para a oferta de novas oportunidades.

Em que, H_0 significa que em média a utilização da aplicação de tradução não contribui para uma melhor inclusão da comunidade surda.

Hipótese Alternativa (H_1) - A utilização da aplicação desenvolvida facilita a comunicação da comunidade surdas e oferece um leque de novas oportunidades.

Em que, H_1 significa que em média a utilização da aplicação de tradução contribui para uma melhor inclusão da comunidade surda e ofereceu um leque de novas oportunidades facilitando a comunicação entre a comunidade surda e a ouvinte.

5.2 Grandezas de Avaliação

De modo a avaliar o resultado desta aplicação recorreu-se à taxa de erro e à taxa de aceitação. Relativamente à taxa de aceitação, depois da gravação dos gestos/palavras, é realizado o treino dos modelos de classificação, é necessário que tais modelos obtenham um grau de precisão aceitável e acima do definido.
Quando é passado à parte da previsão de palavras, é realizada a taxa de erro, isto é, é imitado o mesmo gesto/palavra uma série definida de vezes e é contabilizada a quantidade de vezes que essa palavra/gesto foi detetada incorretamente.

5.2.1 Taxa de Erro

A grandeza de avaliação optada foi a taxa de erro para avaliar a precisão da aplicação. Depois do modelo de classificação ter sido treinado é necessário avaliar a percentagem de precisão do modelo.
Para isso, é realizado um teste onde a mesma palavra ou frase é repetida uma séria definida de vezes e é contabilizado a quantidade de vezes que o modelo treinado acertou.
Por fim, o número de vezes acertado é dividido pelo número total de vezes imitadas para essa mesma palavra e verifica-se se essa palavra se encontra dentro do limite estabelecido para a taxa de erro.

5.2.2 Comparação de classificadores – Teste de Sinal

Com este método é possível testar vários classificadores e verificar qual ou quais os que têm uma precisão e exatidão melhor comparativamente aos restantes. Com esta análise irá ser mais fácil recomendar algoritmos de classificadores dependendo do utilizador e da sua utilização.

5.2.3 Validação cruzada (k-fold)

Por fim, irá ser utilizada o método k-fold, onde um conjunto de informação está dividido em k-folds, ou seja, pedaços de informação e em cada iteração, um pedaço diferente é testado e os outros são usados para treinar os classificadores.[4]

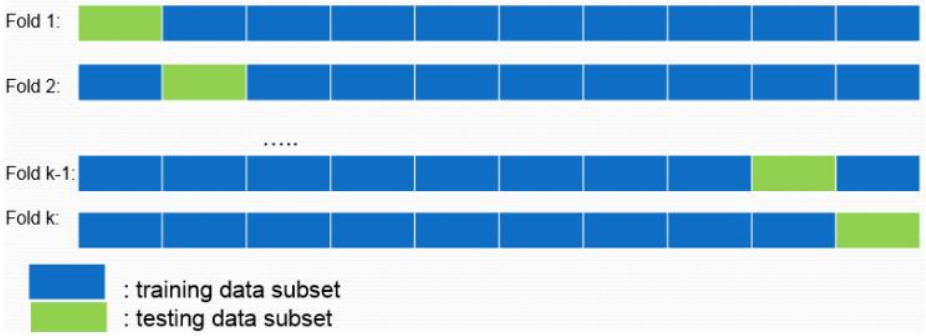

Figura 35 - Exemplo de k-fold

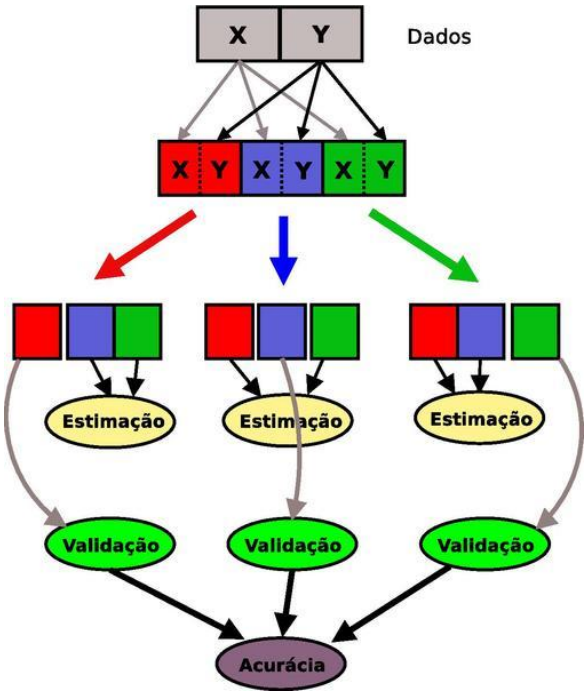

Figura 36 – Outro exemplo de como k-fold é executado

Como é possível verificar na Figura 35 e Figura 36, o k-fold consiste na divisão de um conjunto de dados em k subconjuntos do mesmo tamanho e, a partir daí, um subconjunto é utilizado para teste, enquanto que os k-1 restantes são utilizados para estimação dos parâmetros,

[4] http://www.icmla-conference.org/icmla11/PE_Tutorial.pdf

fazendo-se o cálculo de precisão do modelo. Este processo é realizado k vezes alternando o subconjunto de teste de forma circular.
Ao final das k iterações calcula a precisão sobre os erros encontrados, através de uma equação, obtendo assim uma medida mais confiável do classificador. [5]

5.3 Metodologia de Avaliação

Irá ser definida uma taxa de aceitação mínima para que, cada palavra ou frase seja, correta. De forma a poder afirmar que certo algoritmo se encontra correto é definido um valor que tem de ser atingido, dessa forma é possível encontrar os melhores algoritmos ou os mais adequados para cada tipo de situação.
Os valores a serem analisados fazem parte não só dos resultados depois da realização de treino com os diferentes modelos, como também dos resultados obtidos dos testes de previsão.

5.4 Resultados

De modo avaliar corretamente a análise dos resultados, é apresentado este subcapítulo onde são discutidos os resultados obtidos consoante o número de gestos a serem testados.
Para que exista uma maior coerência e de forma a realizar os testes com a maior parecença possível com a realidade, os testes foram realizados e gravados pelo autor. Desta forma, como não foram usados vídeos da gravação de uma palavra nem realizado por pessoas experientes em língua gestual, os movimentos dos gestos/palavras não são perfeitos o que permite uma melhor análise dos resultados.

5.4.1 Plano Experimental

Cada gesto/movimento foi gravado cinco vezes de um total de 50 palavras/gestos diferentes, identificados na Tabela 15, fazendo um total de 250 gravações (50 palavras/gestos * 5 vezes cada palavra/gesto).

Tabela 15 – Conjunto de palavras gravadas

bom dia	obrigado	Portugal	avô	começar	janeiro	dia	boa noite
email	secundário	avó	já	curso	mês	Porto	ensino
nome	viver	próximo	onde	principal	olá	segunda -feira	terça-feira
quarta-feira	sexta-feira	domingo	fevereiro	escola	março	abril	maio
junho	julho	agosto	setembro	outubro	dezem-bro	azul	Ver-melho
assim	amarelo	verde	castanho	rosa	preto	branco	irmão
irmã	tio	tia	primo				

[5] https://pt.wikipedia.org/wiki/Validação_cruzada

Durante a gravação existiram os diversos cuidados referidos na secção 4.5.3, assim como houve o maior cuidado possível em aprender os movimentos antes de os replicar. De forma a treinar os gestos/palavras foi utilizado o website SpreadTheSign[6] para que os movimentos sejam o mais realistas possíveis.
Cada uma das palavras da Tabela 15 foi gravada cinco vezes, de forma a que o classificador obtenha um maior número de exemplos para cada palavra, simulando a realidade, onde a mesma palavra pode ter certos desvios e os pontos captados serem relativamente diferentes. Contudo, a gravação de cada uma das palavras demora entre 2 e 5 minutos (dependendo da máquina).
Durante duas semanas foram feitas as gravações para as 50 palavras, no entanto, houve a necessidade de acrescentar e/ou fazer alterações na forma como as palavras eram gravadas. Inicialmente a gravação continha o conjunto de dados da configuração de mão exatamente como era recebido como na Figura 37.

bom_dia,Jorge,29,0,244.938,278.305,223.903,338,0.0,[('B', 0.0112323575), ('S', 0.0011349985), ('A', 0.00044186137)],87.29805304062064

O primeiro parâmetro refere-se ao nome do movimento
O segundo parâmetro refere-se ao nome do utilizador
O terceiro parâmetro refere-se ao número de frames que foram captados
O quarto parâmetro refere-se aos vários pontos chave gravados frame a frame
O quinto parametro refere-se ao conjunto de dados relativos às 3 maiores probabilidades da mão se encontrar naquele frame
O quarto e o quinto parâmetro repetem-se até ao final da gravação de cada palavra

Figura 37 – Exemplo da gravação inicial de uma palavra

Devido ao facto de o classificador conseguir ler apenas números, qualquer coisa depois do terceiro parâmetro necessita de ser transformado num *float*.
Para isso, chegou-se à ideia de retirar todos os parênteses e pelicas, bem como transformar as letras em números seguindo o seguinte padrão, A equivale a 001, B a 002 e assim sucessivamente, seguido da sua probabilidade como no seguinte exemplo:

bom_dia,Jorge,29,0,450,0,0,338,223.95273328804703,338,0.0,002,0.0112323575,019,0.0011349985,001,0.00044186137,87.29805304062064,272.4595827282133…

Finalmente, com as 50 palavras gravadas novamente da forma correta, já era possível realizar o treino com os diferentes modelos de classificação: SVM, LSTM e Fully Connected Model.
É necessário realçar que o treino dos modelos LSTM e Fully Connected Model foram relativamente rápidos, visto que ambos são similares, contudo o treino do SVM demorou 4 dias.

5.4.2 Resultados obtidos

Para os 250 exemplos, o modelo que obteve melhores resultados no treino foi o Fully Connected Model com uma média de precisão de 94% e com um desvio padrão de 0.00 (Figura 38). Este modelo de classificação foi também o mais rápido a realizar o treino em relação aos restantes dois.

[6] https://spreadthesign.com/pt.pt/search/

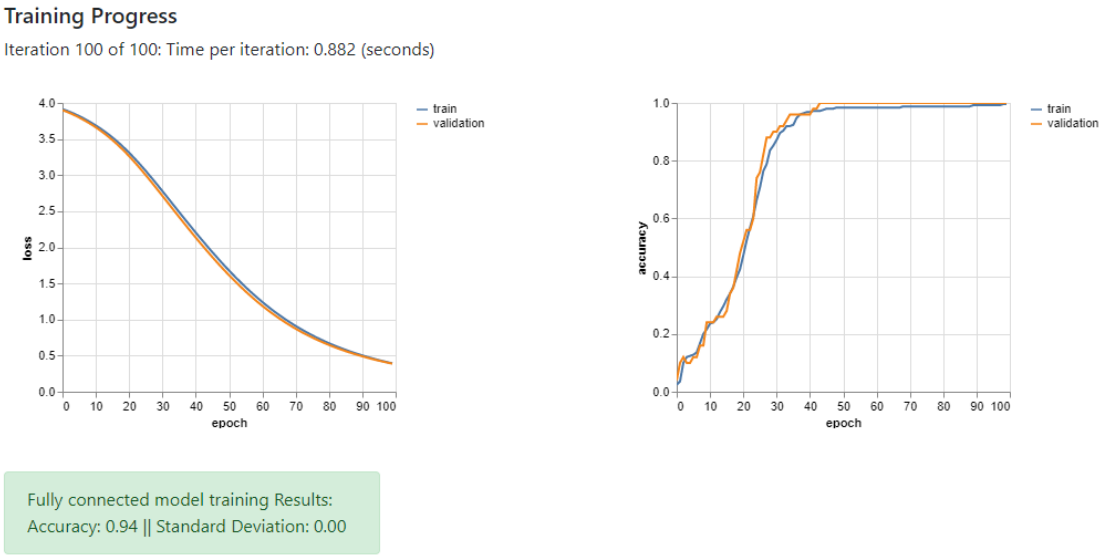

Figura 38 - Resultados do modelo de classificação Fully Connected Model

Quanto ao modelo que mais foi influenciado pela negativa com a adição da configuração de mão foi o LSTM. Como é possível verificar na Figura 39, a precisão do LSTM tem como média de precisão 6% e um desvio padrão de 0 para 100 iterações. Quanto à duração do treino, foi bastante parecida ao modelo FCM, demorou apenas mais alguns minutos

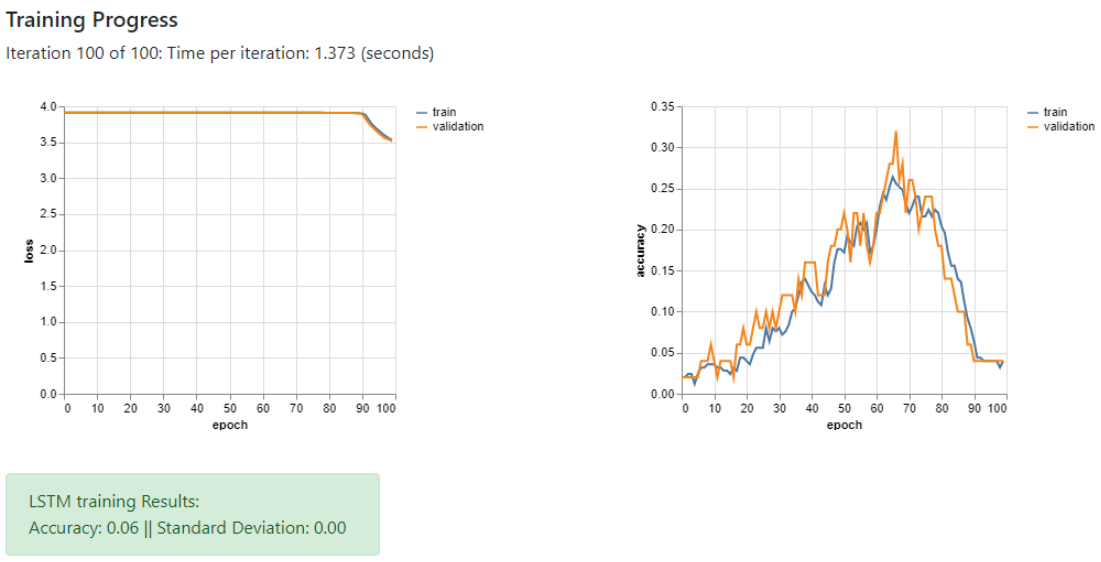

Figura 39 – Resultados do modelo de classificação LSTM

O SVM obteve um muito bom usando a configuração de mão, a precisão aumentou obtendo uma precisão de 90% e um desvio padrão de 0.01 (Figura 40). Em relação à previsão, este foi o algoritmo que conseguir acertar um maior número de vezes, sendo também o classificador mais indicado para fazer a previsão de palavras.

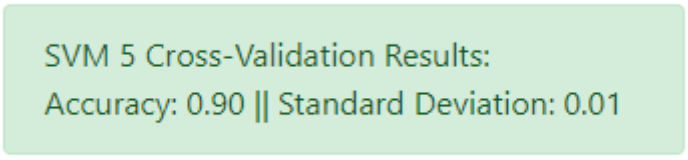

Figura 40 – Resultado do modelo de classificação SVM

O motivo pelo qual o modelo SVM ser preferido em relação ao Fully Connected Model (FCM) embora tenha obtido um resultado ligeiramente inferior no treino, é o facto de que os resultados obtidos na previsão pelo FCM são significativamente inferiores ao SVM.

Finalmente, é possível afirmar que modelo SVM e o FCM beneficiaram do uso da configuração da mão dos frames, enquanto que o modelo LSTM obteve resultados significativamente piores.

Pode-se supor que os resultados são devidos ao facto do modelo SVM ser bastante mais compatível com um maior uso de características aquando a realização do treino do que o LSTM.

Quanto à realização da previsão de gestos, o modelo SVM, como esperado, foi o que conseguiu prever um maior número de palavras com uma precisão maior.

Conclui-se assim que, apesar do aumento significativo no tempo para realizar o treino, o modelo de classificação SVM obteve melhores resultados em ambos os testes e beneficia, ao contrário do modelo de classificação LSTM, do uso de mais características, neste caso o uso da configuração de mão em cada frame.

5.4.3 Descobertas Finais

Depois de os valores obtidos para o modelo de classificação serem bastante baixos, foi feita uma pesquisa de forma a tentar encontrar uma solução com melhores resultados.

Foi encontrada uma resposta a uma pergunta idêntica que aconselha o uso do formato de gravação usado para o modelo de classificação do SVM.

Os resultados obtidos foram os das Figura 41 e Figura 42.

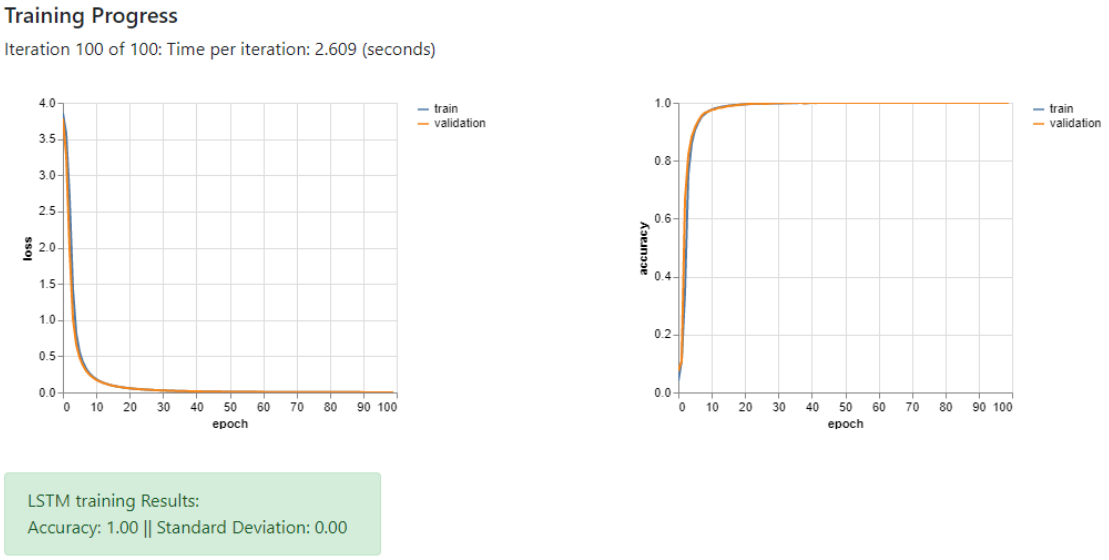

Figura 41 - Resultado do modelo LSTM usando o formato de gravação SVM

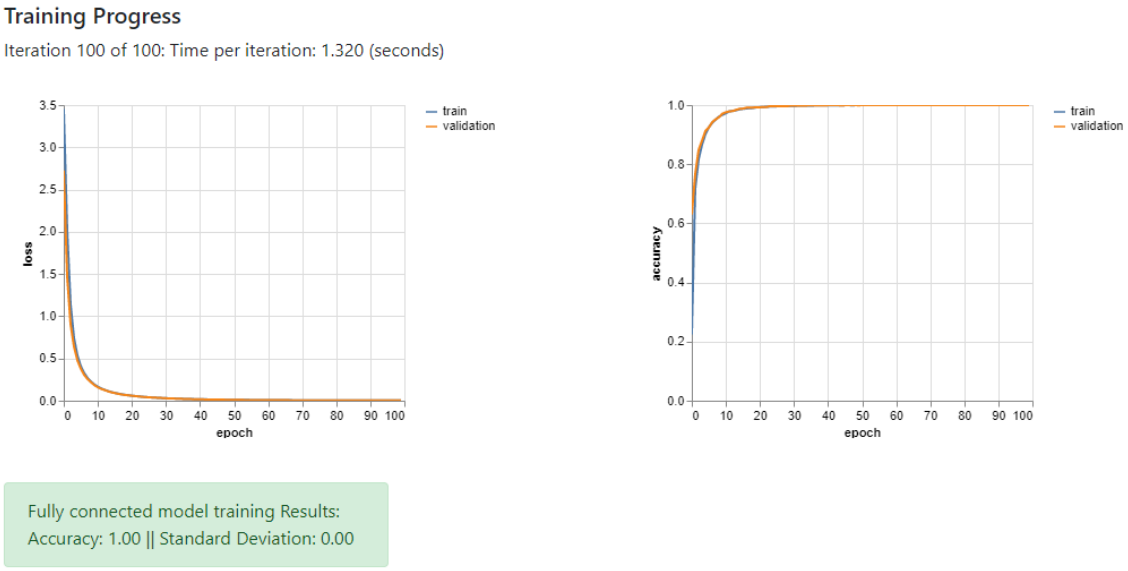

Figura 42 – Resultado do modelo FCM usando o formato de gravação SVM

Apesar dos resultados obtidos serem perfeitos, o modelo pode-se encontrar sobre ajustado e ter dificuldades na generalização, isto é, quando são introduzidos novos movimentos na previsão, este modelo de classificação treinado pode não ser capaz de detetá-los devido à incapacidade de detetar novos exemplos nunca antes vistos.

6 Conclusão

Neste capítulo são apresentadas as conclusões retiradas do projeto. São relatados os objetivos alcançados e ainda uma secção que demonstra as limitações encontradas ao longo do percurso, assim como como melhorarias futuras para o projeto.

6.1 Principais conclusões

De acordo com os objetivos principais desta dissertação, foi realizada uma investigação sobre a tradução de língua comum para língua gestual no capítulo 2. Neste capítulo, são apresentadas diferentes técnicas para o processamento de vídeo e tracking de objetos, um estudo de como fazer a representação dos dados obtidos para machine learning e as possíveis tecnologias a serem utilizadas na implementação da solução.

No capítulo 3, é realizada uma análise de valor onde é definida a proposta de valor e o valor da solução para o cliente, bem como os benefícios e sacrifícios. Através do método AHP, foi realizada uma comparação com o tempo de desenvolvimento, o custo monetário e a conveniência das diferentes ideias. Foi também elaborado um modelo CANVAS que descreve a organização e a sua definição de valor.

No capítulo 4 é apresentada toda a análise e implementação da solução, contando com a apresentação de alguns diagramas para melhor compreensão da aplicação, bem como uma melhor visão da implementação da solução e dos seus componentes.

Por fim, no capítulo de avaliação da solução, são apresentados obtidos nos testes e concluiu-se que os resultados com o uso da classificação da mão foram positivos num dos modelos apresentando um melhor resultado com o uso deste não só no treino, mas também na previsão.

6.2 Objetivos alcançados

O objetivo principal deste projeto consiste na implementação de uma aplicação capaz de traduzir, através de um vídeo, palavras ou frases de língua gestual para língua oral. Tal objetivo foi alcançado com sucesso, houve também um acrescento para realizar a classificação de mão em cada frame e verificar se os modelos beneficiavam de tal característica.
De uma maneira geral, todos os objetivos planeados foram atingidos exceto a implementação de um sistema de login (como referido na secção 4.5.7).

7 Bibliografia

[1] OpenCV, “About,” OpenCV, 2019. [Online]. Available: https://opencv.org/about/. [Acedido em 27 janeiro 2020].

[2] J. Perkel, “When it comes to reproducible science, Git is code for success,” nature index, 11 Junho 2018. [Online]. Available: https://www.natureindex.com/news-blog/when-it-comes-to-reproducible-science-git-is-code-for-success. [Acedido em 10 Dezembro 2019].

[3] “Bitbucket: What is Bitbucket?,” Atlassian, 7 Março 2018. [Online]. Available: https://confluence.atlassian.com/confeval/development-tools-evaluator-resources/bitbucket/bitbucket-what-is-bitbucket. [Acedido em 10 Dezembro 2019].

[4] "Open Pose," Awesome Open Source, 26 abril 2020. [Online]. Available: https://awesomeopensource.com/project/CMU-Perceptual-Computing-Lab/openpose. [Accessed 30 abril 2020].

[5] Z. Cao, T. Simon, S. E. Wei e Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” IEEE, California, 2019.

[6] A. Solano, "Human pose estimation using OpenPose with TensorFlow (Part 1)," Ar Vr Journey, 2 outubro 2017. [Online]. Available: https://arvrjourney.com/human-pose-estimation-using-openpose-with-tensorflow-part-1-7dd4ca5c8027. [Accessed 30 abril 2020].

[7] Heroku, "What is heroku," Heroku, [Online]. Available: https://www.heroku.com/what#. [Accessed 20 maio 2020].

[8] Heroku, "About Us," Heroku, [Online]. Available: https://www.heroku.com/about. [Accessed 20 maio 2020].

[9] K. Rusev, "What is Heroku Used For?," MentorMate, 15 maio 2018. [Online]. Available: https://mentormate.com/blog/what-is-heroku-used-for-cloud-development/. [Accessed 20 maio 2020].

[10] R. Gandhi, "Support Vector Machine — Introduction to Machine Learning Algorithms," Towards Data Science , 7 junho 2018. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. [Accessed 15 maio 2020].

[11] Editores da Wikipedia, “Máquina de vetores de suporte,” Wikipedia, 7 janeiro 2020. [Online]. Available: https://pt.wikipedia.org/wiki/Máquina_de_vetores_de_suporte. [Acedido em 24 abril 2020].

[12] Y. Bengio, P. Simard and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Transactions on Neural Networks,* vol. 5, no. 2, 1994.

[13] "Long short-term memory," Wikipedia, 20 fevereiro 2020. [Online]. Available: https://en.wikipedia.org/wiki/Long_short-term_memory. [Accessed 20 abril 2020].

[14] J. Brownlee, "A Gentle Introduction to Long Short-Term Memory Networks by the Experts," Machine Learning Mastery, 20 fevereiro 2020. [Online]. Available:

https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/. [Accessed 25 abril 2020].

[15] S. Ojha e S. Sakhare, “Image processing techniques for object tracking in video surveillance,” em *Institute of Electrical and Electronics Engineers Inc.*, Pune, India, 2015.

[16] J. Shen, D. Yu, L. Deng and X. Dong, "Fast Online Tracking with Detection Refinement," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 1, pp. 162-173, 2018.

[17] S. R. Maiya, “DeepSORT: Deep Learning to Track Custom Objects in a Video,” Nanonets, 19 julho 2019. [Online]. Available: https://nanonets.com/blog/object-tracking-deepsort/#kalman-filters. [Acedido em 1 dezembro 2019].

[18] S. Mallick, "GOTURN : Deep Learning based Object Tracking | Learn OpenCV," Learn OpenCV, 22 Julho 2018. [Online]. Available: https://www.learnopencv.com/goturn-deep-learning-based-object-tracking/. [Accessed 2 Dezembro 2019].

[19] D. Jurić, “Object Tracking: Kalman Filter with Ease,” CodeProject, 15 janeiro 2015. [Online]. Available: https://www.codeproject.com/articles/865935/object-tracking-kalman-filter-with-ease. [Acedido em 10 dezembro 2019].

[20] MissingLink , "Object Tracking in Deep Learning," MissingLink, [Online]. Available: https://missinglink.ai/guides/computer-vision/object-tracking-deep-learning/. [Accessed 4 dezembro 2019].

[21] Z. G. L. O. F. R. B. U. Alex Bewley, “SIMPLE ONLINE AND REALTIME TRACKING,” Cornell University, Nova Iorque, 2017.

[22] Abewley, “SORT,” GitHub, 17 Junho 2018. [Online]. Available: https://github.com/abewley/sort. [Acedido em 10 Dezembro 2019].

[23] P. Dwivedi, "People Tracking using Deep Learning," towardsdatascience, 7 fevereiro 2019. [Online]. Available: https://towardsdatascience.com/people-tracking-using-deep-learning-5c90d43774be. [Accessed 10 dezembro 2019].

[24] A. Sagar, “Pedestrian Tracking in Real-Time Using YOLOv3,” towardsdatascience, 28 julho 2019. [Online]. Available: https://towardsdatascience.com/pedestrian-tracking-in-real-time-using-yolov3-33439125efdf. [Acedido em 10 dezembro 2019].

[25] N. N. Bhat, Y. V. Venkatesh, U. Karn e D. Vig, “Understanding of a Convolutional Neural Network,” em *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013*, Antalya, Turkey, 2013.

[26] U. Karn, “An Intuitive Explanation of Convolutional Neural Networks,” the data science blog, 11 Agosto 2016. [Online]. Available: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/. [Acedido em 9 Janeiro 2020].

[27] B. H. Hyeonseob Nam, "Learning Multi-Domain Convolutional Neural Networks for Visual Tracking," 27 outubro 2015. [Online]. Available: https://www.arxiv-vanity.com/papers/1510.07945. [Accessed 9 dezembro 2019].

[28] M. Venkatachalam, "Recurrent Neural Networks," towards data science, 1 março 2019. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce. [Accessed 12 fevereiro 2020].

[29] M. Nguyen, "Illustrated Guide to Recurrent Neural Networks," towards data science, 20 setembro 2018. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9. [Accessed 12 fevereiro 2020].

[30] M. Harvey, “Continuous video classification with TensorFlow, Inception and Recurrent Nets,” hackernoon, [Online]. Available: https://hackernoon.com/continuous-video-classification-with-tensorflow-inception-and-recurrent-nets-250ba9ff6b85. [Acedido em 11 fevereiro 2020].

[31] M. Karchevsky, "Machine Learning Video Analysis: Identifying Fish," toptal, 2018. [Online]. Available: https://www.toptal.com/machine-learning/machine-learning-video-analysis. [Accessed 11 fevereiro 2020].

[32] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Google, Inc., France, 2017.

[33] J. Nicholas, tutorialspoint, 12 Março 2019. [Online]. Available: https://www.tutorialspoint.com/program-to-extract-frames-using-opencv-in-python. [Acedido em 27 01 2020].

[34] A. Saha, “Read, Write and Display a video using OpenCV (C++/ Python),” Learn OpenCV, 5 Junho 2017. [Online]. Available: https://www.learnopencv.com/read-write-and-display-a-video-using-opencv-cpp-python/. [Acedido em 27 janeiro 2020].

[35] R. Edgar, "K-fold cross-validation," userach v11, [Online]. Available: https://drive5.com/usearch/manual/kfold.html. [Accessed 29 Abril 2020].

[36] J. Browlee, "A Gentle Introduction to k-fold Cross-Validation," Machine Learning Mastery, 23 maio 2018. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/. [Accessed 29 abril 2020].

[37] S. Tardif, “Introducing: Bitbucket Cards,” 22 Maio 2012. [Online]. Available: https://blog.bitbucket.org/2012/05/22/introducing-bitbucket-cards/.

[38] M. Shishira, "DeepSORT: Deep Learning to Track Custom Objects in a Video," Nanonets, Julho 2019. [Online]. Available: https://nanonets.com/blog/object-tracking-deepsort/. [Accessed 10 Dezembro 2019].

[39] A. Twin, "Value Proposition," Investopedia, 25 junho 2019. [Online]. Available: https://www.investopedia.com/terms/v/valueproposition.asp. [Accessed 13 fevereiro 2020].

[40] Business Analyst training, "What is FURPS+?," Business Analyst training in Hyderabad, 5 agosto 2014. [Online]. Available: https://businessanalysttraininghyderabad.wordpress.com/2014/08/05/what-is-furps/. [Accessed 18 fevereiro 2020].

[41] Utilizadores da Wikipedia, “FURPS,” wikipedia, 10 dezembro 2017. [Online]. Available: https://pt.wikipedia.org/wiki/FURPS. [Acedido em 18 fevereiro 2020].

8 Anexos

Results for SVM classifier (C = 1, gamma = 1, kernel = linear)

Case 1: **One user to train and the same to test** 5 cross-validation with 40 samples from user 1 as training set and the rest 10 of the same user as testing set each time.

Tabela 16 - Caso 1 - Um utilizador treina o mesmo teste

Experiment	Hands	Shoulder	Elbow	Wrist	Accuracy (mean)	Standard Deviation
1.	Both	Y	Y	Y	0.7999	0.1095
2.	Both	Y	Y	N	0.78	0.0400
3.	Both	Y	N	Y	0.78	0.1720
4.	Both	Y	N	N	0.6	0.1414
5.	Both	N	Y	Y	0.8	0.0
6.	Both	N	Y	N	0.74	0.1356
7.	Both	N	N	Y	0.8	0.0800
8.	Both	N	N	N	-	-
9.	Right	Y	Y	Y	0.88	0.0748
10.	Right	Y	Y	N	0.84	0.0800
11.	Right	Y	N	Y	0.84	0.1019
12.	Right	Y	N	N	0.6	0.1414
13.	Right	N	Y	Y	0.9	0.0894
14.	Right	N	Y	N	0.78	0.0979
15.	Right	N	N	Y	0.9399	0.0489
16.	Right	N	N	N	-	-

Case 2: **One user to train and a different one to test.** 5 cross-validation with 40 samples from user 1 as training set and all the 50 samples from the last user as testing set each time.

Tabela 17 - Caso 2 - Um utilizador treina e um diferente realiza o teste

Experiment	Hands	Shoulder	Elbow	Wrist	Accuracy (mean)	Standard Deviation
1.	Both	Y	Y	Y	0.688	0.0271
2.	Both	Y	Y	N	0.584	0.0233
3.	Both	Y	N	Y	0.624	0.0496
4.	Both	Y	N	N	0.46	0.0593
5.	Both	N	Y	Y	0.668	0.0391
6.	Both	N	Y	N	0.552	0.0411
7.	Both	N	N	Y	0.648	0.0744
8.	Both	N	N	N	-	-
9.	Right	Y	Y	Y	0.696	0.0265
10.	Right	Y	Y	N	0.576	0.0233
11.	Right	Y	N	Y	0.68	0.0399
12.	Right	Y	N	N	0.419	0.0334
13.	Right	N	Y	Y	0.708	0.0203
14.	Right	N	Y	N	0.592	0.0271
15.	Right	N	N	Y	0.759	0.0565
16.	Right	N	N	N	-	-

Case 3: **Several users to train and one of them to test.** 5 cross-validation with 120 samples from all the 3 users as training set and 10 samples from one of them as testing set each time.

Tabela 18 – Caso 3 – Vários utilizadores treinam e um deles realiza o teste

Experiment	Hands	Shoulder	Elbow	Wrist	Accuracy (mean)	Standard Deviation
1.	Both	Y	Y	Y	0.8	0.1264
2.	Both	Y	Y	N	0.779	0.0400
3.	Both	Y	N	Y	0.78	0.0748
4.	Both	Y	N	N	0.72	0.1166
5.	Both	N	Y	Y	0.76	0.1356
6.	Both	N	Y	N	0.7	0.1673
7.	Both	N	N	Y	0.84	0.1356
8.	Both	N	N	N	-	-
9.	Right	Y	Y	Y	0.76	0.0489
10.	Right	Y	Y	N	0.7	0.0894
11.	Right	Y	N	Y	0.74	0.1019
12.	Right	Y	N	N	0.48	0.1720
13.	Right	N	Y	Y	0.8	0.1095
14.	Right	N	Y	N	0.6	0.0894
15.	Right	N	N	Y	0.76	0.1356
16.	Right	N	N	N	-	-

Case 4: **Several users to train and a different one to test.** 5 cross-validation with 80 samples from 2 users as training set and all the 50 samples from the last user as testing set each time.

Tabela 19 – Caso 4 – Vários utilizadores treinam e diferentes utilizadores realizam o teste

Experiment	Hands	Shoulder	Elbow	Wrist	Accuracy (mean)	Standard Deviation
1.	Both	Y	Y	Y	0.612	0.0324
2.	Both	Y	Y	N	0.564	0.0149
3.	Both	Y	N	Y	0.6679	0.0466
4.	Both	Y	N	N	0.4480	0.0652
5.	Both	N	Y	Y	0.648	0.0203
6.	Both	N	Y	N	0.488	0.0324
7.	Both	N	N	Y	0.64	0.0632
8.	Both	N	N	N	-	-
9.	Right	Y	Y	Y	0.664	0.0496
10.	Right	Y	Y	N	0.576	0.0344
11.	Right	Y	N	Y	0.624	0.0445
12.	Right	Y	N	N	0.44	0.0593
13.	Right	N	Y	Y	0.628	0.0411
14.	Right	N	Y	N	0.508	0.0411
15.	Right	N	N	Y	0.636	0.0496
16.	Right	N	N	N	-	-

Conclusion:

Maximum results (for case 1 but also for the other cases are good) were observed keeping only the **right** hand with the **wrist** points.

Right hand (wrist)	Accuracy	Standard Deviation
Case 1	0.9399	0.0489
Case 2	0.759	0.0565
Case 3	0.76	0.1356
Case 4	0.636	0.0496

Other good are those from the **right** hand with the points of **wrist** and **elbow**:

Right hand (wrist)	Accuracy	Standard Deviation
Case 1	0.9	0.0894
Case 2	0.708	0.0203
Case 3	0.8	0.1095
Case 4	0.628	0.0411