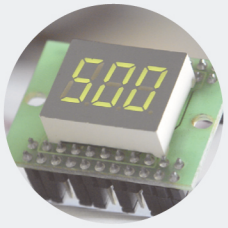
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES






Sistema de deteção e tracking de faces para identificação de características humanas (género, idade, emoções)

BRUNO ANDRÉ GOMES RODRIGUES
novembro de 2018

POLITÉCNICO
DO PORTO

Sistema de deteção e tracking de faces para identificação de características humanas (género, idade, emoções)

Bruno André Gomes Rodrigues
Nº 1110403

Mestrado em Engenharia Eletrotécnica e de Computadores
Área de Especialização de Sistemas Autónomos
Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2018

Dissertação, para satisfação parcial dos requisitos do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Bruno André Gomes Rodrigues

Nº 1110403

Orientador: André Miguel Pinheiro Dias

Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Sistemas Autónomos

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

22 de Novembro de 2018

Agradecimentos

A realização desta dissertação não seria possível sem o apoio e contributo de várias pessoas. A todos eles deixo o meu agradecimento profundo.

Em primeiro lugar, queria agradecer ao meu orientador, Engº André Dias por me ter proporcionado esta oportunidade e pelo suporte prestado ao longo deste percurso.

A todos os membros do laboratório de sistemas autónomos (LSA), por todo o conhecimento transmitido, ajuda e companheirismo.

A todos os meus amigos de longa data pela amizade e companheirismo demonstrado ao longo destes anos.

A minha namorada, Marta Norte, pelo apoio, carinho, paciência e incentivo durante a elaboração desta dissertação.

Por fim, um agradecimento à minha família, em especial aos meus pais, pela educação que me proporcionaram e pelo apoio, incentivo e esforços efetuados ao longo da minha formação académica.

Ao meu irmão ,Engº Adriano Rodrigues pelo incentivo e companheirismo demonstrado ao longo deste processo.

Esta página foi intencionalmente deixada em branco.

Resumo

Atualmente a robótica está cada vez mais presente no dia-a-dia do Homem e os robôs estão cada vez mais a auxiliar o homem. Em consequência disto, a necessidade de interação Homem-máquina está a aumentar e a tornar-se mais complexa. O avanço da tecnologia e das técnicas usadas tem permitido ao robô um avanço no método de interação com o Homem. Através de uma imagem é possível extrair bastantes características de uma cara e determinar aspetos importantes que irão ter impacto no modo de interação com o Homem.

Nesta dissertação, propõe-se abordar o tema de extração de características de uma cara humana através de uma imagem e deste modo permitir ao robô ter uma melhor perceção do Humano. Nesse sentido, no trabalho desenvolvido foram analisadas um conjunto de conceitos e técnicas usadas em processamento de imagem e redes neuronais de modo a permitir o desenvolvimento de um sistema de interação com o Homem sem recurso a soluções clássicas como *Interfaces* gráficas ou botões de resposta.

O sistema deverá ser capaz de extrair características de uma cara e determinar a idade, o género e os estado emocional, utilizando apenas com imagens obtidas de uma câmara de espetro visível. A extração das características e a determinação dos dados definidos anteriormente será efetuado com recurso a redes neuronais. A analise permitiu identificar alguns artigos e trabalhos relevantes e marcantes que extraíssem características de uma cara e determinassem os diferentes dados.

A solução desenvolvida passou por várias fazes de implementação, em que cada uma das fases foram testadas as diferentes redes utilizadas em tempo real, com várias pessoas e em vários meios ambientes. Nesta tese também se estudou o impacto do sistema numa unidade de processamento central e identificou algumas linhas de trabalho que poderão ser integradas na solução atual tendo como impacto principal a melhoria da performance em tempo-real.

Palavras-Chave: Redes neurais, Inteligência artificial, Características faciais, Facetraking

Abstract

In the today's world , robotics is more and more in the daily life of the human kind and robots are more and more helping humans in a big variety of taks. Consequently, the need of Human-machine interaction is growing and becoming more complex. The evolution of technology and the evolution of the techniques used has allowed the robots to improve the way they interact with humans. With a single image is possible to extract many features of a face and determinate important data that will have impact in the way robots interacts with a human.

In this thesis is proposed to analyze the problem of extraction of a face's features using only a image and with this make possible to a robot have a better perception of the human. In the work developed were analyzed a set of concepts, techniques used in image processing and neural networks, in order to develop a interact system that allows robot to interact with humans without using classic solutions like graphic interfaces or buttons.

The system should be capable of extract features of a face and determinate age, gender and emotional state, using only images obtained from a visible spectral camera.The features extraction and the determination of the data previously determined will be done with neural networks. The study allowed to identify some articles and relevant works that extract the features and determinate the data defined previously.

The developed solution passed for many phases of implementation, in each phase were done testes in real time, with different people and in different environments. The tests allow us to see the impact of this system in the central processing unit and identify some lines of work that can be integrated in the actual solution with impact in the performance in real-time .

Key words: Neural networks, Artificial intelligence , Facial features, Facetraking

Conteúdo

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

Esta página foi intencionalmente deixada em branco.

Lista de Siglas e Acrónimos

CNN Convolutional Neural Network

CPU Central Process Unit

D-CNN Deep Convolution Neural Network

DNN Deep Neural Network

FC Fully Connected

HMI Human Machine Interface

LRN Local Response Normalization

MSE Mean Squared Error

NCS Neural Compute Stick

NN Neural Network

RGB Red Green Blue

RNN Recurrent Neural Network

ROS Robotic Operating System

USB Universal Serial Bus

VGG Visual Geometry Group

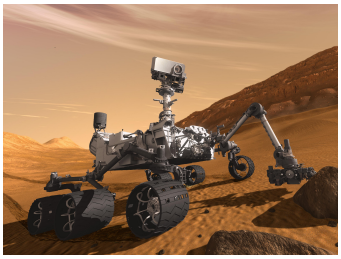
VPU Visual Process Unit

Esta página foi intencionalmente deixada em branco.

Capítulo 1

Introdução

Atualmente, a robótica está cada vez mais integrada na sociedade humana e o seu papel tem sido cada vez mais relevante. Os robôs têm desempenhado tarefas mais complexas e tornaram-se capazes de agir e tomar decisões nos mais variados contextos. Nos últimos anos temos assistido à integração dos robôs na industria (produção, armazéns) , no auxilio e na prestação de serviços a humanos e na investigação/exploração do espaço e outros ambientes bastante adversos (figura 1.1).



(a) Robô da NASA *Opportunity* [1]

(b) Exemplo de um robô prestador de serviços [2]

(c) Exemplo de um robô na Industria [3]

Figura 1.1: Exemplos de robôs que interagem com Homem

As tarefas realizadas por estes robôs exigem que este sejam autónomos e capazes de tomar decisões. Os robôs tem que analisar o ambiente externo e atuar de acordo com este, caso contrário podem não conseguir atingir o seu objetivo e até causar danos ao vários elementos do meio externo. Assim, o modo de operação do robô é uma área bastante importante e tem um grande impacto no seu sucesso. A leitura feita do meio ambiente é um dos aspetos mais relevantes e a informação extraída também. A capacidade de um robô ser capaz de ler características especificas dos elementos do meio é bastante importante, esta capacidade ganha uma atenção especial quando o robô interage com humanos. Os robôs que prestam serviços e interagem com humanos necessitam de con-

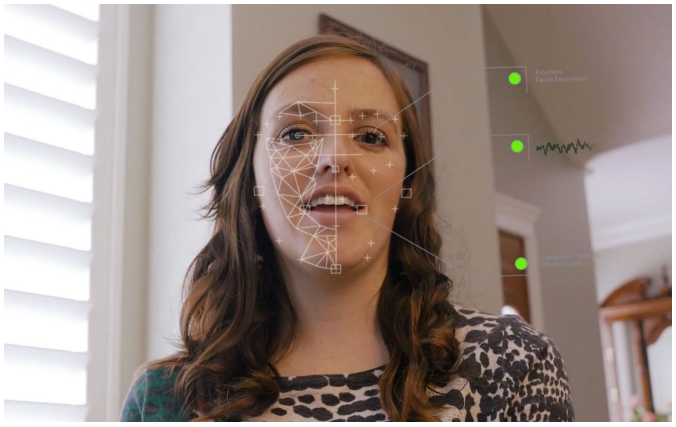


Figura 1.2: Perceção de um cara por parte de um robô. [4]

seguir fazer uma leitura sobre o comportamento do humano e adequar a sua abordagem. A *Interface* que liga o Homem à máquina tem vindo a evoluir com esta necessidade e o modo como a maquina percebe o humano também. A tecnologia permite ao robô melhorar a *Interface* e melhorar a sua interação com o Homem.

Atualmente existem ferramentas que permitem à maquina ser capaz de perceber características tais como idade, género ou até o estado emocional apenas com uma imagem da cara. Será nessa vertente que iremos explorar o tópico, sempre com o objetivo futuro dessa integração permitir melhorar a relação(*Interface*) homem-máquina.

Nesta dissertação propõe-se então abordar-se o problema de ser capaz de, através de uma imagem, extrair informação válida para que o robô possa interagir com o ser humano. Nesse sentido, o trabalho irá endereçar o desenvolvimento de um sistema de perceção, baseado em técnicas de *Deep Learning*, que permitam identificar características como a idade e o género, assim como o estado emocional. A abordagem irá permitir de futuro integrar esse conhecimento no comportamento de um sistema autónomo em diferentes situações do dia-a-dia onde poderá existir uma maior interação homem-máquina.

Figura 1.3: Robô Asimo a prestar um serviço. [5]

1.1 Motivação

No laboratório de sistemas autónomos do Instituto Superior de Engenharia do Porto existem vários projetos na área da robótica. Todo este conhecimento cria um bom ambiente para o desenvolvimento de novos métodos e de novos sistemas para serem integrados em robôs. Deste modo, a principal motivação consiste na exploração de técnicas de *Deep Learning* aplicadas à identificação de características humanas e estado emocional, e que permita de futuro o laboratório poder endereçar áreas de robótica com uma forte componente de interação com o ser humano. Abrindo portas para que os robôs possam ser integrados em ambientes onde usualmente não são vistos. No futuro os poderão ser integrados em ambientes hospitalares, edifícios empresariais, cafés e realizar tarefas tais como receber e dar indicações a pessoas ou até realizar tarefas mais duras e indesejáveis.

A utilização dos robôs para auxilio e ajuda do homem poderá estar a evoluir para um novo patamar. Os robôs poderão estar no futuro a executar tarefas mais delicadas e mais complexas. No futuro poderemos estar a ver uma equipa de robôs num supermercado a indicar o local de um determinado produto e a carregar as compras, num edifício empresarial a receber as pessoas e a leva-las para o local onde são esperadas. Um dia, podemos conseguir integrar os robôs num cenário ainda mais delicado, poderá ser possível ver uma equipa de robôs em hospitais a indicar o lugar que pretendemos ir, transportar e acomodar pacientes e será possível ter uma equipa de robôs disponível 24 horas sobre

24 horas para auxiliar numa situação de emergência.

(a) Exemplo de um robô num supermercado [6]

(b) Exemplo de um robô a auxiliar um paciente [7]

Figura 1.4: Exemplos de robôs que interagem com Homem

1.2 Objetivos

A dissertação endereça o problema de reconhecimento de características e identificação parâmetros na face humana utilizando como recurso uma câmara de espetro visível. O trabalho desenvolvido tem como objetivo melhorar a interação homem-máquina e permitir que um robô seja capaz de identificar parâmetros da cara de uma pessoa e assim adaptar o seu modo de interação. É então necessário que estes objetivos sejam realizados em tempo-real de modo a que o robô seja capaz de adaptar o seu comportamento/interação. Deste modo, o desenvolvimento do projeto implica a concretização dos seguintes objetivos:

- Identificar várias faces numa imagem a uma distância de 4/5 metros;
- Determinar a idade, o género e a emoção de várias pessoas numa imagem;
- Executar todo o processamento em tempo-real, de modo a permitir uma interação contínua com o ser humano.
- Efetuar *Tracking* a cada umas das pessoas na imagem, atribuindo-lhes um ID;
- Implementar e validar a aplicação desenvolvida em CPU e VPU.

1.3 Estrutura da tese

No capitulo 2 são descritos e analisados um conjunto de artigos que foram considerados importantes para deteção de rostos e na determinação de parâmetros importantes como o género, idade e estado emocional. Por fim são analisados algumas técnicas desenvolvidas com o mesmo principio que pretendemos endereçar na dissertação.

No capitulo 3 é descrito um conjunto de conceitos base sobre inteligência artificial, algumas técnicas utilizadas e algumas redes neuronais que têm a mesma abordagem ao problema proposto na dissertação.

No capitulo 4 é descrito qual deverá ser a arquitetura do sistema, as técnicas que pretendemos implementar.

No capitulo 5 irá ser feito a descrição detalha de como foi a implementação de cada bloco e os resultados obtidos para cada bloco e do sistema geral. A conclusão e o trabalho futuro irão ser discutidos no capitulo 6.

Esta página foi intencionalmente deixada em branco.

Capítulo 2

Estado da Arte

Neste capitulo são descritos e analisadas um conjunto de projetos e soluções comerciais que estão relacionadas com o problema descrito no capitulo anterior. Estes projetos e soluções foram analisados e o seu estudo foi relevante para o desenvolvimento do projeto final desta tese de mestrado.

2.1 Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks

O projeto desenvolvido por Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, e Yu Qiao no artigo [8] tem como objetivo criar uma rede capaz de detetar faces em imagens. Este rede deverá ser robusta ou seja, ser capaz de identificar as faces independentemente de fatores externos como a luz. O problema surge pois uma das bibliotecas mais usadas ,*Haar Cascades* do *OpenCV*, é muito vulnerável a fatores externos e não consegue identificar faces a longas distâncias (apenas consegue até 2/3 metros). O método proposto por esta equipa passa por 3 processos. O primeiro processo descobre possíveis candidatos a face e a sua janela na imagem. A rede que efetua este processo é chamada de P-Net. O segundo processo elimina possíveis falsos candidatos e a rede que realiza este processo é chamada de R-Net. O ultimo processo é detetar pequenas regiões na cara com mais detalhe, esta ultima camada deteta posições especificas da face (olhos, nariz e boca) e é chamada de O-Net. A figura 2.1 mostra um esquema do método proposto.

Cada um dos processos apresentados anteriormente é realizado recorrendo a um rede neuronal. As redes utilizadas estão representadas na figura 2.2 e são redes do tipo VGG (Visual Geometry Group). A rede VGG é um tipo de rede neuronal utilizada para classificar imagens, no capitulo é explicado com melhor detalhe o funcionamento da

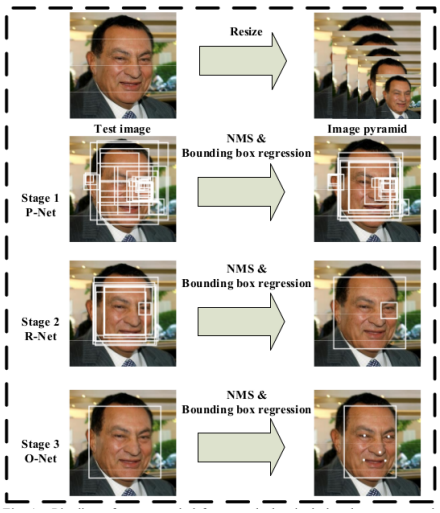


Figura 2.1: Método proposto no artigo *Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks.* [8]

mesma.

Nesta figura, o "Conv" indica uma convulsão e o "MP" indica um *Max polling*. A rede desenvolvida para o processo R-Net tem como *Output* a posição da cara. A rede desenvolvida para o processo O-Net tem como *Output* a posição especifica de elementos da face como os olhos, o nariz e os dois extremos da boca.

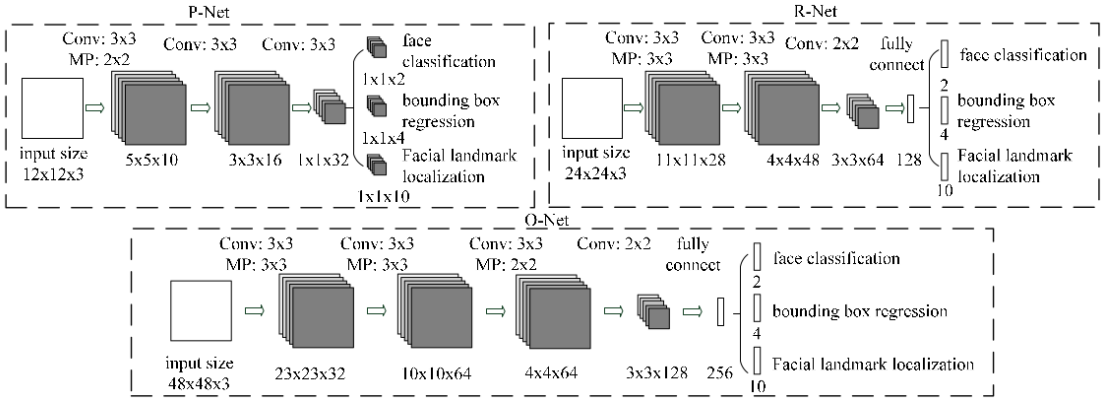


Figura 2.2: Arquitetura das redes propostas no artigo *Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks.*[8]

Os resultados apresentados indicam níveis de precisão que rondam os 95% na deteção da face. As posições especificas dos elementos da face apresentam um erro médio de 6.9%. Os testes realizados pelo autor a este sistema permitiram verificar que a deteção da face

pode ir até uma distância de 10 metros, dependo do tipo de câmara/resolução a ser utilizada.

2.2 Emotion Recognition using Deep Convolutional Neural Networks

Este trabalho foi desenvolvido pelos autores, Enrique Correa, Arnoud Jonker, Michaël Ozo, Rob Stolk [9]. Esta equipa questionou-se sobre como seria possível uma rede neuronal artificial ser capaz de interpretar expressões faciais num humano e deste modo descobrir qual a emoção que seria percetível. Para a classificação das imagens (rostos das pessoas) foram selecionadas sete emoções que são: feliz, triste, neutro, surpreso, nervoso, desgosto e com medo. No seu trabalho eles estudaram um conjunto de redes neuronais, de modo a conseguirem perceber quais as redes que seriam mais adequadas para este tipo de problema. A análise das redes teve em consideração aspetos como, as técnicas utilizadas e o tamanho das redes (número de camadas) e deste modo foram selecionadas três redes para se estudar o desempenho das mesmas. É importante mencionar que as 3 redes escolhidas são redes VGG e apenas diferem em algumas camadas. Em seguida foram recolhidas imagens com caras de pessoas e foi feito a classificação das mesmas segundo as sete emoções mencionadas. É importante mencionar que das imagens recolhidas e classificadas, existem mais imagens com caras de pessoas no estado feliz e neutro. Estas imagens e as suas respetivas classificações foram utilizadas para treinar três redes.

Depois de treinar as três redes, e analisada a precisão obtida em cada um das redes, estes engenheiro concluíram que a melhor rede teria a arquitetura visível na figura 2.3. Analisando a precisão obtida, estado a estado, pode-se concluir que a rede é mais precisa no estado feliz e neutro. Isto deve-se ao facto de estes dois estados serem aqueles com mais imagens no conjunto de imagens utilizadas no treino da rede.

Depois de criada a rede, foi feito um treino utilizando imagens de 5 fontes: IMDB, Wikipedia, FG-NET, MORPH, CACD, LAP. O conjunto das 5 fontes permitiu fazer o treino com um total de 660 000 imagens. Os testes finais mostram um erro obtido de 0.25 e um erro absoluto médio de 2.68 anos. Este valores foram vencedores do concurso mas o trabalho realizado mostrou também um conjunto de abordagens que vieram a ser implementadas em outros projetos.

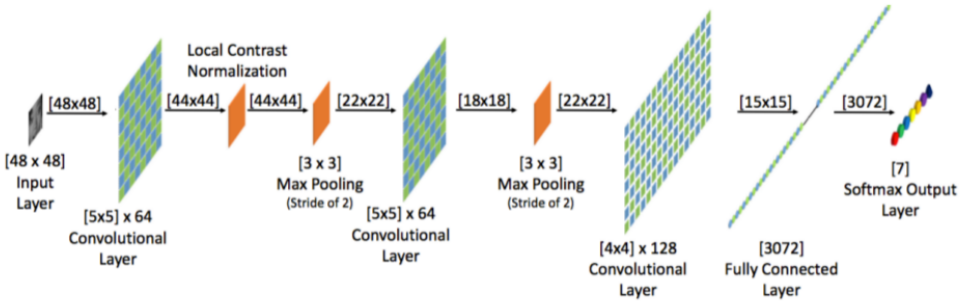


Figura 2.3: Arquitetura da rede escolhida no trabalho *Emotion Recognition using Deep Convolutional Neural Networks.*[9]

2.3 Gender Recognition Through Face Using Deep Learning

Neste projeto, os engenheiros Amit Dhomne, Ranjit Kumar e Vijay Bhan desenvolveram uma rede capaz de identificar o género da pessoa através da cara [10]. Esta equipa utilizou uma rede *VGG*, pois esta rede é bastante adequada para este tipo de problemas. A rede *VGG* utilizada neste projeto é bastante simples e utiliza apenas 4 técnicas: Convulsão, ReLu (Unidade Linear Retificada), Max pool (Max pooling) e LRN (Local Response Normalization). Estas 4 técnicas, utilizadas de maneira sequenciada permitem identificar um objeto na imagem e conhecendo algumas características do mesmo, categoriza-lo. A abordagem neste projeto é idêntica, mas ao invés de categorizar um objeto categoriza uma cara em homem ou mulher. Na figura 2.4 mostra a arquitetura desenvolvida para este projeto é possível verificar a utilização das 4 técnicas mencionadas anteriormente. Tal como no artigo anterior também nesta arquitetura são utilizadas 3 convoluções. A

Figura 2.4: Arquitetura da rede desenvolvida para o trabalho *Gender Recognition Through Face Using Deep Learning.* [10]

utilizações sequenciada de 3 convoluções é bastante comum nas redes VGG para extração de características da cara que irão permitir à rede calcular as probabilidades dos vários estados. Em seguida a rede criada foi treinada com cerca de 100 000 imagens. Por fim, a rede foi testada e foi possível verificar uma precisão de 95%.

2.4 Deep expectation of real and apparent age from a single image without facial landmarks

Este projeto foi desenvolvido pelos engenheiros Rasmus Rothe, Radu Timofte e Luc Van Gool [11]. Estes 3 engenheiros participaram no concurso ChaLearn LAP em 2015, no qual foram os vencedores. O concurso ChaLearn LAP em 2015 tinha como objetivo identificar as caras numa imagem e calcular a idade da mesma. De modo a superar este objetivo, os engenheiros criaram o sistema DEX (Deep EXpectation). A equipa estudou um conjunto de tipologias de redes, vaŕias redes que estimavam a idade e estudou várias abordagens a este problema. A abordagem escolhida passava por utilizar dados da idade real da pessoa (tendo em conta a data de nascimento) e a idade aparente da pessoa (tendo em conta a aparência da cara). A solução final encontrada é uma rede CNN (Convolution Neural Network) com arquitetura VGG com 16 camadas, mas esta arquitetura é utilizada para classificação de imagens e o calculo da idade é um problema de regressão linear. A equipa decidiu então utilizar o sistema VGG-16 para categorizar a idade para valores inteiros entre 0 e 100 e de seguida acrescentar um ultima camada. Esta ultima camada é uma *Euclidean loss function* que junta os valores de saída da rede anterior e calcula apenas um valor se saída que é a idade final calculada. A figura 2.5 mostra um esquema do funcionamento da rede.

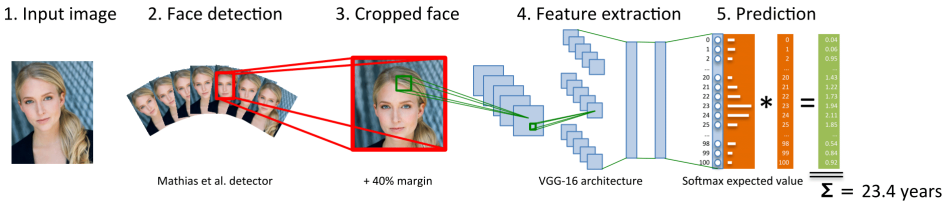


Figura 2.5: Esquema de funcionamento *Deep EXpectation*.[11]

A rede desenvolvida foi então treinada com imagens da *IMDB-WIKI*. Por fim foram efetuados testes que demonstraram um erro 0.2649 que garantiu o primeiro lugar, quando o segundo classificado obteve um erro de 0.270687.

2.5 DAGER: Deep Age, Gender and Emotion Recognition Using Convolutional Neural Networks

O projeto *DAGER* tem como objetivo encontrar as várias caras que existem numa imagem e, para cada uma das caras calcular a idade, o género e o estado emocional. Este

trabalho foi desenvolvido por Afshin Dehghan, Enrique G. Ortiz, Guang Shu, Syed Zain Masood que são quatro engenheiros do *Computer Vision Lab*, da empresa *Sighthound Inc.* [12]. Estes engenheiros criaram e treinaram quatro redes diferentes, uma para a cara, uma para a idade, uma para o género e uma para a emoção, ao invés de criar só uma rede que calcula-se os quatro parâmetros simultaneamente. Esta a abordagem ao problema permite estudar e manipular melhor cada um dos aspetos (cara, idade, género e emoção) e assim efetuar as alterações necessárias à rede ou à base de dados utilizada para que deste modo consigam obter resultados superiores as restantes redes/-projetos já existentes. Na figura 2.6 é possível ver um esquema que mostra a sucessão acontecimentos até ao calculo da idade, género e emoção.

Figura 2.6: Esquema de funcionamento *DAGER.*[12]

A primeira fase é detetar as várias caras na imagem, para tal foi treinada um rede com 4 000 imagens onde nesta se podem encontrar 40 000 caras. As imagens usadas foram o mais diversas possível, para que deste modo a rede que deteta as caras seja o mais robusta possível. Depois de identificada a cara é feito um alinhamento da mesma e de seguida está utilizada pelas restantes três redes. O alinhamento da cara é feito pois esta equipa verificou que com isto conseguiam obter melhor resultados nas restantes redes. No cálculo da idade foram utilizados dois métodos, o cálculo real da idade e o cálculo da idade segundo aparência da pessoa. No cálculo real da idade é utilizada uma rede e esta é treinada com uma base de dados com a imagem da cara da pessoa e a respetiva idade real. A realização de testes permitiu verificar que este método obteve um erro absoluto médio inferior aos restantes métodos utilizados atualmente. No cálculo da idade segundo a aparência da pessoa, existe um conjunto de dez valores dados para a idade de uma cara e é calculado o valor médio da idade, este juntamente com a imagem da cara é utilizado para treinar um rede neuronal. A realização de teste segundo este método permitiu verificar que o erro obtido no cálculo da idade foi o segundo melhor em comparação com os restantes. A rede que foi integrada no projeto é treinada com ambos os valores da idade (real e aparente), deste modo a idade final calculada tem em consideração a idade real mas também a idade que essa pessoa aparenta.

Na deteção da emoção, tal como o trabalho *Emotion Recognition using Deep Convolutional Neural Networks*, a equipa categorizou as emoções em sete: feliz, triste, neutro,

surpreso, enervado, desgosto e medo. Esta criou um rede capaz de detetar qual desta emoções se encontra na cara e treinou com 2 156 imagens. Na base de dados usada existe um numero idêntico de caras para cada emoção. Os resultados obtidos no teste desta rede mostram que esta rede tem mais precisão que a rede neuronal desenvolvida e treinada pela *Microsoft*. É também possível verificar que esta rede tem uma performance boa a identificar qualquer emoção, o que poderá estar relacionado com o facto de existirem um número idêntico de caras para cada emoção. Na deteção do género, a equipa criou uma rede e treinou-a 17 492 caras marcadas com o respetivo género. Os resultados obtidos mostras que esta rede tem um precisão(91%) maior que algumas das redes atuais, desenvolvidas pela *Microsoft* (90.86%), Kairos (84.66%),Face++ (83.04%) . Por fim, com a analise de alguns testes em imagens verificou-se que as 4 redes em conjunto são bastantes precisas e bastante robustas. A deteção é feita até a um longa distância e não se verifica um diminuição na precisão no cálculo da idade, género ou emoção. A figura 2.7 mostra um dos teste realizados. Não foi possível verificar o funcionamento do programa em tempo-real.

Figura 2.7: Teste realizado ao sistema *DAGER*.[12]

2.6 Moodme

A empresa *Moodme* tem um conjunto de aplicações para retirar dados de expressões faciais [13]. Esta empresa utiliza redes neuronais para extrair parâmetros faciais. Nas várias aplicações é possível verificar uma caracterização bastante complexa de *Landmarks* cara. É visível que existe a extração bastante detalhada de características da cara. A informação retirada permite que as redes calculem dados como: idade, género, emoção, nível de atenção, numero de vezes em que a cara aparece e consegue identificar a cara (se esta estiver no banco de imagem). A figura 2.8 mostra um exemplo de um teste

realizado a uma aplicação.

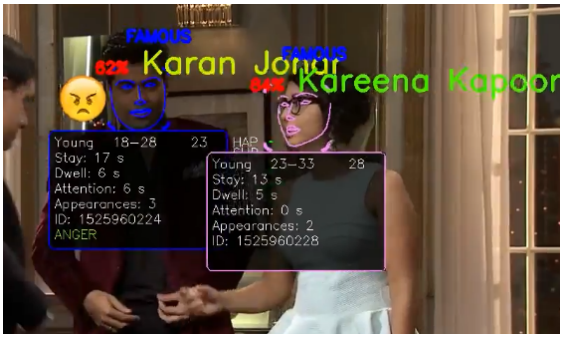


Figura 2.8: Teste realizado ao sistema *Moodme*.[13]

Este teste foi realizado com um vídeo e foi testado em tempo real.

2.7 Faceplusplus

A *Faceplusplus* é um empresa que oferece um conjunto de soluções para visão computacional para extração de dados da cara e do corpo de uma pessoa, utilizando técnicas de processamento de imagem e redes neutrais [14]. O *Faceplusplus*, para a deteção e localização de faces utiliza uma técnica de *facial bounding boxes*, o que permite localizar a face com alta precisão. Depois de detetada o rosto, existe um conjunto de técnicas de *Machine Learning* para analisar o rosto e dados relacionados com esta. Os dados obtidos calculam o género, a idade, a postura da cabeça, o estado emocional, o grupo étnico e o estado dos olhos. A figura 2.9 mostra um teste realizado ao software da *Faceplusplus*. O teste realizado mostra um conjunto de dados que são coincidentes com a realidade. É

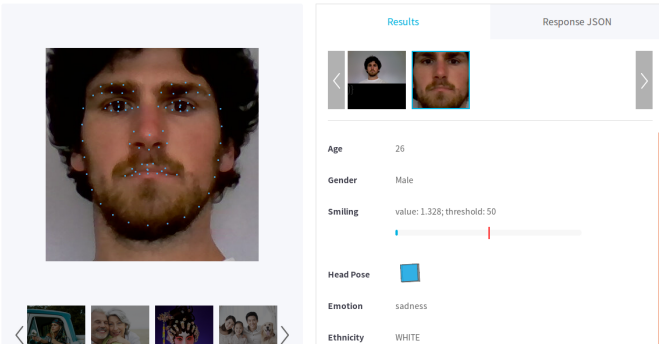


Figura 2.9: Teste realizado ao sistema *Faceplusplus*.

também possível verificar pela figura 2.9 que existe uma extração bastante complexa de

Landmarks da cara e que estes são utilizados para calcular os restantes dados. Nenhum teste foi realizado em tempo-real.

2.8 Interfaces homem-máquina

A Interface homem-máquina (HMI) é a parte de um sistema ou máquina que comunica com o utilizador. Segundo a ISO 9241-110 [15], o termo interface de utilizador é definido como "todas as partes de um sistema interativo (de software ou hardware) que fornecem informações e controle necessários para que o utilizador realize uma determinada tarefa com o sistema interativo." Em suma, pode-se considerar que a interface Homem-máquina é a plataforma na qual o sistema/máquina oferece informações ao utilizador e simultaneamente permite que este controle as ações que a máquina irá realizar. Um exemplo bastante simples é um interruptor e uma lâmpada. Neste caso, a interface utilizada é o interruptor, no qual o utilizador decide se quer ligar ou desligar a lâmpada. Ao longo do anos as interfaces homem-máquina têm evoluído bastante e forma como o utilizador comunica com a maquina tem se alterado bastante. Desde os tempos em que os painéis de controlo eram um conjunto de interruptores para controlar operações da máquina, até aos dias de hoje em que o controlo pode ser através de câmaras que captam os movimentos do corpo e reagem a isso, microfones que captam comandos de voz, ou até através de sensores complexos que fazem leituras dos sinais elétricos produzidos pelo cérebro e com isso sabem qual a ordem do utilizador. A investigação e o avanço tecnológico tem proporcionado ao Homem um avanço na interação com os robôs, sendo que cada vez mais interação Homem-máquina se assemelha cada vez mais com a interação Homem-homem (figura 2.10).

Figura 2.10: Exemplo de uma interação homem-maquina com o robô Asimo. [16]

O facto de cada vez mais a interação homem maquina ser mais idêntica homem-

homem tem também sido propulsionada pelo o facto de haver um aumento dos robôs integrados no dia-a-dia do Homem. A evolução da interação homem-máquina passa por o robô conseguir retirar mais informação do humano e do seu comportamento, se este muda o tom de voz, se a sua expressão facial muda, qual o seu género ou idade. A informação obtida pelo o robô permite que este mude o seu comportamento em função da resposta dada pelo humano. Podemos assim verificar a importância de conseguir retirar informação de uma imagem da face. A leitura feita a um rosto pode ser crucial para uma melhor interação Homem-máquina, por exemplo a abordagem feita a uma criança de 10 anos não deverá ser a mesma que a abordagem a um adulto de 24 anos. O robô, se verificar que o estado emocional mudou de feliz para triste, deverá alterar a sua abordagem/comportamento.

Capítulo 3

Fundamentos Teóricos

Neste capitulo é apresentado e explicado um conjunto conceitos base, que foram utilizados nesta tese e que vão permitir uma melhor compressão da mesma. Este trabalho foca-se principalmente em redes neuronais. Inicialmente irão ser abordados um conjunto de conceitos gerais sobre inteligência artificial e de seguida irão ser explicados um conjunto de conceitos mais específicos de inteligência artificial relacionados com processamento de imagem.

3.1 Rede Neuronal

Uma rede neuronal é um conjunto de perceptrões. Um perceptrão calcula a soma ponderada de vários *Inputs*, aplica uma função e assim calcula o seu *Output*. A figura 3.1 mostra um exemplo de um perceptrão, em que o *Input* esta definido como x, os pesos para cada *Input* como w e o *Output* como y. Na imagem 3.1, depois de atribuído os pesos a cada *Input* temos o correspondente a um neurónio, ou seja a soma ponderada e a função de ativação.

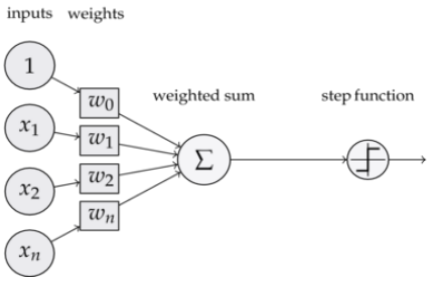


Figura 3.1: Exemplo de um perceptron. [17]

O perceptron, matematicamente e segundo a nomenclatura usada na imagem pode ser definido como :

$$Xw = y \tag{3.1}$$

A equação 3.1 sobre a forma matricial, irá ter a formulação(3.2). Aqui temos uma matriz de *Input* e por fim são calculados vários *Outputs* que correspondem ao vetor de saída *y*.

$$\begin{bmatrix} 1 & x_{11} & ... & x_{1d} \\ 1 & x_{21} & ... & x_{2d} \\ ... & ... & ... & ... \\ 1 & x_{n1} & ... & x_{nd} \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \\ ... \\ w_d \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ ... \\ y_n \end{bmatrix} \tag{3.2}$$

Um conjunto de perceptrons em paralelo recebe um conjunto de *Inputs* e calcula um conjunto de *Outputs*, se estes *Outputs* forem *Inputs* de outros perceptrons temos então uma segunda camada de perceptrons temos então uma rede neuronal com várias camadas ocultas.

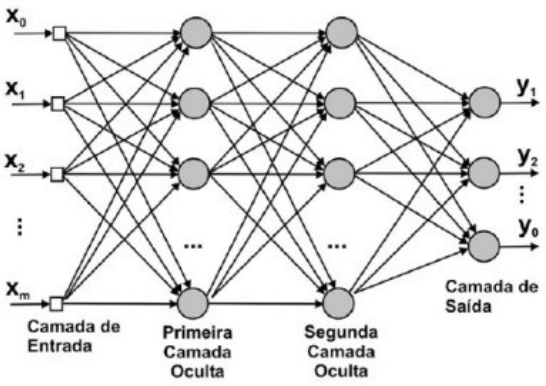


Figura 3.2: Exemplo de uma rede neuronal. [18]

A formulação matemática de uma rede neuronal é idêntica à de um perceptron(3.2) mas, neste caso têm que se considerar as ponderações das camadas ocultas(W). A equação 3.3 mostra a transformação da equação para uma rede neuronal, aqui o w representa a ultima camada antes do *Output*.

$$(XW)w = y \tag{3.3}$$

Na forma matricial, as ponderações das camadas ocultas passam de um vetor a uma matriz, como de pode ver na figura 3.4.

$$\begin{bmatrix} 1 & x_{11} & ... & x_{1d} \\ 1 & x_{21} & ... & x_{2d} \\ ... & ... & ... & ... \\ 1 & x_{n1} & ... & x_{nd} \end{bmatrix} \times \begin{bmatrix} w_{01} & w_{01} & ... & w_{0m} \\ w_{11} & w_{11} & ... & w_{1m} \\ ... & ... & ... & ... \\ w_{d1} & w_{d1} & ... & x_{dm} \end{bmatrix} \times \begin{bmatrix} w_{01} \\ w_{11} \\ ... \\ w_{d1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ ... \\ y_n \end{bmatrix} \tag{3.4}$$

O numero de camadas ocultas pode elevar-se e criar redes neurais bastante complexas. As redes neurais com um elevado numero de camadas ocultas são chamadas de DNN (*Deep Neural Networks*).

3.2 Utilização de Redes Neuronais para Machine Learning

O *Machine Learning* é uma das áreas da inteligência artificial que utiliza um conjunto de algoritmos para analisar dados, aprender com estes dados e prever ou determinar algo em novos dados. No caso especifico das redes neuronais, *Machine Learning* é utilizado para analisar vários dados, calcular as ponderações necessárias para a rede neuronal (corresponde á variável *w*) e por fim dado um novo *Input* a rede calcula um novo *Output* com uma determinada precisão. Um exemplo poderá ser, se tivermos dados sobre vaŕias características de imóveis e o preço a que estes foram vendidos. Se estes dados forem analisados e recorrendo a *Machine Learning* ensinar-mos uma rede neural. A rede neural será capaz de prever um preço de um novo imóvel, considerando que são fornecidas as características necessárias do mesmo. Em *Machine Learning* existem vários algoritmos de aprendizagem, e considerando o seu modo de aprendizagem podem ser divididos em 3 categorias:

- Aprendizagem supervisionada;
- Aprendizagem não-supervisionada;
- Aprendizagem semi-supervisionada.

A aprendizagem supervisionada é quando os dados que utilizamos para a aprendizagem têm simultaneamente os *Inputs* e os *Outputs*. No caso dos imóveis, é quando temos as características e o preço a que foi vendido. A aprendizagem não supervisionada é quando os dados que vão ser utilizados tem apenas os *Inputs*. No caso dos imóveis seria apenas as caraterísticas das habitações. A aprendizagem semi-supervisionada é quando temos bastantes *Inputs* e apenas alguns *Outputs*. De todos os vários tipos de aprendizagem, o mais comum é a aprendizagem supervisionada. Os algoritmos de aprendizagem podem

também estar agrupados, considerando outros fatores tais como a sua semelhança na forma como funcionam e nesse caso podemos ter:

- Algoritmos de classificação;
- Algoritmos de regressão;
- Árvore de decisão;
- *Deep Learning*.

Das 4 categorias mencionadas anteriormente iremos explorar a categoria do *Deep Learning*, pois esta área tem um enorme potencial para o futuro da inteligência artificial.

3.3 Utilização de Redes Neuronais para Deep Learning

Em *Machine Learning*, é treinada uma rede para que esta com determinados dados de *Input* calcular um *Output*. Neste caso é necessário que os dados (utilizados para treinar a rede e para *Input*) estejam pré tratados, mas e se existir a possibilidade de introduzir uma imagem? Se houver a possibilidade de extrair um conjunto de *Features* de uma imagem e conseguir identificar os objetos que lá estão. É aqui que surge o *Deep Learning*. O *Deep Learning* é uma área especifica do *Machine Learning* que trabalha com *Deep Neural Networks* (DNN), que são redes neuronais mas com múltiplas camadas ocultas. Em *Deep Learning*, as redes são capazes de extrair *Features* da imagem (ou outro *Input*, como por exemplo uma faixa sonora) recorrendo ao uso de várias camadas ocultas. De seguida introduzir os dados obtidos pelas camadas ocultas em uma última camada que vai categorizar os dados obtidos. As camadas ocultas iniciais (mais próximas do *Input*) identificam parâmetros tais como traços, variação de luminosidade e as camadas ocultas finais(mais próximas do *Output*) juntam os vários traços identificados anteriormente e reconhecem formas geométricas.

Um exemplo prático é, suponhamos que utilizamos uma DNN a identificar bicicletas. Inserimos nesta rede uma imagem, a rede extrai um conjuntos de *Features* desta imagem e no fim os dados são categorizados em bicicleta ou não bicicleta. A figura 3.3 mostra um esquema representativo.

Na figura 3.3 os blocos a cinzentos marcam as etapas automatizadas do processo.

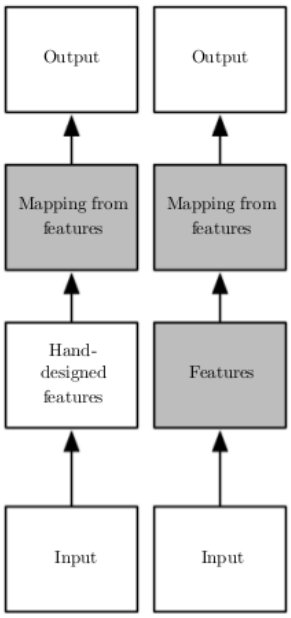


Figura 3.3: Principais diferenças entre *Deep Learning* e *Machine Learning*.

3.4 Convolutional Neural Networks

As *Convolutional Neural Networks* (CNN) são redes bastantes utilizadas em inteligência artificial quando se trabalha com imagens. As CNN são bastante fáceis de treinar quando temos um elevado número de imagens rotulados com os diferentes tidos de categorias-alvo. As vantagens em utilizar CNN são a capacidade de extrair características relevantes e depender de um menor numero de parâmetros do que redes totalmente conectadas com o mesmo numero de camadas ocultas. Cada camada oculta não é conectada com todas as unidades da camada seguinte, logo à um menor numero de pesos (corresponde ao W na equação 3.4) a serem calculados. Logo é mais fácil e rápido treinar uma rede CNN. As CNN são formadas por sequencias de camadas, cada uma destas camadas desempenha um função especifica e extrai um tipo especifico de *Features*.Na figura 3.4 está ilustrado um exemplo de uma *Convolutional Neural Networks*.

De seguida irá ser explicado quais os vários tipos de camadas ocultas e como estas funcionam.

3.4.1 Convolução

A camada convolucional tem um conjunto de neurónios e cada neurónio é um filtro aplicado á imagem que entra. Se considerarmos que a imagem de entrada com o for-

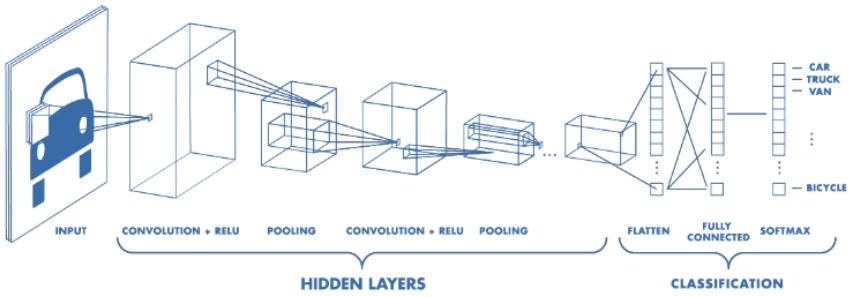


Figura 3.4: Exemplo de uma CNN. [19]

mato RGB e tem as dimensões 224×224, temos dados de então dados sobre a forma 224×224×3 (uma para cada cor). Cada um dos filtros da camada convolucional vai processar a imagem ponto a ponto. Se definirmos que queremos analisar uma área em redor de um determinado ponto de 3 *Pixels*, a dimensão de cada filtro será de 3×3×3 (3 *Pixels* de largura, 3 *Pixels* de altura, isto para cada camada). O filtro irá dar origem a uma estrutura conectada localmente que percorre os dados de entrada (224×224×3). O somatório do produto ponto a ponto entre os valores do filtro e a posição dos dados de entrada é então conhecida como convolução. A figura 3.5 mostra um esquema representativo de uma convolução.

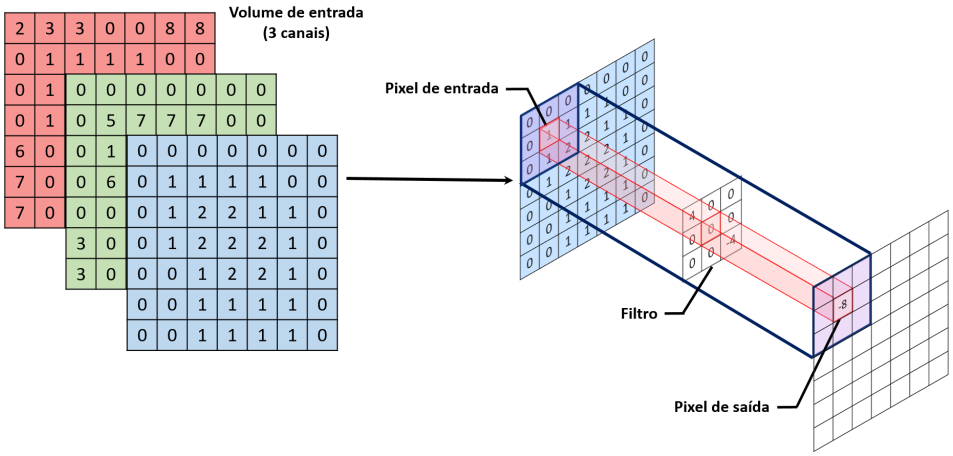


Figura 3.5: Exemplo de uma convolução.

Os valores que resultam da convolução passam depois por uma função de ativação. Existem 3 parâmetros que definem o tamanho dos dados resultantes de uma camada

convolucional: profundidade, passo e *zero-padding*. A profundidade dos dados resultantes dependem do número de filtros utilizados. Cada um destes filtros irá extrair características diferentes nos dados de entrada. O passo indica qual o tamanho do salto na operação de convolução. Se o passo for igual a 1, o filtro salta uma posição de cada vez. Se o passo for igual a 2, o filtro salta duas posições de cada vez. Quanto maior o valor do passo, menor será a altura e o comprimento dos dados resultantes mas deste modo existem características importantes que poderão ser perdidas. A figura 3.6 mostra os dois casos quando o passo é igual a 1 e 2.

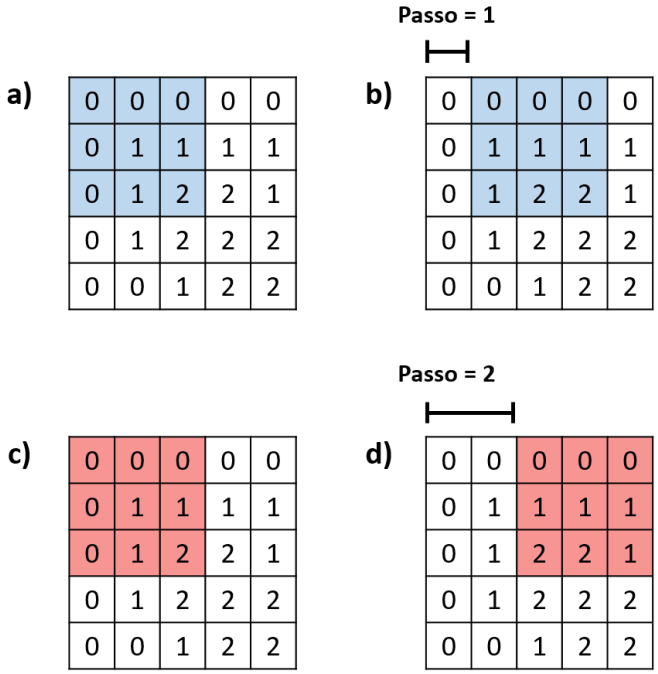


Figura 3.6: Exemplo demonstrativos dos vários passos.

O *zero-padding* consiste em preencher com zeros a borda dos dados de entrada. A vantagem em utilizar esta operação é poder controlar a altura e largura dos dados de saída. Deste modo é possível fazer com que eles fiquem com os mesmos valores dos dados de entrada. As equações 3.5 e 3.6 mostram como é possível calcular a altura e a largura dos dados de saída.

$$Altura = \frac{A - F + 2P}{S} + 1 \tag{3.5}$$

$$Largura = \frac{L - F + 2P}{S} + 1 \tag{3.6}$$

Nas equações 3.5 e 3.6, o A corresponde à altura dos dados de entrada, o L à largura dos dados de entrada, o F é o tamanho dos filtros utilizados, o S é o valor do passo e o P é o valor do *zero-padding*.

3.4.2 Pooling

Numa CNN é usual, depois de uma camada convolucional, haver uma camada de *pooling*. A camada de *pooling* é utilizada para reduzir as dimensões dos dados de entrada (dados de saída de uma outra camada), com esta redução diminui o custo computacional e evita o problema de *overfitting*. A operação de *pooling* consiste em agrupar os valores pertencentes a uma determinada região dos dados,gerados pela camada de convulsão, e substitui-los por alguma métrica que exista nessa região. Usualmente a substituição é feita pelo valor máximo encontrado nessa região e a essa técnica chamamos de *Maxpooling*. A figura 3.7 mostra um exemplo demonstrativo de um *Maxpool* a uma imagem de 4×4, com um filtro de 2×2.

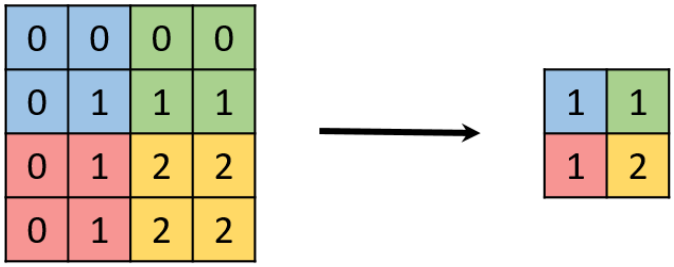


Figura 3.7: Exemplo de um *Maxpool*.

O *Maxpooling* é útil para eliminar valores desprezíveis, acelerando a computação necessária para as próximas camadas. Outras funções de *pooling* usadas são a média e a média ponderada baseada na distância partindo do *pixel* central. A altura e largura dos dados de saída podem ser calculados com as seguintes formulas(3.7 e 3.8).

$$Altura = \frac{A - F}{S} + 1 \tag{3.7}$$

$$Largura = \frac{L - F}{S} + 1 \tag{3.8}$$

As variáveis A e L correspondem à altura e largura dos dados de saída. A variável F as dimensões do filtro utilizado. A variável S corresponde ao passo utilizado.

3.4.3 Relu

A camada ReLU (unidade de retificação linear) de um modo simples recebe um determinado dado de entrada, caso este seja positivo não o altera, caso seja negativo altera-o para 0. Estas são das camadas não-lineares mais simples de se usar.

3.4.4 Full connected

Os *Outputs* obtidos pelas camadas de *pooling* e camadas de convulsão representam características da imagem de *Input* da rede. A camada *Full connected*(FC) utiliza estas características para classificar a imagem em um categoria, das categorias para qual a rede foi treinada. A camada FC devido as suas características é normalmente uma camada usada no fim da rede. Esta rede tem o nome *Full connected* pois esta conecta os neurónios da camada anterior com os neurónios da camada seguinte. Matematicamente, podemos considerar um neurónio de uma camada *Full connected* como está nas equações 3.9 e 3.10.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.9}$$

$$y_k = \varphi(u_k + b_k) \tag{3.10}$$

Nestas equações, o x_k correspondem aos valores de entrada,om w_{kj} correspondem ao pesos atribuídos ao neurónio k, b_k corresponde ao *bias* responsável por realizar o deslocamento da função de ativação definida por φ. Umas das técnicas bastante usadas entre as camadas FC é o *dropout*. O *dropout* permite reduzir o tempo de treino e o *overfitting*. Resumidamente, esta técnica remove iterativamente uma percentagem aleatória de neurónios de uma camada e volta a adiciona-los na iteração seguinte.

3.4.5 LeNet

A rede LeNet foi proposta por Yann LeCun em 1988 [20] e foi um dos primeiros projetos de CNN. Esta rede foi uma das responsáveis por impulsionar o campo de Deep Learning. As primeiras versões da rede LeNet foram utilizadas para reconhecimento de caracteres em cartas e outros documentos escritos à mão. Este tipo de rede utiliza 3 tipos de camadas, camadas convolucionais, camadas de *pooling* e camadas *Full connected*. As camadas estão organizadas por esta ordem, primeiro uma camada convolucional, depois uma camada de *pooling*, outra camada convolucional e outra de *pooling* e por fim duas

camadas camadas *Full connected*. As camadas de camadas *pooling* estão sempre depois de uma camada convolucional para reduzir a dimensão dos dados resultantes da convolução. O *Output* desta rede é a probabilidade de a imagem de *Input* pertencer a uma das classes para qual a rede foi treinada.A figura 3.8 mostra um exemplo de uma rede LeNet.

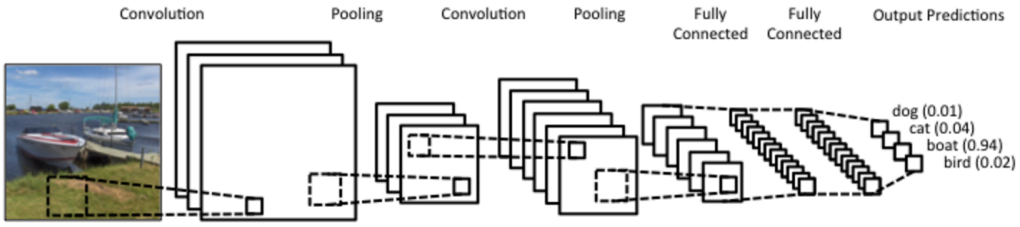


Figura 3.8: Exemplo de uma rede LeNet. [21]

Ao longo do tempo foram surgindo novas arquiteturas que inspiraram na rede LeNet.

3.4.6 AlexNet

A rede AlexNet corresponde a uma rede LeNet mas com mais camadas e foi criado por Alex Krizhevsky [22]. Esta possui 5 tipos de camadas convolucionais, camadas *Maxpooling* e 3 camadas *Full connected*. Os avanços nesta rede permitiram classificar imagens em 1000 possíveis categorias. Esta rede ganhou o concurso de ILSVRC 2012 (ImageNet Large Scale Visual Recognition Challenge). No ILSVRC 2012, várias equipas competem para desenhar e implementar o melhor modelo para classificação, deteção e localização de objetos em imagens. A AlexNet atingiu o primeiro lugar desse desafio, e com uma diferença bem significativa para o segundo classificado, com erro de 15.4% contra 26.2%. A figura 3.9 mostra um exemplo de uma rede AlexNet.

Figura 3.9: Exemplo de uma rede AlexNet. [23]

A rede AlexNet foi um avanço significativo, tendo a mesma inspirado dezenas de outras redes convolucionais para reconhecimento de padrões em imagens.

3.4.7 Visual Geometry Group

A rede VGG proposta por Karen Simonyan e Andrew Zisserman [24] foi a primeira rede a utilizar filtros de pequenas dimensões em cada camada convolucional ao contrário da rede AlexNet. Usualmente nas redes eram utilizados filtros de grandes dimensões (9x9 e 11x11) para capturar características nas imagens. A grande contribuição da VGG foi a ideia de que múltiplas convoluções 3x3 em sequência podiam substituir efeitos de filtros de máscaras maiores (5x5 e 7x7).Isto resulta em um menor custo computacional comparando com redes do tipo AlexNet. A figura 3.10 mostra um exemplo de uma rede VGG com 16 camadas.

Figura 3.10: Exemplo de uma rede VGG. [23]

Na figura 3.10 é possível ver uma sucessão de blocos onde, os blocos amarelos correspondem a convoluções, os blocos azuis correspondem a *pool*, os blocos verdes correspondem a *Full connected* e os vermelhos correspondem a *softmax*.

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Projeto

Neste capitulo é apresentado e descrito a arquitetura do *Software* desenvolvido para resolver o problema de deteção do rosto, *Facetraking* e determinação da idade, género e estado emocional do rosto. Para tal irá ser apresentado a estrutura de *Software* a ser desenvolvida, detalhando os blocos que a compõem numa arquitetura. Por fim irá ser descrito onde e como vai ser implementado a arquitetura proposta.

4.1 Arquitetura de software

Depois de estudados vários artigos e trabalhos que abordassem a problemática, analisadas várias soluções existentes foi possível ter uma ideia geral de como se irá abordar o problema e qual será a melhor maneira de se criar uma solução. Os dados de entrada do sistema que será desenvolvido deverão ser as imagens obtidas através de uma câmara de espetro visível. Os dados de saída deverão ser os dados de cada cara (idade, género e estado emocional) e um identificador (ID).

Para que este processo aconteça, foi projetada uma arquitetura para o sistema. Na figura 4.1 encontra-se um esquema onde é possível ver os blocos necessários para transformar os dados de entradas e assim obter os dados de saída. O bloco da figura representa um parte do código desenvolvido que é responsável por realizar um conjunto de tarefas especificas.

Na arquitetura da figura 4.1 é realizado a captura de uma imagem, de seguida deverá ser efetuada a deteção das caras na mesma, as caras deverão ser enviadas para cada um dos 3 blocos e estes irão efetuar os processos necessários e extrair a informação necessária das mesmas. Depois disto, existe uma associação dos dados obtidos de cada bloco e de seguida obtemos os dados de saída do sistema. Os dados de saida irão voltar a ser

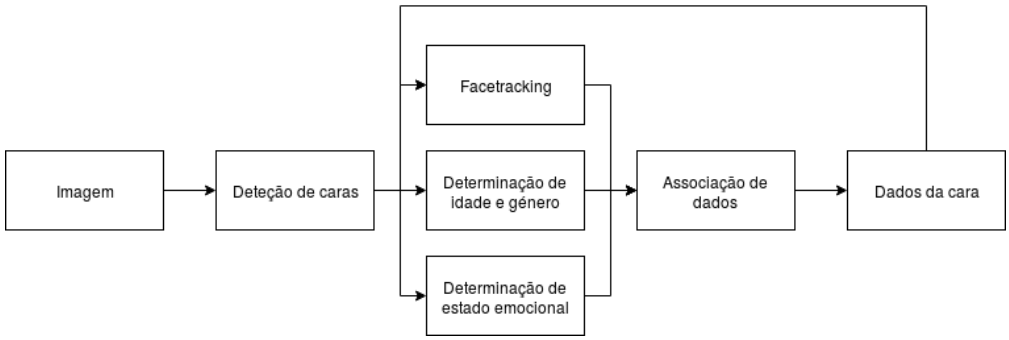


Figura 4.1: Arquitetura do *Software*.

utilizados na imagem seguinte pelo bloco de facetracking. A arquitetura da imagem 4.1 o bloco Deteção da cara, Determinação da idade e do género e Determinação do estado emocional irão utilizar redes neuronais mas serão 3 redes separadas. As redes irão ser separadas pois, deste modo é possível trabalhar cada rede separadamente e afinar e corrigir cada rede sem influenciar o resultado de outra rede. O problema resolvido em cada bloco é diferente do problema dos restantes e depende da fatores diferentes.

4.1.1 Deteção da cara

No bloco Deteção de cara deverão ser detetadas e localizadas todas as caras da imagem e de seguida recortar cada um dos rosto da imagem. Este problema é um problema bastante comum de classificação de imagens e é bastante usual em rede neurais. A figura 4.2 mostra um exemplo do que este bloco deverá fazer, à esquerda temos a imagem de entrada e à direita temos as imagens de saída com as caras.

Este bloco tem uma importância bastante relevante para o restante sistema, pois a correta identificação das imagem irá levar a um bom desempenho dos restantes blocos. Este bloco deverá ter alguma robustez e assim ser capaz de identificar as caras mesmo havendo algumas variações de luminosidade. A distância é também um fator importante e este deverá conseguir identificar as caras a distancias que poderão ir até cerca de 4/5 metros. Os vários artigos e trabalhos estudados no capitulo 2 utilizaram a biblioteca *Haar Cascades* do *OpenCV*. A biblioteca *Haar Cascades* é uma solução exequível mas não consegue efetuar a deteção de caras a uma distancia tão elevada. No capitulo 2 encontra-se mencionado o artigo [8]. Este bloco deverá criar uma rede similar e utilizar os mesmos tipos de blocos ou idênticos (convolução, *Maxpooling* e *Full Connected*). A qualidade das imagens das caras é também um aspeto bastante importante, pois a performance dos restantes blocos depende deste. Esse será um parâmetro a analisar na

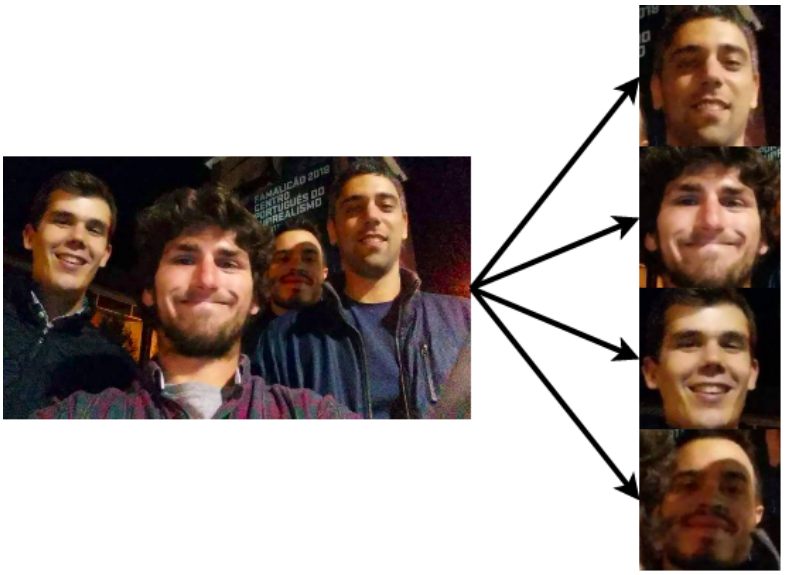


Figura 4.2: Exemplo do que deverá ser feito no bloco Deteção de caras.

construção deste bloco.

4.1.2 Determinação da idade e género

No bloco Determinação da idade e género deverá ser determinada a idade aproximada da face e o género da mesma. A determinação da idade e do género estão no mesmo bloco pois estas utilizam parâmetros similares da cara. A determinação da idade e do género necessitam de informação não só do centro da cara mas da área envolvente como por exemplo o cabelo. Assim será mais indicado ter uma imagem da cara como se pode ver na figura 4.3(b).

(a) Centro da cara (b) Cara completa

Figura 4.3: Exemplos de dados de entrado no bloco Determinação da idade e género

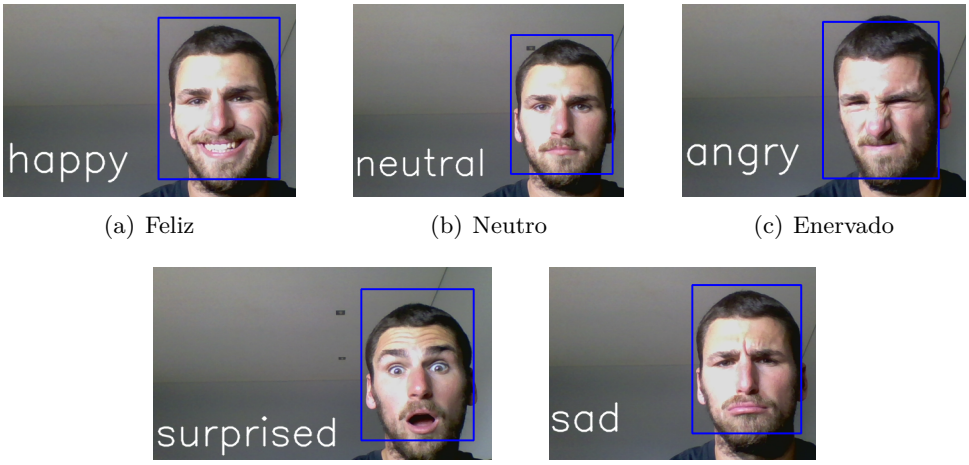
A determinação da idade utiliza somente a imagem da cara, tal como estudado no

artigo [11] no capitulo 2, permite-nos apenas descobrir a idade aparente pois não existe mais nenhum dado de entrada. O problema da determinação da idade, utilizando apenas a imagem da cara, pode ser abordado da seguinte maneira. A rede neuronal permite extrair informação importante de uma imagem (tais como rugas e traços faciais no zona da boca e olhos) e conseguir detetar detalhes que o olho humano não consegue mas essa informação tal como um humana é apenas baseada na imagem. A imagem indica apenas o aspeto visual e este não se altera de igual modo em todas as pessoas. Existem pessoas com 30 anos de idade que aparentam ter 40 anos de idade e vice-versa. Logo os dados que serão utilizados para treinar a rede deverão ter o valor real e valores aparentes da idade da pessoa. Há outros fatores tais como luminosidade e maquilhagem que podem levar alterar o valor de saída determinado pela rede neuronal. O problema pode ser abordado como um problema de classificação de imagem onde a rede classifica a cara entra valores inteiros entre 1 e 100. A determinação da idade é um problema de regressão logo é necessário acrescentar à ultima camada da rede VGG, a camada *Euclidean loss function*. Esta camada deverá analisar a saída da rede VGG, as idades e as respetivas probabilidades, e calcular o valor final da idade.

A determinação do género é um problema classificação simples em que se utiliza também um rede VGG. Alguns parâmetros podem ser decisivos na determinação do género tais como o cabelo, a barba e alguns traços da cara. O resultado desta rede é apresentado na figura 4.4.

Figura 4.4: Exemplo do que deverá ser feito no bloco Determinação da idade e género.

Os dados determinados neste bloco serão depois enviados para o bloco Associação de dados, onde serão associado a uma determinada cara e uma identificação.



(a) Feliz (b) Neutro (c) Enervado

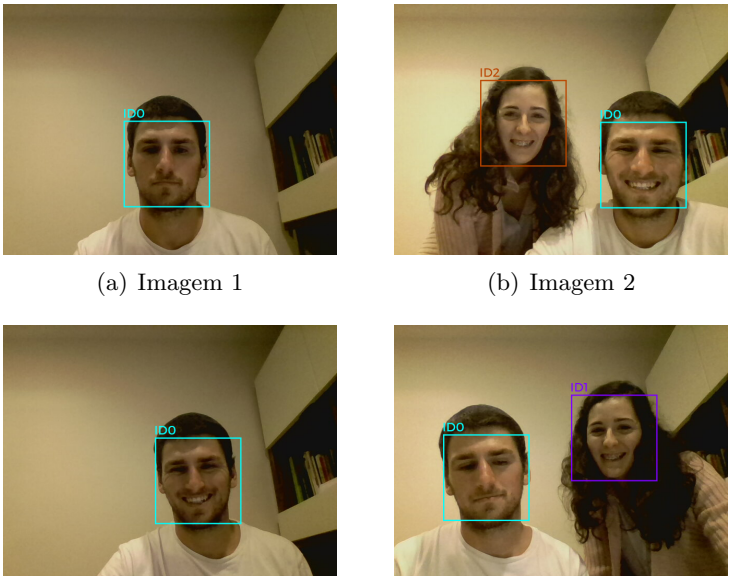
(d) Surpreso (e) Triste

Figura 4.5: Exemplos de vários estados emocionais

4.1.3 Determinação do estado emocional

O bloco Determinação do estado emocional deverá determinar qual o estado emocional que está representado na cara da pessoa. Os estados emocionais podem ser bastante variados mas neste trabalho, tal como no trabalho [9], foram considerados 7: feliz, triste, neutro, surpreso, enervado, desgosto e com medo. A figura 4.5 mostra um exemplo de várias expressões faciais e os seus respetivos estados emocionais. Nas imagens é possível verificar os vários detalhes da cara que são específicos de um determinado estado emocional. O estado emocional feliz está normalmente associado a um sorriso. O estado emocional surpreso está associado a sobrancelhas levantadas e a boca boquiaberta. A informação necessária extrair da cara para determinar o estado emocional, ao contrario do bloco Determinação da idade e género, encontra-se no centro da cara. O envolvente da cara é considerado ruído e deverá ser excluído para não influenciar erradamente determinação do estado emocional.

A rede neuronal que irá fazer a determinação deverá ser semelhante à rede VGG usada no artigo [9]. A rede VGG é a mais adequado a este problema, pois este é um problema de categorização com 7 categorias. As sete categorias corresponde aos sete estados emocionais. Esta rede deverá conseguir extrair informações sobre os zonas e traços da cara tais como boca, nariz, olhos, sobrancelhas e testa e assim determinar a probabilidade de aquela cara corresponder a um estado emocional. A extração de informação deverá ser feito recorrendo à utilização de sucessivas convoluções, deste modo conseguimos extrair toda a informação da cara. Os dados de saída deste bloco deverão



(a) Imagem 1 (b) Imagem 2

(c) Imagem 3 (d) Imagem 4

Figura 4.6: Sequencia de imagens que ilustra o que deverá acontecer no bloco *Facetraking*

depois ser enviados para o bloco Associação de dados.

4.1.4 Facetraking

O bloco *Facetraking* deverá receber as várias caras encontradas pelo bloco Deteção da cara, atribui-lhes um ID. Caso uma cara apareça em vários *Frames* seguidos esta deverá ter sempre o mesmo ID. Por exemplo, obtem-se uma imagem com 3 caras diferentes, a cara A, a cara B e a cara C. A cada uma deve ser atribuído um ID, à cara A o ID 0, à cara B o ID 1 e à cara C o ID 2. Enquanto as caras A, B e C continuarem a aparecer na imagem, o ID correspondente deve aparecer também. Se a cara B sair da imagem o seu ID deve desaparecer e os restantes dois ID's devem continuar a seguir as caras. Se depois de alguns instantes a cara B voltar a aparecer esta deverá ser atribuído um novo ID, neste caso o ID 3.

A atribuição dos ID e o seguimento das caras deverá ser feito do seguinte modo. A cada *Frame* da câmara é enviado o conjunto de caras identificadas e as coordenadas do respetivo retângulo da imagem. Se as caras forem novas e naquela zona da imagem não houver nenhuma cara do *Frame* anterior é porque a cara é nova e deverá ser atribuído um novo ID a cada uma das caras. O ID de cada uma das caras e a respetiva posição da cara deve ser guardado e analisado no *Frame* seguinte. É também criada e guardado

uma região de deteção para cada cara. A região de deteção corresponde à área do retângulo da cara mais uma margem. No *Frame* seguinte são de novo enviadas as caras encontradas e o seu respetivo retângulo na imagem. Caso o retângulo desta cara corresponda a uma região de deteção do *Frame* anterior é então atribuído o mesmo ID da cara que corresponde aquela região de deteção. Se não houver correspondência deverá ser atribuído um novo ID. Se uma região de deteção não tiver correspondência durante alguns *Frame* este ID e os dados correspondentes deverão ser apagados. Os dados de saída deste bloco serão as coordenadas da cara na imagem e o respetivo ID, para ajudar a uma melhor visualização é também atribuída uma cor especifica ao ID.

4.1.5 Associação de dados

No bloco Associação de dados são recebidos os dados enviados pelos blocos anteriores e estes são agrupados. Os dados deverão ser agrupados por para cara, ou seja, devem ser guardados do seguinte modo: primeiro o ID e a respetiva cor, as coordenadas da cara na imagem, o género, a idade e o estado emocional. Deste modo irá facilitar a representação dos dados na imagem e também no terminal do computador. A representação dos dados na imagem deverá colocar um retângulo à volta da cara em cima do retângulo identificar o ID e por baixo a informação sobre género, idade e estado emocional. Os dados representados na imagem devem ter todos a mesma cor para facilitar a identificação na imagem. No terminal do computador a cada *Frame* deveŕa aparecer uma lista com os vários ID e á frente de cada ID os dados da cara. A representação dos dados na imagem deve ser algo como ilustrado na imagem 4.7.

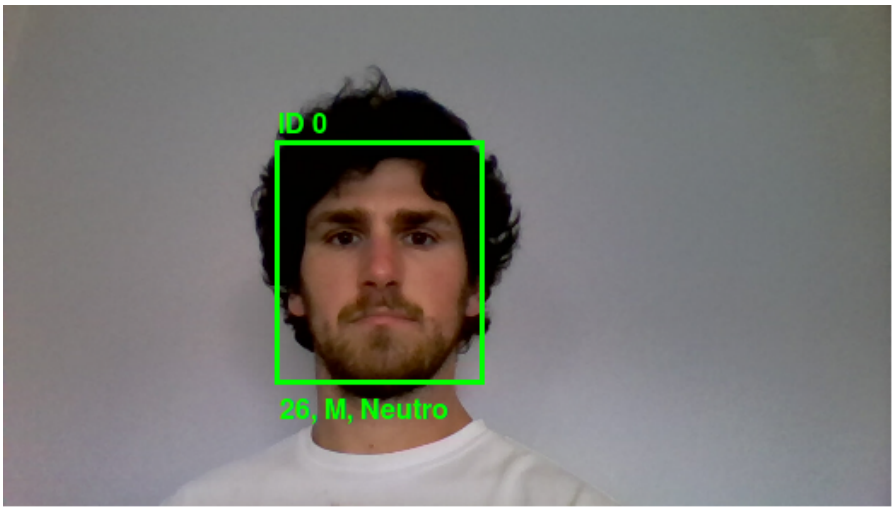


Figura 4.7: Exemplo da representação de dados na imagem.

4.2 Plataforma de implementação

A arquitetura mencionada anteriormente será implementada numa unidade de processamento central (CPU) e numa unidade de processamento de visão (VPU). A unidade CPU que irá ser utilizada será um processador *Intel Core i7-3537U* com 4 *Cores* e 2 *Threads*. A frequência de funcionamento base é de 2,00 GHz e a frequência máxima turbo é de 3,10 GHz. A capacidade máxima de memoria suportada por este processador é de 32 GB. O processador *Intel Core i7-3537U* têm também um processador gráfico integrado que é o *Intel HD Graphics 4000*. O *Intel HD Graphics 4000* trabalha a uma frequência base de 350 MHz e a frequência máxima dinâmica é de 1.20 GHz. O sistema operativo que irá ser utilizado será *Ubuntu 16.04 LTS* e a linguagem que irá ser utilizada será *Python*. No sistema operativo para a utilização das redes neurais e processamento de imagem será necessário o *OpenCV 3.1.0*, *CUDA 8.0*, a biblioteca *caffe* e o *Tensorflow 1.10.1*.

A unidade VPU que irá ser utilizada é uma *Neural Compute Stick* (NCS) desenvolvida pela marca *Intel*. A NCS é uma *Pendrive* que pode ser conectada a uma porta USB 2.0 ou USB 3.0 e que contem uma arquitetura VPU integrada. A arquitetura VPU integrada contém 4 Gbits de LPDDR3 DRAM e aceleradores de visão e imagem. Para se utilizar a NCS é necessário a instalação de uma biblioteca, a *NCSDK*. A biblioteca *NCSDK* permite a conversão de ficheiro em *Tensorflow* ou *caffe* para um formato compatível com a NCS e permite também utilizar um conjunto de funções para executar os ficheiros convertidos na NCS. A versão instalada no sistema operativo foi a *NCSDK 2.0*.

A câmara utilizada foi a *Webcam* integrada no portátil ASUS X Series X550CA que corresponde ao modelo Asus CAMERA HD FIX 3.3V A MIC CL.

Capítulo 5

Implementação e resultados

Neste capitulo será apresentado e descrito a implementação dos vários blocos da arquitetura mencionada no capitulo 4 e os resultados obtidos durante as várias fases de implementação e do sistema final. Na primeira fase será implementado o bloco Deteção de cara. De seguida os blocos *Facetraking*, Determinação de idade e género e Determinação do estado emocional. Na fase final é feito o bloco Associação de dados. Os resultados obtidos em cada bloco serão analisados e por fim será analisado o resultado geral do sistema. As várias secções descritas aqui são ativadas em cadeia, ou seja o envio de um *Frame* ativa o primeiro bloco. Se existirem caras são ativados os restantes blocos que enviam informação para os seguintes. O ciclo repete-se de *Frame* em *Frame*. Nos blocos em que são utilizados redes neuronais, antes do início da captura das imagens da câmara ou de um ficheiro de vídeo é realizado a criação da estrutura/arquitetura da respetiva rede e é carregado o ficheiro treinado da respetiva rede. Depois disto as redes estão prontas a ser utilizadas pelos respetivos blocos.

5.1 Deteção do rosto

O primeiro bloco a ser implementado é o bloco Deteção de cara. Neste bloco é efetuada a deteção da imagem da câmara. A imagem é redimensionada para o formato 320×180. A imagem redimensionada é envia para a rede. A rede utilizada é uma rede criada em *caffe model* com a mesma arquitetura da rede no artigo [8]. A rede desenvolvida utiliza os modelos treinados obtidos pelos treino documentado no mesmo artigo [8]. A rede indica a localização das caras na imagem. A localização das caras são os 4 pontos que definem o retângulo onde está a cara na imagem.

Depois de obtidas a localização das várias caras na imagem, estas são guardadas por

ordem numa matriz. O fluxograma da figura 5.1 mostra o funcionamento do bloco.

Figura 5.1: Fluxograma do bloco Deteção de cara.

O fluxograma repete-se todas as vezes que existe um novo *Frame* da câmara. Caso não sejam detetadas caras na imagem a matriz que guarda a localização das caras fica vazia e não é ativado o inicio dos blocos seguintes. A figura 5.2 mostra um exemplo de código utilizado neste bloco. Na primeira linha é realizada a captura de um *Frame*. Na segunda linha é realizado o redimensionamento do *Frame* capturado. Na terceira linha é enviada a imagem para a rede e esta retorna as várias posições das faces na imagem.

```
ret, frame = cap.read()
draw = cv2.resize(frame, (320,180))
results = mxmd.detect_face(draw)
```

Figura 5.2: Exemplo de código utilizado no bloco de deteção de rostos.

Um dos aspetos mais importantes a verificar é a distância máxima à câmara a que

a rede consegue detetar caras. A distância máxima obtida, com boas condições de luz, é de 4.50 metros , o que é um valor satisfatório. A imagem 5.3 mostra um exemplo de

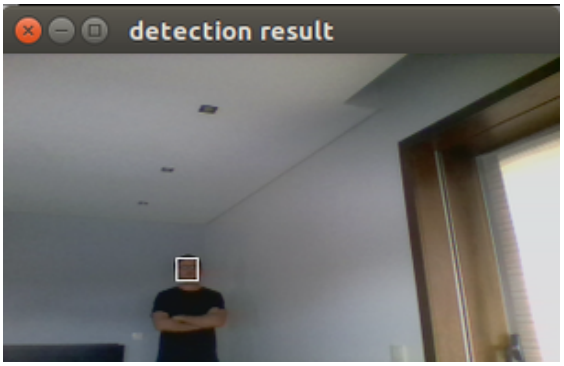
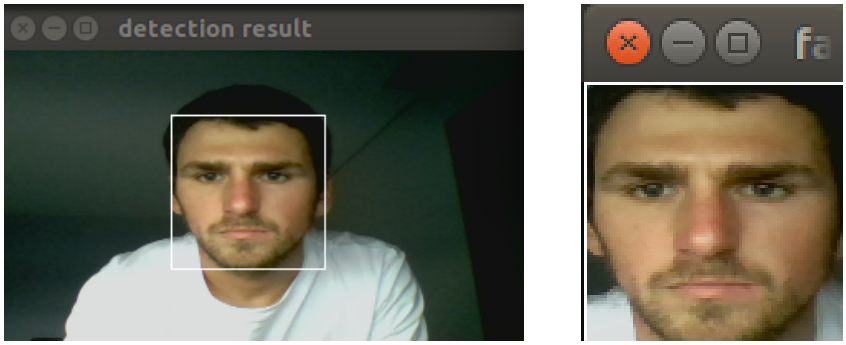


Figura 5.3: Resultado obtido no teste de distância do bloco deteção de cara.

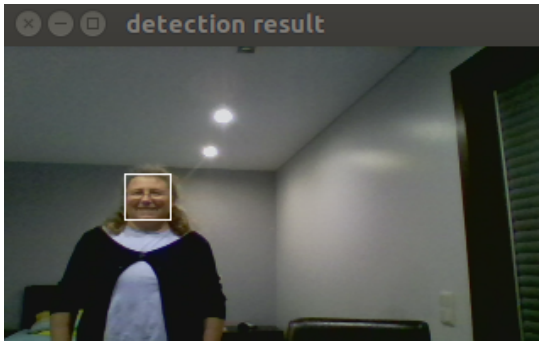
uma deteção feita aproximadamente a 4.5 metros de distância da câmera. De seguida foram realizados testes a várias distancias e analisados a qualidade da imagem da cara. A qualidade da imagem da cara é um fator importante e tem impacto no resultado dos restantes blocos.

Na figura 5.4 é possível verificar que a imagem da cara tem bastante qualidade e é possível ver todos os traços da cara. Visualmente é possível identificar facilmente características tais como idade, género e estado emocional. De seguida foi realizado um teste a uma distancia de 2.3 metros. Na figura 5.5 é possível verificar que a qualidade da imagem baixou bastante e já não se consegue ver alguns detalhes da cara. Neste caso, visualmente é mais complicado identificar as características pretendidas. Por fim realizou-se um teste a uma distância mais próxima da distância limite(4.5 metros). A

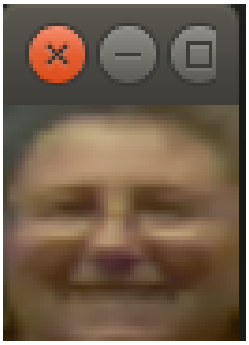


(a) Resultado 1 (b) Cara do resultado 1

Figura 5.4: Resultados obtidos no bloco a um distância de 0.5 metros

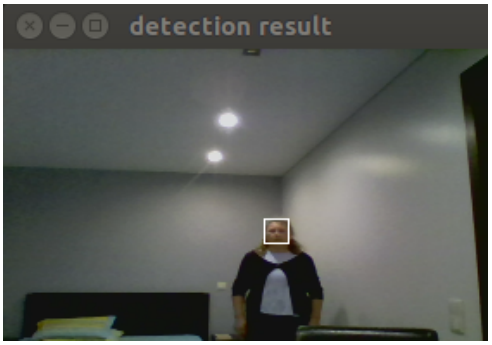


(a) Resultado 2

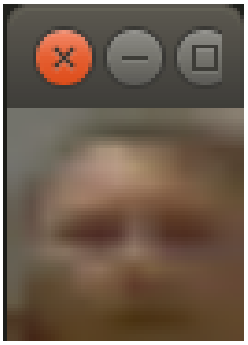


(b) Cara do resultado 2

Figura 5.5: Resultados obtidos no bloco a um distância de 2.3 metros



(a) Resultado 3



(b) Cara do resultado 3

Figura 5.6: Resultados obtidos no bloco a um distância de 4.3 metros

distância neste teste foi de 4.3 metros Na imagem 5.6 a qualidade é muito reduzida e não se consegue visualizar os detalhes da cara. Visualmente não se consegue identificar as características pretendidas. Seria possível aumentar a distancia de deteção de caras e obter caras a uma distancia mais longa com melhor qualidade, para tal seria necessário o uso de uma câmara de espetro visível melhor. Uma câmara de espetro visível com maior alcance e uma melhor resolução iria ter um impacto positivo nos resultados deste bloco e dos restantes.

5.2 Determinação da idade e género

Depois de se ter obtido as várias caras, é efetuada a determinação da idade e do género. O bloco recebe as várias posições(x,y) no plano da imagem detetadas anteriormente. A matriz com as posições é lida e para cada uma das faces é realizado o seguinte procedimento. Primeiro, a face é recortada da imagem. A face que é recortada é a face completa (orelha e cabelo) pois existem fatores que são importantes para as determinações realizadas aqui. De seguida a face é inserida na rede e esta determina a idade e o género. Por fim os valores são guardados. A rede construida para este bloco tem a mesma arquitetura da rede do artigo [11] e foi construida utilizando *Tensorflow*. O modelo de treino usado é o mesmo que foi utilizado no artigo. O fluxograma da figura 5.7 ilustra o que acontece neste bloco.

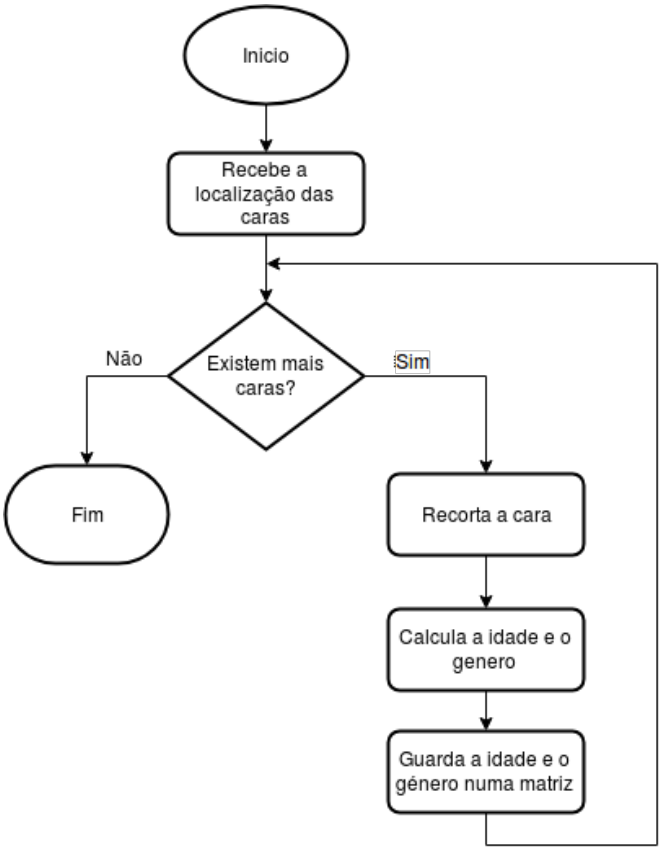


Figura 5.7: Fluxograma do bloco Determinação de idade e género.

A ordem com que é guardado os valores obtidos deste bloco é a mesma ordem que tem a matriz com as localizações, ou seja o último valor guardado corresponde à última

face. Deste modo será mais fácil efetuar a associação de dados. A figura 5.8 mostra um exemplo do código desenvolvido para este bloco. Na figura 5.8 é realizado o calculo

```
results_ag = self.model.predict(face_imgs)
predicted_genders = results_ag[0]
ages = np.arange(0, 101).reshape(101, 1)
predicted_ages = results_ag[1].dot(ages).flatten()
```

Figura 5.8: Exemplo de código utilizado no bloco de determinação de idade e género.

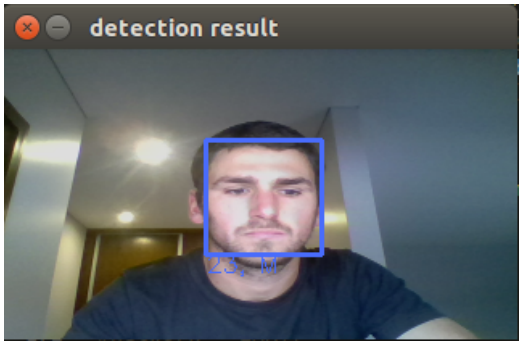
para determinação de idade e género e de seguida os valores são guardados. Na primeira linha é enviado para o modelo que foi inicializado previamente e este irá realizar uma previsão. A previsão irá indicar o valor da idade com maior probabilidade e irá indicar um valor de probabilidade para o género. A imagem 5.9 mostra os resultados obtidos neste bloco.

Na imagem é possível verificar que os resultados são próximos do valor real que é 25 anos (23 anos a menos de um metro e 27 a 3 metros de distância) e o género é o correto. Os valores obtidos para a idade alteraram um pouco com a distância. Em ambos os casos o imagem da cara tem o mesmo tamanho e formato, 64×64 logo a variação da idade foi por causa de dois fatores. Os dois fatores são a variação de luminosidade e da variação da qualidade da imagem entra a figura 5.9(a) e a figura 5.9(b). A variação de luminosidade tem um grande impacto neste bloco. A luminosidade pode levar a uma mudança na aparência da cara e deste modo levar a rede a atribuir uma valor para a idade diferente. Por fim realizaram-se também testes com pessoas de outros género e com varias caras em simultâneo.

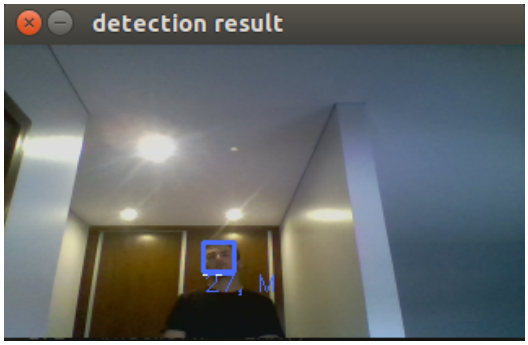
Na figura 5.10 é possível verificar uma correta atribuição do género nas duas caras e uma correta atribuição da idade no caso da cara masculina. No caso da cara feminina a idade indica é apenas uma ano menos, ou seja a pessoa feminina na imagem tinha 24 anos. Na imagem é possível verificar que a foto foi tirada num ambiente com boa luminosidade o que teve um impacto positivo nos resultados obtidos. Em distâncias superiores a 3 metros de distancia a qualidade da imagem não era a mais indicada o que levava a rede a atribuir um valor de idade entre os 35 e 40 anos.

Por fim realizaram-se teste onde se fixou um cara na imagem por 30 segundo e se verificou o género e a idade atribuído pela rede. A rede desenvolvida leu imagem de um vídeo com poucas movimentações na face.O rosto usado para este teste é de um homem de 25 anos. A imagem 5.11 mostra um *Frame* do vídeo.

O teste realizado permitiu a obtenção de 818 *Frames*. Os resultados obtidos foram representados em gráficos. O gráfico da figura 5.12 mostra os resultado obtidos para o



(a) Resultado obtido a 0.5 metros com um *Output* de 23 anos e do sexo masculino



(b) Resultado obtido a 3 metros com um *Output* de 27 anos e do sexo masculino

Figura 5.9: Resultados obtidos no bloco

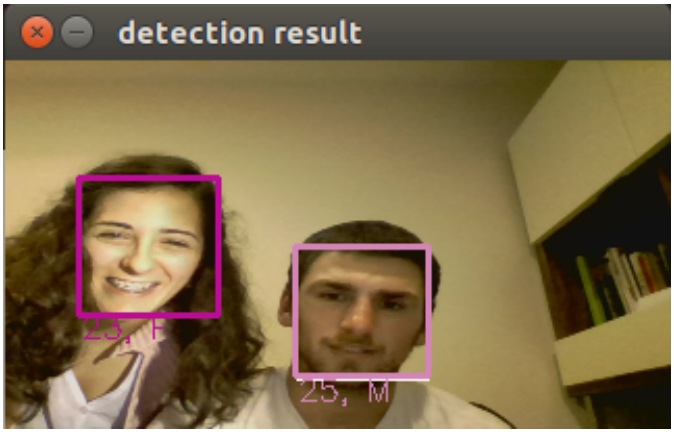


Figura 5.10: Resultados obtidos com várias caras

género. No gráfico da figura 5.12 verifica-se que em todos os *Frames* houve uma atribuição correta do género. A imagem 5.13 mostra os resultado obtidos para a atribuição de idade.

O histograma da figura 5.13 mostra que na maioria dos casos foi atribuída uma idade próxima dos 29 anos. A idade real é 25 anos temos assim diferença de 4 anos do valor real. A idade atribuída pela rede é uma idade aparente e apenas baseada na imagem.

Figura 5.11: *Frame* obtido do teste realizado à idade e ao género.

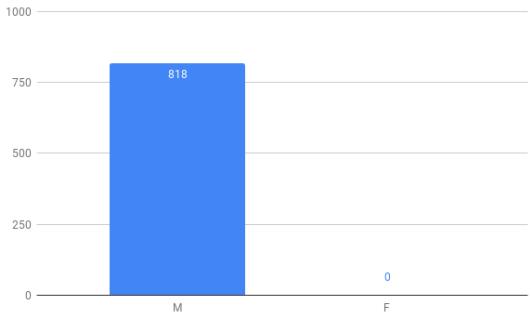


Figura 5.12: Resultados obtidos no teste realizado ao género.

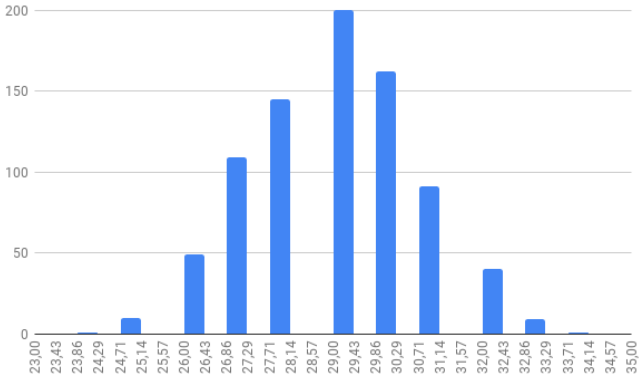


Figura 5.13: Resultados obtidos no teste realizado á idade.

5.3 Determinação do estado emocional

A matriz com a posição das faces é também utilizado para realizar a determinação do estado emocional. O bloco recebe a matriz com as localizações das faces e é efetuada a leitura das várias mesmas. Para cada uma das posições é realizado os seguintes procedimentos. Inicialmente é recortada a cara da imagem. Neste caso é apenas recortado o centro da cara, pois o restante é considerado ruído e poderá interferir com os resultados obtidos. De seguida, a cara é inserida na rede e é obtido os vários valores de probabilidades para cada estado emocional com que a rede foi treinada (feliz, triste, neutro, surpreso, enervado, desgosto e com medo). Por fim é guardado o valor com maior probabilidade numa matriz. A rede criada para este bloco tem a mesma arquitetura que a rede do artigo [9] e foi construida utilizando *Tensorflow*. Nesta rede foi utilizado o mesmo modelo treinado obtido no artigo. O fluxograma da figura 5.14 demonstra o funcionamento deste bloco.

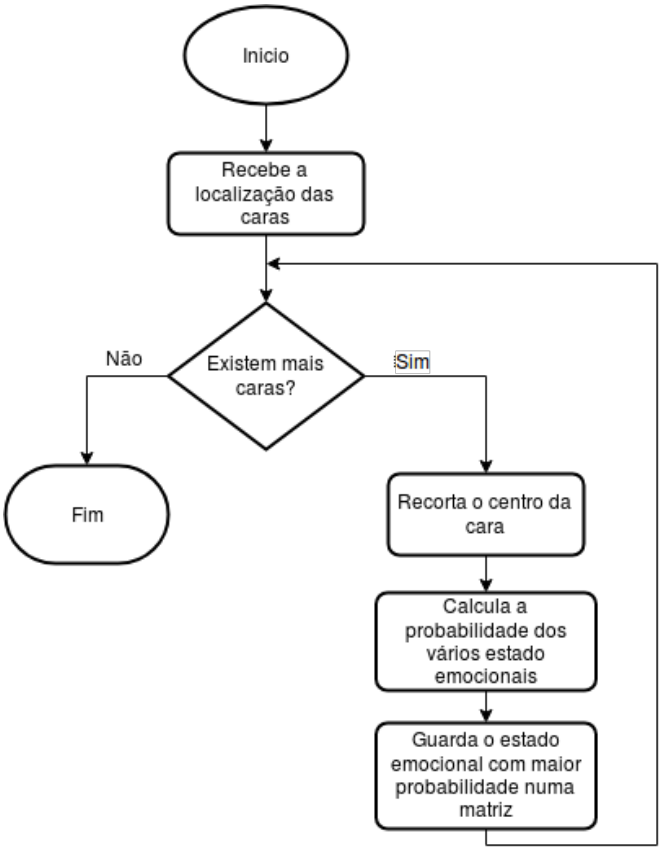


Figura 5.14: Fluxograma do bloco Deteção de cara.

Tal como no bloco anterior, os dados obtidos por este bloco são guardados pela mesma ordem. A figura 5.15 mostra um exemplo do código desenvolvido para este bloco onde é calculado a probabilidade dos vários estados emocionais e é guardado o estado com maior probabilidade.

```
result_em = network.predict(image_em)
if result_em is not None:
  maxindex = np.argmax(result_em[0])
emotion_d.append(EMOTIONS[maxindex])
.
```

Figura 5.15: Exemplo de código utilizado no bloco de determinação do estado emocional.

Na primeira linha do código da figura 5.15 é enviado para o modelo que determina o estado emocional uma imagem da cara e este retornas os vários estados e os respetivos estados emocionais. Na segunda e terceira linha é verificado qual o estado com probabilidade mais alta. Na ultima linha é guardado valor com probabilidade mais alta. A figura 5.16 mostra os resultados obtidos a distancias curtas (entre 0.5 metros e um 1 metro).

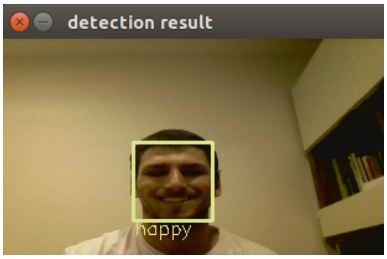
Aqui é possível verificar que mesmo com má luminosidade na figura 5.16(a) existe uma atribuição correta do estado emocional. Do mesmo modo. na figura 5.16(b) existe um atribuição correta. De seguida realizou-se testes a uma distância superior (entre 2 metros e 3 metros), representado na figura 5.17.

Aqui é possível verificar que apesar de as imagens serem a uma distancia maior que nos restantes teste continua a haver uma atribuição correta de estado emocional. A luminosidade no cenário utilizado era indicada e ajudou para os bons resultados obtidos na identificação do estado emocional. Por fim realizou-se testes a uma distância intermédia (entre 1 metro e 2 metros), tal como é possível ver na figura 5.18.

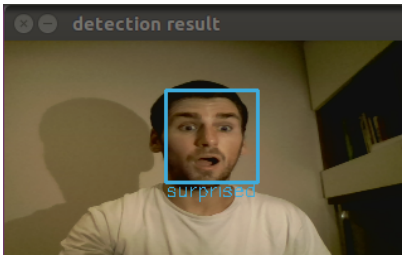
Neste teste verificou-se também um atribuição correta mesmo com uma mudança de género da pessoa. A integração do bloco de determinação do estado emocional foi bem sucedido e não se verificou um grande impacto nos resultados com as caras identificadas à distancia que estava entre 2 metros e 3 metros. Em distancias superiores a 3 metros, a qualidade da imagem da cara não era a mais indicada e a rede atribuía o estado emocional neutro em todas as situações.

Por fim realizaram-se teste onde se fixou um cara num estado emocional por dez segundo e se verificou o estado emocional atribuído pela rede. o primeiro estado a ser testado foi o estado emocional Feliz. A rede desenvolvida leu imagem de um vídeo com poucas movimentações na face. A imagem 5.19 mostra um *Frame* do vídeo.

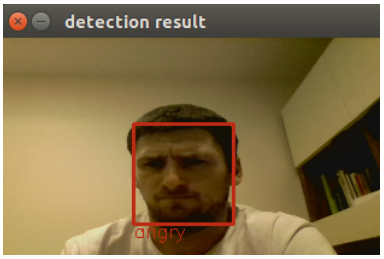
O vídeo usado tinha onze segundos e 357 *Frames*, para cada *Frame* registou-se o



(a) Feliz

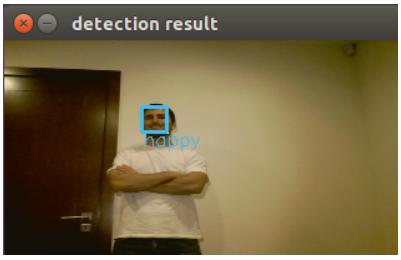


(b) Surpreso

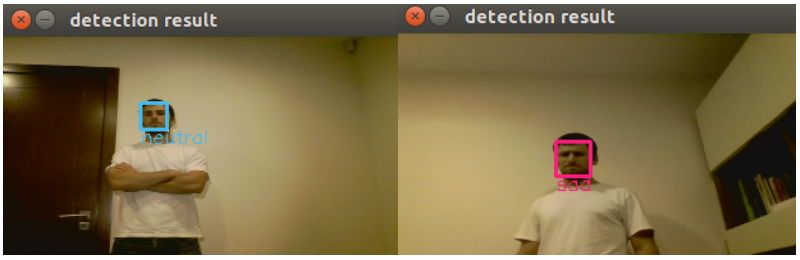


(c) Nervoso

Figura 5.16: Resultados obtidos a um distancia curta(entre 0.5 metros e um 1 metro)

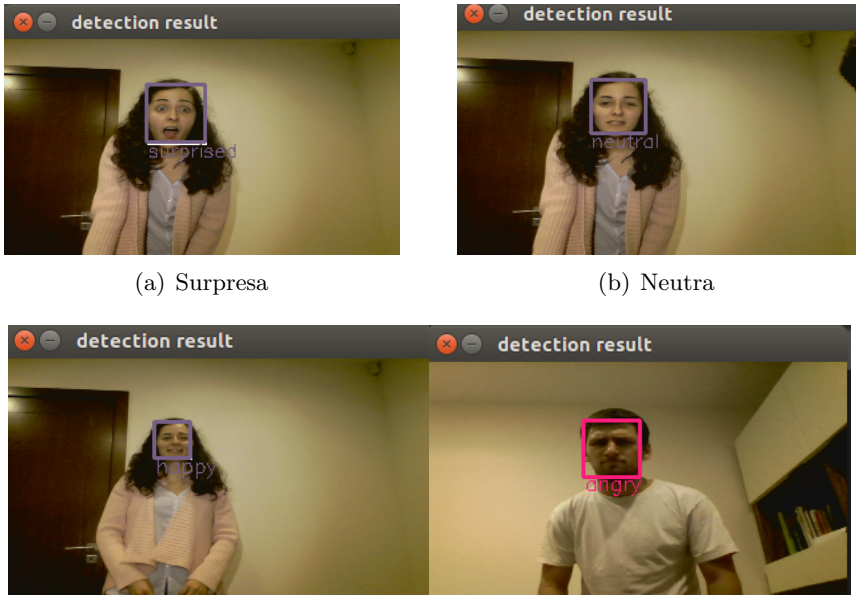


(a) Feliz



(b) Neutro (c) Triste

Figura 5.17: Resultados obtidos a um distância longa(entre 2 metros e 3 metros)



(a) Surpresa (b) Neutra

(c) Feliz (d) Nervoso

Figura 5.18: Resultados obtidos a um distância intermédia(entre 1 metro e um 3 metros)

Figura 5.19: *Frame* obtido do teste realizado ao estado emocional feliz.

estado emocional detetado. A figura 5.20 mostra o gráfico com os resultados obtidos. Nesta figura pode-se verificar que em todos os *Frames* o estado obtido foi o estado feliz. De seguida realizou-se o mesmo teste mas para o estado neutro. O vídeo utilizado tinha a duração de 12 segundos e um total de 74 *Frames*. A imagem 5.21 mostra um gráfico com os resultados obtidos neste teste.

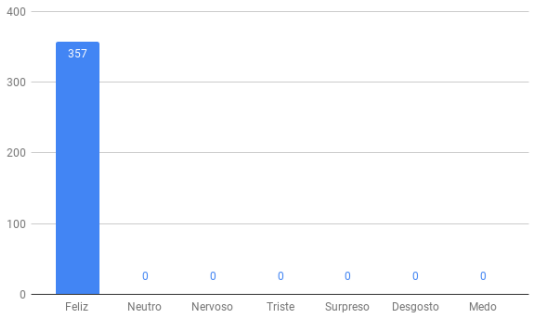


Figura 5.20: Resultados obtidos no teste realizado ao estado Feliz.

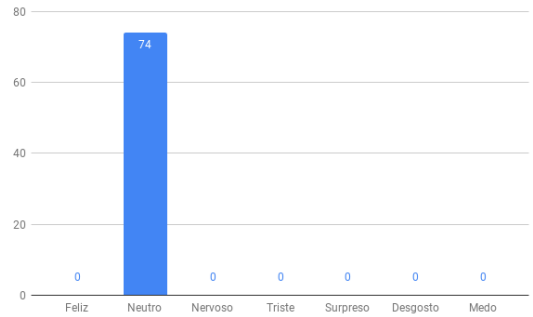


Figura 5.21: Resultados obtidos no teste realizado ao estado Neutro.

O gráfico da figura 5.21 mostra que em todos os *Frames* foi detetado o estado Neutro.

5.4 Facetracking

De seguida a localização das faces é enviada para o bloco *Facetracking*. Aqui é efetuada a leitura da matriz e para cada face são efetuados os seguinte procedimentos. Primeiro é efetuada a verificação, se a posição da imagem está dentro de alguma região de interesse. A região de interesse é uma área da onde foi localizada uma face e foi atribuído um ID se voltarem a aparecer nos próximos *Frames* nesta zona é associada a esta região de interesse e é atribuído o mesmo ID. Se a localização da cara pertencer a alguma região de interesse, então é atribuído o mesmo ID e de seguida é recalculada e guardada a nova região de interesse desse mesmo ID. Se a cara não pertencer a nenhuma região de interesse então é criado um novo ID e é calculado e guardado uma região de interesse para este ID. Por fim é verificado quais são os ID que não tiveram correspondência e é verificado se estes ultrapassaram um numero de *Frames* sem haver correspondência. Se sim, o ID e a região de interesse é apagado da lista. Se não, o ID volta a ser guardado.

O fluxograma da figura 5.22 mostra o funcionamento deste bloco. Por fim, temos uma lista com todos os ID's (os da imagem e os que não tiveram correspondência mas ainda poderão vir a ter). A lista com todos os ID's irá ser utilizada no próximo *Frame* nas duas verificações que são feitas e na imagem irá ser utilizada para o bloco associação de dados para efetuar associação de dados. Na figura 5.23 é possível ver-se uma parte do código desenvolvido que é responsável por receber as várias posições dos rostos, atualizar a lista dos ID e por fim remover os ID sem correspondência. Na primeira linha do exemplo do código da figura 5.23 é criado uma lista que irá auxiliar na verificação e associação de ID's aos rostos. Esta lista têm um tamanho igual ao número de caras que é recebido nesta função e para cada uma das posições da lista é colocado o valor *True*. Nas 4 linhas seguintes cada uma das faces é enviada para uma outra função *"update"* que irá retornar a correspondência a um ID caso exista, ou o valor -1 caso não exista. Na posições em que já existe correspondência não será necessário mas nas restantes é necessário criar um novo ID. Quando não é necessário criar um ID para esse rosto então o valor da lista nessa posição é alterado para *False*. Nas 5 linhas seguintes é verificado na lista todas as posições da lista que têm o valor *True*, para cada uma delas é criado um novo ID e estas são guardadas na lista dos rostos para uma verificação futura. Por fim é chamada a função *"removeFaces"* que irá verificar na lista dos rostos quais são aqueles que já não têm correspondência à varias ciclos e esses rostos apaga-os da lista. A figura 5.24 mostra um sequência de imagens que foram os resultados obtidos neste bloco.

Na figura 5.24(a) vemos a atribuição do primeiro ID e de seguida aparece um outro rosto à qual é atribuído um novo ID(figura 5.24(b)). De seguida o rosto desaparece mas

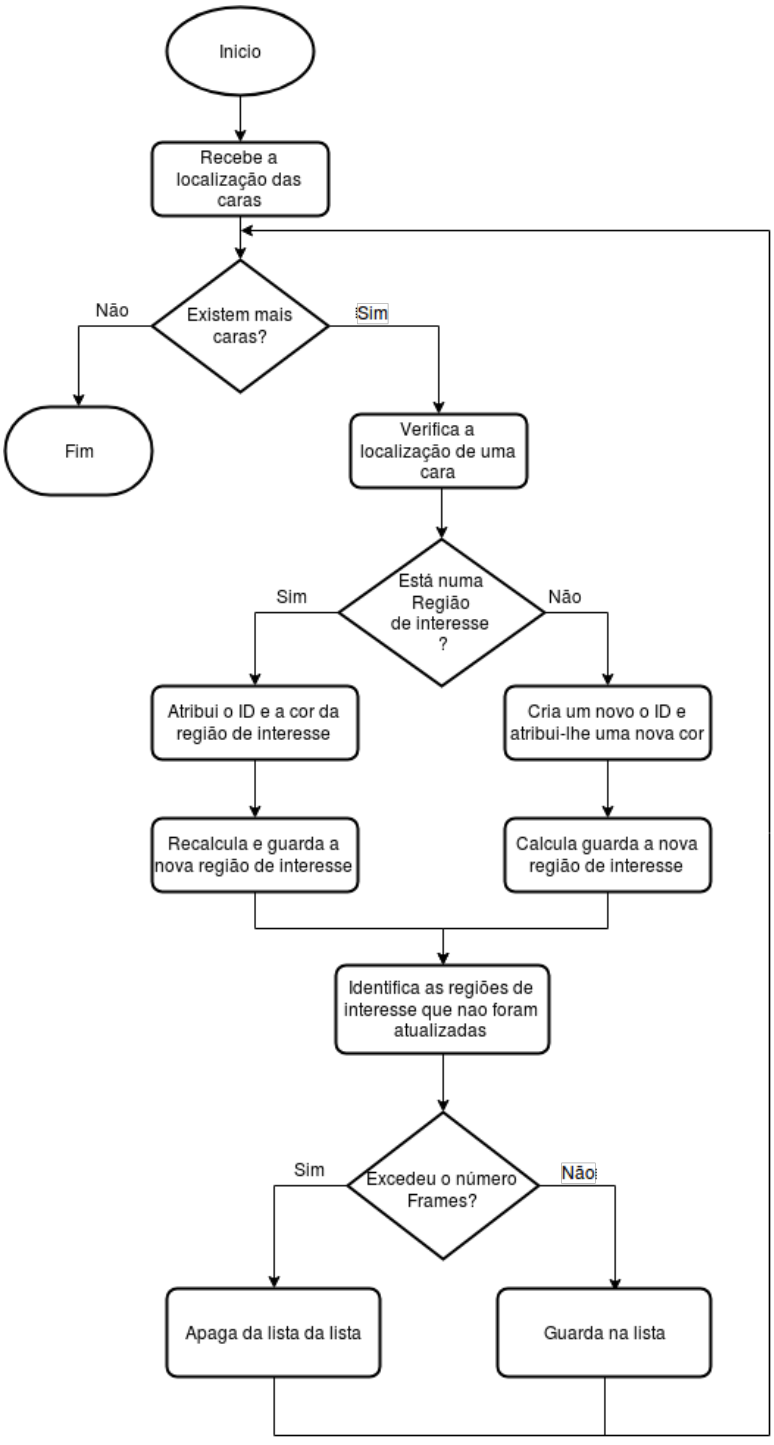
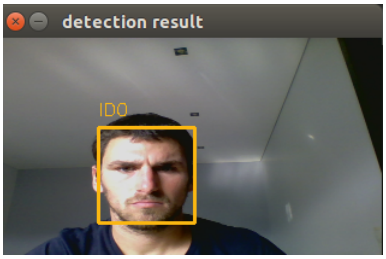


Figura 5.22: Fluxograma do bloco *Facetracking*.

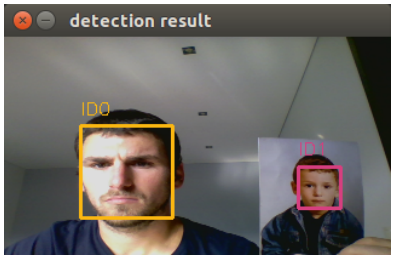
```
def update(self, newFaces):
    toAdd = [True] * len(newFaces)
    for i, f in enumerate(self.faces):
        v = f.update(newFaces)
        if v != -1:
            toAdd[v] = False
    for i, f in enumerate(newFaces):
        if toAdd[i]:
            nf = f.clone()
            nf.id = self.newId()
            self.faces.append(nf)
    self.removeFaces(5)
```

Figura 5.23: Exemplo de código utilizado no bloco de *Facetracking*.

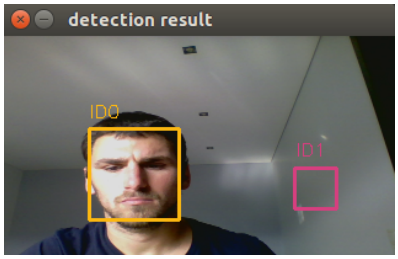
no *Frame* continua a aparecer o retângulo do rosto até que desaparece(figura 5.24(c) e 5.24(d)). Por fim o rosto volta a aparecer, depois de alguns instantes (10 segundos) e é então atribuído um novo ID (figura 5.24(e)).



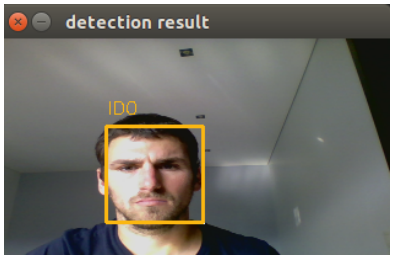
(a) Primeira imagem da sequência



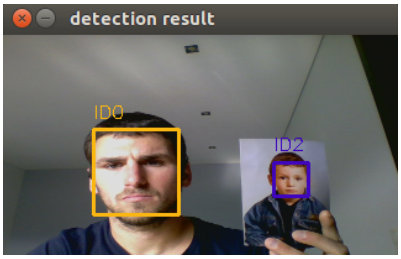
(b) Segunda imagem da sequência



(c) Terceira imagem da sequência



(d) Quarta imagem da sequência



(e) Ultima imagem da sequência

Figura 5.24: Sequência de imagens que mostra os resultados obtidos no bloco *Facetracking*

5.5 Associação de dados

Por fim o bloco de Associação de dados recebe todas as informações geradas pelos blocos anteriores e agrupa-as. O bloco inicialmente recebe a posição das caras geradas no bloco Deteção de cara, depois a matriz com as idades e os géneros e por último a matriz com os ID e a informação associada a este. De seguida, a informação recebida é associada, ou seja é percorrida a matriz com os ID e é verificado qual a cara correspondente na matriz obtida no bloco Deteção de cara. A correspondência é feita através da posição da na imagem. Depois de se saber qual é a face na matriz das posições é verificado na restantes matrizes quais são os dados do género, idade e estado emocional que correspondem a essa cara. Este processo é simples pois, tal como mencionado anteriormente, os valores destas matrizes são guardados com a mesma ordem da matriz com as posições das faces. Por exemplo, é verificado que o ID 0 corresponde à terceira cara da matriz das caras, nas restantes matrizes basta ir á terceira posição e verificar qual o valor que lá está. Por fim temos uma matriz com toda a informação junta, os vários ID, a cor associada, a localização da cara, a idade, o género e o estado emocional.

O fluxograma da figura 5.25 mostra o funcionamento deste bloco.

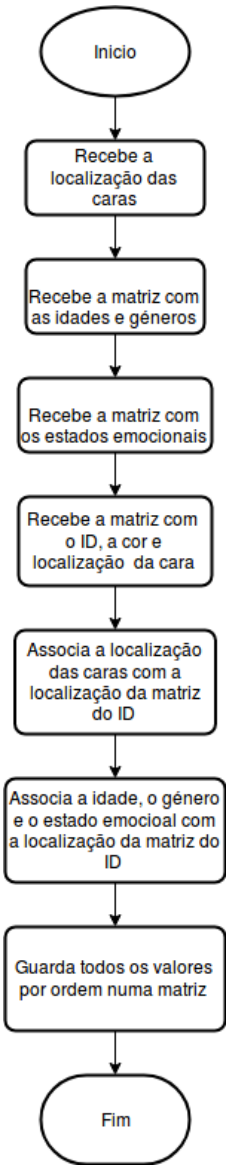
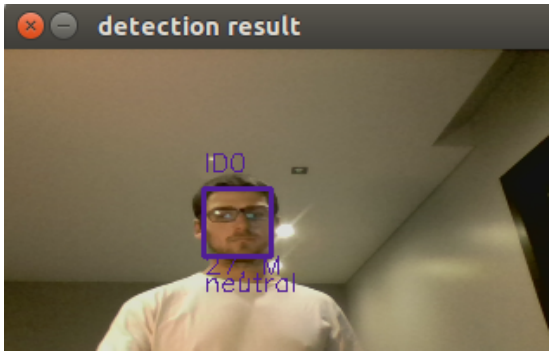


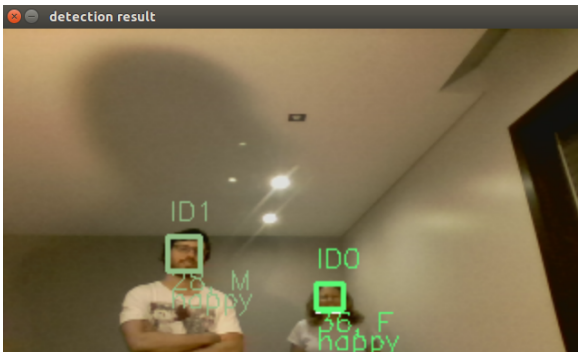
Figura 5.25: Fluxograma do bloco Associação de dados.

5.6 SeeGAgEmotion

Por fim foram feitos testes gerais ao sistema e foi analisada a performance do sistema. Nestes testes forem verificados o comportamento do *Facetracking*, os resultados com várias distâncias na mesma imagem e o resultados de geral da determinação da idade, género e estado emocional. Na figura 5.26 mostra os resultados obtidos nos testes realizados com uma só pessoas e duas pessoas na imagem. A imagem 5.26(a) é o resultado

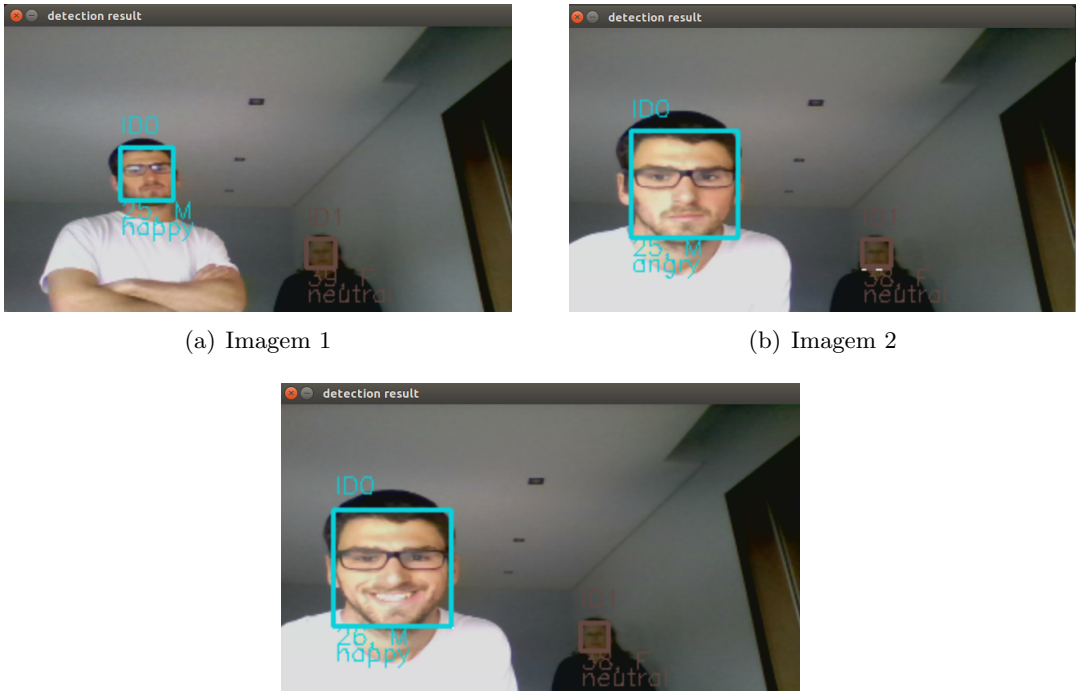


(a) Resultado 1



(b) Resultado 2

Figura 5.26: Resultados obtidos ao sistema



(a) Imagem 1

(b) Imagem 2

(c) Imagem 3

Figura 5.27: Resultados obtidos ao sistema no segundo teste

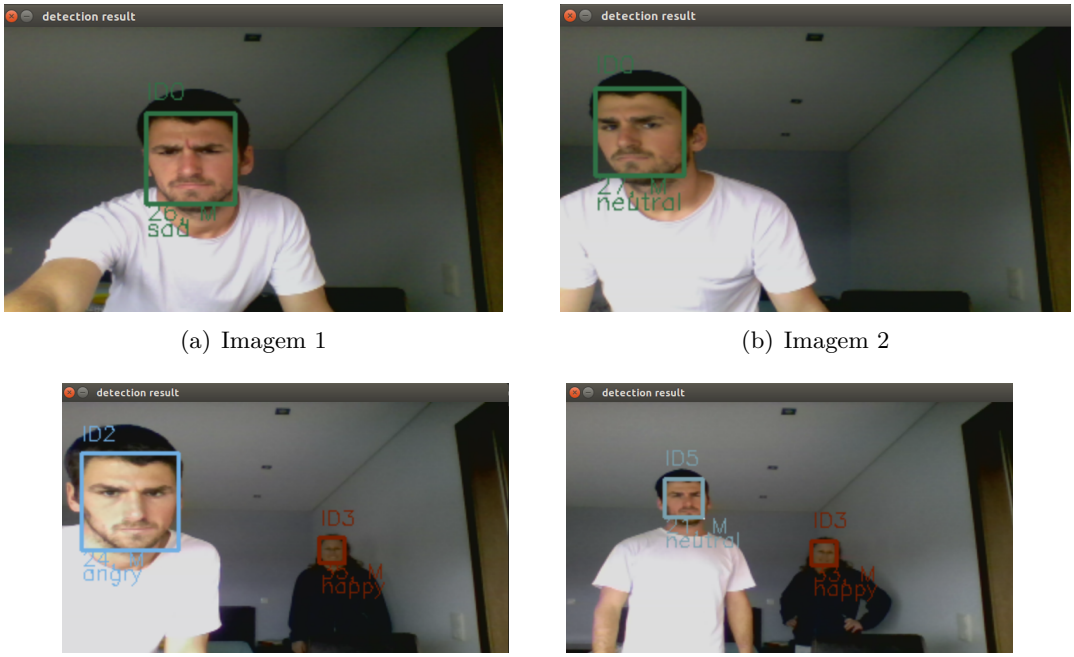
obtido a um distancia curta num local bem iluminado. Nesta imagem verifica-se uma atribuição correta de ID, género, estado emocional e a idade está apenas a mais dois anos do valor real. Na figura 5.26(b) estão duas pessoas a uma distância entre os 2 metros e os 3 metros e também nesta imagem verifica-se atribuição correta dos dados.

De seguida realizaram-se testes nos quais uma das pessoas afastava-se e aproximava-se da imagem e outra mantinha-se mais distante (figura 5.27).

Na figura 5.27 verificou-se que o *Facetracking* funcionou corretamente e houve uma atribuição correta de dados, havendo uma pequena oscilação na idade. Há um exceção na primeira imagem (imagem 5.27(a)) em que na cara com ID0 tem o estado emocional *Happy* (Feliz) quando o estado real deveria ser neutro. Isto acontece pois por vezes dois estados podem ter probabilidades muito próximas mas a escolhida é a com maior valor. Por vezes por pequenas diferenças de valores poderá ser escolhido o estado errado.

Em seguida foram realizados teste em que as pessoas entravam e saiam da imagem e mesmo quando se mantinham na imagem alteravam a sua posição (Figura 5.28).

Nas imagens da figura 5.28 verificou-se que por vezes se as pessoas se moverem demasiado depressa ficam fora da região de interesse do ID anterior e então é atribuído



(a) Imagem 1

(b) Imagem 2

(c) Imagem 3

(d) Imagem 4

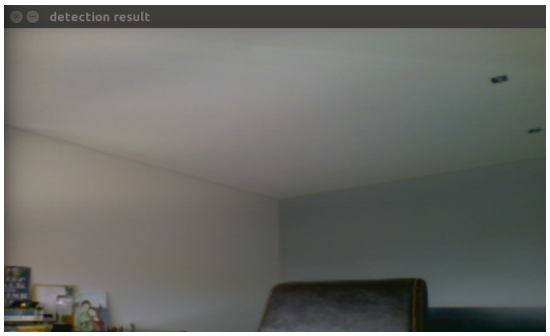
Figura 5.28: Resultados obtidos ao sistema no segundo teste

um novo ID (figura 5.28(c) e figura 5.28(d)).

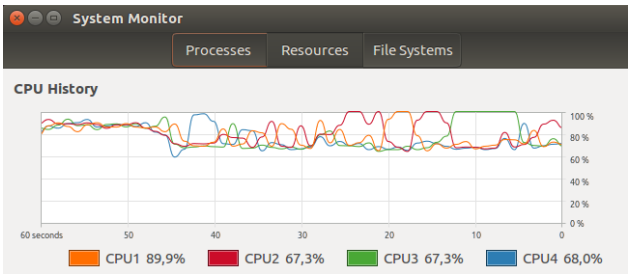
Nos vários teste realizados consegui-se verificar que na maioria dos casos houve uma atribuição correta de estados e características, por vezes existiam variações de luminosidade no ambiente externo que provocavam atribuições erradas. Verificou-se que por vezes se duas caras estiverem muito próximas na imagem e uma delas desaparecer então a região de interesse da pessoa que saiu converge para a região de interesse da pessoa que ficou e esta fica com dois ID. Neste caso os restantes os valores/características atribuídas continuam igual pois é a mesma cara que entra na rede.

5.7 Resultados do CPU

De seguida efetuou-se testes com imagens onde se fez variar o numero de caras na imagem e testou-se a resposta do CPU. Inicialmente verificou a resposta do CPU ao sistema total, ou seja, desde a entrada da imagem até à obtenção dos dados de saída do sistema. Depois da realização destes testes, realizou-se teste onde se verificou o tempo que era gasto no CPU em cada um dos blocos. Nestes teste não houve controlo sobre o modo como as tarefas eram atribuídas aos *Cores*. No primeiro teste executou-se o programa e mediu-se



(a) Imagem testada



(b) Gráfico do CPU

Figura 5.29: Testes realizados com uma imagem sem rostos

qual a percentagem de CPU usada quando não haviam caras nas imagens (5.29).

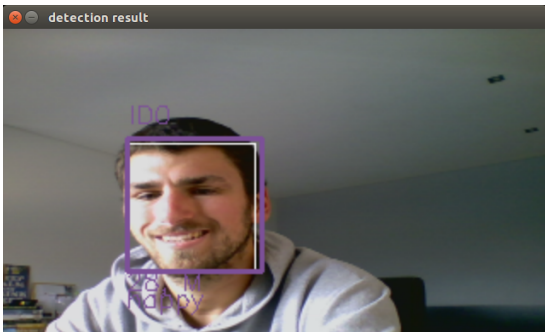
Na figura 5.29(b) verifica-se que dos quatro *Cores* do CPU, o primeiro encontrava-se a 89,9% e os restantes entre 67% e 68%. Neste momento apenas uma rede encontra-se a ser executa que é a rede de deteção de caras. A rede de deteção de caras não identifica caras na imagem logo não ativa os restantes blocos.

De seguida, o programa foi executado com imagens com apenas uma cara(figura 5.30).

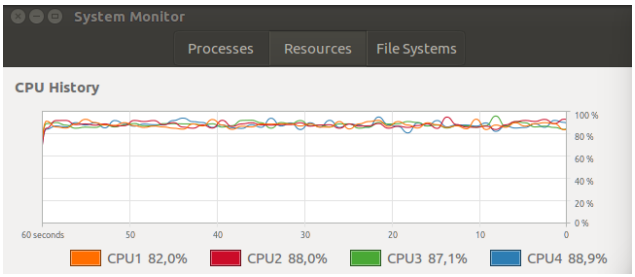
Neste teste verificou-se que a percentagem de CPU usada subiu(figura 5.30(b)). No CPU temos um *Core* com 82% e os restantes com valores entre os 88% e os 89%. Neste caso os restantes blocos foram ativos, o justifica a subida da percentagem. Depois deste teste, foi realizado um teste com imagens com duas caras(figura 5.31).

Neste caso voltou-se a verificar um aumento da percentagem de CPU usado mas apenas num dos *Cores* usados(Figura 5.31(b)). O aumento da percentagem de CPU pode ser justificado pelo aumento de caras. De seguida realizou-se um teste com quatro caras na imagem(5.32).

Neste teste verificou-se um aumento ainda maior da percentagem de CPU usado e os *Cores* passaram a ter valores entre 90% e 95%(figura 5.32(b)). A execução deste teste abrandou o funcionamento do sistema e fez com que o processamento de *Frames*



(a) Imagem testada



(b) Gráfico do CPU

Figura 5.30: Testes realizados com um rosto

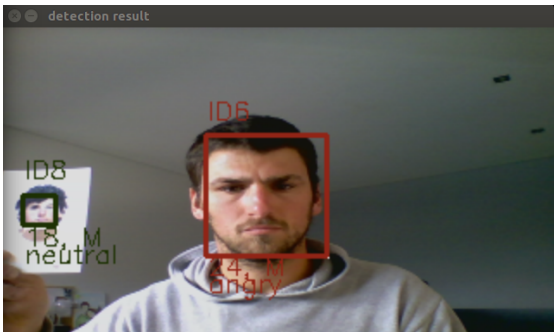
pelo programa fosse mais lento. Por fim, executou-se um teste com doze caras na imagem(figura 5.33).

O último teste mostrou uma subida ligeira na percentagem de CPU usado mas os valores encontravam-se, também, entre os 90% e os 95% (figura 5.33). Verificou-se também que neste teste, o funcionamento do sistema abrandou bastante e cada *Frame* demorava cerca de 4 a 6 segundos para ser analisado.

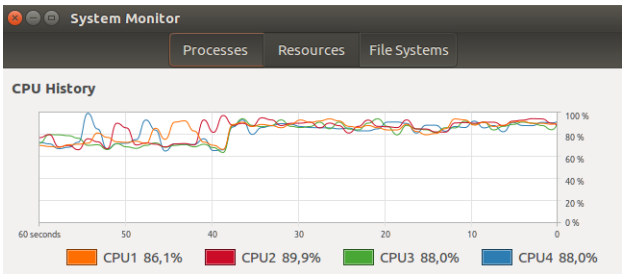
Isto pode ser justificado pelo aumento significativo de caras nas imagens e pelo facto o de o CPU estar a atingir a sua capacidade máxima. Deste modo, a única possibilidade para o sistema seria demorar mais tempo a analisar cada *Frame*.

De modo a conseguir-se analisar de modo transversal todos os testes realizados desenhou-se um gráfico onde se pode verificar a percentagem de CPU usado dependendo do nº de caras na imagem (figura 5.34).

No gráfico da figura 5.34 é possível verificar se que a percentagem de CPU utilizada foi subindo à medida que o número de caras foi aumentando, até atingir a sua capacidade máxima. Quando o número de caras foi igual ou superior a 4, os vários *CPU* estiveram sempre acima dos 90% e foi neste momento que o sistema atingiu a sua capacidade máxima. Quando sistema atingiu a sua capacidade máxima, simultaneamente o tempo



(a) Imagem testada



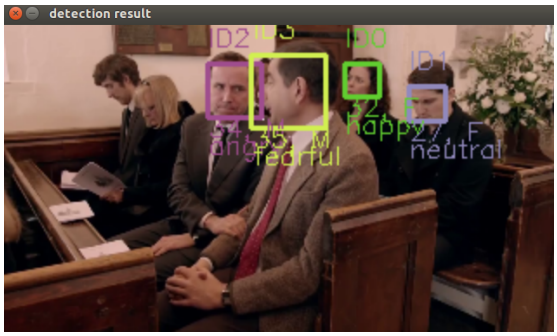
(b) Gráfico do CPU

Figura 5.31: Testes realizados com imagens com dois rostos

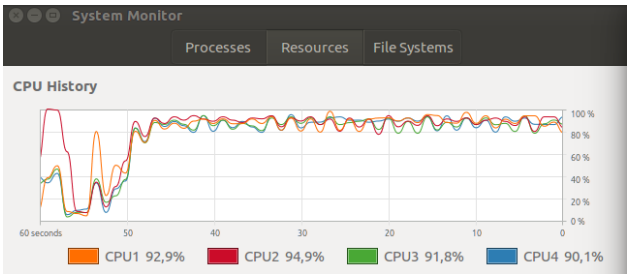
para processar cada imagem aumentou. À medida que o número caras aumentava (de 4 para 12 caras), o tempo para processar a imagem também. Na situação de 4 caras verificou-se que o sistema levava cerca de 1 a 2 segundos para analisar um *Frame* e na situação de 12 rostos o tempo para analisar um *Frame* subiu para 4 a 6 segundos.

De seguida foi analisado o tempo que cada rede neural utilizada para processar os dados. Ou seja o tempo que cada bloco ia ocupar no CPU. O primeiro bloco a ser executado é o bloco deteção de rosto e foi este o primeiro bloco a ser testado. Neste variou-se o número de rostos, começou-se com um rosto na imagem até doze rostos na imagem. O teste permitiu a obtenção dos vários tempos que o bloco Deteção de rosto necessitou para realizar a identificação dos rostos na imagem, para cada número de rostos calculou-se o tempo médio para cada número de rostos. O gráfico da figura 5.35 mostra a relação entre número de rostos e tempo necessário para executar o bloco Deteção de rosto.

No gráfico 5.35 é possível verificar um aumento do tempo à medida que o numero de rostos aumenta. De seguida realizou-se testes para analisar o tempo que era necessário para determinar o estado emocional das faces encontradas nas várias imagens. Neste caso também se fez variar o número de faces nas imagens e também se calculou o tempo



(a) Imagem testada

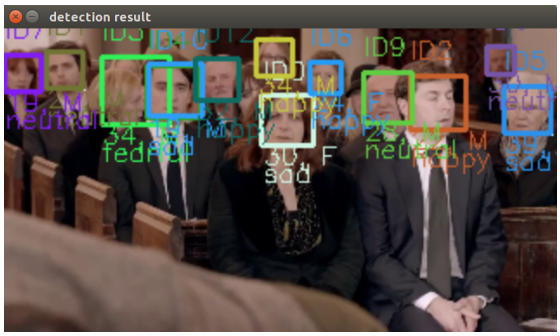


(b) Gráfico do CPU

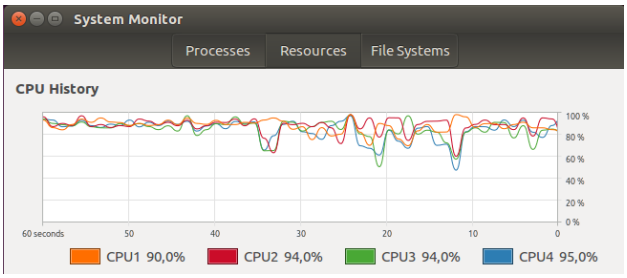
Figura 5.32: Testes realizados com imagens com quatro rostos

médio para cada número de rostos. A figura 5.36 mostra o gráfico obtido.

O gráfico mostra um aumento linear do tempo à medida que aumenta o numero de caras na imagem. O tempo necessário para analisar uma face, independentemente do número de faces na imagem, é aproximadamente 0.02987 segundos. Á medida que o número de faces aumenta na imagem o tempo aproximado para analisar essas faces será 0.02987× nº de faces. De seguida realizou-se testes para se verificar o tempo necessário para determinar o género e a idade das vaŕias faces na imagem. A metodologia utilizada foi idêntica à utilizada nos testes anteriores. A figura 5.37 ilustra os resultado obtidos nos testes realizados. O gráfico da figura 5.37 mostra uma subida de valores à medida que o número de faces aumenta na imagem, é também possível reparar que quando se aumenta de 7 faces na imagem para 12 faces na imagem o tempo não sobe tanto como anteriormente. Comparando os dados obtidos neste bloco com os dados obtidos no bloco anterior(determinação do estado emocional) verifica-se que este necessita de mais tempo. Quando na imagem estão 12 faces o bloco determinação de idade e género demora 3.34 segundos e o bloco determinação do estado emocional demora 0.35 segundos. Depois deste teste, realizou-se um teste onde se pretendia verificar o tempo necessário para executar os blocos todos, ou seja o tempo que passa desde que a imagem entra até



(a) Imagem testada



(b) Gráfico do CPU

Figura 5.33: Testes realizados com imagens com doze rostos

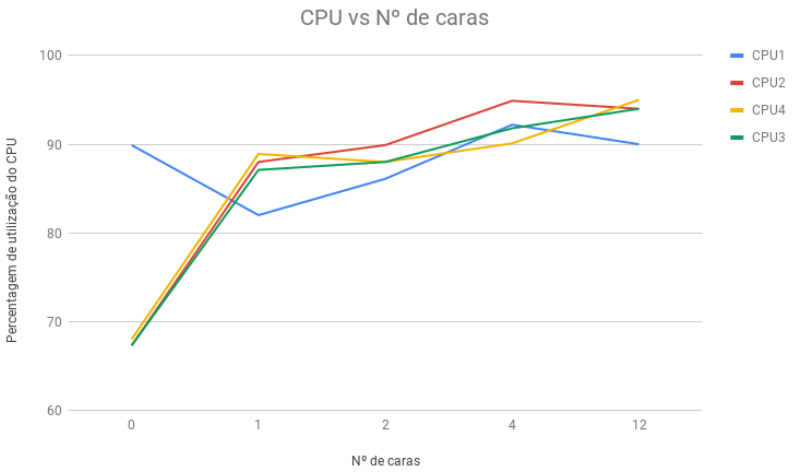


Figura 5.34: Percentagem de utilização de CPU vs Nº de caras.

que se obtém os dados de saída. A figura 5.38 ilustra os resultado obtidos nos testes realizados. No gráfico da imagem 5.38 verifica-se que o tempo aumenta à medida que o numero de faces aumenta na imagem, com exceção entre 4 e 6 faces na imagem. É

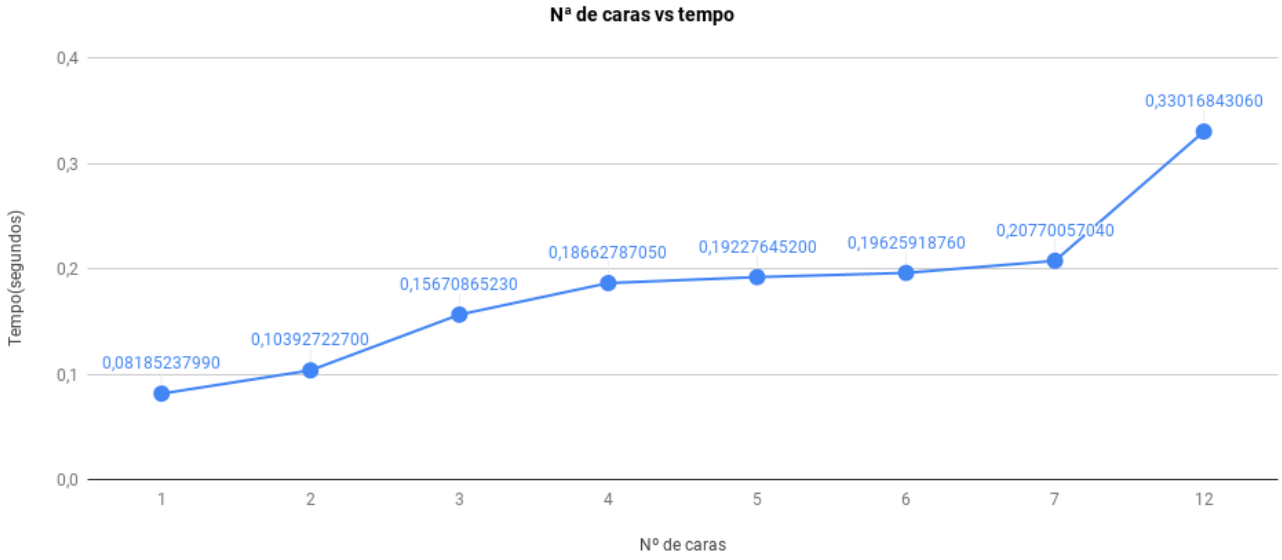


Figura 5.35: Tempo necessário para executar o bloco Deteção de cara variando o nº de rostos na imagem.

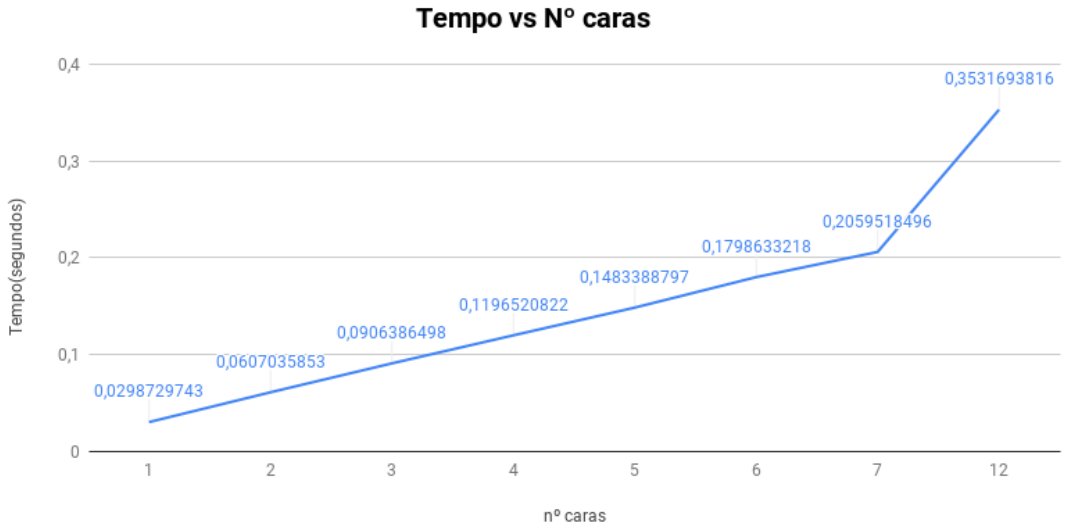


Figura 5.36: Tempo necessário para executar o bloco Determinação de estado emocional o nº de rostos na imagem.

possível verificar um comportamento idêntico ao gráfico da figura 5.37, pois o tempo gasto neste bloco tem um grande impacto no tempo do sistema total.

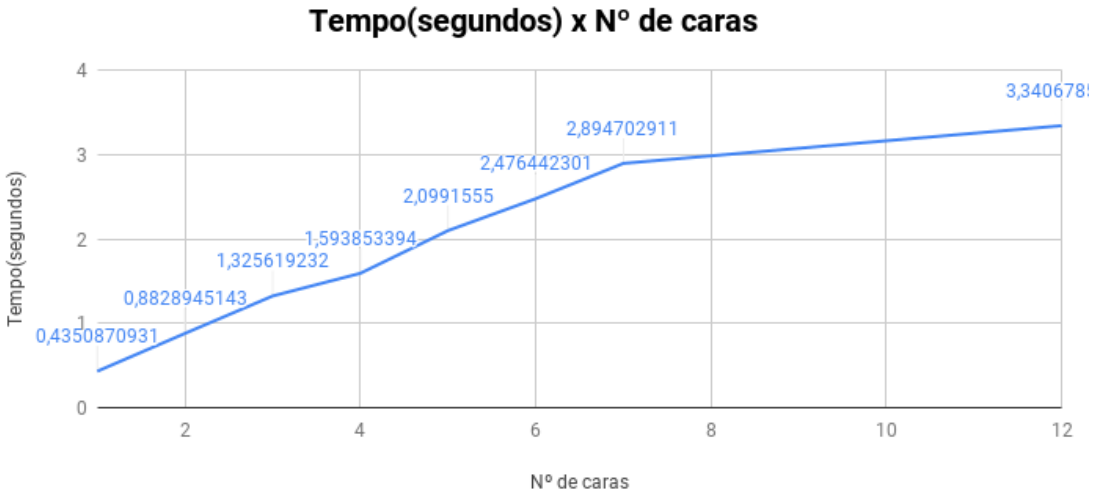


Figura 5.37: Tempo necessário para executar o bloco Determinação de idade e género variando o nº de rostos na imagem.

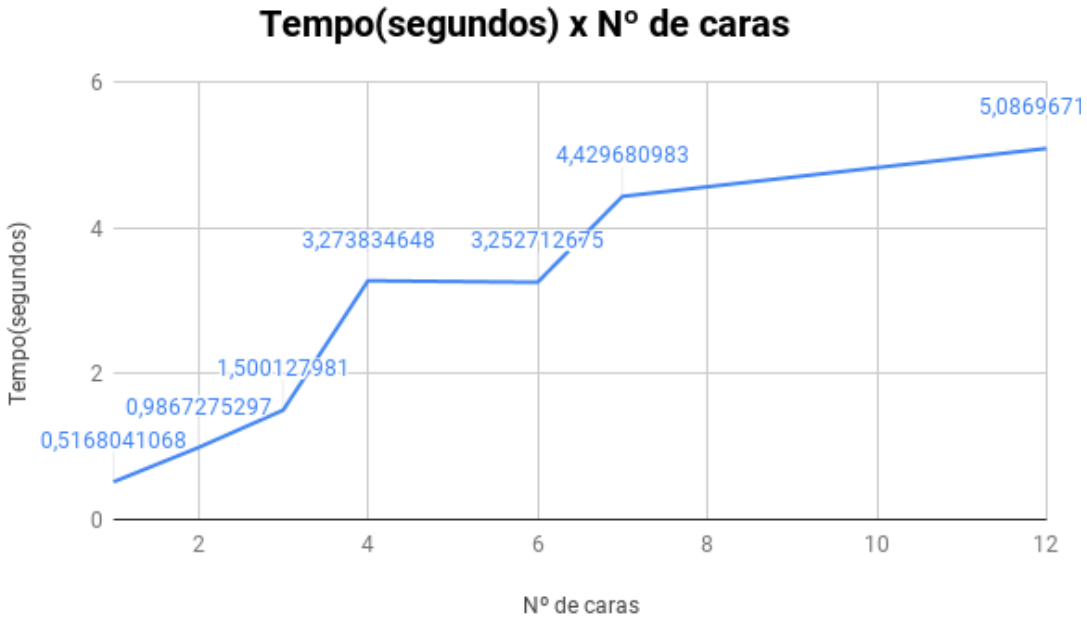


Figura 5.38: Tempo necessário para executar a solução desenvolvida variando o nº de rostos na imagem.

5.8 Resultados e implementação em VPU

Por fim, realizaram-se testes com imagens onde se fez variar o numero de caras na imagem e testou-se a resposta do VPU. Inicialmente verificou a resposta do CPU ao bloco de deteção de rostos, ou seja, desde a entrada da imagem até à obtenção das coordenadas dos vários rostos das imagens. Os testes realizados tinham como objetivo registar o tempo necessário para executar este bloco em VPU. A execução da rede criada neste bloco em VPU exigiu que houvesse uma conversão do ficheiro *Caffe Model* para um ficheiro do tipo graph (que é suportado por este VPU). A conversão é feita através do comando *mvNCCompile*. De seguida foi necessário criar um programa para atribuir ao VPU (ligado por USB) o ficheiro graph criado. O NCS da *Intel* que foi utilizado só permite executar uma rede de cada vez, logo só é possível treinar uma rede de cada vez. A imagem 5.39 mostra um exemplo do código criado.

```
11  devices = mvnc.EnumerateDevices()
12  if len( devices ) == 0:
13      print( 'No devices found' )
14      quit()
15  device = mvnc.Device( devices[0] )
16  device.OpenDevice()
17  # Lê o ficheiro GRAPH
18  with open( graphfile, mode='rb' ) as f:
19      blob = f.read()
20
21  graph = device.AllocateGraph( blob )
22  input_image = cv2.imread('10.jpeg')
23  graph.LoadTensor(input_image.astype(np.float16), 'user object')
24  output = graph.GetResult()
```

Figura 5.39: Exemplo de código criado para executar um ficheiro Graph em VPU.

No código da imagem 5.39, da linha 11 à linha 15 é verificado o dispositivo NCS ligado ao computado e é realizado uma conexão a este. Da linha 18 á linha 21 é lido o ficheiro Graph e este é enviado para o NCS. Por fim é enviado uma imagem para o NCS e este envia o *Output* determinado pela rede.

No exemplo seguinte foi enviado para o NCS o gráfico responsável pela deteção dos rostos nas imagens. A figura 5.40 mostra um gráfico tempo necessário variando o número de caras que estavam na imagem.

No gráfico da figura 5.40 é possível verificar que o tempo aumenta ligeiramente à medida que o número caras aumenta. É também possível verificar que de quando se aumenta o número de caras de 1 para 2 o tempo diminui ligeiramente, este acontecimento repete-se algumas vezes no gráfico e não foi verificado nos testes realizados no CPU.

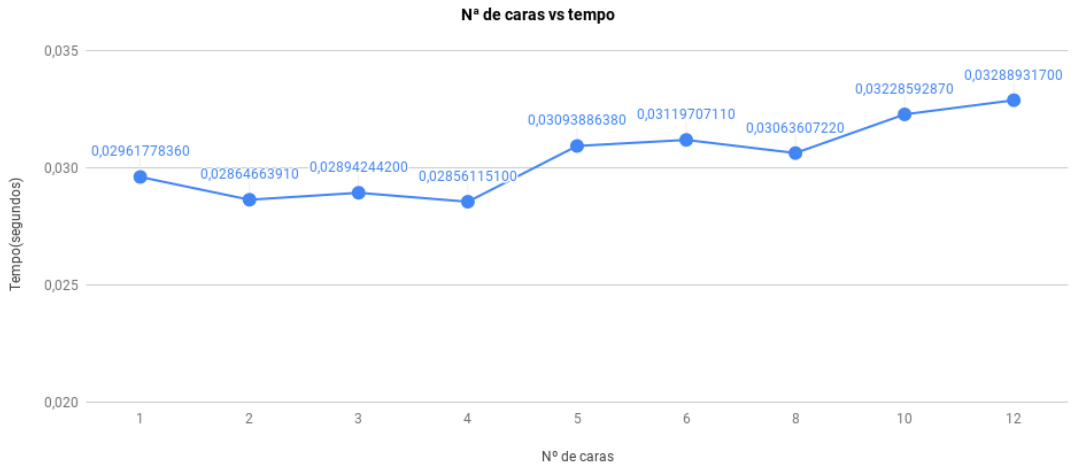


Figura 5.40: Tempo necessário para executar o bloco de deteção de rostos variando o nº de rostos na imagem.

5.9 Comparação de resultados de VPU e CPU

A realização dos vários testes permitiu obter dados nas duas plataformas testadas(VPU e CPU) dos vários blocos e das várias redes. Nos subcapítulos anteriores observaram-se os dados obtidos nos vários testes e verificou-se alguns aspetos importantes. Neste capitulo irão ser comparados os tempos obtidos na execução das redes em VPU e CPU.

O primeiro bloco a ser comparado foi o bloco de deteção de rostos. Os tempos obtidos na execução deste bloco foram colocados no mesmo gráfico de modo a facilitar a comparação. A imagem 5.41 mostra um gráfico com os resultados obtidos em VPU(vermelho) e os resultados obtidos em CPU (azul).

Neste gráfico é possível observar que os resultados obtidos foram bastante melhores em VPU do que os resultados obtidos em CPU. Os tempos de execução para VPU foram sempre menores em VPU, mesmo quando em VPU existiam 12 rostos numa imagem e em CPU existiam apenas 1 rosto na imagem o tempo de execução em VPU era menor que o tempo de CPU. É também possível verificar que o impacto que o aumento do número de caras tinha em VPU era era muito menor que em CPU. Em CPU a linha sobe muito mais acentuadamente do que em VPU.

Figura 5.41: Comparação dos dados obtidos nos testes realizados ao VPU(vermelho) e nos testes realizados ao CPU(azul).

Capítulo 6

Conclusão e Trabalho Futuro

Esta dissertação abordou o desenvolvimento de um sistema capaz de identificar e seguir uma cara numa imagem e determinar características da uma cara tais como a idade, género e estado emocional apenas com o uso de uma câmera visível.

Numa fase inicial, realizou-se o estudo para se identificar os melhores trabalhos realizados para cada uma das características a determinar e qual o tipo de rede neural mais indicada. Nesta fase analisaram-se também alguns trabalhos idênticos e que extraiam um conjunto de características da cara. Deste estudo consegui-se verificar que o tipo de rede utilizado pela maioria das redes eram redes VGG ou divergentes desta. A rede VGG era transversal nos vários trabalhos e a versatilidade desta permitia com algumas alterações de blocos determinas diversas características. Foi também possível verificar as redes que tinham sido estudadas para a realização do trabalho desses artigos eram idênticas ou semelhantes as projetadas nos artigos, conclui-se então que a arquitetura a implementar deveria ser semelhantes às referenciadas. A boa performance da rede neural depende mais das base de dados e imagens utilizados no treino. Este estudo permitiu também identificar algumas técnicas simples na manipulação da imagem que poderiam melhorar a performance do sistema.

Em seguida projetou-se como se iria realizar a implementação, qual seria a arquitetura do *software* e qual seria o fluxo de informação entre os vários blocos. Nesta fase foi também necessário projetar como se iria realizar o *Facetraking* e como seria feita a associação de dados para que os dados fossem todos agrupados. Depois de projetado tudo, iniciou-se a construção das redes projetadas nos artigos referenciados. As redes foram testadas com os treinos obtidos nos artigos pois verificou-se que estes tinham realizados treinos com uma grande variedade de imagens e os resultados tinham sido bastante satisfatórios. Acrescentar mais imagens a estes treinos não iria ter grande impacto na rede. Durante

a implementação dos vários blocos foram se realizando vários teste para se analisar os dados obtidos e se seria necessário melhorar algum bloco anterior. Verificou-se ao longo da implementação um bom desempenho dos blocos na atribuição das características. Verificou-se também o impacto de alguns fatores tais como a luminosidade, e a variação da mesma na distância, que poderiam levar a rede a não atribuir o estado mais correto. Verificou-se também que o bloco *Facetraking* por vezes não tinha o comportamento mais adequado e será necessário melhora-lo.

Como trabalho futuro pretende-se a implementação dos restantes blocos que utilizam redes artificiais do sistema desenvolvido em VPU e analisar a performance nos dois sistemas(VPU e CPU) para cada bloco. No futuro também seria importante a implementação do sistema utilizando múltiplos NCS, em que cada uma deles executava cada uma das redes utilizadas. Deste modo será possível verificar qual a performance do sistema desenvolvido em VPU e assim comparar com os resultados obtidos em CPU.

A implementação deste trabalho em ROS será também uma tarefa a realizar. O ROS é uma plataforma bastante utilizada nos robôs das mais diversas áreas. A implementação deste trabalho em ROS irá facilitar a integração desde sistema em robôs que utilizem ROS.

Por fim a implementação deste sistema numa robô de serviço que tenha interação com humanos. Deste modo é validado o funcionamento do sistema e é cumprido o objetivo geral deste tipo de trabalho que é melhorar a interceção homem-máquina. Deste modo, proporcionar a expansão da área da robótica e uma melhor integração dos robôs na sociedade.

Bibliografia

[1] NASA. Five things about nasa's mars curiosity rover, 2018.

[2] Promobot. Promobot, 2018.

[3] Terabee. Terabee, 2018.

[4] Samantha McLaren. 9 ways ai will reshape recruiting (and how you can prepare), 2018.

[5] New Scientist. Humanoid robot gets job as receptionist, 2018.

[6] Sanbot Innovation Technology. Sanbot, 2018.

[7] Shijie Guo. Riba-ii, the next generation care-giving robot, 2011.

[8] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *CoRR*, abs/1604.02878, 2016.

[9] Enrique Correa, Arnoud Jonker, Michaël Ozo, and Rob Stolk. Emotion recognition using deep convolutional neural networks. Junho 2016.

[10] Amit Dhomne, Ranjit Kumar, and Vijay Bhan. Gender recognition through face using deep learning. *Procedia Computer Science*, 132:2 – 10, 2018. International Conference on Computational Intelligence and Data Science.

[11] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, Julho 2016.

[12] Afshin Dehghan, Enrique G. Ortiz, Guang Shu, and Syed Zain Masood. DAGER: deep age, gender and emotion recognition using convolutional neural network. *CoRR*, abs/1702.04280, 2017.

[13] MoodMe. Moodme, 2018.

[14] Face++. Face++, 2018.

[15] ISO. Iso 9241-110:2006, 2006.

[16] Eileen Brown. Asimo agile and responsive robot is poised to replace humans, 2014.

[17] Rajalingappaa Shanmugamani. *Deep Learning for Computer Vision*. Packt Publishing, Reading, Massachusetts, 2018.

[18] Anderson Barbosa; Marcílio Freitas; Francisco Neves. Confiabilidade estrutural utilizando o método de monte carlo e redes neurais, 2005.

[19] Prabhu. Understanding of convolutional neural network (cnn)—deep learning, 2018.

[20] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–, London, UK, UK, 1999. Springer-Verlag.

[21] An intuitive explanation of convolutional neural networks, 2016.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[23] Sadek. Convolutional neural network, 2017.

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.