



## Monitorização do estado de um operador

**PEDRO GOMES PINHEIRO RODRIGUES**

Outubro de 2021

# **Monitorização do estado de um operador**

**Pedro Gomes Pinheiro Rodrigues**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Sistemas de Informação e Conhecimento**

**Orientador: Isabel Praça**

**Co-orientador: Ana Raquel Faria**

Instituto Superior de Engenharia do Porto

Porto, Outubro 2021



# Resumo

A digitalização da indústria tem trazido um novo paradigma de interação entre o operador e a máquina. Com a introdução da robótica nas linhas de produção a interação entre o operador e os robots torna-se frequente levando a que em algumas situações seja muito próxima.

Esta evolução representa um enorme desafio não apenas do ponto de vista tecnológico, mas também do ponto de vista da segurança dos operadores. A monitorização do comportamento humano e a perceção do estado emocional do operador pode ser um fator decisivo na deteção precoce e prevenção de acidentes. Atualmente, existem diversos meios que podem ser utilizados nessa monitorização e deteção, sendo que os meios não invasivos trazem vantagens ao não implicar adaptação do operador a mais tecnologias.

A utilização de inteligência artificial para detetar as emoções dos operadores torna-se cada vez mais frequente. Esta monitorização permite analisar o comportamento dos mesmos e criar uma estimativa da probabilidade de ocorrer um acidente. Operadores que apresentem sinais de stress ou de cansaço tem maior risco de ter um acidente. Esta análise é obtida através da obtenção de imagens da face do operador e, através de algoritmos as imagens são classificadas em emoções. Para além da monitorização das emoções, também a monitorização dos movimentos dos corpos é utilizada para rastrear o operador e identificar potenciais acidentes.

Na presente dissertação pretende-se combinar estes diferentes tipos de monitorização de comportamentos de risco, tais como a deteção dos estados emocionais associados à fadiga e ao stress.

**Palavras-chave:** Inteligência artificial, robótica, aprendizagem máquina, deteção emoções, deteção movimentos, indústria



# Abstract

The digitalization of the industry has brought a new paradigm between the operator and the machine. With the introduction of the robotic in the production lines, the interaction of the operator with the machine becomes more frequent and in some situations this interaction may be very close.

This evolution represents a big challenge not only from the technology point of view, but also from the point of view of the operator safety. Monitoring the human behaviour and the perception of the operator emotional state could be a decisive factor in early detection and prevention of labour accidents. Currently, there are several means that can be used to monitoring the human behaviour, however the non-invasive means have more advantages by the fact they do not require the operator adaption to new technologies.

The artificial intelligence has become more used in monitoring the human emotions. Monitoring the human emotions turns possible to estimate a probability to occur an accident between the operator and the machine. The operators showing stress or tiredness have more probability to have an accident. This process is done capturing images of the operator and through algorithms the images are classified in emotions. Also, the operator's movements are captured to identify potential accidents.

The current work intends to combine different types of monitoring of risk behaviours of the operator, such as the operator emotional state associated with fatigue or stress.

**Keywords:** Artificial intelligence, robotic, machine learning, emotion detection, movements' detection, industry



# Índice

<b>1</b>	<b>Introdução</b> .....	<b>1</b>
1.1	Enquadramento.....	1
1.2	Problema .....	2
1.3	Objetivos .....	3
1.4	Constrangimentos e limitações.....	3
1.5	Estrutura do Relatório .....	4
<b>2</b>	<b>Estado de arte</b> .....	<b>5</b>
2.1	Contexto atual.....	6
2.2	Ferramentas atuais .....	7
2.2.1	Real-Time-Action-Recognition.....	8
2.2.2	Behavior detection and control via Computer Vision .....	9
2.2.3	Emotion-Recognition.....	10
2.2.4	Open Face .....	10
2.2.5	EmoPy.....	11
2.3	Comparação entre ferramentas .....	12
<b>3</b>	<b>Análise de valor</b> .....	<b>15</b>
3.1	Identificação do problema .....	15
3.2	Proposta de valor .....	18
3.3	Quality Function Deployment (QFD).....	19
3.4	TOPSIS .....	21
3.5	Experimentação e avaliação .....	24
3.5.1	Especificação da hipótese.....	25
3.5.2	Definição dos indicadores .....	25
3.5.3	Metodologia de avaliação .....	25
<b>4</b>	<b>Deteção de emoções e comportamento de risco</b> .....	<b>27</b>
4.1	Solução .....	31
4.1.1	Caso de uso 1 - Identificação das emoções do operador .....	32
4.1.2	Caso de uso 2 - Identificar risco de colisão do operador com robot .....	33
<b>5</b>	<b>Implementação da solução</b> .....	<b>35</b>
5.1	Solução implementada .....	35
5.2	Implementação .....	39
5.3	Validação .....	39
<b>6</b>	<b>Conclusão</b> .....	<b>43</b>

6.1	Resultados obtidos .....	43
6.2	Trabalho futuro.....	44
<b>7</b>	<b>Referências.....</b>	<b>45</b>

# Lista de Figuras

Figura 1 - Análise execução dos algoritmos (Hidalgo, et al., 2020) .....	8
Figura 2 - Performance do "SORT" (Bewley, et al., 2016) .....	9
Figura 3 - Nº de robots instalados anualmente (International federation of robots, 2020) .....	16
Figura 4 - Robots industriais por tipo de indústria (International federation of robots, 2020) .....	16
Figura 5 - Mercado mundial de IIoT (Statista, 2021).....	17
Figura 6 - Proposta de valor CANVAS.....	19
Figura 7 - Analise QFD .....	21
Figura 8 - Expressões faciais (Arriaga, et al., 2017).....	28
Figura 9 - "Fúria" (Ayman, et al., 2020).....	28
Figura 10 - "Repulsa" (Ayman, et al., 2020) .....	29
Figura 11 - "Tristeza" (Ayman, et al., 2020) .....	29
Figura 12 - Diagrama componentes .....	32
Figura 13 – Diagrama de sequência do UC1 .....	33
Figura 14 - Diagrama de sequência do UC2 .....	34
Figura 15 - Treino do sistema.....	35
Figura 16 - modelConfig.txt.....	36
Figura 17 - Resultado da aplicação .....	36
Figura 18 - Menu da aplicação .....	37
Figura 19 – Conta cada uma das emoções.....	37
Figura 20 - Calcula o comportamento de risco .....	38
Figura 21 - Resultado do treino.....	38
Figura 22 – Fúria.....	40
Figura 23 – Tristeza .....	40
Figura 24 – Repulsa .....	40
Figura 25 - Aplicação em execução no modo treino.....	41
Figura 26 - Resultado da aplicação com o <i>dataset</i> de comportamento de risco.....	41
Figura 27 - Resultado obtido com o <i>dataset</i> sem comportamento de risco .....	42
Figura 28 – Fúria.....	49
Figura 29 – Repulsa .....	49
Figura 30 – Felicidade .....	50
Figura 31 – Tristeza .....	50



# Lista de Tabelas

Tabela 1 - Comparação de modelos neuronais (Perez, 2018) .....	11
Tabela 2 - Comparação de diferentes tecnologias de detecção de movimentos.....	13
Tabela 3 - Escala de avaliação .....	20
Tabela 4 - Matriz de decisão .....	21
Tabela 5 - Matriz normalizada .....	22
Tabela 6 - Matriz normalizada pesada .....	22
Tabela 7 - Multiplicação de cada elemento pelo peso da respectiva coluna .....	22
Tabela 8 - Solução ideal e solução ideal negativa .....	23
Tabela 9 - Separação da solução ideal positiva.....	23
Tabela 10 - Separação da solução ideal negativa .....	24
Tabela 11 - Cálculo da proximidade relativa à solução ideal .....	24
Tabela 12 - Resultados das experiências preliminares .....	30



# Acrónimos e Símbolos

## Lista de Acrónimos

---

CNN	Convolutional Neural Network
CPU	Central Processing Unit (Unidade de processamento central)
CUDA	Compute Unified Device Architecture
EUA	Estados Unidos da América
FACS	Facial Action Coding System
GPU	Graphic Processing Unit
HOG	Histograms of Oriented Gradients
IA	Inteligência artificial
IFR	International Federation of Robots)
IIoT	Industrial Internet of Things
ML	Machine learning (aprendizagem máquina)
QFD	Quality Function Deployment
SMS	Short Message Service
SORT	Simple Online and Realtime Tracking
UC	Use Case (Caso de uso)



# 1 Introdução

A tecnologia de informação e comunicação está em constante evolução na indústria e novas tecnologias surgem, tais como a internet das coisas, inteligência artificial e computação na nuvem (ZHENG, et al., 2018). Com esta evolução, a interação entre os operadores e os robots torna-se frequente e o risco de ocorrerem acidentes aumenta (Cohen, et al., 2018).

Com esta dissertação, pretende-se testar tecnologias de detecção de emoções para identificar comportamentos de risco.

## 1.1 Enquadramento

Com a atual revolução industrial, os robots estão cada vez mais presentes na indústria. Espera-se que esta revolução traga alterações no paradigma de produção nos próximos anos (Indri, et al., 2018).

Do ponto de vista da segurança, esta transformação traz novos problemas e novos desafios tanto a nível da cibersegurança como ao nível da segurança física do operador (Robotic Industries Association, 2018). Os operadores interagem diretamente com os robots o que aumenta o risco de ocorrer um acidente entre o operador e o robot. A constante monitorização do operador e do ambiente que o rodeia permite recolher informação suficiente para alimentar algoritmos de machine learning (ML) capazes de classificar os comportamentos do operador através das expressões faciais (Cohen, et al., 2018). Para a detecção das expressões faciais, (Ekman & Friesen, 1976) propôs um sistema capaz de detetar sete expressões faciais: raiva, repulsa, assustado, triste, feliz, surpreendido e neutro.

As expressões faciais são uma ferramenta importante na sociedade e com a crescente interação entre humanos e robots, torna-se fundamental que os humanos consigam exprimir as suas emoções através das máquinas. Com o crescente avanço no campo da inteligência artificial, é possível desenvolver sistemas capazes de detetar e classificar em emoções as expressões humanas. Também, os avanços na área das redes neuronais e da capacidade de computação permitiram melhorar a precisão dos algoritmos de classificação (Rocha & Praça,

2020). Estas tecnologias aplicadas na indústria atual permitem monitorizar o estado de um operador constantemente de forma não intrusiva. Os dados podem ser obtidos de diversas fontes, desde câmaras a biossensores, e depois serem integrados em sistemas de inteligência artificial (IA). Com base nas emoções detetadas, é analisada a possibilidade de ocorrer um acidente sendo que as emoções negativas normalmente estão associadas a um maior risco de acidente e decréscimo de produtividade enquanto as emoções positivas normalmente estão associadas a um melhor desempenho e produtividade do operador (Golan, et al., 2019).

## **1.2 Problema**

Com a quarta revolução industrial espera-se grandes mudanças no paradigma da produção durante os próximos anos. A robótica tem um papel importante nesta revolução ao estar cada vez mais presente, com robots cada vez mais inteligentes e com capacidade de aprender tarefas e interagir com outras máquinas e com humanos (Indri, et al., 2018).

O aumento do uso da robótica na indústria permite reduzir o número de tarefas repetitivas ao automatizá-las tornando o processo de produção mais eficiente (Williams & Allen, 2017). Atualmente, a introdução da inteligência artificial (IA) na robótica permite que este processo seja mais eficiente na execução de processos automatizados (Ribeiro, et al., 2021).

Contudo, a utilização de robots na indústria não elimina a necessidade de mão de obra humana (Ustundag & Cevikcan, 2018), tornando-se necessário a colaboração entre humanos e robots no mesmo espaço de trabalho (Ustundag & Cevikcan, 2018). Este novo paradigma requer novas abordagens na interação entre o operador e as máquinas (Cohen, et al., 2018). (Cohen, et al., 2018) propõe um sistema capaz de analisar o operador e o ambiente que o rodeia e, com recurso a IA melhorar o desempenho do operador e a segurança do mesmo.

Com esta nova interação entre humanos e robots, levantam-se novos problemas especialmente na segurança dos operadores. A necessidade de colaboração de operadores e robots no mesmo espaço físico representa um risco de colisão entre o operador e o robot. Torna-se necessário criar um sistema capaz de identificar possíveis colisões (S. ROBLA-GÓMEZ, 2017).

Este é um problema cada vez mais frequente e a deteção das expressões faciais permite criar sistemas para detetar comportamentos de risco que possam resultar em acidentes entre um operador e um robot (Cohen, et al., 2018). Existem várias ferramentas que permitem detetar as emoções, mas é necessário relacioná-las com o estado emocional do operador. Geralmente as emoções negativas como a tristeza ou a ansiedade estão associadas a comportamentos de risco que podem resultar em acidentes (Golan, et al., 2019). A fadiga é outro estado emocional que representa o perigo de ocorrência de acidente (Golan, et al., 2019). (Clarke & Clarke, 1997) propôs uma solução para detetar a fadiga no ramo automóvel baseada na análise dos movimentos dos olhos para identificar os sinais de cansaço (Clarke & Clarke, 1997).

## 1.3 Objetivos

Pretende-se com a presente dissertação, explorar ferramentas de deteção de emoções para obter um protótipo de uma solução informática capaz de identificar possíveis acidentes. A solução deve funcionar de forma não intrusiva para que não seja necessária a adaptação do operador a novas ferramentas. É necessário relacionar as emoções obtidas com comportamentos de risco.

Atualmente existem ferramentas que permitem identificar as emoções do operador. A ferramenta “Emotion-Recognition” é uma ferramenta desenvolvida em Python que permite obter as emoções da face do operador (Ayman, et al., 2020). Esta ferramenta foi disponibilizada por (Ayman, et al., 2020) e é um trabalho inspirado no trabalho de (Arriaga, et al., 2017). O “Emotion-Recognition” é uma ferramenta grátis para uso comercial e privado, a licença permite a modificação do código e a sua distribuição (Ayman, et al., 2020).

A ferramenta “Emotion-Recognition” permite identificar várias emoções como a fúria, repulsa, medo, tristeza, felicidade, surpresa e neutro. As emoções são identificadas através de imagens capturadas por uma camara que captura as faces. Com recurso a IA, as imagens são classificadas em emoções. Através das emoções obtidas, pretende-se relacionar as emoções com comportamentos de risco. Se o operador apresentar um comportamento de risco então o sistema deve lançar um alerta.

O objetivo desta dissertação é testar a integração das ferramentas de deteção de emoções para identificar comportamentos de risco dos operadores.

É necessário capturar as imagens através de uma camara para identificar as emoções. Com as emoções identificadas é necessário relacionar as mesmas com comportamentos de risco.

Para além da identificação das emoções é necessário rastrear o movimento do operador para identificar possíveis colisões entre o operador e um robot. O “Real-Time-Action-Recognition” é uma ferramenta que permite identificar em tempo real o movimento do operador. Esta ferramenta disponibilizada por (Ildoonet, 2018), utiliza OpenCV, Tensorflow e Filterpy para rastrear os movimentos dos corpos.

## 1.4 Constrangimentos e limitações

A solução deve ser capaz de monitorizar o operador e o meio que o rodeia de forma não intrusiva. É necessário que a solução seja capaz de rastrear o movimento do operador e detetar as emoções através da expressão facial do operador. Esta deteção recorre ao ML para categorizar as imagens em emoções.

O algoritmo de ML necessita de um conjunto de dados de treino. Estes dados são imagens para treinar o algoritmo. Através destes dados de treino, o algoritmo é capaz de categorizar a imagem da face do operador obtida pela camara numa das sete emoções: fúria, repulsa, medo, tristeza, felicidade, surpresa ou neutro.

Contudo, a quantidade e a qualidade dos dados são um constrangimento pois têm um grande impacto na fidelidade dos resultados a serem obtidos.

Neste projeto, é usado um conjunto de dados de treino disponibilizado no repositório da ferramenta “Emotion-Recognition” para identificar as emoções dos operadores.

Para o rastreamento da posição do operador é utilizada a ferramenta “Real-Time-Action-Recognition” que contém um conjunto de dados de treino.

## 1.5 Estrutura do Relatório

O presente relatório descreve o trabalho realizado no âmbito do TMDEI para conclusão do curso Mestrado em Engenharia Informática – Sistemas de Informação e Conhecimento.

O capítulo 2 descreve o estado de arte. Neste capítulo contextualiza-se o leitor com a evolução da robótica e da IA ao longo dos anos. Apresenta-se o que é utilizado atualmente na indústria para monitorizar o operador e identificar as emoções. É descrito um caso prático de uma solução implementada na indústria e são apresentadas algumas ferramentas atuais para deteção de expressões faciais e rastreamento dos operadores. Estas ferramentas são descritas e analisadas em detalhe. Ainda neste capítulo, é feita uma análise e comparação entre as ferramentas descritas, tendo em consideração as características mais relevantes para a resolução do problema em questão.

No capítulo 3, descreve-se a análise de valor. Nesta seção descreve-se como é que a solução consegue trazer valor ao negócio e resolver o problema. Começa por identificar o problema em questão e de seguida, contextualiza-se o leitor com o problema da interação dos operadores com os robots, que esta solução se propõe a resolver. É demonstrado o atual crescimento da robótica na indústria para fundamentar a necessidade de obter uma solução para resolver este problema. É apresentada a proposta de valor. A proposta de valor responde às perguntas:

- Que valor entregamos ao cliente?
- Qual o problema do cliente que pretendemos resolver?
- Quais as necessidades do cliente que satisfazemos?
- Que produtos ou serviços oferecemos para cada segmento de clientes?

É também neste capítulo que é apresentado o CANVAS e descreve-se a análise Quality Function Deployment. Nesta análise pretende-se analisar os requisitos, identificar as funcionalidades que o produto necessita e como e quem as vai implementar.

Este capítulo descreve os cálculos feitos para o método TOPSIS. Este método põe em hipótese duas soluções para chegar à solução ideal. De seguida descreve a experiência e quais os indicadores para avaliar a implementação e o cumprimento dos requisitos.

## 2 Estado de arte

Com a evolução da robótica na indústria, a interação entre o humano e máquina acontece mais frequentemente e com isso surge a necessidade de criar sistemas capazes de detectar possíveis acidentes entre máquinas e humanos através da análise de expressões faciais e detecção de movimentos dos corpos. A detecção de expressões faciais e movimentos dos seres humanos é um tema cada vez mais presente indústria (Rocha & Praça, 2020). Começam a ser desenvolvidos sistemas de inteligência artificial capazes deste tipo de interação humano máquina tornando possível prever possíveis situações de perigo, com recurso a modelos de classificação, como o *Facial Action Coding System* (FACS) que permite classificar as expressões faciais em sete tipos: felicidade, tristeza, surpresa, medo, fúria, repugno e desprezo (Ekman & Friesen, 1976).

Detetar emoções através de imagens é uma tarefa trivial para os olhos humanos, no entanto é bastante desafiante para as máquinas. Atualmente, existem várias técnicas de reconhecimento de emoções que recorrem a ML e capturas de imagens para obter os movimentos dos músculos da cara e movimento dos olhos. Qualquer sistema de ML requer um sistema de treino e um conjunto de dados de teste para treino. Contudo, a detecção destas emoções não é fácil pois, existem pequenos movimentos de músculos da face que diferem entre cada uma das expressões. Esta diferença pode ser mais ou menos acentuada de pessoa para pessoa. Pode também obter-se os movimentos das mãos e braços e categorizá-los. Esta tarefa é complexa, no entanto o ML e as Redes Neurais têm obtido bons resultados nesta classificação (Raut, 2018).

A definição de IA remonta aos anos 1940s, a Alan Turing, que trabalhou ao lado dos britânicos na descodificação da máquina Enigma durante a 2ª Guerra Mundial. Nos anos 1950s, Alan Turing descreveu como criar máquinas inteligentes e como testar a inteligência das máquinas, com o teste de Turing. O teste de Turing consiste num teste realizado com um humano a interagir com outro humano e uma máquina. Se o humano não conseguir distinguir a máquina do humano então a máquina é considerada inteligente. Este teste continua a ser utilizado para avaliar a inteligência de um sistema de IA (Haenlein & Kaplan, 2019).

Atualmente a IA é uma ciência cognitiva que contém diversas áreas como processamento de imagem, processamento de linguagem natural, robótica e aprendizagem máquina. IA torna-se importante no meio industrial e foca-se principalmente em desenvolver, validar e implantar algoritmos de ML que permitam resolver os problemas atuais da indústria (Jay Lee, 2018).

Com o FACS, é possível transformar uma imagem num número, ou unidade de ação, que representa a posição dos pontos da face naquele momento. O resultado da combinação de todos os números é a expressão facial. Cada unidade de ação representa as micro alterações na face, e a combinação destas unidades representa uma expressão. A definição de pontos na face é importante para o reconhecimento facial. Existem bibliotecas que permitem fazer esta identificação, como por exemplo a *Dlib* que utiliza 68 pontos faciais para a identificação da

expressão. Esta biblioteca permite obter as coordenadas (x, y) dos pontos detetados, divididos em áreas como os olhos (esquerdo e direito), sobrancelhas, boca, nariz e maxilar (Raut, 2018).

Viola-Jones é o algoritmo mais conhecido para deteção de objetos criada por Paul Viola e Michael Jones. Inicialmente este método foi desenvolvido para detetar objetos e mais tarde permitiu detetar trânsito pedonal. Atualmente, pode ser utilizado para deteção das faces. Esta técnica é conhecida como *Haar-cascade classifier*. O funcionamento da Viola-Jones baseia-se no *Haar feature Selection*, para detetar os objetos. O retângulo vai aumentando o tamanho em 1.2 vezes até encontrar um padrão que corresponda aos olhos e ao nariz. É criado um vetor que caracteriza a imagem e processa com o classificador *AdaBoost*. Este método permite processar imagens em tempo real com grande precisão (Surinta & Khruahong, 2019). Este método recorre ao *Haar-cascade* para detetar os objetos por sobreposição e treinar o algoritmo para treinar os positivos verdadeiros com as variações crescentes dos falsos negativos (Lee-Morrison, 2019).

Dalal e Triggs propuseram uma nova abordagem, *histograms of oriented gradients* (HOG), que utiliza a forma do objeto para calcular a taxa de variação do objeto. O método cria subamostras da imagem com variações das bordas do objeto e cria um histograma com a orientação do objeto, que alimenta o algoritmo de aprendizagem (Surinta & Khruahong, 2019).

Mais tarde, apareceu um novo método, o *max-margin object detection*, que não é baseado em subamostras de imagem. Este método pode ser utilizado para otimizar outros métodos de deteção de objetos como o HOG (King, 2015).

Para além da deteção das faces, é crucial detetar os movimentos dos objetos e corpos e os seus contornos. Os *feature descriptors* são algoritmos capazes de identificar os contornos e um objeto e fornecer uma descrição visual da imagem. Nas últimas duas décadas, houve uma evolução substancial destes métodos de descrição visual de imagens. (Yang, et al., 2020).

O algoritmo de Harris é uma técnica popular devido à suas características que o tornam eficiente mesmo quando a imagem sofre uma rotação, uma escala ou mesmo havendo ruído na imagem (Shah & Shah, 2012). O “OpenCV” é uma ferramenta que disponibiliza este algoritmo (Raut, 2018).

A desafiante tarefa de deteção e classificação de movimentos humanos através de vídeo tem uma vasta aplicabilidade em diversas áreas como sistemas de videovigilância, sistemas de assistência à condução para deteção de peões e deteção de comportamentos humanos (Jin, et al., 2019).

## 2.1 Contexto atual

Na indústria atual, um operador consegue ser constantemente monitorizado por um sistema computadorizado capaz de receber dados de várias fontes e integrá-los com sistemas de IA. Com este tipo de monitorização é possível obter o estado atual do operador (o seu estado psicológico e fisiológico) (ZHENG, et al., 2018). É possível identificar a fadiga de um operador pela observação da falta de concentração e ao identificar o desejo de dormir. A fadiga reduz o

nível de segurança e por isso é um comportamento normalmente observado utilizando tecnologia de biossensores. As emoções são outra característica observada através da monitorização da interação entre o operador e máquina. Emoções positivas estão normalmente relacionadas com um melhor desempenho e produtividade do operador. Já as emoções negativas podem significar um decréscimo na produtividade e também um aumento de comportamentos de risco por parte do operador (Golan, et al., 2019).

A monitorização da interação de um operador com os robots pode ser feita com recurso a três elementos: observação, análise e reação. O processo de observação faz o rastreio do movimento do operador e de todo o ambiente que o rodeia. O elemento de observação do operador deve ser composto por três agentes (*input agent*, *momentary profile agente* e *profile trajectory agent*) (Cohen, et al., 2018).

O *input agent* é responsável por recolher os dados de linguagem corporal do operador, gestos, movimentos do corpo, movimentos dos olhos, expressões faciais, através do áudio ou até a pulsação. Estes dados são obtidos através de camaras e sensores e são transformados em vetores de informação multidimensionais. O *momentary profile agent* recebe os vetores de dados criados no agente anterior e cria um vetor com a representação do estado do operador. O *momentary profile* contém os dados de indicadores fisiológicos (batimentos cardíacos, nível de transpiração, temperatura e nível do stress), das posições do corpo e movimentos, movimento dos olhos e músculos faciais. O *profile trajectory agent* é responsável por analisar os dados obtidos no *momentary profile* e identificar os comportamentos de risco. Contém algoritmos capazes de analisar e encontrar padrões nos dados recolhidos. Se existir uma sequência de momentos de fadiga no *momentary profile* então o *profiler* poderá identificar que o estado de cansaço no operador que poderá resultar num acidente (Cohen, et al., 2018).

## 2.2 Ferramentas atuais

No contexto da indústria atual, existem diversas ferramentas que permitem monitorizar os operadores através da deteção das emoções e dos movimentos.

Nesta seção são analisadas algumas das ferramentas que permitem detetar a posição e as expressões faciais dos operadores, nomeadamente o “*Real-Time-Action-Recognition*”, “*Behavior detection and control via Computer Vision*”, “*Emotion-recognition*”, “*Open face*” e “*EmoPy*”.

A escolha destas ferramentas teve em conta alguns critérios. Um dos critérios é ter ferramentas que detetem o corpo do operador e outras que detetem a face e consigam obter o estado emocional. Outro critério necessário é ter ferramentas que permitam a combinação entre elas para obter uma solução capaz de monitorizar a posição e o movimento do operador e também o seu estado emocional.

Para esta seleção foram analisados alguns aspetos das ferramentas tais como, a quantidade de emoções que esta consegue identificar e a quantidade de pontos que consegue detetar.

### 2.2.1 Real-Time-Action-Recognition

O *Real-Time-Action-Recognition* é uma ferramenta capaz de detetar os movimentos de uma pessoa. Esta ferramenta recorre ao *tf-pose-estimation* para estimar a posição do corpo da pessoa. O *tf-pose-estimation* utiliza o algoritmo *OpenPose* implementado em *TensorFlow* e otimizado para obter uma melhor performance na utilização do CPU em processamento em tempo-real (Ildoonet, 2018).

O *OpenPose* é o primeiro algoritmo que permite a deteção em tempo real de várias pessoas em simultâneo e capaz de detetar o corpo humano, a face e as mãos. Na Figura 1 são representados os gráficos comparativos entre diferentes algoritmos: *OpenPose*, *Alpha-Pose* e *Mask R-CNN*. Pode-se constatar que o algoritmo *OpenPose* mantém o tempo de execução constante à medida que o número de pessoas por imagem aumenta enquanto que, os algoritmos *Alpha-Pose* e *Mask R-CNN* tem um aumento do tempo de execução sendo que este último é o que apresenta a subida mais acentuada (Hidalgo, et al., 2020).

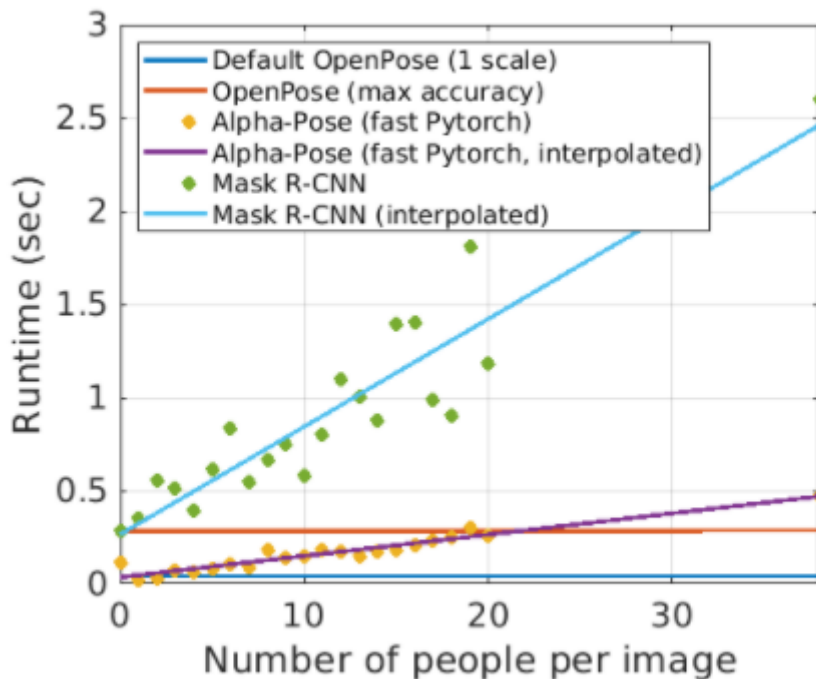


Figura 1 - Análise execução dos algoritmos (Hidalgo, et al., 2020)

O *OpenPose* permite correr nos sistemas operativos Ubuntu, Windows e Mac OSX, e é compatível com vários tipos de hardware como *CUDA GPUs*, *OpenCL GPUs* e computadores com apenas *CPU*. Este sistema caracteriza-se por ter três blocos: deteção de corpo e pés, deteção das mãos e deteção da face (Cao, et al., 2019).

Quanto ao rastreamento feito pelo *Real-Time-Action-Recognition*, esta tecnologia recorre ao algoritmo *Simple Online and Realtime Tracking* (SORT) desenhado para aplicações de rastreamento online. Este algoritmo utiliza imagens passadas e imagens atuais para criar uma identidade do objeto em tempo real. De acordo com o *Benchmark* publicado na *MOTChallenge*

em 2017, o algoritmo "SORT" apresentou a melhor performance em relação aos outros algoritmos analisados (Figura 2) (Bewley, et al., 2016).

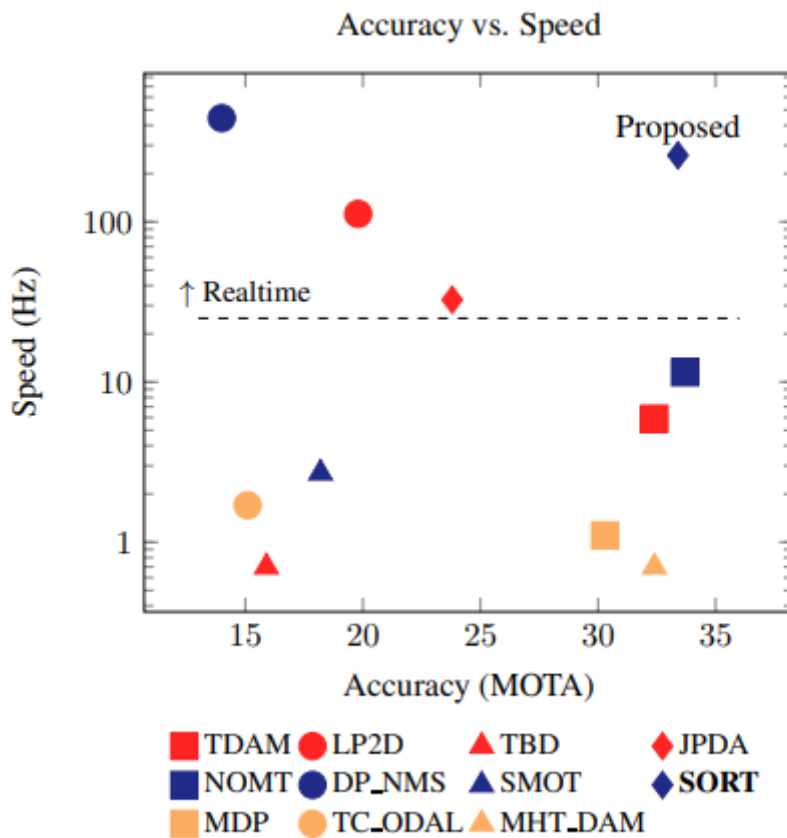


Figura 2 - Performance do "SORT" (Bewley, et al., 2016)

Na Figura 2 pode-se analisar o comparativo entre diferentes ferramentas *open source* de detecção de múltiplos objetos. Cada marcador na figura representa a velocidade e precisão de um marcador (Bewley, et al., 2016).

## 2.2.2 Behavior detection and control via Computer Vision

O Behavior detection and control via computer vision é uma ferramenta que permite detetar movimentos das mãos e as expressões faciais através da imagem obtida de uma camara. Esta ferramenta permite detetar os movimentos das mãos, a posição dos dedos, o número de dedos, número de mãos e a área das mãos (Andresen, 2013).

A deteção das mãos é feita com recurso a *Deep Learning*. A utilização de *Deep Learning* para deteção de movimentos tem vindo a ser cada vez mais frequente devido ao aumento da capacidade de processamento dos computadores. A *Convolutional Neural Network* (CNN) é bastante robusta e por isso tornou-se popular pois, consegue ter uma boa performance. Mais tarde surgiu a R-CNN com um método de treino mais rápido do que o algoritmo original e permite treinar redes neuronais mais complexas (Chen, et al., 2016).

A detecção das expressões faciais utiliza uma biblioteca *face\_recognition*, implementada em Python. É possível com esta ferramenta detetar faces nas imagens e obter a posição dos olhos, da boca, do nariz e do queixo. Para detetar a face, esta ferramenta transforma a imagem em preto e branco, depois é analisado pixel a pixel para encontrar os pontos pretos. São analisados os pontos que rodeiam o ponto para encontrar a forma e a direção da imagem. É substituído o pixel por uma seta (*gradient*). No final a imagem é transformada numa representação que contém a estrutura básica da face. De seguida, procura partes da imagem que correspondam a padrões conhecidos, obtidos pelos dados de treino (Geitgey, 2016).

### 2.2.3 Emotion-Recognition

O *Emotion-Recognition* é uma ferramenta que permite ler as emoções expressadas pela face humana através do processamento de imagens. Esta ferramenta consegue combinar diferentes algoritmos com técnicas de processamento de imagens. Consegue também identificar várias emoções, tais como, raiva, repulsa, assustado, triste, feliz, surpreendido e neutro (Ayman, et al., 2020).

São requisitos para a utilização desta ferramenta o *OpenCV*, o *Keras*, o *Numpy*, *Pandas*, *Imutils* e o *scikit\_learn* (Ayman, et al., 2020).

As probabilidades de cada emoção detetada são apresentadas numa janela com as capturas das imagens obtidas pelas camaras (Ayman, et al., 2020).

De acordo com os testes realizados por (Ayman, et al., 2020), esta ferramenta conseguiu ter uma precisão de 66% na detecção das emoções.

O *Emotion-Recognition* é um projeto baseado no projeto *Real-Time Face Detection* (Arriaga, et al., 2017) que é uma implementação de uma *Convolutional Neural Network*.

### 2.2.4 Open Face

O *Open Face* permite detetar a cara toda incluindo os olhos. Permite detetar os pontos da cara, posição da cabeça e detecção dos *Action Unit* da face. Esta ferramenta consegue ainda fazer o rastreamento do movimento dos olhos (Tadas Baltrušaitis, 2020).

Esta ferramenta surgiu da necessidade de analisar o comportamento da face humana de forma automática e não necessita de nenhum hardware específico para além de uma simples camara (Tadas Baltrušaitis, 2020).

O *Open Face* é uma ferramenta desenvolvida em C++ e foi originalmente criado por Tadas Baltrušaitis em colaboração com *CMU MultiComp Lab* (Tadas Baltrušaitis, 2020).

Relativamente à licença, esta ferramenta possui uma licença académica não comercial, no entanto para fins comerciais esta ferramenta não é grátis (Flintbox, 2019).

## 2.2.5 EmoPy

O *EmoPy* é uma ferramenta que utiliza redes neurais para prever as emoções humanas através de imagens das faces. Esta ferramenta explora o campo de reconhecimento de expressões faciais e utiliza conjuntos de dados públicos grátis para criar os modelos. Consegue detetar várias emoções através das expressões, tais como, a raiva, repulsa, medo, tristeza, felicidade, surpreendido e neutro (Perez, 2018).

É possível usar diferentes modelos de redes neurais no *EmoPy*. Um dos modelos disponíveis no *EmoPy* é o CNN. Este modelo divide a imagem em pequenas partes, como por exemplo uma sobrancelha, para detetar os padrões. O treino é feito utilizando imagens estáticas (Perez, 2018).

O *Time-Delayed 3D-Convolutional Neural Network* é outro modelo disponível. Este modelo não utiliza imagens estáticas para o treino. Utiliza uma série de imagens antigas e uma série com as imagens mais recentes e analisa a evolução para determinar o estado emocional (Perez, 2018).

Outro modelo disponível é o *Convolutional Long Short Term Memory*. Este modelo também utiliza séries de imagens antigas e séries de imagens mais recentes para o treino. No entanto, neste modelo as imagens não precisam de estar relacionadas como no *Time-Delayed* (Perez, 2018).

Também o modelo *Transfer Learning* está disponível no *EmoPy*. Este modelo não utiliza imagens temporais. Utiliza modelos de redes neurais pré treinados para deteção de objetos como ponto de partida. Aplica estes modelos e retreina-os para classificação de expressões faciais (Perez, 2018).

De acordo com (Perez, 2018), foram testados dois modelos, *ConvolutionalLstmNN* e *TransferLearningNN*, sendo que o *ConvolutionalLstmNN* teve melhor classificação. A Tabela 1 apresenta os resultados obtidos na experiência.

Tabela 1 - Comparação de modelos neuronais (Perez, 2018)

Modelo de Rede Neural	7 emoções		3 emoções	
	Precisão do treino	Precisão de validação	Precisão do treino	Precisão de validação
<b>ConvolutionalLstmNN</b>	0.6187	0.4751	0.9148	0.6267
<b>TransferLearningNN</b>	0.5358	0.2933	0.7393	0.4840

Estes testes foram realizados para medir a precisão de deteção das emoções em classificações de sete emoções e depois para classificar apenas três emoções. Os valores de precisão do treino são superiores aos de precisão de validação o que significa que estes modelos são precisos a classificar as expressões nas imagens de treino, mas menos precisos a classificar imagens novas (Perez, 2018).

## 2.3 Comparação entre ferramentas

A Tabela 2 apresenta uma comparação entre diferentes ferramentas existentes. Com esta análise pretende-se selecionar a que apresenta os melhores resultados para implementar na presente solução.

Foram selecionadas duas ferramentas que permitem rastrear o movimento do corpo dos operadores, o *Real-Time-Action-Recognition* e a *Behavior detection and control via Computer Vision*. Para a deteção de emoções, foram selecionadas as ferramentas *Emotion-Recognition*, *Open Face* e *EmoPy*, que permitem detetar as expressões faciais dos operadores.

Na Tabela 2, foram analisadas as ferramentas em alguns aspetos relevantes para o problema. Esta análise focou-se na avaliação das ferramentas pela sua capacidade de detetar emoções e os pontos que esta ferramenta consegue detetar. As ferramentas *Emotion-recognition* e *EmoPy*, são semelhantes nas emoções que cada uma consegue identificar. O *Open Face* consegue identificar as expressões faciais.

Na Tabela 2, pode-se verificar que o *Real-Time-Action-Recognition* consegue identificar vários pontos do corpo e várias pessoas em simultâneo. Esta ferramenta permite o rastreamento do movimento do operador. O *Behavior detection and control via Computer Vision* consegue identificar os movimentos das mãos, a posição dos dedos, o número de dedos, número de mãos e a área das mãos. O *Emotion-Recognition*, o *Open Face* e o *EmoPy*, permitem detetar a face. No entanto, o *Open Face* permite detetar vários pontos da cara e o *EmoPy* consegue detetar as partes da cara como os olhos, as sobrancelhas, a boca, etc.

Tabela 2 - Comparação de diferentes tecnologias de detecção de movimentos

<b>Programa</b>	<b>Tipo</b>	<b>Emoções</b>	<b>Pontos detetados</b>
<b>Real-Time-Action-Recognition (Ildoonet, 2018)</b>	Corpo	N/A	Vários pontos corpo; Multi-pessoas
<b>Behavior detection and control via Computer Vision (Andresen, 2013)</b>	Múltiplos elementos	N/A	Diversos pontos de vários elementos
<b>Emotion-recognition (Ayman, et al., 2020)</b>	Face	Raiva, repulsa, assustado, triste, feliz, surpreendido, neutro	Face em retângulo / sem pontos na cara
<b>Open face (Tadas Baltrušaitis, 2020)</b>	Face	Action units	Diversos pontos na cara
<b>EmoPy (Perez, 2018)</b>	Face	Raiva, repulsa, medo, tristeza, felicidade, surpreendido e neutro	Deteta várias partes da cara como olhos, boca, sobrancelhas, etc.



## 3 Análise de valor

A presente seção apresenta uma análise do produto sobre o ponto de vista da análise de valor, na qual são identificadas e analisadas as oportunidades de negócio. São apresentadas respostas para as perguntas “o que é o produto?”, “quem é o público alvo?”, “que valor traz ao negócio?” e “o que torna o produto único?”.

### 3.1 Identificação do problema

A indústria tem evoluído imenso nos últimos tempos, nomeadamente com a introdução de robótica na indústria. Um dos desafios que surgiram com esta evolução é garantir a segurança na interação dos humanos com as máquinas (Rocha & Praça, 2020).

O termo robot apareceu pela primeira vez em 1917, num romance do autor checo Karel Capek intitulado “Opilec”. Neste romance, os robots eram pessoas artificiais feitas com esqueletos metálicos. Atualmente, o robot já não é uma mera peça de ficção e, existem diversas áreas onde robots já são utilizados nomeadamente na área médica (DOBRESCU & DOBRESCU, 2017).

Atualmente, assistimos à quarta revolução industrial. É espectável que esta revolução traga grandes mudanças durante os próximos anos a nível do paradigma de produção. A robótica tem um papel fundamental nesta revolução. Estando cada vez mais presente, tecnologias inovadoras e robots cada vez mais inteligentes, com capacidade para aprender tarefas e para interagir com outras máquinas e com humanos, permitindo assim tornar a produção mais eficiente (Indri, et al., 2018).

No entanto, a nível social tem-se constatado que a robótica tem transformado o mercado de trabalho, penalizando mais as tarefas mais rotineiras (Vries, et al., 2020).

Na Figura 3, pode-se verificar o aumento do uso de robótica desde 2010. No entanto, foi atingido um pico no ano 2018 e em 2019 o número de instalações de novos robots diminuiu. De acordo com a fonte IFR (International federation of robots, 2020), esta diminuição está relacionada com a diminuição da procura de equipamentos eletrónicos derivado ao conflito económico entre os EUA e a China, sendo que a China é o país que investe mais em robótica aplicada à indústria da eletrónica de consumo (International federation of robots, 2020).

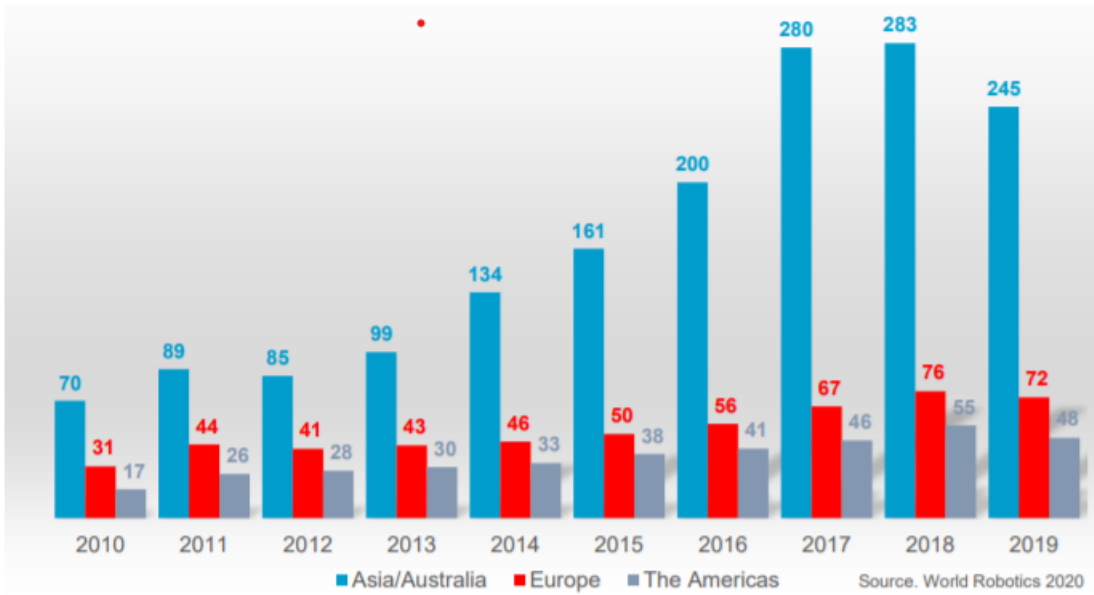


Figura 3 - Nº de robots instalados anualmente (International federation of robots, 2020)

A presença da robótica na indústria é maior na indústria automóvel e na elétrica/eletrónica como pode ser verificado na Figura 4. Segue-se a indústria metalúrgica, a indústria química e a indústria alimentar (International federation of robots, 2020).

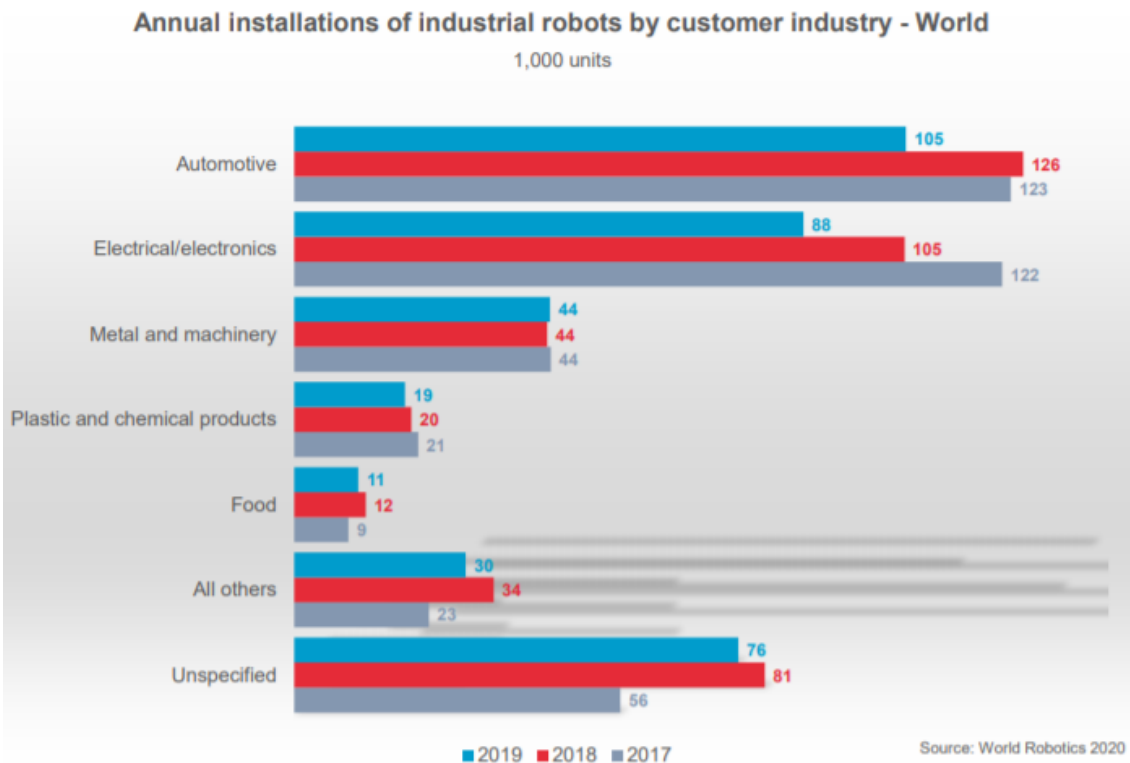


Figura 4 - Robots industriais por tipo de indústria (International federation of robots, 2020)

No contexto atual da pandemia do COVID-19, prevê-se uma diminuição da implementação da robótica em 2020. No entanto, a atual crise pode ser um impulsionador para a digitalização e o

aumento de oportunidades para o crescimento da robótica a médio e longo prazo (International federation of robots, 2020).

Associado à introdução da robótica na indústria, está o aumento da utilização de tecnologias associadas como a Internet das Coisas Industrial (IIoT do termo inglês Industrial Internet of Things) (Robotic Industries Association, 2018). Na Figura 5, está representado o aumento do mercado de IIoT mundial desde 2017. Segundo dados da Statista (Statista, 2021), espera-se que esta tendência continue pelo menos até 2025 (Statista, 2021). Esta tendência implica o aumento do número de sensores inteligentes instalados nas linhas de produção capazes de obter um maior volume de dados. No entanto, uma maior produção de dados implica um maior investimento em tecnologias de Big Data, para organizar e transformar todos os dados obtidos e criar informação útil e em tempo útil. Na sequência da interligação dos robots com os sistemas internos de dados, o risco de ataques informáticos aumenta e torna-se necessário investir em cibersegurança (Robotic Industries Association, 2018).

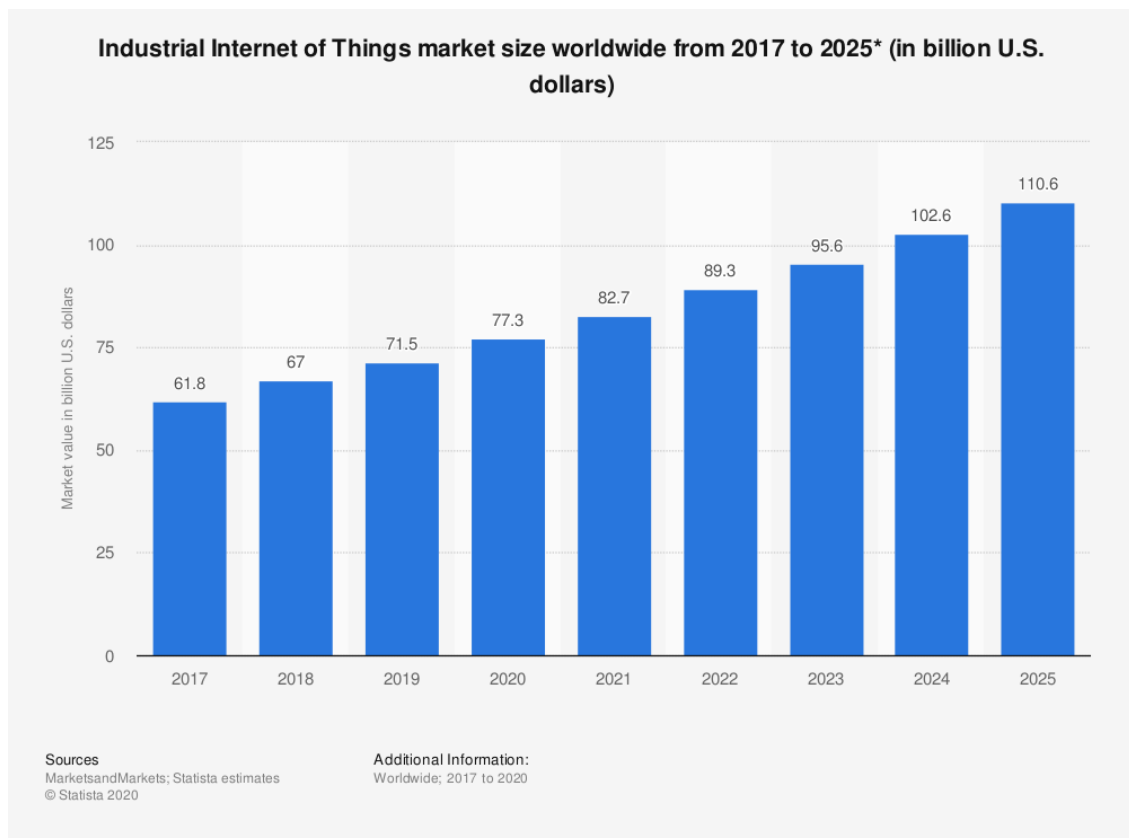


Figura 5 - Mercado mundial de IIoT (Statista, 2021)

Na presente dissertação pretende-se analisar diferentes ferramentas que permitem a monitorização de um operador de forma não intrusiva, e assim desenvolver um protótipo que permita identificar comportamentos de risco do operador que possam resultar em acidentes na linha de produção.

Esta solução deve ser capaz de identificar os estados emocionais do operador, tais como, a fadiga, o stress ou a distração e, também, os movimentos do operador que possam resultar em colisão com um braço robótico. Pretende-se com a solução combinar diferentes tipos de

deteção de comportamentos de risco, através da análise da posição do operador, gestos e estados emocionais em conjunto com tecnologias de ML para obter uma previsão de comportamentos de risco.

## 3.2 Proposta de valor

O valor de um produto varia de cliente para cliente. É comum analisar e avaliar as características de um produto a nível da sua performance, da capacidade ou o produto ser apelativo ou em relação ao seu preço. No entanto, aumentar o valor não é apenas otimizar o custo de produção. Por vezes o preço pode aumentar desde que a sua utilidade para o cliente aumente (DRM Associates, 2019).

A proposta de valor representa um conjunto de serviços ou produtos que trazem valor ao cliente. Os produtos ou serviços podem trazer valor pela novidade, pela performance, pela customização, funcionalidade, design, marca/status, preço, redução de custos e entre outras características. Existem 4 perguntas que a proposta de valor deve responder (Nicola, 2020):

- Que valor entregamos ao cliente?
- Qual o problema do cliente que pretendemos resolver?
- Quais as necessidades do cliente que satisfazemos?
- Que produtos ou serviços oferecemos para cada segmento de clientes? (Nicola, 2020)

O produto descrito na presente dissertação adequa-se essencialmente à indústria 4.0. Pretende-se oferecer valor à indústria que opere com recurso a robótica ao permitir uma maior segurança dos operadores quando trabalham em conjunto com as máquinas, ao combinar diferentes técnicas de monitorização de movimentos e expressões faciais dos operadores para detetar comportamentos de risco.

Com a evolução e digitalização da indústria a interação do homem e robot tem-se tornado num enorme desafio do ponto de vista da segurança do operador. É importante a constante monitorização do comportamento do operador para que sejam detetados comportamentos de risco, a posição do operador ou gestos que possam resultar em acidentes.

Na Figura 6 está representado o diagrama CANVAS de proposta de valor.

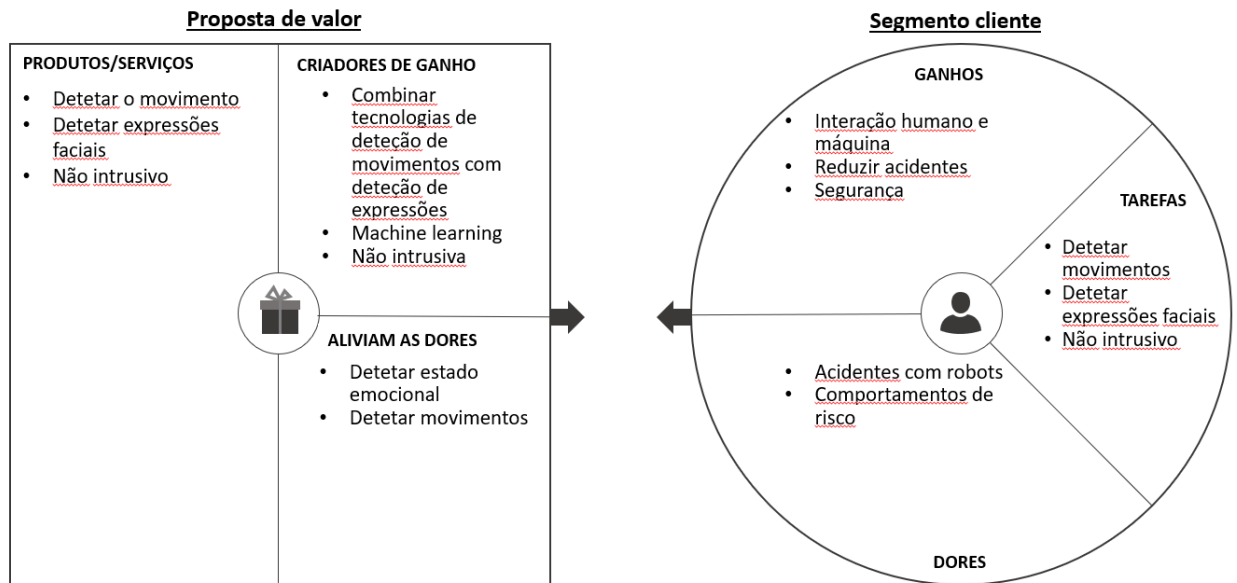


Figura 6 - Proposta de valor CANVAS

### 3.3 Quality Function Deployment (QFD)

O Quality Function Deployment (QFD) começou no Japão em 1972 na fábrica da Mitsubishi's Kobe Shipyard. Foi utilizada uma matriz que continha as necessidades do cliente numa coluna vertical e os métodos numa linha horizontal. Desde então que este método tem sido aperfeiçoado para considerar a funcionalidade do produto, potenciais pontos de falha, novas tecnologias e redução de custo (Warwick Manufacturing Group, 2007).

O QFD permite desenhar um produto tendo em conta as necessidades do cliente envolvendo todas as partes interessadas no produto desde o produtor até ao fornecedor. A sigla QFD significa "Quality Function Deployment" (em português "qualidade, funcionalidade e implantação") e em japonês "Implantação" refere-se a uma extensão de atividades, portanto, QFD significa que as responsabilidades da produção de um produto com qualidade é uma responsabilidade de todos os elementos da empresa. QFD define-se como um sistema de controlo de qualidade e cada um dos termos representa uma característica do sistema:

- Quality: Reunir os requisitos do cliente
- Function: Focar a atenção no que deve ser feito
- Deployment: Quem vai executar e quando

Este sistema permite obter produtos com mais qualidade e mais rapidamente a um baixo custo, e ao mesmo tempo permite desenhar um produto à medida das necessidades do cliente e também controlar o processo de desenvolvimento do produto para que seja possível melhorar o produto ou o processo de desenvolvimento (Warwick Manufacturing Group, 2007).

A implementação do sistema QFD é feita em 4 fases, sendo a primeira a identificação das necessidades do cliente, transformando em funcionalidades. Neste projeto, o cliente pretende

que a aplicação seja capaz de identificar as emoções de um operador para que seja possível detetar possíveis acidentes. Foram então identificadas as seguintes funcionalidades:

- Detetar a face
- Detetar o estado emocional raiva
- Detetar o estado emocional assutado
- Detetar o estado emocional triste
- Detetar o estado emocional feliz
- Detetar o estado emocional surpreendido
- Detetar movimento do corpo
- A deteção deve ser feita de forma pouco intrusiva

Após identificadas as funcionalidades deve ser definido como implementar estas funcionalidades. No contexto do presente trabalho, foram identificadas as tecnologias: ML, image capture, python e deteção de emoções (Quality One, 2020).

Na Figura 7 está representado o QFD para o trabalho em desenvolvimento. A coluna WHATs contém as funcionalidades e no meio é feita a relação com o “Como fazer” (tecnologias necessárias) numa escala de 0, 1, 3 e 9 em que 0 significa que não existe relação entre a funcionalidade e a tecnologia, 1 significa que existe uma relação fraca, 3 existe uma relação moderada e 9 é uma relação forte. No topo da “casa” é feita a relação entre as tecnologias conforme definido na Tabela 3 (Quality One, 2020).

Tabela 3 - Escala de avaliação

++	Strong positive
+	Positive
.	None
-	Negative
--	Strong negative

Conseguimos analisar que existe uma forte relação entre o “ML” e a “Image capture” e a “Deteção de emoções”. O ML é necessário para detetar padrões nas imagens captadas que permitam descobrir as emoções nas faces dos operários. A captura de imagens está também ligada pois as imagens são a base para a deteção das expressões faciais. Quanto a relação com o Python, grande parte das tecnologias de deteção funcionam em Python, no entanto existem algumas tecnologias com base noutras linguagens de programação pelo que não tem de ser necessariamente usado o Python. A coluna “Customer rating” contém definição da importância de cada uma das funcionalidades para o cliente. Estes valores são fornecidos pelo cliente e com base nestes valores é feito o cálculo da importância relativa. Este cálculo é feito pela soma de cada coluna multiplicada pelo “Customer rating” (Quality One, 2020).

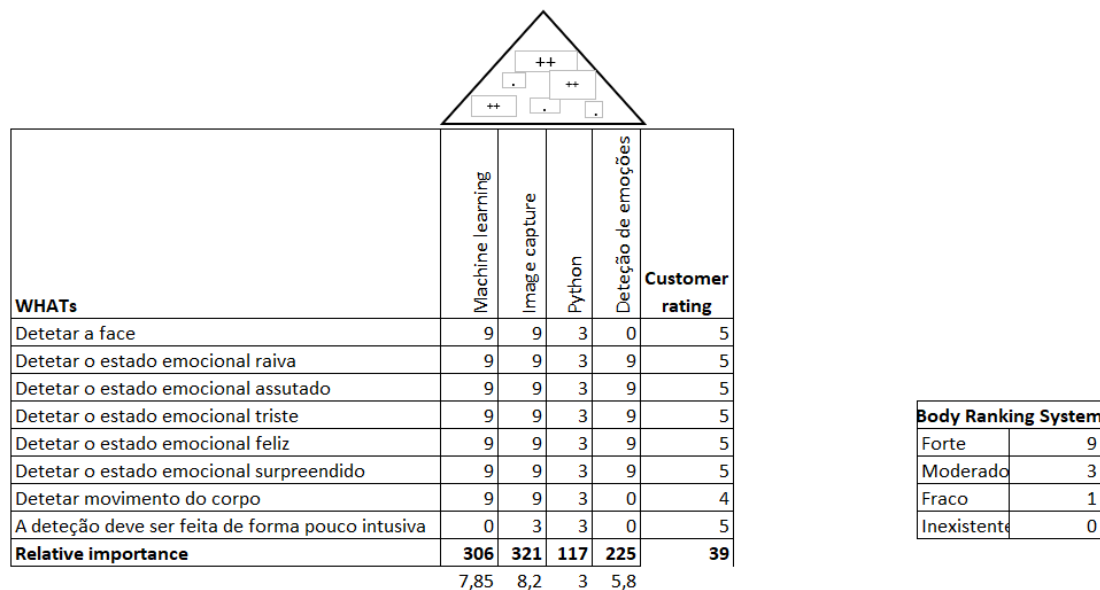


Figura 7 - Analise QFD

### 3.4 TOPSIS

O método TOPSIS põe em hipótese duas alternativas: alternativa ideal e a alternativa ideal negativa. A alternativa ideal é a que apresenta o melhor nível para os atributos considerados. Quanto à alternativa ideal negativa, esta apresenta os piores valores para os atributos considerados. O TOPSIS seleciona a alternativa que esteja mais próxima da alternativa ideal e mais afastada da alternativa ideal negativa.

Na presente dissertação foram consideradas três ferramentas alternativas para avaliar qual responde ao maior número de critérios. Foram considerados os critérios: “deteção da face”, “deteção da posição do operador” e “deteção do movimento do operador”. As ferramentas consideradas são: “Real-Time-Action-Recognition”, “Behavior detection and control via Computer Vision” e “Emotion-recognition”. Com base nas especificações destas ferramentas e os critérios que cada uma consegue responder, foi elaborada uma matriz de decisão com as respetivas classificações (Tabela 4).

Tabela 4 - Matriz de decisão

Pesos	0,2	0,4	0,4	1
	Face	Posição	Movimento	
<b>Real-Time-Action-Recognition</b>	0	3	3	
<b>Behavior detection and control via Computer Vision</b>	1	1	2	
<b>Emotion-recognition</b>	2	0	0	

O primeiro passo é normalizar a matriz de decisão ao transformar os valores de dimensão em atributos não dimensionais para permitir a comparação. A normalização é feita com a fórmula:

$$r_{ij} = \frac{x_{ij}}{(\sum_i x^2_{ij})} \text{ for } i = 1, \dots, m; j = 1, \dots, n$$

A Tabela 5 apresenta a matriz normalizada.

Tabela 5 - Matriz normalizada

	Face	Posição	Movimento
<b>Real-Time-Action-Recognition</b>	0	9	9
<b>Behavior detection and control via Computer Vision</b>	1	1	4
<b>Emotion-recognition</b>	4	0	0
<b>Soma</b>	5	10	13
<b>RaizQSoma</b>	2,23606798	3,16227766	3,60555128

A Tabela 6 representa os valores da matriz normalizada pesada.

Tabela 6 - Matriz normalizada pesada

	Face	Posição	Movimento
<b>Real-Time-Action-Recognition</b>	0	0,9486833	0,83205029
<b>Behavior detection and control via Computer Vision</b>	0,4472136	0,31622777	0,5547002
<b>Emotion-recognition</b>	0,89442719	0	0

O segundo passo deste método é multiplicar os valores da matriz pelo peso dado a cada atributo. O peso foi atribuído conforme o grau de importância dado a cada atributo. A detecção da posição e do movimento do operador tem um grau de importância maior do que detectar a face do operador. Assim, foi atribuído o valor de 0.4 à detecção da posição e do movimento e 0.2 à detecção da face.

Com base nestes valores foi criada a Tabela 7.

Tabela 7 - Multiplicação de cada elemento pelo peso da respectiva coluna

	Face	Posição	Movimento
<b>Real-Time-Action-Recognition</b>	0	0,37947332	0,33282012
<b>Behavior detection and control via Computer Vision</b>	0,08944272	0,12649111	0,22188008
<b>Emotion-recognition</b>	0,17888544	0	0

O terceiro passo é calcular o valor ideal e ideal negativo. A solução ideal é calculada com a seguinte fórmula:

$$A^* = \{V_1^*, \dots, V_n^*\}, \text{ onde } V_j^* = \{\max_i(V_{ij}) \text{ if } j \in J; \min_i(V_{ij}) \text{ if } j \in J'\}$$

A solução ideal negativa é calculada com a seguinte fórmula:

$$A' = \{V_1', \dots, V_n'\}, \text{ onde } V_j' = \{\min_i(V_{ij}) \text{ if } j \in J; \max_i(V_{ij}) \text{ if } j \in J'\}$$

Na Tabela 8, a vermelho estão identificados os valores da solução ideal e a azul os valores da solução ideal negativa.

Tabela 8 - Solução ideal e solução ideal negativa

	Face	Posição	Movimento
<b>Real-Time-Action-Recognition</b>	0	0,37947332	0,33282012
<b>Behavior detection and control via Computer Vision</b>	0,08944272	0,12649111	0,22188008
<b>Emotion-recognition</b>	0,17888544	0	0

No passo quatro é feito o cálculo da separação da solução ideal positiva e solução ideal negativa.

O cálculo da separação ideal positiva é feito através da seguinte fórmula:

$$S_i^* = \left[ \sum_j (V_j^* - V_{ij})^2 \right]^{1/2}, i = 1, \dots, m$$

A fórmula para calcular a separação do valor ideal negativo é semelhante:

$$S_i' = \left[ \sum_j (V_j' - V_{ij})^2 \right]^{1/2}, i = 1, \dots, m$$

Na Tabela 9 está representado o cálculo da solução ideal positiva.

Tabela 9 - Separação da solução ideal positiva

	Face	Posição	Movimento	Soma	Si*
<b>Real-Time-Action-Recognition</b>	0,032	0	0	0,032	0,178885
<b>Behavior detection and control via Computer Vision</b>	0,008	0,064	0,01230769	0,084308	0,290358
<b>Emotion-recognition</b>	0	0,144	0,11076923	0,254769	0,504747

A Tabela 10 apresenta o cálculo da separação da solução ideal negativa.

Tabela 10 - Separação da solução ideal negativa

	Face	Posição	Movimento	Soma	Sí'
<b>Real-Time-Action-Recognition</b>	0	0,144	0,11076923	<b>0,254769</b>	0,504747
<b>Behavior detection and control via Computer Vision</b>	0,008	0,016	0,04923077	<b>0,073231</b>	0,270612
<b>Emotion-recognition</b>	0,032	0	0	<b>0,032</b>	0,178885

O quinto e último passo do TOPSIS é o cálculo da proximidade relativa à solução ideal. Este cálculo é feito com a fórmula:

$$C_i^* = \frac{S_i'}{S_i^* + S_i'}, 0 < C_i^* < 1$$

Aplicada esta fórmula, obtemos os valores apresentados na Tabela 11. Deve ser selecionado o valor mais próximo de 1.

Tabela 11 - Cálculo da proximidade relativa à solução ideal

Real-Time-Action-Recognition	0,73833085
Behavior detection and control via Computer Vision	0,4824001
Emotion-recognition	0,26166915

Em conclusão, nesta análise TOPSIS, foram dados pesos maiores às características de deteção da face e deteção de movimento pois, estas são as características com mais importância para este projeto. O método de TOPSIS seleciona a alternativa que está mais próxima do valor da solução ideal e mais afastado do valor da solução ideal negativa. Analisando a Tabela 11, podemos verificar que a ferramenta “Real-Time-Action-Recognition” tem o valor mais próximo de 1 o que significa que este é o valor mais próximo da solução ideal e o valor que está mais afastado do valor da solução ideal negativa. Com base nesta análise, podemos concluir que a ferramenta “Real-Time-Action-Recognition” é a escolha mais adequada para este problema.

### 3.5 Experimentação e avaliação

Desenvolver *software* com qualidade exige a utilização de um sistema de controlo de qualidade. A qualidade pode ser definida como um conjunto de características que devem ser cumpridas de modo que o produto cumpra as necessidades do cliente (Escudeiro & Bidarra, 2008).

A definição da qualidade pode ser iniciada pela correta definição do problema e dos objetivos. Na presente dissertação, pretende-se apresentar uma solução para reduzir os acidentes de trabalhos resultantes da interação entre humanos e robots na indústria. O objetivo é combinar

diferentes tecnologias de deteção de emoções e deteção da posição de pessoas e, com recurso a ML, obter um *software* capaz analisar comportamentos e estados emocionais e prever possíveis acidentes. Outro ponto necessário do ponto de vista da qualidade é a definição da hipótese. A definição da hipótese deve ser uma afirmação preditiva, quantificável, que seja possível verificar e testar. Analisar os resultados de uma experiência requer a identificação dos indicadores e a definição de uma metodologia de avaliação (Escudeiro, 2020).

### **3.5.1 Especificação da hipótese**

Este trabalho foca-se em experimentar combinar ferramentas para monitorização da interação de um operador com a máquina na indústria. Pretende-se combinar diferentes ferramentas informáticas de deteção de emoções e movimentos de um operador e avaliar o desempenho a determinar possíveis acidentes entre um operador e uma máquina.

Nesta experiência foi definida como hipótese a utilização do “Real-Time\_Action-Recognition” combinado com o “Emotion-Recognition”. Com a combinação destas duas ferramentas obtemos um software capaz de identificar a posição do operador e o seu estado emocional. De seguida consegue-se avaliar a precisão na identificação da posição e das emoções do operador.

### **3.5.2 Definição dos indicadores**

De forma a avaliar os resultados da experiência é necessário identificar os indicadores que permitem avaliar a experiência. Com base na solução que se pretende, um indicador necessário deve ser a precisão na identificação das emoções raiva, repulsa, medo, tristeza, felicidade, surpreendido e neutro. Com este indicador, espera-se conseguir avaliar se a ferramenta consegue identificar de forma acertada a emoção correspondente à face do operador.

É necessário outro indicador para a análise da precisão na identificação da posição do operador para avaliar a capacidade da ferramenta em identificar a posição do operador corretamente.

A identificação do risco de acidente requer também um indicador que permita avaliar se a ferramenta identifica corretamente as situações de risco.

O sistema deve conseguir uma precisão superior a 60% para que seja funcional.

Os dados para análise devem ser obtidos num ambiente de simulação.

### **3.5.3 Metodologia de avaliação**

A recolha dos dados deve ser feita num ambiente de experiência controlado.

Através de uma webcam, é feita a captura de imagens e depois processadas pelos algoritmos de identificação de imagens para obter o resultado.

A avaliação da identificação das emoções e da posição do operador é definida com variáveis categóricas: acerta ou não acerta. São necessárias várias execuções para obter um número de vezes que a aplicação identificou corretamente. Com esta experiência consegue-se determinar a precisão da solução.

Devem ser ainda identificadas as variáveis que influenciam o resultado (análise multivariada). A identificação destas variáveis pode ser feita com a repetição da experiência com diferentes configurações. Por exemplo, testar com o operador sem óculos e novamente a utilizar óculos e comparar os resultados. Se a solução identificar as emoções de forma diferente então usar óculos será uma variável que influencia o resultado.

## 4 Detecção de emoções e comportamento de risco

A deteção das emoções permite obter o estado emocional do operador no momento. No entanto é necessário relacionar as emoções do operador com comportamentos de risco. Tendo como base as seis emoções de Ekman, consegue-se obter um sistema capaz de detetar o stress (Hua Gao, 2015). Este sistema deteta as emoções de raiva, repulsa, medo, triste, feliz, surpreendido, neutro (Ekman & Friesen, 1976). Através destas emoções é possível relacionar as emoções “fúria” e “repulsa” com o stress (Hua Gao, 2015) (Zhang, et al., 2020) e assim identificar o risco de o operador estar em stress. Segundo (Giannakakis, et al., 2017), quando o operador manifesta sintomas de stress, identifica-se uma maior percentagem de emoções “fúria” e “repulsa” (Giannakakis, et al., 2017) (Sharma & Gedeon, 2014).

O stress representa um comportamento de risco no contexto da interação do operador com os robots. No entanto, existem outros comportamentos que representam também o risco de ocorrer um acidente como a sonolência e o cansaço. Segundo (Beattie, et al., 2015), o indivíduo com privação de sono tem uma probabilidade elevada de apresentar uma face “triste”. Desta forma, consegue-se relacionar a face “triste” com a sonolência e o cansaço e assim criar uma associação das faces “fúria”, “repulsa” e “tristeza” com comportamentos de risco.

A deteção das emoções é feita com recurso a ML e utiliza uma base de dados pública para treino fer2013 (Arriaga, et al., 2017).

A Figura 8 representa as emoções, de cima para baixo, “fúria”, “felicidade”, “tristeza” e “surpreendido” respetivamente (Arriaga, et al., 2017).



Figura 8 - Expressões faciais (Arriaga, et al., 2017)

As expressões faciais “felicidade” e “surpreendido” não estão associadas a nenhum comportamento de risco (Giannakakis, et al., 2017) (Sharma & Gedeon, 2014) pelo que neste problema, o foco será em identificar as expressões faciais “fúria”, “repulsa” e “tristeza”.

No contexto deste problema, entende-se como “fúria” a expressão facial semelhante à Figura 9.

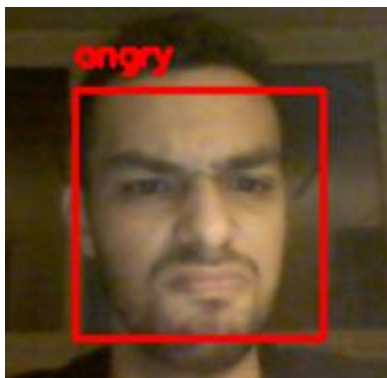


Figura 9 - "Fúria" (Ayman, et al., 2020)

Relativamente à “repulsa”, a expressão facial encontrada deve ser semelhante à Figura 10.



Figura 10 - "Repulsa" (Ayman, et al., 2020)

Por fim, a "tristeza" corresponde a uma expressão facial semelhante à Figura 11.

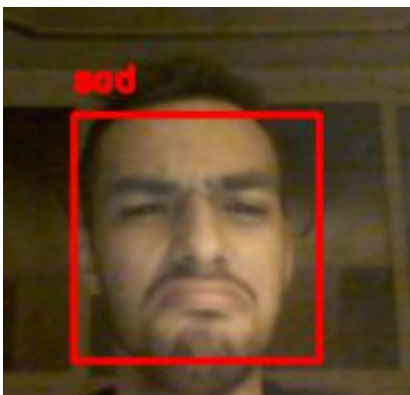


Figura 11 - "Tristeza" (Ayman, et al., 2020)

Se pelo menos uma destas emoções for detetada com frequência no operador num determinado intervalo de tempo, então existe uma elevada probabilidade de que o operador apresentar um comportamento de risco.

É necessário definir a frequência de cada uma destas três emoções, que permita identificar o comportamento de risco. O valor da percentagem de cada emoção, foi afinado através da realização de várias experiências à aplicação, para detetar o estado de stress. As experiências realizadas tiveram como objetivo, chegar a um valor de percentagem de cada uma das três emoções, que permita a aplicação detetar o comportamento do operador como sendo de risco. Foi utilizado um dataset público (Kaggle, 2021) com imagens de pessoas em estado de stress para a realização da experiência. Foram incluídas no dataset, imagens de pessoas em diferentes estados emocionais, como "tristeza", "fúria", "repulsa" e também foram incluídas imagens de pessoas no estado emocional "felizes", para conseguir validar se a aplicação consegue identificar o risco. No anexo 1, pode-se encontrar algumas imagens do dataset. Inicialmente, foi definida a percentagem das emoções da seguinte forma: "fúria" 100%, "repulsa" 100% e "tristeza" 100%. Nas iterações seguintes, foi alterada a configuração da percentagem das emoções até a aplicação começar a detetar o comportamento de risco.

Ao utilizar as imagens do dataset como dados de entrada, o resultado esperado é a aplicação identificar o comportamento como sendo de risco. O resultado obtido da aplicação em cada

iteração foi registrado numa tabela. Foram realizadas várias experiências, com diferentes configurações das frequências das emoções de risco, e registrados todos os valores obtidos para posterior análise. Os dados foram recolhidos e analisados para perceber qual a configuração que melhor deteta o comportamento de risco.

A Tabela 12 contém os resultados obtidos da aplicação em cada uma das iterações. A “Perc. Emoção”, contém a configuração da percentagem, de cada uma das emoções mencionadas, que é considerada para categorizar o comportamento como sendo de risco.

Na primeira iteração, foi considerada a mesma percentagem para as três emoções de 100%, o que significa que com esta configuração, a aplicação só iria considerar que o operador possivelmente estava a ter um comportamento de risco, se fossem detetadas estas emoções em todos os frames do dataset de teste. Neste caso, a aplicação não detetou o comportamento de risco.

Nas iterações seguintes, foi reduzida a percentagem de emoções necessárias para considerar comportamento de risco. A diminuição foi feita iteração a iteração até chegar a um valor em que a aplicação conseguiu detetar o comportamento de risco e assim chegar ao resultado esperado.

Tabela 12 - Resultados das experiências preliminares

Perc. Emoção			Detetou comportamento de risco
Tristeza	Fúria	Repulsa	
100%	100%	100%	Não
90%	90%	90%	Não
80%	80%	80%	Não
50%	50%	50%	Não
50%	40%	20%	Não
20%	40%	20%	Não
30%	30%	15%	Não
30%	30%	10%	Não
30%	20%	10%	Não
20%	20%	10%	Não
20%	15%	7%	Não
15%	15%	7%	Sim

Ao fim de 12 iterações, a aplicação conseguiu detetar o comportamento de risco. Assim, a configuração definida na 12ª iteração foi a melhor para obter um resultado positivo no teste.

A configuração é a seguinte:

- “Tristeza” = 15%
- “Fúria” = 15%
- “Repulsa” = 7%

Esta configuração deverá ser utilizada na aplicação para detetar o comportamento de risco.

## 4.1 Solução

Uma possível abordagem para esta solução, passa pela utilização da ferramenta “Emotion-Recognition” para identificar as emoções do operador. Esta ferramenta consegue identificar as emoções através das imagens capturadas pelas camaras. A emoção identificada é recebida por um outro componente que vai relacionar a emoção com comportamentos de risco.

O primeiro caso de uso que se pretende resolver com este projeto é identificar o comportamento de risco através da observação do operador. A camara capta as imagens da face e o “Emotion-Recognition” deve identificar as emoções. Se as emoções identificadas estiverem associadas a comportamentos de risco, a solução deve escrever o alerta.

Para além da identificação das emoções é necessário identificar a posição e o movimento do operador. Este requisito dá origem a um novo caso de uso. Analisar o movimento do operador e identificar se o mesmo está em risco de colidir com um robot.

Está representado na Figura 12 o diagrama de componentes que demonstra como os componentes estão interligados. O componente “DetetarEmoções” recolhe as imagens capturadas pela camara e identifica a face do operador. Com recurso a ML, caracteriza a expressão obtida numa das sete emoções: fúria, repulsa, medo, tristeza, felicidade, surpresa ou neutro. O componente “IdentificaRisco” recebe as emoções identificadas pelo componente “DetetarEmoções” e avalia se estas emoções representam um comportamento de risco.

O componente “DetetarEmoções”, desenvolvido em python, utiliza o opencv para detetar a face do operador. Para a identificação das emoções, esta ferramenta utiliza a base de dados pública fer2013 como modelo de treino para o algoritmo de deep learning Keras fazer a classificação das expressões faciais em sete emoções raiva, repulsa, assustado, triste, feliz, surpreendido, neutro.

O componente “DetetarMovimentos” recolhe as imagens captadas por uma camara e identifica a posição do operador. O componente “IdentificarRisco” recebe a informação do componente “DetetarMovimentos” e avalia se a trajetória do operador está em risco de colidir com um robot.

O componente “DetetarMovimentos”, desenvolvido em python, recorre ao OpenCV para identificar o corpo do operador na imagem e utiliza o Tensorflow para identificar o movimento do corpo.

O componente “IdentificarRisco” é desenvolvido em python. Este componente recebe a informação da emoção do operador ou a posição do operador dos componentes “DetetarEmoções” e “DetetarMovimentos” respetivamente e avalia se a informação obtida está dentro dos parâmetros de comportamento de risco.

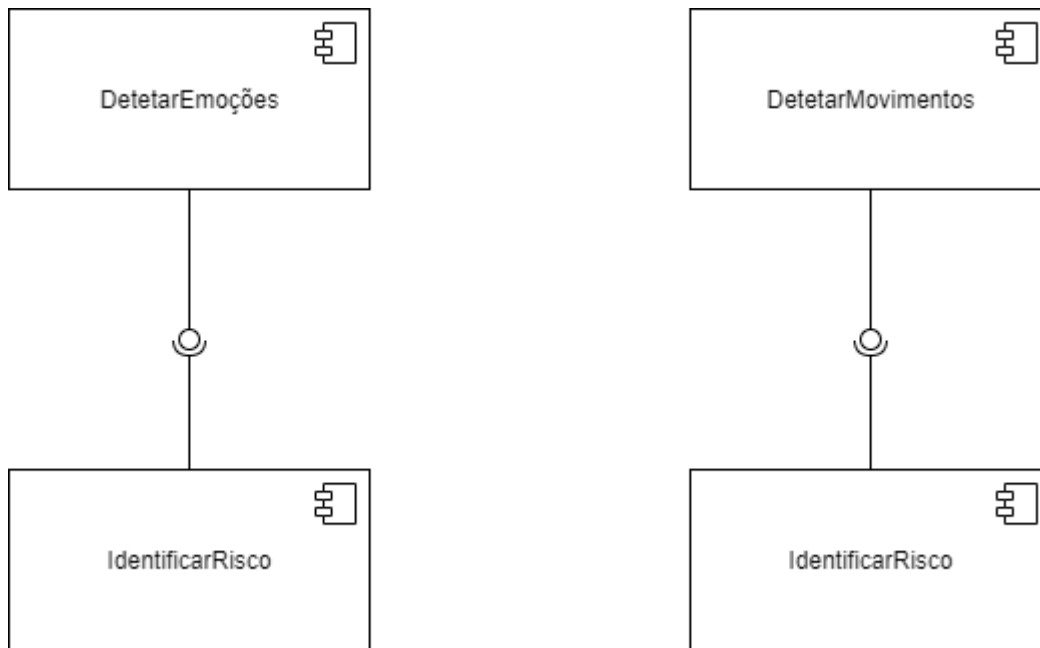


Figura 12 - Diagrama componentes

Com a especificação dos casos de uso, consegue-se avaliar a implementação. Conseguir identificar as emoções é um dos requisitos da solução e pode ser medido pela quantidade de emoções acertadas que a solução consegue identificar.

#### 4.1.1 Caso de uso 1 – Identificação das emoções do operador

Neste caso de uso, o componente “DetetarEmoções” inicia a execução numa máquina.

O componente permanece em execução e identifica as expressões faciais do operador através das imagens recebidas pela camera.

Das emoções detetadas, o componente regista as emoções “tristeza”, “repulsa” e “fúria”.

O componente calcula a percentagem das vezes que estas três emoções foram detetadas.

O componente “IdentificarRisco” vai receber a percentagem das emoções detetadas. O sistema deve considerar comportamento de risco se o operador apresentar uma percentagem da emoção “tristeza” igual a 15% ou uma percentagem da emoção “fúria” igual a 15% ou então uma percentagem da emoção “repulsa” igual a 7%. Estes valores foram obtidos através de experiência em que foram testadas diferentes configurações. Esta configuração foi a que demonstrou ter uma melhor eficiência a detetar o comportamento de risco.

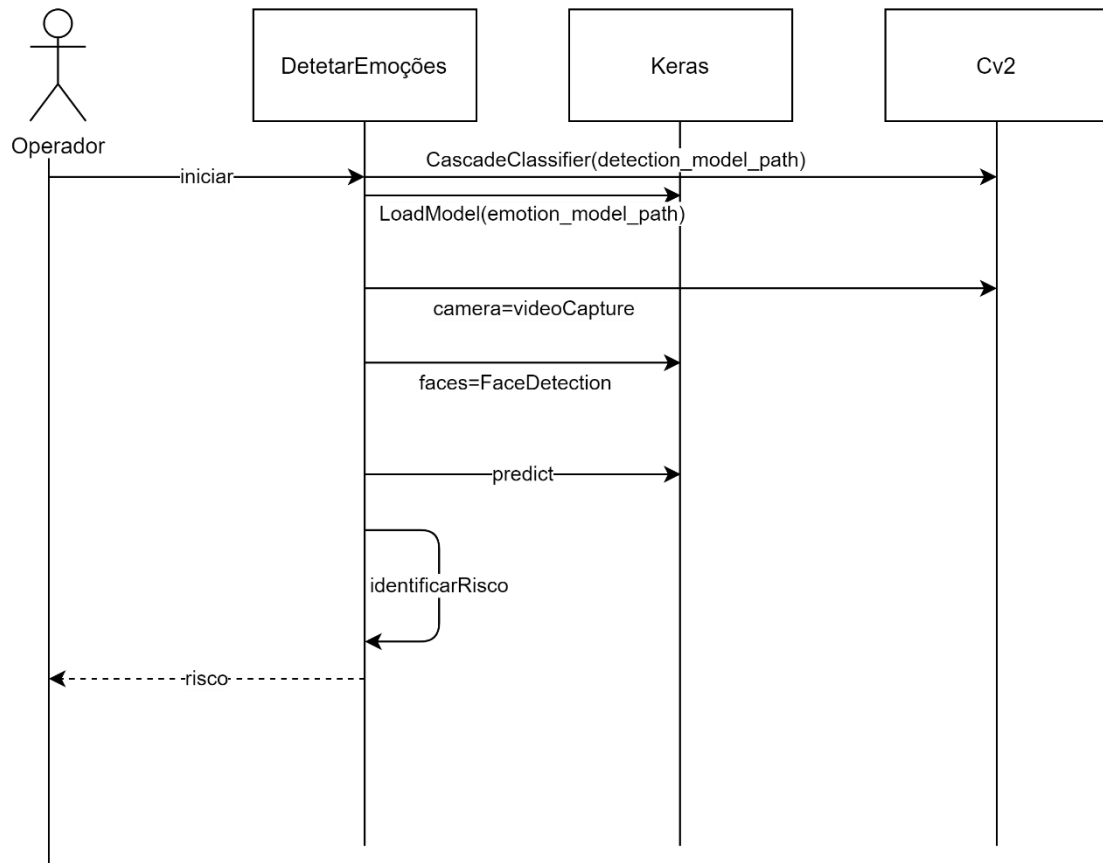


Figura 13 – Diagrama de sequência do UC1

A Figura 13 representa o diagrama de sequência do caso de uso “Identificação das emoções do operador”.

#### 4.1.2 Caso de uso 2 – Identificar risco de colisão do operador com robot

O componente detetar movimentos inicia a execução e identifica a posição do operador através das imagens recebidas da camara.

Identifica as coordenadas do operador.

O componente “IdentificarRisco” recebe as coordenadas do operador e se estas coincidirem com o trajeto de um braço robótico então identifica o risco de colisão e retorna mensagem de perigo.

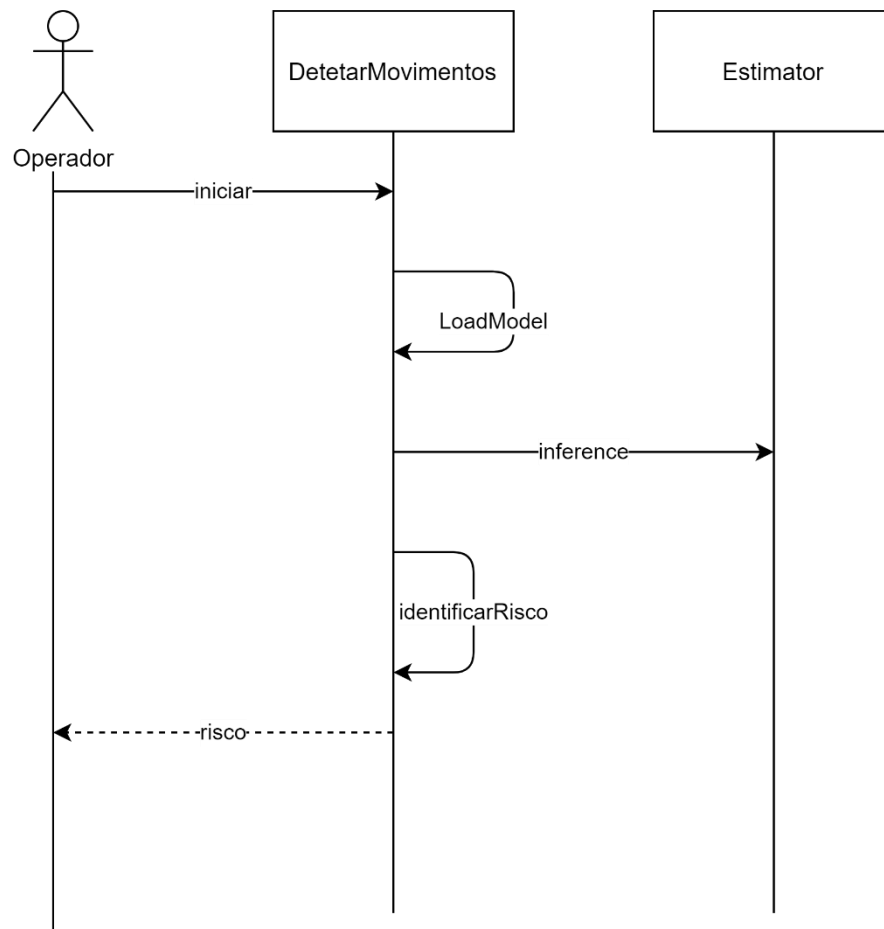


Figura 14 - Diagrama de sequência do UC2

A Figura 14 representa o diagrama de sequência do caso de uso “Identificar risco de colisão do operador com robot”.

# 5 Implementação da solução

Existem várias ferramentas que permitem identificar as emoções do operador conforme referido nos capítulos anteriores. Da comparação entre estas várias ferramentas, foi definida a solução que aparenta ser a mais adequada ao problema em questão.

## 5.1 Solução implementada

A deteção de emoções é feita com a ferramenta “Emotion-Recognition”, que permite detetar a face dos operadores e identificar a emoção atual do operador. Com base nesta ferramenta, foi desenvolvido um modelo que permite identificar o comportamento de risco do operador através das emoções detetadas. Este modelo baseia-se na relação entre as emoções “fúria”, “tristeza” e “repulsa” com comportamentos de risco. A deteção de algum destas emoções indica a probabilidade de o operador ter um comportamento de risco.

O desenvolvimento da aplicação foi feito orientado aos testes, utilizando um *dataset* para testar a aplicação e realizar experiências para chegar a uma configuração do modelo ótima. O *dataset* contém imagens de pessoas que apresentam comportamentos de risco, tais como *stress*, cansaço, agressividade e tristeza. A aplicação processa cada frame do dataset e identifica a emoção da pessoa. Com este *dataset*, o resultado esperado é a aplicação identificar o comportamento de risco. O modelo foi ajustado em cada experiência até obter uma configuração em que a aplicação foi capaz de identificar o comportamento como sendo de risco.

A Figura 15 apresenta um diagrama com os componentes que compõem o sistema.

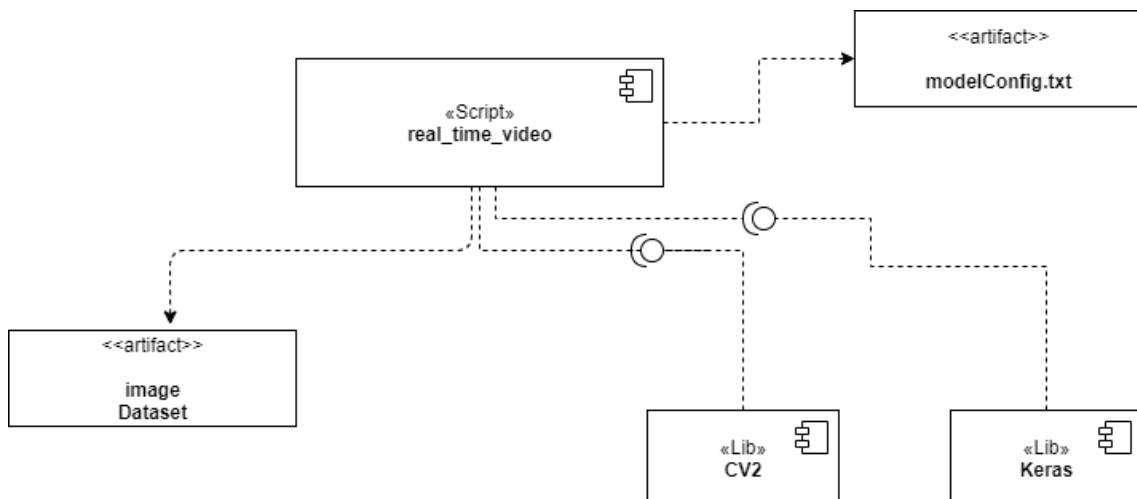
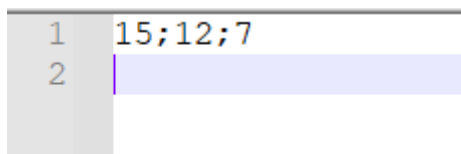


Figura 15 - Treino do sistema

O script `real_time_video` é um script em Python, que lê as imagens obtidas pela câmara e identifica face e respetiva emoção da pessoa na imagem. Este script contém o

EmotionRecognition que faz a identificação das faces das pessoas e as suas emoções, recorrendo às bibliotecas OpenCV e Keras. Neste caso, as imagens são obtidas através de um vídeo criado a partir de um *dataset* de imagens selecionadas.

O modelConfig.txt é um ficheiro que contém a percentagem de cada uma das emoções consideradas de comportamento de risco, que devem ser analisadas pela aplicação. A aplicação lê estas configurações e, com os valores definidos, vai identificar se o comportamento identificado corresponde a risco ou não. Os valores deste ficheiro devem ser afinados através de experiências. A Figura 16 representa o ficheiro com a configuração utilizada.



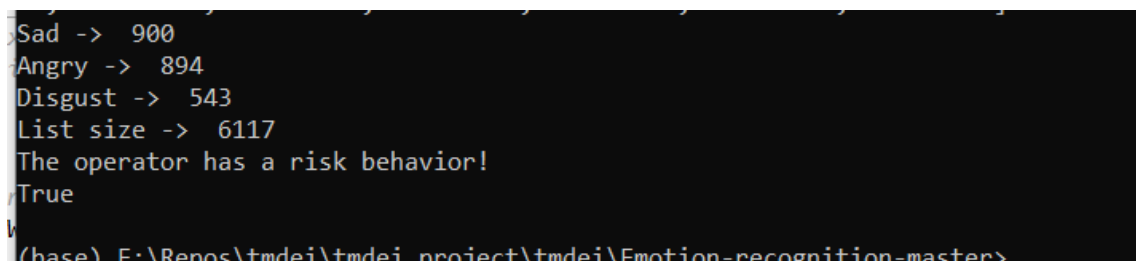
```
1 15;12;7
2
```

Figura 16 - modelConfig.txt

Na primeira coluna do ficheiro, é definida a percentagem de emoção “tristeza” que deve ser considerada, na segunda coluna a percentagem da emoção “fúria” que deve ser considerada e na coluna 3, a percentagem da emoção “repulsa” que deve ser considerada.

O imageDataset é um vídeo criado a partir de imagens de um *dataset*. As imagens utilizadas, contém pessoas que apresentam emoções “fúria”, “tristeza” e “repulsa”, que são as consideradas como emoções que indiciam um potencial comportamento de risco. Também foram adicionadas imagens de pessoas que aparentam felicidade para uma melhor validação se a aplicação consegue identificar o comportamento corretamente. A quantidade de imagens de pessoas felizes é inferior às restantes para garantir que o resultado do dataset deve ser o comportamento de risco. Este dataset foi utilizado para treinar o sistema até que este seja capaz de detetar o comportamento de risco corretamente. Uma vez que o sistema esteja treinado, pode começar a ser utilizado para detetar o comportamento através de imagens obtidas pela câmara.

A Figura 17 apresenta o resultado obtido após a execução da aplicação.



```
Sad -> 900
Angry -> 894
Disgust -> 543
List size -> 6117
The operator has a risk behavior!
True
(base) E:\Beno\tmlde\tmlde_project\tmlde\Emotion-recognition-master>
```

Figura 17 - Resultado da aplicação

A aplicação imprime o número de emoções do tipo “tristeza”, “fúria” e “repulsa” detetadas, assim como o número total de emoções detetadas no *dataset*. A aplicação avalia, através da configuração do modelConfig.txt, se o comportamento é de risco e, em caso afirmativo,

retorna a mensagem “The operator has a risk behavior!” (“o operador apresenta um comportamento de risco!”).

Quando a aplicação arranca, é apresentado o menu que permite seleccionar que a aplicação corra no modo de treino ou não (Figura 18).

```
Select run mode:
1. Train model
2. Run
3. Exit
Select option:3
```

Figura 18 - Menu da aplicação

Ao correr a aplicação no modo de treino, a aplicação carrega o *dataset* e faz a deteção das emoções. No modo Run, a aplicação ativa a câmara e obtém as imagens em direto da câmara.

A análise do risco é feita com base na percentagem de emoções detetadas. A aplicação lê do ficheiro de configuração a percentagem das emoções que deve considerar para que seja identificado o comportamento de risco. Esta ação é feita no código da Figura 19.

```
#Get configuration to analyse the risk behavior
with open(cfg_file) as cfg:
    cfg_reader = csv.reader(cfg, delimiter=';')
    for row in cfg_reader:
        sadPerc = int(row[0])
        angryPerc = int(row[1])
        disgustPerc = int(row[2])

#print(emotionList)
countSad = emotionList.count('sad')
countAngry = emotionList.count('angry')
countDisgust = emotionList.count('disgust')
count = len(emotionList)
print('Sad -> ', countSad)
print('Angry -> ', countAngry)
print('Disgust -> ', countDisgust)
print('List size -> ', count)

probSad = countSad * 100 / count
probAngry = countAngry * 100 / count
probDisgust = countDisgust * 100 / count
```

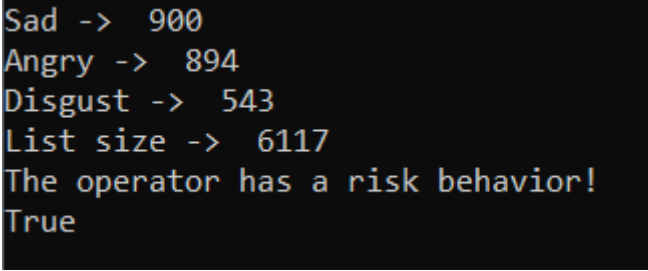
Figura 19 – Conta cada uma das emoções

Na imagem anterior é calculada a percentagem detetada de cada uma destas emoções (“tristeza”, “fúria” e “repulsa”). Esta percentagem é depois comparada com a configuração da aplicação. O código na Figura 20 verifica se cada uma destas três emoções ultrapassa valor do parâmetro definido na configuração.

```
67 isRisk=False
68 if probSad>sadPerc or probAngry>angryPerc or probDisgust>disgustPerc:
69     isRisk = True
70     print('The operator has a risk behavior!')
71 print(isRisk)
72
```

Figura 20 - Calcula o comportamento de risco

Os valores das variáveis “probPerc”, “angryPerc” e “disgustPerc” são os parâmetros configurados na aplicação. Se a percentagem de alguma das três emoções for superior ao valor do respetivo parâmetro, a aplicação considera como comportamento de risco. Os valores destes parâmetros foram definidos através de experiências realizadas previamente com um *dataset* de treino.



```
Sad -> 900
Angry -> 894
Disgust -> 543
List size -> 6117
The operator has a risk behavior!
True
```

Figura 21 - Resultado do treino

A Figura 21 apresenta o resultado impresso pela aplicação. Neste caso, o comportamento do operador foi classificado como sendo de risco. Este é o resultado esperado uma vez que foi utilizado o *dataset* de treino.

Ao extrapolar este comportamento da aplicação para o mundo real, é possível atuar perante o comportamento de risco do operador da máquina. Existem várias ações possíveis de tomar quando este comportamento é detetado. Num ambiente real, a aplicação deve enviar uma mensagem para o operador e para o seu supervisor. O alerta do operador pode ser feito através do envio de uma mensagem para o ecrã e, simultaneamente, ativar um aviso sonoro. O aviso sonoro funciona como um estímulo para o operador, como um mecanismo para obter a atenção do operador e obter o foco dele no equipamento que está a operar e na tarefa. Pode até ser implementado um sistema para interromper a execução do robot se o sistema robótico permitir uma interligação com este sistema.

## 5.2 Implementação

A utilização da linguagem de programação Python é uma escolha óbvia visto que o “Emotion-Recognition” está implementado em Python. Para além disso, o Python disponibiliza diversas bibliotecas que permitem analisar a informação de uma forma rápida e simples.

Na presente solução, é utilizado o artefacto `imageDataset`, que permite treinar a aplicação. Este artefacto é um dataset de imagens selecionadas para obter como resultado o comportamento de risco, que contém imagens de expressões faciais representativas das emoções “fúria”, “tristeza”, “repulsa” e “felicidade”. Incluir estas quatro expressões no dataset é importante para obter uma amostra que permita validar se a aplicação consegue identificar o comportamento de risco corretamente. É considerado comportamento de risco quando existe uma determinada percentagem de emoções “fúria”, “tristeza” ou “repulsa”, é identificada. A percentagem necessita de ser configurada.

A configuração das percentagens de cada emoção a considerar são definidas no ficheiro `modelConfig.txt`. Este ficheiro permite que a aplicação possa ser configurada para o ambiente de execução. O ficheiro de texto contém os valores separados por “;”.

## 5.3 Validação

A implementação de uma solução requer que seja definido um plano para validar e testar a aplicação de forma a validar se o resultado obtido corresponde ao esperado. É necessário, fazer uma validação da implementação para perceber se a solução cumpre os requisitos propostos. Esta validação é feita através do feedback recebido do Stackholder, que valida se o resultado da aplicação cumpre os requisitos pretendidos.

Para além da validação pelo Stackholder, foram realizados testes para verificar se a aplicação consegue atingir o resultado esperado. O desenvolvimento da aplicação baseou-se na utilização de um *dataset*. Com este *dataset*, o resultado esperado é a identificação do comportamento de risco.

Foram realizados vários testes e registado o número de emoções detetadas por tipo de emoção. Considerando as imagens do *dataset*, foi possível validar se a aplicação consegue identificar e catalogar cada emoção corretamente.

As experiências realizadas permitiram ajustar a aplicação até esta conseguir chegar ao resultado esperado.

Foram realizados testes funcionais para validar as funcionalidades implementadas do ponto de vista dos requisitos. A utilização de um *dataset* foi importante para a realização destes testes ao disponibilizar um conjunto de dados de entrada e um resultado esperado.

O primeiro teste focou-se na validação da deteção do comportamento de risco. Foi utilizado um *dataset* que resultaria na deteção do comportamento de risco. Este *dataset* foi compor

maioritariamente com imagens de emoções relacionadas com o comportamento de risco (“tristeza”, “fúria” e “repulsa”).

As Figura 22, Figura 23 e Figura 24 são exemplos das emoções representativas do comportamento de risco. Estas imagens estão incluídas no *dataset*.



Figura 22 – Fúria



Figura 23 – Tristeza



Figura 24 – Repulsa

Foi executada a aplicação com este *dataset* (Figura 25) e foi comparado o resultado com o resultado esperado. O resultado do teste foi positivo uma vez que retornou o comportamento de risco.

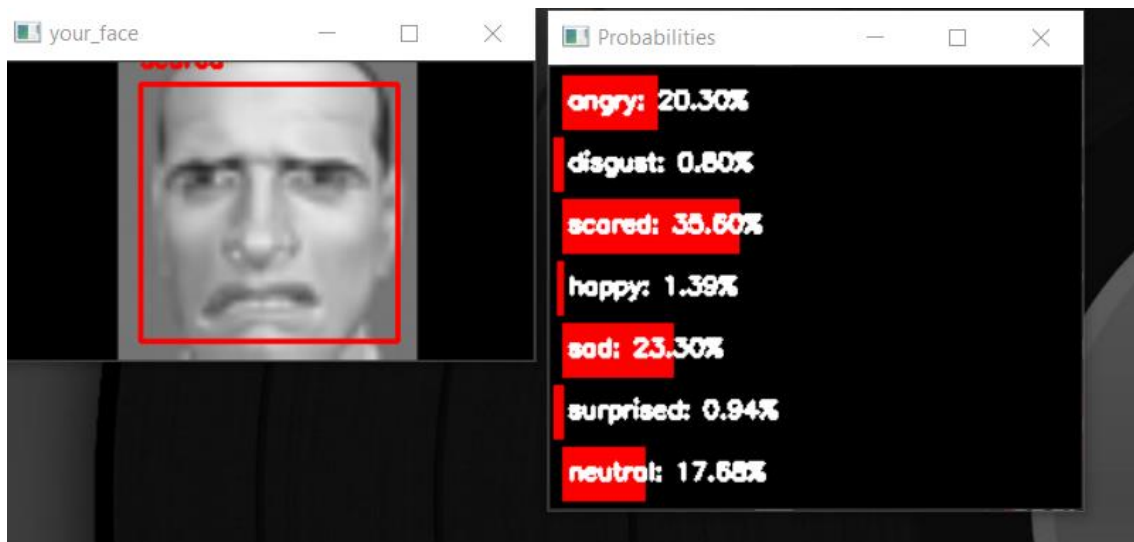


Figura 25 - Aplicação em execução no modo treino

A aplicação analisou todo o *dataset* e identificou as emoções de cada frame do dataset. No fim retornou que existe comportamento de risco (Figura 26).

```
Sad -> 900  
Angry -> 894  
Disgust -> 543  
List size -> 6117  
The operator has a risk behavior!  
True
```

Figura 26 - Resultado da aplicação com o *dataset* de comportamento de risco

Um segundo teste foi efetuado utilizando um *dataset* de imagens que não representam o comportamento de risco. Este segundo teste foi importante para validar se aplicação deteta o comportamento de risco apenas nas situações em que este está presente.

Foi criado um *dataset* de imagens representativas de emoções que não estão relacionadas com o comportamento de risco. À semelhança do outro *dataset*, foram também incluídas algumas imagens representativas do comportamento de risco para tornar o teste mais rigoroso. O resultado esperado com este *dataset* é que a aplicação não identifique o comportamento de risco e não deve apresentar a mensagem “The operator has a risk behavior!”. Após a execução a aplicação retornou o resultado esperado e o teste foi bem-sucedido. A Figura 27 apresenta o resultado obtido da aplicação.

```
Sad -> 450
Angry -> 264
Disgust -> 0
List size -> 5220
False
```

Figura 27 - Resultado obtido com o *dataset* sem comportamento de risco

Os testes funcionais foram realizados com sucesso e foram importantes para garantir que a aplicação retorna o resultado esperado.

Quanto aos testes não funcionais, foram realizados testes de confiabilidade para validar se a aplicação consegue executar sem falhas num determinado período de tempo e retornar o resultado esperado. Estes testes foram realizados com o *dataset*. Aplicação conseguiu correr e analisar todo o *dataset* sem ocorrer nenhuma falha na sua execução.

Do ponto de vista da estabilidade, foram realizados testes para avaliar se a aplicação consegue executar por longos períodos de tempo sem falhar. A aplicação foi executada durante 10 minutos no modo “Run”. Durante este tempo a aplicação obteve imagens da camera e identificou as emoções da face captada na imagem sem falhas.

Relativamente à escalabilidade, esta aplicação só permite ser utilizada por um utilizador e numa máquina pelo que não faz sentido testar a escalabilidade da aplicação.

O caso de uso 2 permite analisar a posição do operador e identificar quando este está posicionado num local em que existe o risco de colisão com um robot. Foi feita a análise e design da solução utilizando a ferramenta “Real-Time-Action-Recognition” que permite identificar a posição do corpo do operador. Contudo, este caso de uso não foi implementado neste trabalho. O desenvolvimento deste caso de uso necessita da utilização de dois *datasets* que identifiquem uma situação em que o operador está dentro da área de segurança e outro em que está fora. O resultado esperado para o primeiro *dataset* é que a aplicação não retorne que o operador está numa zona de perigo de colisão com um robot. No segundo *dataset*, o resultado esperado é que a aplicação identifique que o operador está em perigo de colisão com um robot. A zona de colisão deve ser configurada na aplicação para que seja possível identificar a zona em que o operador deve estar posicionado.

## 6 Conclusão

O presente projeto permitiu utilizar uma ferramenta de detecção de emoções humanas para identificar o comportamento de risco de um operador. Com esta implementação, foi possível obter uma solução que permite diminuir o número de acidentes de trabalho, entre operadores e robots, provocados por comportamentos de risco praticados pelo operador. A aplicação desenvolvida é capaz de identificar o comportamento de risco com base nas emoções detetadas.

Um dos objetivos propostos para este problema era obter uma solução capaz de detetar o comportamento de risco de um operador com base nas suas emoções. A utilização da ferramenta “Emotion-Recognition”, permitiu obter uma solução que deteta as emoções do operador. Através das emoções é feita uma relação entre as emoções que indiciam um possível comportamento de risco. A utilização de um *dataset* permitiu treinar e testar a aplicação de forma a esta devolver resultados precisos.

Foi analisada e desenhada a solução para detetar a posição do operador e o seu movimento com a utilização da ferramenta “Real-Time-Action-Recognition”. Com a solução proposta é possível identificar se a posição do operador coincide com o trajeto de um robot. No entanto, esta solução não foi implementada.

### 6.1 Resultados obtidos

A implementação do caso de uso 1 foi feita com a utilização de um *dataset* com imagens de faces que apresentam emoções características do comportamento de risco. A utilização desta metodologia proporcionou um ambiente de teste, em que existe um resultado esperado.

Este processo de desenvolvimento permite desenvolver a aplicação através de uma metodologia orientada aos testes. O primeiro passo deste processo de desenvolvimento consiste na especificação dos testes. Para o caso de uso 1, foram definidos dois *datasets* que serviram como dados de entrada. Um dos *datasets* tem como resultado a identificação do comportamento de risco. A seleção das imagens deste *dataset* é um ponto fundamental pois, devem representar uma amostra de expressões representativas do comportamento de risco. O outro *dataset* utilizado contém imagens não representativas do comportamento de risco e o resultado esperado com este *dataset* é que a aplicação não identifique o comportamento de risco e não retorne a mensagem a indicar que o operador apresenta um comportamento de risco. Os testes realizados com cada um dos *datasets* foram bem-sucedidos.

O desenvolvimento do caso de uso 2 deve seguir a mesma metodologia orientada aos testes. É necessário criar um caso de teste para a situação em que o operador está em risco de colisão com o braço robótico e outro caso de teste em que o operador está em segurança. Estes testes devem ser feitos através da utilização de *datasets*. Os resultados obtidos permitem avaliar e medir o desenvolvimento do caso de uso.

## 6.2 Trabalho futuro

No presente trabalho foi implementada a função para detetar o comportamento de risco através das emoções detetadas. A deteção do comportamento de risco pode ser complementada de diversas formas num ambiente real. Podem ser implementadas soluções para notificar o operador e o seu supervisor da existência de comportamento de risco para que estes possam tomar as devidas ações para manter a segurança do operador. Quando o comportamento de risco é detetado, a aplicação deve imprimir uma mensagem no ecrã do operador e simultaneamente iniciar um sinal sonoro. O uso do sinal sonoro é importante para obter a atenção do operador. O operador deve corrigir o seu comportamento.

Para além da notificação do operador, é importante que o seu supervisor seja informado para que possa tomar as devidas precauções. Em caso de comportamento de risco detetado, a aplicação envia um email ou uma *sms* para o supervisor que, deverá aproximar-se do operador e avaliar o seu comportamento para tomar as devidas precauções.

A aplicação deve interromper a execução do sistema robótico para evitar qualquer tipo de acidente. O normal funcionamento do sistema robótico só deve ser repostado após uma avaliação do supervisor e confirmação.

Esta solução pode ser complementada com a utilização da ferramenta “*Real-Time-Action\_Recognition*” que permite detetar e rastrear a posição do operador. Futuramente esta solução pode ser implementada de forma a detetar possíveis colisões do operador com o sistema robótico. Ao implementar esta solução, será mais um nível de deteção de acidentes pois não estará apenas a observar o comportamento do operador, mas também a sua posição. Caso o operador se desloque para um local em que o sistema robótico estará a operar, a aplicação deve imediatamente interromper a atividade do sistema robótico e notificar o operador através de uma mensagem no seu ecrã de trabalho e simultaneamente iniciar um sinal sonoro para alertar do perigo.

## 7 Referências

Warwick Manufacturing Group, 2007. *Quality Function Deployment*. UK: s.n.

Andresen, S., 2013. <http://sa-cybernetics.github.io/>. [Online] Available at: <https://github.com/TheoLvs/computer-vision-control#behavior-detection-and-control-via-computer-vision> [Acedido em 03 02 2021].

Arriaga, O., Plöger, P. G. & Valdenegro, M., 2017. Real-Time Convolutonal Neural Networks for Emotion and Gender Classification. 23 10.

Ayman, O., Junior, G. & Srikanth, A., 2020. *GitHub*. [Online] Available at: <https://github.com/omar178/Emotion-recognition> [Acedido em 17 02 2021].

Beattie, L., Kyle, S. D., Espie, C. A. & Biello, S. M., 2015. Social interactions, emotion and sleep: A systematic review and research agenda. *Sleep Medicine Reviews*, pp. 83-100.

Bewley, A. et al., 2016. SIMPLE ONLINE AND REALTIME TRACKING. *2016 IEEE International Conference on Image Processing (ICIP)*.

Biswas, S. & Chatterjee, S., 2019. *GitHub*. [Online] Available at: <https://github.com/susantabiswas/realtime-facial-emotion-analyzer> [Acedido em 17 02 2021].

Cao, Z. et al., 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 6-7.

Carneiro, D. R., 2013. An agent-based architecture for online. 06.

Chen, T.-Y., Wu, M.-Y., Hsieh, Y.-H. & Fu, L.-C., 2016. Deep Learning for Integrated Hand Detection and Pose Estimation. *International Conference on Pattern Recognition (ICPR)*, Volume 23, pp. 4-8.

Clarke, J. R. & Clarke, S. M., 1997. *Sleep detection and driver alert apparatus*. Estados Unidos da América, Patente Nº US5689241A.

Cohen, Y., Golan, M., Singer, G. & Faccio, M., 2018. Workstation–Operator Interaction in 4.0 Era: WOI 4.0. *ScienceDirect*.

DOBRESCU, E. M. & DOBRESCU, E. M., 2017. THE FUTURE OF THE ARTIFICIAL INTELLIGENCE IN ECONOMICS AND MANAGEMENT. *Review of General Management*.

DRM Associates, 2019. *NPD Solutions*. [Online] Available at: <http://www.npd-solutions.com/bok.html> [Acedido em 13 01 2021].

Ekman, P. & Friesen, W. V., 1976. Measuring facial movement. *Environmental Psychology and Nonverbal Behavior*, p. 55–75.

Escudeiro, N., 2020. *TMDEI Experimentação e Avaliação*, s.l.: s.n.

Escudeiro, P. & Bidarra, J., 2008. Quantitative Evaluation Framework (QEF). *Revista Ibérica de Sistemas e Tecnologias de Informação*, 1(1), pp. 16-27.

Flintbox, 2019. *2017-163 - OpenFace- Facial Landmark Detection and Facial Expression Analysis Software*. [Online] Available at: <https://cmu.flintbox.com/#technologies/5c5e7fee-6a24-467b-bb5f-eb2f72119e59> [Acedido em 26 02 2021].

Geitgey, A., 2016. *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning*. [Online] Available at: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78> [Acedido em 03 02 2021].

Giannakakis, G. et al., 2017. Stress and anxiety detection using facial cues from videos. *Biomedical Signal Processing and Control*, Volume 31, pp. 89-101.

Golan, M., Cohen, Y. & Singer, G., 2019. A framework for operator – workstation interaction in Industry 4.0. *International Journal of Production Research*, 58(8), pp. 2421-2432.

Haenlein, M. & Kaplan, A., 2019. A Brief History of Artificial Inteligence: On the Past, Present, and Future of Artificial Inteligence. *Haas School Of Business University Of California Berkeley*.

Hidalgo, G. et al., 2020. *github*. [Online] Available at: <https://github.com/CMU-Perceptual-Computing-Lab/openpose> [Acedido em 02 02 2021].

Hua Gao, A. Y. J.-P. T., 2015. DETECTING EMOTIONAL STRESS FROM FACIAL EXPRESSIONS FOR DRIVING SAFETY. *2014 IEEE International Conference on Image Processing (ICIP)*, 29 01.

Ildoonet, 2018. *github*. [Online]  
Available at: <https://github.com/ildoonet/tf-pose-estimation/blob/master/etc/ros.md>  
[Acedido em 01 02 2021].

Indri, M., Grau, A. & Ruderman, M., 2018. Guest Editorial Special Section on Recent Trends and Developments in Industry 4.0 Motivated Robotic Solutions. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 04.

International federation of robots, 2020. *ifr*. [Online]  
Available at:  
[https://ifr.org/img/worldrobotics/Executive\\_Summary\\_WR\\_2020\\_Industrial\\_Robots\\_1.pdf](https://ifr.org/img/worldrobotics/Executive_Summary_WR_2020_Industrial_Robots_1.pdf)  
[Acedido em 23 01 2021].

Jay Lee, H. D. J. S. V. P., 2018. Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 10, pp. 20-30.

Jin, C.-B., Do, T. D., Liu, M. & Kim, H., 2019. *Real-Time Action Recognition Using Multi-level Action Descriptor and DNN*, s.l.: IntechOpen.

Kaggle, 2021. *Kaggle*. [Online]  
Available at: <https://www.kaggle.com/ananthu017/emotion-detection-fer>  
[Acedido em 15 08 2021].

King, D. E., 2015. Max-Margin Object Detection. *Max-Margin Object Detection*, 31 01.

Lee-Morrison, L., 2019. *Portraits of Automated Facial Recognition*. Bielefeld, Alemanha: transcript Verlag.

Nicola, S., 2020. *Análise de Valor*, s.l.: ISEP Instituto Superior de Engenharia do Porto.

Patacchiola, M. & Cangelosi, A., 2017. Head pose estimation in the wild using Convolutional Neural Networks and adaptive gradient methods. 01 11.

Perez, A., 2018. *GitHub*. [Online]  
Available at: <https://github.com/thoughtworksarts/EmoPy>  
[Acedido em 27 02 2021].

Quality One, 2020. *Quality Function Deployment (QFD)*. [Online]  
Available at: <https://quality-one.com/qfd/>  
[Acedido em 31 12 2020].

Raut, N., 2018. Facial Emotion Recognition Using Machine. *Facial Emotion Recognition Using Machine*, 05.

Ribeiro, J., Lima, R., Eckhardt, T. & Paiva, S., 2021. Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review. *Procedia Computer Science*, pp. 51-58.

Robotic Industries Association, 2018. Six future trends in robotic automation. *Innovations Back to basics*, p. 64.

Rocha, R. & Praça, I., 2020. FullExpression - Using transfer learning in the.

S. ROBLA-GÓMEZ, V. M. B. E. G.-S. C. T.-F. J. P.-O., 2017. Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments. 14 11.

Shah, D. & Shah, N., 2012. *Human Behavior Recognition*, s.l.: Department of Electrical Engineering Electronics & Communication Engineering Program Institute of Technology, Nirma University.

Sharma, N. & Gedeon, T., 2014. Modeling observer stress for typical real environments. *Expert Systems with Applications*, 41(5), pp. 2231-2238.

Statista, 2021. *Industrial Internet of Things market size worldwide from 2017 to 2025\**. [Online] Available at: <https://www.statista.com/statistics/611004/global-industrial-internet-of-things-market-size/> [Acedido em 24 01 2021].

Surinta, O. & Khruahong, S., 2019. Tracking People and Objects with an Autonomous Unmanned Aerial Vehicle using Face and Color Detection.

Tadas Baltrušaitis, 2020. *GitHub*. [Online] Available at: <https://github.com/TadasBaltrusaitis/OpenFace> [Acedido em 17 02 2021].

Ustundag, A. & Cevikcan, E., 2018. *Industry 4.0: Managing The Digital Transformation*. Switzerland: Springer Series in Advanced Manufacturing.

Vries, G. J. d., Gentile, E., Miroudot, S. & Wacker, K. M., 2020. The rise of robots and the fall of routine jobs. *Labour Economics*, 66(101885).

Williams, D. D. & Allen, I. L., 2017. Using artificial intelligence to optimize the value of robotic process automation.

Yang, W., Zhang, X. & Li, J., 2020. A local multiple patterns feature descriptor for face recognition. *Neurocomputing*, 15 01, pp. 109-122.

Zhang, H. et al., 2020. Video-Based Stress Detection through Deep Learning. 28 09.

ZHENG, P. et al., 2018. Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives.

## Anexo 1



Figura 28 – Fúria



Figura 29 – Repulsa



Figura 30 – Felicidade



Figura 31 – Tristeza