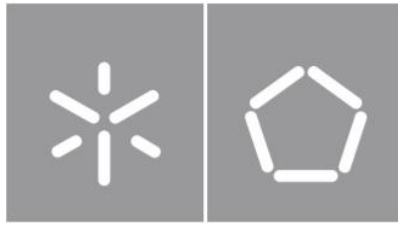




Universidade do Minho
Escola de Engenharia

Luís Miguel dos Santos Silva

**Desenvolvimento de Sistema
Mecatrónico para a Identificação e
Gestão de Peças de Vestuário para
Cegos**



Universidade do Minho

Escola de Engenharia

Luís Miguel dos Santos Silva

**Desenvolvimento de um Sistema
Mecatrónico para Identificação e
Gestão de Peças de Vestuário para
Cegos**

Dissertação de Mestrado

Mestrado Integrado em Engenharia Eletrónica

Industrial e de Computadores

Controlo, Automação e Robótica

Trabalho efetuado sob a orientação de

Professor Doutor João Sena Esteves

Professor Doutor Vítor Carvalho

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.



**Atribuição
CC BY**

<https://creativecommons.org/licenses/by/4.0/>

Agradecimentos

Neste momento especial, gostaria de agradecer a todas as pessoas com quem me deparei ao longo deste percurso que agora termina, em especial aos orientadores, Professor João Sena Esteves e Professor Vítor Carvalho, pelo tempo despendido nas diversas reuniões, guiando-me sempre na escolha das melhores ideias para o projeto. Ao aluno de Doutoramento, Daniel Rocha, que apresenta uma importância fulcral no projeto, por me ter “lançado” o tema, integrando-me da melhor maneira na equipa do *Fantastic Lab*, que me permitiu ter sempre um acompanhamento mais rigoroso ao longo do desenvolvimento do projeto. Um agradecimento a todos os colegas e amigos, em especial ao André Lopes e João Cruz, por estarem sempre disponíveis para ajudar e por terem estado sempre presentes em todos os momentos ao longo dos últimos cinco anos. Devo também os agradecimentos à FCT – Fundação para a Ciência e Tecnologia por ter financiado parcialmente este trabalho no âmbito do Projeto UIDB/00319/2020 (Financiamento Plurianual das Unidades de I&D). Por último, agradecer à minha família que sempre foi o meu suporte e me apoiou incondicionalmente em todas as minhas decisões ao longo desta jornada.

A todos eles, um Muito Obrigado.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Data: Guimarães, 31 janeiro de 2023

Aluno: Luís Miguel dos Santos Silva

Resumo

Desenvolvimento de um Sistema Mecatrónico para Identificação e Gestão de Peças de Vestuário para Cegos

Nos últimos anos, a integração de indivíduos com deficiência visual na sociedade tem vindo a ganhar cada vez mais atenção e a ser fortemente estimulada por diversas atividades. Neste contexto, a tecnologia apresenta uma importância fulcral. Contudo, apesar de todos os esforços que têm vindo a ser realizados, ainda existem alguns desafios no que se refere a tarefas básicas específicas na rotina diária das pessoas cegas, nomeadamente, a gestão e a identificação de peças de roupa, que se pode tornar uma tarefa complexa e bastante demorada. Relativamente a esta tarefa em concreto, as pessoas cegas tornam-se dependentes de familiares ou amigos para uma escolha exata das peças de roupa desejadas.

Com base na problemática apresentada, nesta dissertação propõe-se o desenvolvimento do protótipo de um sistema mecatrónico capaz de auxiliar pessoas cegas na identificação e escolha de peças de vestuário. Esta proposta encontra-se integrada num projeto em desenvolvimento que tem como objetivo proporcionar uma melhoria na qualidade de vida e bem-estar das pessoas cegas. Desenvolveu-se um automatismo mecatrónico que possibilita a identificação de peças de roupas do utilizador. Usaram-se sensorização e motorização diversificadas. Para a interação com o utilizador, existe uma interface desenvolvida a partir da implementação de um servidor que permite a gestão dos pedidos por parte dos utilizadores. Para a obtenção de dados importantes das peças de roupa, implementou-se um algoritmo para a segmentação e classificação das peças de roupa e um algoritmo para a determinação das cores predominantes de cada peça. Obteve-se uma exatidão entre 75% e 100% no reconhecimento de diferentes tipos de peças de roupa.

Palavras-chave: Aprendizagem Máquina, Peças de Roupa, Pessoas Cegas, Sistema Mecatrónico.

Abstract

Development of a Mechatronic System for Identification and Management of Garments for Blind People

In the last years, the integration of handicapped individuals in society has gained significant attention and is being strongly stimulated by several activities. In this context, technology has major importance. However, besides all efforts that have been made, still exist some challenges related to specific basic tasks in blind people's daily routines, namely, the management and identification of personal garments that could become a complex and time-consuming task. For this specific task, these people depend on their relatives or friends for choosing the exact clothes desired.

In this way, and based on the problems presented, this document proposes the development of a prototype of a mechatronic system capable to assist blind people in the identification and choice of personal garments. This proposal is integrated in a project under development, and the proposed solution seeks to provide an improvement in the quality of life and well-being of blind people. A mechatronic automatism was developed that allows the identification of user garments using sensors. For user interaction, there was an interface developed with the implementation of a server responsible for managing the requests from the user. For the obtainment of important data from the garments, were implemented algorithms for segmentation and classification of garments and for detecting the predominant colors of each garment. An accuracy between 75% and 100% was achieved in the recognition of different types of clothing.

Keywords: Blind People, Garments, Mechatronic System.

Índice

Abreviaturas e Siglas	x
Índice de Figuras	xi
Índice de Tabelas	xiii
1. Introdução	1
1.1. Enquadramento	1
1.1.1. Integração da comunidade cega na sociedade	1
1.1.2. A tecnologia e a sua aceitação na vida quotidiana	2
1.2. Motivação e Objetivos	3
1.3. Organização da Dissertação	3
2. Estado da Arte	4
2.1. Soluções tecnológicas para cegos	4
2.1.1. Soluções para cegos a nível de vestuário	4
2.2. Soluções existentes no mercado	7
2.2.1. Armários inteligentes	8
2.2.1.1. Amazon Echo Look	8
2.2.1.2. Fashion API	9
2.2.1.3. Smart Closet	9
2.2.1.4. TailorTags	9
2.2.1.5. Smart Wardrobe	10
2.3. Segmentação de imagem	11
3. Fundamentos teóricos	14
3.1. Machine Learning	14
3.2. Transfer Learning	15
3.3. Redes Neurais	16
3.3.1. Redes Neurais Convolucionais	17
3.4. Faster R-CNN	18
3.4.1. Region Proposal Network	19
3.4.1.1. Region of Interest	20
3.4.2. Mask R-CNN	20
3.4.2.1. Detectron2	22
3.5. Métricas avaliadoras do Modelo	23

3.5.1. AP (Precisão Média)	23
3.5.2. IoU (Intersect Over Union)	25
4. Sistema desenvolvido	27
4.1. Visão Geral do Sistema	27
4.2. Modelo físico	28
4.3. Hardware utilizado	30
4.3.1. Placa de desenvolvimento	30
4.3.2. Leitor NFC	31
4.3.3. Atuadores	32
4.3.4. Câmara	33
4.3.5. Iluminação	33
4.4. Arquitetura de software	34
4.4.1. Software de controlo de hardware	34
4.4.2. Servidor HTTP	38
4.4.3. Algoritmos de processamento de imagem	41
4.4.3.1. Algoritmos de Machine Learning	41
4.4.3.2. Algoritmo de deteção de cor	47
5. Testes e resultados	51
5.1. Testes Realizados	51
5.2. Validação preliminar	69
6. Conclusões e Trabalho Futuro	72
Referências Bibliográficas	74

Abreviaturas e Siglas

ACAPO – Associação dos Cegos e Amblíopes de Portugal

API – Application Programming Interface

CMOS – Complementary Metal–Oxide–Semiconductor

CSI – Channel state information

IoU – Intersect Over Union

I2C – Inter-Integrated Circuit

HTTP – Hypertext Transfer Protocol

LED – Light-emitting diode

QR – Quick Response

RAM – Random-access memory

RGB – Red, Green, and Blue

RPN – Region Proposal Network

ROI – Region of Interest

SIFT – Scale-invariant feature transform

SQL – Structured Query Language

SVM – Support vector machine

SPI – Serial Peripheral Interface

TTL – Transistor-Transistor Logic

UART – Universal Asynchronous Receiver-Transmitter

UID – Unique Identifier

Índice de Figuras

Figura 1 - Ciclo do Transfer Learning	16
Figura 2 - Modelo de uma Rede Neuronal Convolutacional	17
Figura 3 - Camada de Region Proposal	20
Figura 4 - Arquitetura Mask R-CNN	22
Figura 5 - Sumário das previsões do modelo	24
Figura 6 - Cálculo do IoU	26
Figura 7 - Visão Geral do Sistema	27
Figura 8 - Armário usado para teste	29
Figura 9 - Arquitetura de <i>hardware</i>	30
Figura 10 - Raspberry Pi 3B+	31
Figura 11 - Leitor PN532 NFC	31
Figura 12 - Motor de Passo 28BYJ-48 e driver ULN2003	32
Figura 13 - Câmara OV5647	33
Figura 14 - Fluxo do <i>software</i> de controlo	35
Figura 15 - Função Encode_State	36
Figura 16 - Fluxograma do Sistema	38
Figura 17 - Visão Geral do Servidor	39
Figura 18 - Parâmetros do Modelo treinado	40
Figura 19 - Fluxo do menu principal	41
Figura 20 - Segmentação e Classificação no <i>VGG Annotator</i>	43
Figura 21 - Função <i>get_garments_dicts</i>	44
Figura 22 - Parâmetros do modelo	45
Figura 23 - Fluxo da função segment	47
Figura 24 - Peça de roupa segmentada e respetiva máscara binária	48
Figura 25 - Algoritmo de Detecção de Cor	50
Figura 26 - Foto da zona de cima da peça	52
Figura 27 - Foto da zona de baixo da peça	52
Figura 28 - Foto total da peça	52
Figura 29 - Foto da zona de baixo da peça	53
Figura 30 - Foto da zona de cima da peça	53
Figura 31 - Foto total da peça	53
Figura 32 - Fotos captadas fora do ambiente controlado (esquerda e centro) e Foto captada dentro do ambiente controlado (direita)	54
Figura 33 - Fotos captadas fora do ambiente controlado (esquerda e centro) e Foto captada dentro do ambiente controlado (direita)	54
Figura 34 - Fotos captadas fora do ambiente controlado (esquerda e centro) e Foto captada dentro do ambiente controlado (direita)	54
Figura 35 - Imagem segmentada	55
Figura 36 - Perda Total do modelo às 5000 iterações	56
Figura 37 - Perda Total do modelo às 3000 iterações	58
Figura 38 - Máscaras Previstas (à esquerda) e Máscaras obtidas (à direita) para um THR=0.5	59
Figura 39 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)	60

Figura 40 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)	61
Figura 41 - Imagem classificada e Segmentada (à esquerda); Máscara Binária (à direita)	61
Figura 42 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)	62
Figura 43 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)	62
Figura 44 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)	63
Figura 45 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)	63
Figura 46 - Distribuição das cores principais na peça de roupa	64
Figura 47 - Imagem Original (à esquerda); Segmentação (à direita)	65
Figura 48 - Distribuição das cores	65
Figura 49 - Imagem Original (à esquerda); Segmentação (à direita)	66
Figura 50 - Distribuição das cores	66
Figura 51 - Imagem Original (à esquerda); Segmentação (à direita)	67
Figura 52 - Distribuição das cores	67
Figura 53 - Imagem Original; Distribuição de cores	68
Figura 54 - Imagem Original; Distribuição de cores	68
Figura 55 - Imagem Original; Distribuição de cores	69

Índice de Tabelas

Tabela 1 - Características das soluções no mercado	11
Tabela 2 - Precisão Média para 5000 iterações	56
Tabela 3 - Precisão Média para 3250 iterações	57
Tabela 4 - Precisão Média para 3000 iterações	57
Tabela 5 - Precisão Média para 2500/2750 iterações	58
Tabela 6 - <i>Intersect Over Union</i> obtida nos treinos	60

1. Introdução

Neste capítulo, são evidenciadas as motivações que levaram à escolha do tema desta dissertação, bem como a problemática em que o mesmo se enquadra no contexto social e comunitário. Para além disso, são elencados os objetivos principais do trabalho desenvolvido e, por último, é apresentada a forma como se encontra estruturada a dissertação.

1.1. Enquadramento

Ao longo dos últimos anos tem estado cada vez mais na ordem do dia a problemática da integração na sociedade dos indivíduos com algum tipo de deficiência, assistindo-se a diversas iniciativas destinadas a que estes grupos se sintam autónomos e plenamente integrados nas comunidades que os rodeiam, sendo o caso particular das pessoas cegas.

1.1.1. Integração da comunidade cega na sociedade

No caso específico das pessoas cegas, apesar de um crescente aumento de investigação e procura de soluções para a integração destes indivíduos na vida ativa das comunidades, ainda persistam diversos desafios e problemáticas. A conceção e o desenvolvimento de soluções é um processo moroso e complexo pois existem diversos entraves, desde a falta de pessoal qualificado até aos custos elevados associados às implementações dos projetos. Apesar destas dificuldades, uma constante divulgação de ideias permite que os indivíduos e as diversas instituições tenham cada vez mais consciência desta realidade e das novas possibilidades que se abrem, de modo a que o Mundo seja cada vez mais inclusivo, integrando não só a comunidade cega, mas também todos os indivíduos com outros tipos de deficiência. Qualquer processo que permita a inclusão da comunidade cega é considerado um avanço, por não se limitar apenas a facilitar a vivência do indivíduo em certos momentos, perante as suas limitações, mas também por propiciar oportunidades de igualdade sem que venham ao de cima as suas diferenças. É neste contexto específico que a tecnologia se tem tornado num instrumento fulcral, pois permite o acesso à informação e ao conhecimento. Mais concretamente, no caso dos indivíduos com deficiência

visual, já existem soluções que auxiliam no acesso à informação com base na síntese de voz, através do tato, por meio de gestos no ecrã, bem como pela reprodução de sons.

O trabalho desenvolvido no âmbito desta dissertação encontra-se intrinsecamente ligado à causa social do apoio à comunidade cega. No que concerne especificamente à seleção de roupa, os indivíduos cegos não conseguem escolher plenamente e de forma autónoma a roupa pretendida, nem determinar o seu estado de conservação, estando, por isso, dependentes de auxílio externo. A proposta apresentada nesta dissertação visa tornar possível a realização desta tarefa básica do quotidiano, contribuindo, assim, para a melhoria não só da sua autoestima, mas também da sua qualidade de vida.

1.1.2. A tecnologia e a sua aceitação na vida quotidiana

Nos dias de hoje, as soluções usadas nas chamadas “casas inteligentes” são já um tema de curiosidade e discussão entre os consumidores mais atentos. Estas soluções incluem o conceito de guarda-roupa inteligente, que visa tornar funcional e organizado o início do dia a dia do indivíduo, reduzindo-se, assim, o tempo gasto na escolha e seleção de peças de vestuário. Identificaram-se alguns estudos que avaliam o nível de aceitação pelas pessoas do conceito de guarda-roupa inteligente. Em [1], os consumidores aceitam de bom grado esta conceção, assente em dois fatores fundamentais: a facilidade de uso e a utilidade no quotidiano. Os resultados do estudo evidenciam precisamente esta ideia, tendo cerca de 84% dos participantes indicado a facilidade de uso e a utilidade como elementos fulcrais para a aceitação do modelo tecnológico de guarda-roupa inteligente. Torna-se, por isso, evidente que uma solução de um guarda-roupa que englobe toda a tecnologia que permita facilitar a vida aos utilizadores é bastante aceite pela comunidade. O guarda-roupa inteligente é um conceito no que permite ao utilizador definir um certo estilo, de forma a que as peças combinem entre si.

É com base numa crescente procura destas novidades tecnológicas que surgem cada vez mais soluções que permitem auxiliar os utilizadores numa das principais tarefas do seu quotidiano.

1.2. Motivação e Objetivos

Tendo por base a temática da integração da comunidade cega na sociedade, o trabalho realizado no âmbito desta dissertação teve como objetivo auxiliar esta comunidade tão necessitada numa tarefa bastante importante do quotidiano, mediante o projeto e o desenvolvimento de um protótipo de um guarda-roupa com a capacidade de auxiliar pessoas cegas na identificação e gestão das suas peças de vestuário.

1.3. Organização da Dissertação

Esta dissertação encontra-se dividida em cinco capítulos. No capítulo 2 é apresentado o Estado da Arte, onde se referem propostas existentes no âmbito da temática de soluções tecnológicas para cegos. O capítulo 3 expõe uma visão geral do projeto, com a apresentação das ferramentas a usar na sua implementação, bem como a referência aos requisitos e especificações do sistema a desenvolver. Esse capítulo aborda a arquitetura de *hardware* e os componentes utilizados. Também descreve detalhadamente a estrutura do projeto a nível de *software*. No capítulo 5 apresentam-se os resultados obtidos a nível de *hardware* e de *software*. Por último, no capítulo 6 destacam-se as principais conclusões e apontam-se algumas perspetivas de trabalho futuro.

2. Estado da Arte

Neste capítulo, é apresentada uma visão geral do estado atual da tecnologia relativamente ao auxílio de cegos, com ênfase nas soluções a nível de vestuário. De seguida, são apresentadas diversas soluções existentes no mercado que não apresentam como público-alvo pessoas cegas, mostrando um pouco do que tem vindo a ser disponibilizado ao público em geral, no sentido de auxiliar e facilitar na seleção de peças de vestuário.

2.1. Soluções tecnológicas para cegos

No que se refere, em específico, às soluções que auxiliam os cegos no seu quotidiano, existem diversas propostas que permitem dar uma ideia do que tem vindo a ser realizado, em diversos âmbitos, recorrendo a tecnologia. Nguyen *et al.* [2] apresentaram uma bengala que auxilia as pessoas cegas na difícil tarefa de detetar obstáculos durante a locomoção. Por outro lado, outra das grandes dificuldades das pessoas cegas prende-se com a interpretação dos diversos tipos de informação que surge no quotidiano. Com base nesta necessidade, foi proposto em [3], por Apu *et al.*, um sistema de texto e voz de tradução de código Braille. Numa outra vertente, neste caso, no auxílio da navegação e orientação, foi sugerido, em [4], por Badawi *et al.*, uma solução que ajuda os cegos na orientação no interior de edifícios.

2.1.1. Soluções para cegos a nível de vestuário

Neste tópico, serão referenciadas diversas propostas, que se enquadram no auxílio na escolha e seleção de vestuário.

A deteção de cor de uma peça de roupa de forma direta, a forma e a textura das diversas peças de vestuário, são tarefas impossíveis ou bastante complexas para um cego. Foi com base nessa questão que Tateno *et al.* apresentaram em [5] uma solução que mitiga este problema. Assim, foi proposto um método que permite a conversão de informação visual adquirida a partir da roupa em exposição verbal (usando Redes Neurais Profundas). Deste modo, o sistema apresenta a capacidade de captar fotos de combinações, sendo, posteriormente, submetidas a um pré-

processamento responsável pela remoção da face da pessoa e extração das características principais. Após esse tratamento das imagens, é aplicado um modelo matemático que permite a implementação do sistema como um todo, de forma a que este reconheça e caracterize automaticamente a peça de roupa apresentada.

Para tornar mais simples a tarefa das compras de roupa *online*, foi proposto em [6] um sistema que simplifica as compras *online* para cegos. Esta ideia começou a ganhar forma após diversas entrevistas a alunos cegos do Colégio de Educação Especial da Universidade de Pequim. Nessas interações, os alunos expuseram os diversos desafios e dificuldades que encontravam aquando da compra em lojas de roupa *online*, sendo que os principais problemas relatados se referem à demasiada publicidade existente, a uma interface pouco intuitiva e à existência de poucas subdivisões das diversas peças de vestuário. Nesse sentido, com base neste estudo, os autores propuseram uma aplicação *web* que mitigasse estas questões e auxiliasse os cegos de forma a tomarem a decisão correta aquando da compra de qualquer tipo de peça. Assim, apresentaram diversas alterações, tendo em vista a melhoria da interface com o utilizador. Uma delas prendeu-se com a escolha da característica a seleccionar, passando a existir seis subdivisões (estilo, tamanho, material, cor, preço e tipo), tornando o filtro de procura das peças de roupa cada vez mais eficaz. Após uma primeira filtragem, já quando o utilizador se encontra na página das peças, a informação essencial relativa às características de cada peça (nome, tipo, foto, preço e volume de vendas) passou a encontrar-se disponibilizada imediatamente abaixo da foto respetiva, tornando a leitura e procura bastante mais simples.

Outro sistema que facilita a vida dos cegos na escolha e seleção da roupa é apresentado em [7]. Neste caso, trata-se de um sistema automático de reconhecimento de padrões, baseado num *kit* de montagem com uma câmara. Mais concretamente, este protótipo apresenta a capacidade de reconhecer quatro tipos de categorias (xadrezado, listado, sem padrão e irregular) e onze cores diferentes. O sistema proposto integra uma câmara, um microfone, um computador e uns auscultadores para uma descrição áudio dos padrões e cores. No caso da câmara, encontra-se embutida nuns óculos especiais a fim de capturar as fotos necessárias para a identificação das peças. No que se refere à interface com o utilizador, pode ser realizada através de fala a partir do microfone, sendo que as descrições de cada peça são comunicadas ao utilizador através dos auscultadores. O sistema apresentado em [8] baseia-se numa solução que auxilia na combinação de peças de roupa. O sistema integra a tecnologia *NFC* com o *smartphone* permitindo a escolha das roupas desejadas. Nesse sentido, a proposta apresentada em [9] vai de encontro a esta

anterior, pois, além de usar a tecnologia *NFC*, integra-a com a tecnologia QR (*Quick Response*) com o principal objetivo de desenvolver um sistema de combinação de peças de roupa com audiodescrição. Em [10] é apresentada uma solução que compara cores e padrões de duas peças de roupa e indica se existe correspondência. Este sistema engloba uma câmara, responsável pela captura das fotos, um computador onde é aplicado o algoritmo de visão em cada foto e um microfone que comunica os resultados ao utilizador. Para a identificação de cores são usados os modelos RGB e HSI.

Yang *et al.* apresentam em [11] um protótipo que permite a identificação de 4 tipos de padrões e 11 cores diferentes em peças de roupa. O sistema integra uma câmara, um microfone, um computador e uns auscultadores sem fios. Para a identificação das características são implementados diversos algoritmos: *Radon Signature*, *Statistical Descriptor* e *Scale Invariant Feature Transform*. Para a identificação das cores das peças de roupas é usado o modelo HSI.

Em [12] é proposto um sistema que reconhece 4 tipos de padrões (xadrez, listado, irregular e sem padrão) e 11 cores diferentes. Este sistema integra uma câmara que permite a captura das imagens, sendo usado o algoritmo de *Support Vector Machine* para a classificação das diferentes peças de roupa. Outro sistema apresentado em [13] apresenta objetivos semelhantes sendo usado o método de *Sequential Minimal Optimization* para a classificação de peças. A proposta apresentada em [14] assemelha-se bastante às duas anteriores, sendo usados os algoritmos de *Statistical Feature (STA)* e o *SIFT (Scale Invariant Feature Transform)*.

Medeiros *et al.* apresentam em [15] um protótipo com uma câmara endoscópica anexada no topo de um dedo que permite a captura de imagens à superfície das peças e respetivo reconhecimento de cores e padrões.

Por último, foi apresentado em [16] e [17] um sistema que permite ao utilizador ter um guarda-roupa virtual com as suas peças de vestuário, sendo possível a adição de peças de roupa com base na leitura de etiquetas com a tecnologia *NFC*. Esta aplicação móvel permite a gestão total das peças de vestuário pessoais, possibilitando também a deteção da cor predominante de cada uma das peças através da captura ou *upload* de uma foto. No que se refere ao sistema em si, este encontra-se dividido numa API, responsável pelo processamento dos pedidos por parte do utilizador, e uma aplicação de *front-end* que representa a interface visível ao utilizador. Este sistema baseia-se numa plataforma de *Arduino* com recurso a uma *shield NFC* que permite a leitura e escrita dos dados característicos de cada peça (tamanho, lavagem, estampagem, época,

tipo, cor e estilo), apresentando uma aplicação *web* que possibilita ao utilizador a gestão dos seus dados pessoais, das várias combinações e das várias peças de vestuário do guarda-roupa. O protótipo a apresentar nesta dissertação é uma continuação desta última solução, pelo que a mesma adquire uma especial relevância neste trabalho.

No seguimento da solução anterior, foram abordadas em [18] as linhas gerais da possível implementação de um sistema mecatrónico que integraria essa proposta. Este estudo apresenta uma importância fulcral na realização desta dissertação, pois permitiu apresentar uma ideia dos problemas existentes, bem como definir os objetivos e os requisitos principais.

Considerando que se pretende englobar no projeto o reconhecimento e identificação de cores de cada peça de roupa, é necessário identificar, ao nível do estado da arte, as diversas soluções que existem neste âmbito. Deste modo, relativamente a soluções que auxiliam na identificação de cores em diversas situações na rotina diária dos cegos, existem algumas implementações, desde aplicações móveis a pequenos dispositivos.

Nesse sentido, surge a *V7 Aipoly* [19], que é uma aplicação móvel que permite a identificação em tempo real de objetos, textos e cores. Este sistema comunica com o utilizador através de áudio. Noutra vertente, surgem pequenos dispositivos portáteis como o *ColorTest2000* e o *Colorino*. Relativamente ao *ColorTest2000* [20], este permite a identificação de mais de 1700 cores diferentes de itens do quotidiano. No caso do *Colorino* [21], permite a distinção de cerca de 150 tons de cores podendo auxiliar os utilizadores na escolha da roupa bem como no seu procedimento de lavagem. Tal como referido anteriormente, é importante reconhecer a existência de diversas soluções que auxiliam na identificação de cores, pois nesta dissertação, além da projeção do protótipo físico, será desenvolvido um algoritmo de deteção de cor de cada peça de roupa.

2.2. Soluções existentes no mercado

Neste tópico, são apresentadas as diversas soluções existentes no mercado e disponíveis para os utilizadores, tendo como ênfase aplicações móveis de ajuda na escolha de combinações.

2.2.1. Armários inteligentes

Considerando o aumento da procura por soluções que se enquadram no conceito de armários inteligentes, são cada vez mais as implementações que surgem no mercado. No entanto, cada uma das soluções encontradas apresenta capacidades diferenciadoras, sendo que se encontram divididas em implementações físicas e em implementações virtuais.

2.2.1.1. Amazon Echo Look

Em 2017, foi desenvolvido pela *Amazon* o *Echo Look* que se baseia num *kit* de montagem com uma câmara que permite a captura de fotos de peças de vestuário de modo a catalogá-las, apresentando também conselhos de utilização da roupa, de acordo com diversos fatores, nomeadamente, a meteorologia e a perceção do gosto pessoal do utilizador. O funcionamento desta solução baseia-se na captura de fotos ou gravação de vídeo a corpo inteiro, desfocando totalmente o fundo com base num algoritmo de visão por computador. Neste sentido, o *kit* pode ser montado num armário ou numa estante e, usando comandos de voz, é possível capturar as fotos ou vídeos. Deste modo, recorrendo a inteligência artificial, identifica peças de roupa, recomenda possíveis visuais e permite a criação de um armário virtual com as peças de vestuário de cada utilizador, facilitando a sua gestão total. Além do mais, permite a partilha dos *outfits* com outras pessoas, bem como, o uso da funcionalidade de verificação de estilo que combina algoritmos de *Machine Learning* com conselhos de especialistas, de modo a apresentar sugestões de moda mais personalizadas. Nessa funcionalidade em específico, o utilizador submete duas fotos com *outfits* diferentes de modo a que sejam avaliados tanto pelo algoritmo como por especialistas externos, sendo que, ao longo do tempo, as decisões tomadas se tornam cada vez mais inteligentes. Tal como referido anteriormente, toda esta interação com o utilizador pode ser realizada através de um controlo de voz ou com recurso à aplicação disponível para *smartphone* [22].

2.2.1.2. Fashion API

Em 2019, foi concebida uma aplicação móvel conhecida como *Fashion API* [23] que permite a gestão do guarda-roupa com base na leitura de um código QR. Esta solução permite ao utilizador a criação de um guarda-roupa virtual que efetua a gestão total, ou seja, adição ou remoção das diversas peças de vestuário. A adição de peças baseia-se na leitura de um código QR presente na etiqueta de cada peça, a partir da câmara do *smartphone* onde se encontra instalada a aplicação *mobile*. Além da gestão do guarda-roupa, este conceito de armário inteligente oferece conselhos de utilização de roupa com base nas peças de vestuário existentes, sugere a compra de novas roupas de acordo com as necessidades do utilizador e ajuda no planeamento de *outfits* a usar. Esta tecnologia é oferecida ao utilizador a partir de uma aplicação *mobile* que apresenta uma interação intuitiva e rápida com a presença de *dashboards* e dados estatísticos que permitem ter conhecimento das peças mais usadas, bem como dos *outfits* que se encontram a ser sugeridos.

2.2.1.3. Smart Closet

A solução de *Smart Closet* [24] apresentada em 2018 baseia-se, tal como a solução anterior, numa aplicação móvel que gere o guarda-roupa com base na compra de peças de vestuário em diversas lojas. Deste modo, esta aplicação permite ao utilizador a gestão total das peças de roupa do seu armário, bem como a obtenção de estatísticas detalhadas, tais como a peças mais usadas. No entanto, esta solução não apresenta qualquer tipo de inteligência, ou seja, não recorre a nenhum algoritmo de *Machine Learning* ou de visão de computador de forma a identificar autonomamente peças de roupa ou aconselhamento de moda com base em peças do utilizador.

2.2.1.4. TailorTags

A implementação exposta pela *TailorTags* [25] apresenta uma solução baseada em etiquetas inteligentes de modo a identificar e catalogar peças de roupa. Neste caso, são usados sensores, em miniatura, que se encontram embebidos nas peças de vestuário pessoais de forma a que estas sejam detetadas e identificadas automaticamente sem qualquer tipo de interferência do utilizador.

Além do mais, esta implementação apresenta inteligência artificial que permite a capacidade de reconhecer o estilo pessoal do utilizador, sugerindo conselhos de utilização de roupa de acordo com a hora do dia e com as condições meteorológicas.

De modo a que o utilizador possa efetuar a escolha e explorar o guarda-roupa virtual, *TailorTags* apresenta uma aplicação *mobile* para *smartphone* bastante simples e intuitiva que permite a gestão total das peças de vestuário de forma remota.

2.2.1.5. Smart Wardrobe

Por último, a proposta de solução apresentada por Goh *et al.* em [26] baseia-se num sistema que integra etiquetas com a tecnologia RFID que permite identificação de forma única de peças de vestuário. Neste sentido, esta solução é mais vocacionada para pessoas daltónicas, tendo como principal objetivo a facilitação e redução do tempo gasto na escolha de roupas. O funcionamento parte da leitura de etiquetas RFID, pois permite a obtenção dos dados característicos de cada peça de roupa. Estas etiquetas encontram-se coladas junto das etiquetas da roupa, apresentando uma tecnologia RFID que permite a identificação e catalogação a partir de um leitor que apresente essa capacidade. Este sistema encontra-se dividido num módulo de *hardware* e num módulo de *software*. No que se refere à parte de *software*, o sistema é controlado por uma aplicação que permite a gestão total das peças, tendo o utilizador a possibilidade de adicionar, listar e remover peças que pretenda. Além do mais, o sistema apresenta a capacidade de sugerir peças de vestuário com base em alguns critérios: estilo, cor, material e disposição do utilizador, partindo de seis estados de espírito (calmo, feliz, zangado, triste, cansado e inquieto). Algo que se deve ter em consideração é que, ao contrário da solução anterior, esta não apresenta qualquer tipo de algoritmo de inteligência artificial, sendo que apenas gere o guarda-roupa e sugere combinações como base em dados fornecidos pelo utilizador. Além de tudo, o sistema também permite a criação de relatórios estatísticos que permitem ao utilizador ter uma noção das peças de roupa mais utilizadas bem como da quantidade de dinheiro despendida em cada categoria de roupa.

Como se pode verificar, a maior parte das soluções abordadas focam-se quase exclusivamente na resposta a necessidades de pessoas sem qualquer tipo de deficiência. Assim sendo, é com base nas lacunas das soluções apresentadas que se propõe este tema, ou seja, um sistema

mecatrónico que auxilie as pessoas cegas na escolha da sua roupa do dia a dia. A Tabela 1 apresenta uma comparação entre as diferentes soluções apresentadas.

Tabela 1 - Características das soluções no mercado

Solução	Descrição		
<i>Amazon Echo Look</i> [22]	Kit de montagem com uma câmara que capta fotos e grava	Interação realizada através de controlo de voz ou aplicação móvel	Cria um <i>lookbook</i> pessoal com as peças de cada utilizador
<i>Fashion API</i> [23]	Aplicação móvel que gere o guarda-roupa com base na leitura de códigos QR	Ajuda na compra de novas roupas de acordo com as existentes	Sugere conselhos de moda com base no gosto pessoal
<i>Smart Closet</i> [24]	Aplicação móvel que gere o guarda-roupa com base nos gostos pessoais	Adiciona peças de roupa a partir da captura de foto	Planeia peças a usar de acordo com eventos calendarizados
<i>Tailor Tags</i> [25]	Sistema que usa etiquetas inteligentes para detetar peças de vestuário	Sugere conselhos de moda com base no estilo de cada utilizador	Sugere combinações de acordo com as condições climáticas
<i>Smart Wardrobe</i> [26]	Sistema que usa etiquetas RFID para obter características de cada peça	Sugere combinações de acordo com o estilo de cada utilizador	Sugere peças de roupa com o estado de espírito de cada utilizador

2.3. Segmentação de imagem

A segmentação de imagem baseia-se na identificação e anotação de diversos objetos em imagens, para separação em classes. Assim, existem diversas aplicações que recorrem a este método, entre as quais sistemas de assistência à condução de veículos ou de deteção de doenças. No âmbito da segmentação, é possível subdividir entre segmentação semântica e de instância. Considerando que um dos focos deste trabalho é a identificação de peças de roupa para o auxílio de pessoas cegas, é necessário que, aquando da captura da foto da peça, sejam realizadas ações de pré-processamento - permite que a imagem chegue ao utilizador o mais limpa possível. Deste modo, é necessário realizar uma segmentação da imagem e posterior remoção de fundo de modo a que o algoritmo de deteção de cor tenha os melhores resultados, analisando apenas os *pixels* relativos à peça em questão. Este processo de segmentação apresenta uma grande influência no modo como os restantes algoritmos se irão comportar, tendo este que ser maximizado para apresentar

os melhores resultados possíveis, minimizando os possíveis erros associados. Assim, para a segmentação de imagem a escolha recaiu na rede *Mask R-CNN*, uma Rede Neuronal Convolutiva que permite a detecção de objetos e a geração das respectivas máscaras de segmentação.

Assim sendo, realizou-se um estudo no sentido de verificar o que tem sido implementado no que se refere à segmentação de imagens de peças de roupa. Em [27] apresenta-se um método que auxilia os analistas de moda e os consumidores na identificação do tipo de roupa, ao longo dos anos. Nesse sentido, é proposta uma abordagem que permite aumentar os resultados na segmentação usando um sistema que engloba a Multi-Attention *Mask R-CNN* através de mecanismos que possibilitam a extração de informação das peças de roupa. Esta implementação apresenta uma maior eficácia na segmentação, em especial, nos limites entre peças de roupa, quando em comparação com outros tipos de implementações da *Mask R-CNN*. Em [28], é apresentada outra solução que se foca na resolução de um problema relacionado com a identificação e detecção de peças de roupa em imagens. Com isto, é proposto um método que tem a capacidade de diferenciar diversos tipos de roupa e que possam ter padrões ou texturas semelhantes. Assim, a implementação baseia-se em dois módulos: um primeiro, que faz uso da Rede totalmente Convolutiva para uma segmentação espacial dos limites e, um segundo, que deteta características particulares para um melhor auxílio no reconhecimento de certos limites. Ao nível dos resultados, é evidenciada uma melhoria em relação ao existente, pois, de certa forma, com esta implementação, foi reduzida a confusão existente em fotos com peças de roupa visualmente semelhantes, mas de diferentes materiais. Já em [29], é proposto o DeepFashion2, tratando-se de um método que permite extrair o máximo de dados possíveis de fotos com peças de roupa. Esta proposta é apresentada com base nas falhas existentes no sistema DeepFashion, tendo como grande objetivo a segmentação da forma mais exata possível de todas as peças de roupa existentes em cada foto, bem como a sua identificação. Assim, foi produzido um *dataset* com cerca de 801 000 imagens contendo peças de roupa devidamente anotadas, com identificação do tipo e do estilo, segmentadas através de máscara e de caixas delimitadoras. No que se refere ao modelo, o sistema usado, o Match R-CNN, é uma estrutura baseada em *Mask R-CNN*. No que se refere aos resultados, estes apresentam uma ligeira melhoria ao nível do Precisão Média quando em comparação com o modelo da *Mask R-CNN* e o *Match R-CNN*. Em [30], é proposto outro modelo baseado em DeepLabV3+ que permite segmentar as peças de roupa que se encontram englobadas num fundo complexo, em que os limites das peças se possam confundir. Desta forma, foi redesenhada uma nova arquitetura da Rede Neuronal para que a performance da

segmentação apresentasse melhores resultados, revelando, por isso, uma melhor adaptabilidade aos limites de cada peça de roupa. Após o treino do modelo e o teste no *dataset*, os resultados mostram que, quando em comparação com o modelo do DeepLabv3+, os valores de 97.26 % de *accuracy*, de 93.23 de % de mIoU e de 90.56 % de AP são ligeiramente superiores ao obtidos anteriormente. Em [31] é abordado um método que permite reconhecer conjuntos de peças de roupa e identificar o ano a que se referem. É usada a Rede Neuronal Convolutacional de forma a segmentar o corpo humano e depois para classificar o tipo de peça, nas zonas segmentadas sem que haja influência do fundo. Para validar o modelo foi usado um *dataset* com 9339 imagens de 8 anos e os resultados apresentaram uma boa eficácia. Ao nível da segmentação, foram testados diferentes modelos, tendo o modelo SegNet apresentado um valor de IoU de cerca de 0.951, tendo sido o mais próximo do obtido usando o método apresentado, isto é, um valor de IoU de 0.951. Ao nível de classificação, o método proposto também apresenta os melhores resultados, tendo o valor de *Accuracy* sido 0.805 em contraponto com o valor mais próximo de 0.785 usando o modelo ResNet101. Em [32], é descrito um método que permite detetar de forma eficaz as silhuetas de camisolas usando a segmentação instantânea, com recurso à *Mask R-CNN*. Assim, foram recolhidas cerca de 9000 imagens, tendo sido devidamente segmentadas e classificadas em diversos atributos. Os resultados experimentais demonstraram que o modelo apresentou bons resultados na deteção de camisolas de manga curta e na gola das camisolas, tendo sido obtido um AP de 0.95 e 0.94, respetivamente. Este método apresentado permite mitigar o problema da segmentação da imagem das peças de roupa quando captadas em diferentes posições.

3. Fundamentos teóricos

Neste capítulo, apresentam-se as metodologias usadas na realização do trabalho. Visto serem usadas Redes Neurais para segmentação de imagem, descrevem-se os métodos de *Machine Learning* com foco nas Redes Neurais profundas e como estas se comportam perante diferentes imagens.

3.1. Machine Learning

O *Machine Learning* é um conceito que está cada vez mais na ordem do dia nos últimos anos. Baseia-se na implementação de modelos, em aplicações de *software*, que apresentem a capacidade de prever resultados relativamente aos diversos dados em estudo. Estes algoritmos são modelados através de treino, com base no conhecimento obtido numa panóplia de características do conjunto de dados fornecido. Este método iterativo permite ao modelo a aprendizagem e previsão de forma bastante precisa sobre diversos dados futuros. Este conceito encontra-se cada vez mais em voga no nosso quotidiano, sendo frequentemente usado em anúncios em *websites*, deteção de fraudes, reconhecimento de voz e resultados personalizados em pesquisas. Deste modo, dentro desta temática, é possível encontrar três subníveis que permitem categorizar de diferentes formas: o *supervised learning*, *unsupervised learning* e *reinforcement learning*. No *supervised learning*, o modelo apresenta a capacidade de aprender através da receção do conjunto de dados devidamente anotado, ou seja, com dados necessários para a aprendizagem. Já no *unsupervised learning*, o modelo aprende tendo acesso ao conjunto de dados sem qualquer tipo de anotações, isto é, o sistema tem de ser capaz de descobrir as semelhanças e diferenças nos dados sujeitos a treino. No caso do *reinforcement learning*, o método baseia-se numa unidade central capaz de perceber o ambiente em que se encontra envolvido, tomando diversas decisões no sentido de minimizar o erro através de tentativa e erro.

3.2. Transfer Learning

O *Transfer Learning* é um método de *Machine Learning* que permite o uso de um modelo pré-treinado como ponto inicial para o treino de um modelo numa nova tarefa, permitindo uma otimização e um rápido progresso na nova modelização (Figura 1). A aprendizagem tradicional é isolada, sendo usada para tarefas específicas com um conjunto de dados isolados. Desta forma, não existe transferência de conhecimento entre modelos. É neste sentido que surge o *Transfer Learning*, pois passa a existir transmissão de conhecimento através dos pesos e de recursos específicos obtidos a partir do treino no conjunto de dados anterior. Assim, baseia-se no uso de modelo treinado anteriormente num elevado conjunto de dados, sendo possível a obtenção de melhores resultados com este método, em vez do treino de um modelo a partir do zero, com um pequeno conjunto de dados. Os dados deste modelo pré-treinado são usados de modo a obter elementos do novo conjunto, evitando, assim, que o treino do modelo se inicie totalmente do zero. No treino do novo modelo, é adicionado um novo classificador, de modo a redirecionar o treino para os elementos do novo conjunto de dados. Este método apresenta um elevado número de vantagens, entre as quais, o tempo reduzido de treino e um melhor rendimento da rede num curto conjunto de dados.

A nível do *Transfer Learning*, este pode ser subdividido em três categorias, com base no tipo de tarefa encontrada ou a partir do conjunto de dados disponível: indutivo, não-supervisionado e transdutivo. No caso do *Transfer Learning* indutivo, este baseia-se na capacidade de aprendizagem através de tarefas diferentes, mas em domínios semelhantes. Neste método, os dados finais encontram-se etiquetados, podendo os dados usados no modelo pré-treinado serem etiquetados, no caso do *Multi-task Learning*, ou não, no caso do *Self-taught Learning*. No *Transfer Learning* não supervisionado, os domínios são semelhantes, sendo as tarefas diferentes, apresentando como grande diferença o facto de que tanto o conjunto de dados original como o final não serem etiquetados. No método de *Transfer Learning* transdutivo, as tarefas apresentam semelhanças, sendo os domínios diferentes. Relativamente aos dados, o conjunto original encontra-se etiquetado ao contrário do final.

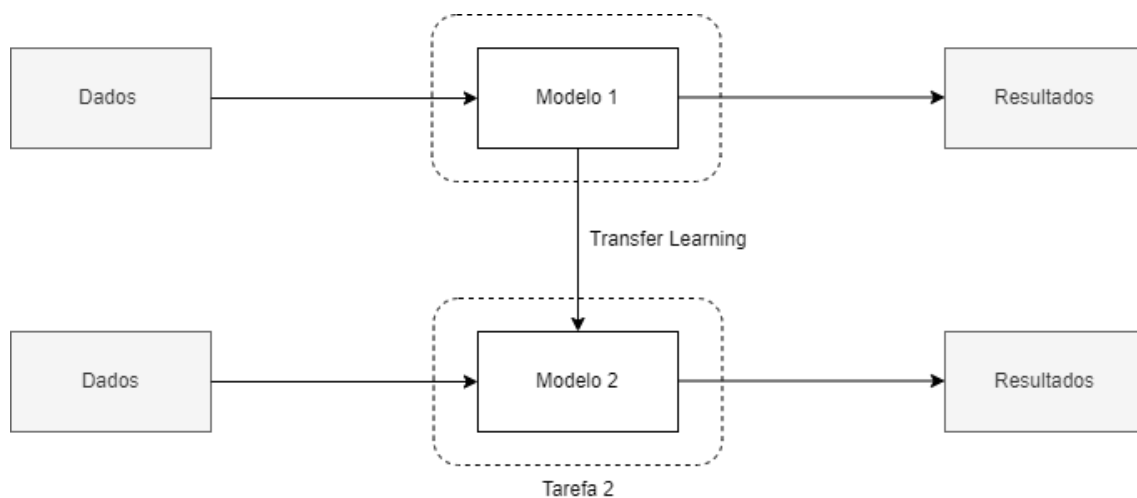


Figura 1 - Ciclo do *Transfer Learning*

3.3. Redes Neurais

As Redes Neurais são modelos computacionais subdivididos em diversas unidades de processamento que apresentam uma arquitetura semelhante às conexões entre os neurónios do cérebro humano. Assim, é possível afirmar que os neurónios surgem como base deste processo, podendo ser divididos em camadas. Desta forma, as camadas encontram-se interligadas, sendo a aprendizagem obtida através da interação entre neurónios. Estas Redes Neurais são projetadas de modo a classificar objetos, a partir da análise das suas características. Através de computação, são realizados diversos cálculos de modo a que as redes possam reconhecer os padrões desejados. As redes têm a capacidade de aprender através de treinos, ajustando o peso para cada interação entre dois neurónios. Esta capacidade de aprendizagem permite a melhoria do desempenho através do processo iterativo de treino. O processo de treino baseia-se na geração de valores de saída que, quando comparados com os valores desejados, determinam o valor do erro. Este valor do erro é usado para novos cálculos, sendo ajustados os valores dos diversos pesos, de forma a minimizar o erro entre resultado obtido e o resultado previsto. Após o treino do modelo, é necessária a sua validação, de forma a verificar que os resultados estão de acordo com o esperado.

3.3.1. Redes Neurais Convolucionais

Este tipo de Redes Neurais (Figura 2) surgiram nos anos 80, sendo atualmente usadas em diversas aplicações de reconhecimento de imagem, apresentando uma importância fulcral em áreas relacionadas com o processamento digital de imagem, na visão computacional e na segmentação de objetos. Deste modo, existem diversos métodos de implementação deste tipo de Redes, sendo cada arquitetura enquadrada em situações específicas de acordo com o problema enfrentado.

As Redes Neurais Convolucionais são um tipo de rede que apresenta um diverso número de camadas, sendo cada camada responsável pela extração dos mais variados dados do conjunto sujeito a teste. Estas Redes revelam uma grande capacidade de simular a “inteligência” do cérebro humano, através do reconhecimento e identificação de diversos elementos numa imagem.

A arquitetura destas Redes apresenta dois grandes módulos: a convolução e a classificação. No caso da convolução, a responsabilidade passa pela extração de certos elementos de imagens, sendo o módulo seguinte responsável pela classificação do conjunto de elementos identificados na etapa anterior.

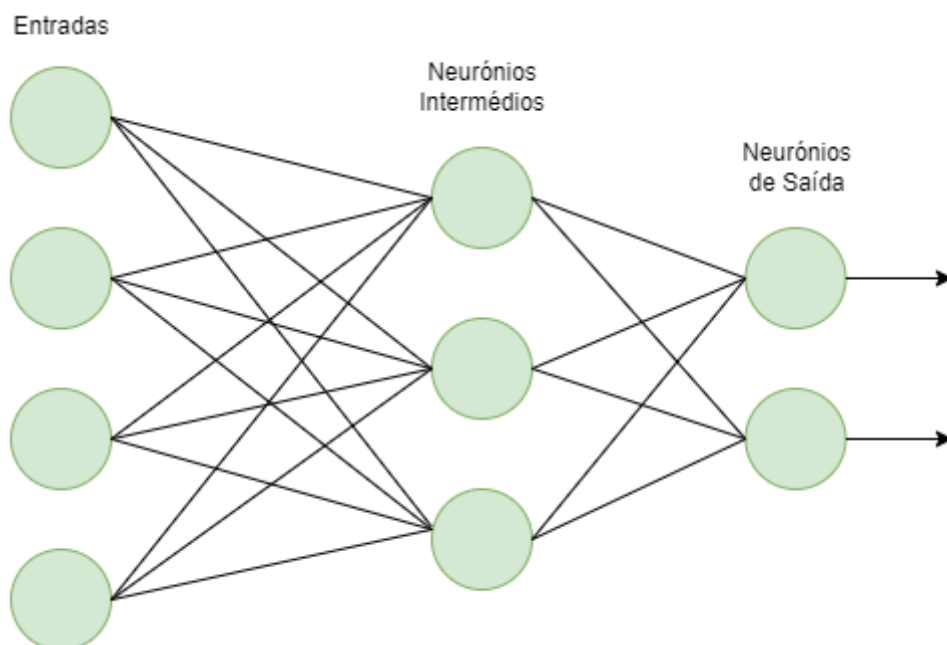


Figura 2 - Modelo de uma Rede Neuronal

Como referido anteriormente, é possível identificar diferentes camadas componentes das Redes Neurais Convolucionais: camada convolucional, camada de agregação e camada totalmente conectada. A camada convolucional é responsável pela extração dos elementos das imagens convertendo-os em dados, sendo, para isso, usados filtros para a abstração e a subdivisão da imagem sujeita a teste. Na camada de agregação, os dados seccionados na camada anterior são recebidos e simplificados em apenas um valor, de forma a que haja uma redução de recursos. Por último, a camada totalmente conectada é responsável pela classificação das informações obtidas nas camadas anteriores.

Desta forma, ao combinar estas três camadas, a Rede Neuronal passa a registar a capacidade de identificar e a reconhecer a região de interesse em cada imagem. Estas redes apresentam uma redução no pré-processamento relativamente a outros algoritmos de classificação, tendo a capacidade de otimizar os filtros através de aprendizagem. No que se refere a cada neurónio, é calculado um determinado valor, de acordo com as entradas para uma função específica. Cada função apresenta valores de entrada de acordo com o valor do peso dado. Este valor é alterado tendo em conta um vetor de pesos, sendo ajustado através da constante aprendizagem da Rede. Os vetores representam uma característica que se pretende evidenciar em detrimento de outras.

3.4. Faster R-CNN

A *Faster R-CNN* é uma Rede Neuronal Convolucional usada na deteção de objetos. A arquitetura desta Rede encontra-se dividida em três etapas: *Region Proposal Network (RPN)*, *Region of Interest Pooling (ROI)* e classificação. O primeiro passo do algoritmo corresponde à obtenção da Região de Interesse, que, na respetiva etapa, recebe como parâmetro de entrada uma imagem. Aqui, é realizada uma análise de forma a encontrar objetos. Assim que é obtida a área onde os objetos se encontram, é atribuída a esta uma denominação de classe *foreground*. Pelo contrário, a área que não apresenta qualquer tipo de objeto é denominada de classe *background*. Nesta etapa de deteção da Região de Interesse, primeiramente, são usadas *anchor boxes* que são caixas delimitadoras de diferentes tamanhos desenhadas na imagem de entrada, de forma a que seja possível a deteção de objetos de diversas dimensões. Após o desenho das *anchor boxes* na imagem sujeita a análise, é necessário o cálculo da *Intersect Over Union*, que corresponde à sobreposição de duas caixas delimitadoras. Desta forma, para cada *anchor box* será obtido o valor

de *IoU*. Assim, caso esse valor seja superior a 0.5, o objeto será classificado com a classe *foreground*. Caso seja inferior a 0.5, é classificado com a classe *background*. As *anchor boxes* que são classificadas com a classe *foreground* apresentam-se como parâmetro de entrada para a próxima etapa da arquitetura, a *ROI pooling*. Nesta etapa, visto que os parâmetros de entrada são as *anchor boxes* de diferentes tamanhos, o objetivo prende-se com a ajuste destas para o mesmo tamanho. Chegando à última etapa do modelo, esta encontra-se dividida em dois módulos: um classificador e uma regressão. A tarefa do classificador é a identificação do objeto obtido nas camadas anteriores. Já a regressão tem como tarefa o desenho das caixas delimitadoras nos objetos classificados previamente.

3.4.1. Region Proposal Network

A *Region Proposal Network* é uma etapa da Rede integrante da arquitetura da *Faster R-CNN*. Nesta etapa, de forma geral, as imagens sujeitas a teste são analisadas de forma a que sejam encontradas áreas onde podem vir a ser encontrados objetos. Deste modo, a imagem em estudo é colocada à entrada, sendo submetida a uma Rede Convolutiva pré-treinada. Desta forma, são captados os elementos da imagem para, através do desenho de *anchor boxes*, se obterem as áreas em que se encontram os objetos de interesse (Figura 3). Com isto, as diversas áreas obtidas, com base no valor de *IoU*, são classificadas. Aqui, existem duas etapas com funções distintas: uma que permite a classificação dos objetos encontrados e outra responsável pelo desenho das caixas delimitadoras. Na classificação, é apenas indicado se o objeto se encontra ou não presente numa área em particular. Tendo em conta os resultados que se obtêm na classificação, esta apresenta a *designação* de classificação binária. Na camada de regressão, é desenhada a caixa delimitadora à volta do objeto identificado pela camada classificadora. A saída desta camada será parâmetro de entrada para a etapa de *ROI (Region of Interest)*.

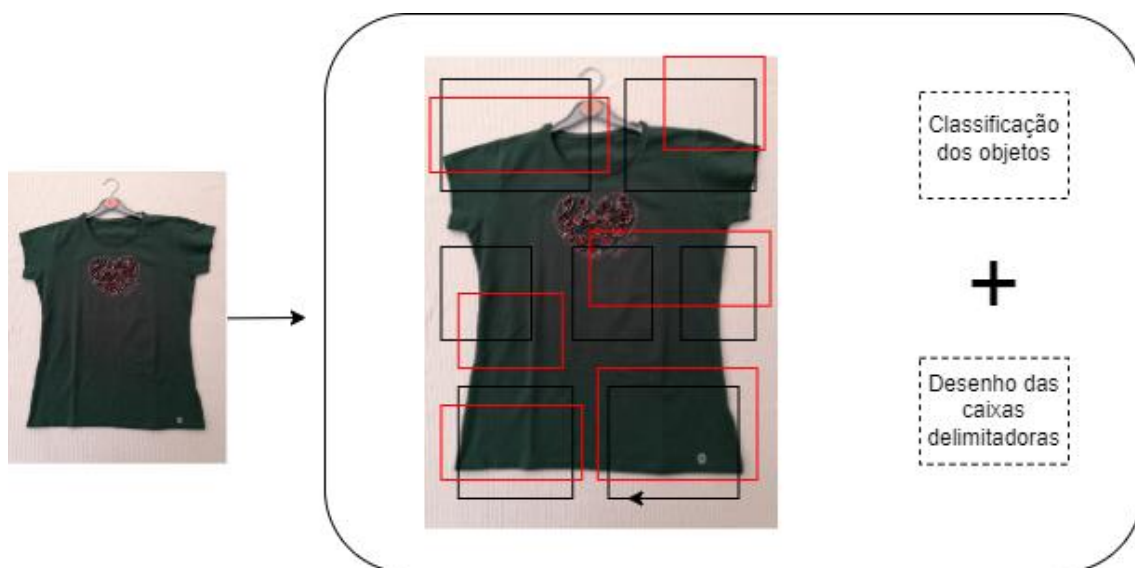


Figura 3 - Camada de *Region Proposal*

3.4.1.1. Region of Interest

Os dados obtidos na *Region Proposal Network* representam as áreas onde os objetos são mais suscetíveis de serem encontrados. No entanto, visto que as *anchor boxes* usadas para esta detecção apresentam dimensões diferentes, é necessário ajustar todas as áreas para uma dimensão igual. É aqui que surge a camada de *Region of Interest pooling*, pois permite fixar um igual tamanho para todas as áreas analisadas. À saída, encontram-se duas camadas totalmente conectadas com ligação a um classificador e a uma regressão. O classificador é responsável pela identificação da classe a que pertence o objeto, enquanto que a regressão permite o desenho da caixa delimitadora em redor do objeto encontrado, um processo semelhante ao da *Region Proposal Network*.

3.4.2. Mask R-CNN

A *Mask R-CNN* trata-se de uma *Region Based Convolutional Neural Network* que é um tipo de modelo de *Machine Learning* usado em aplicações de visão por computador, em específico na detecção de objetos (Figura 4). Basicamente, trata-se de uma extensão da *Faster R-CNN*, que é um algoritmo de detecção de objetos. Assim, a *Mask R-CNN* engloba tanto a detecção de objetos a partir do desenho de caixas delimitadoras em volta do objeto detetado, como a segmentação de

imagem através da segregação dos *pixéis* que pertencem a um objeto em particular. Ao nível da arquitetura de camadas usada nesta rede, assim que uma imagem é colocada na entrada, esta é submetida a uma rede neuronal convolucional pré-treinada, de forma que exista uma extração de recursos da imagem. Estes recursos estarão divididos em mapas e estarão como entrada na *RPN (Region Proposal Network)*. Na *RPN* encontram-se duas camadas convolucionais, sendo que uma permite a classificação binária, isto é, indica se o mapa de recursos de entrada apresenta ou não objetos, enquanto que a outra camada permite o desenho das caixas delimitadoras caso existam objetos. De seguida, no bloco de *Region of Interest*, existem duas entradas, que são as *ROIs* (a área em que o objeto foi capturado) obtidas no bloco anterior e o mapa de recursos da rede pré-treinada. Depois, é realizado um ajuste em cada recurso, de forma a que apresentem uma dimensão fixa. Por último, existem três classificadores: o *softMax classifier*, que indica que objeto se encontra na imagem, o *regressor classifier*, responsável pelo desenho da caixa delimitadora no objeto, e o *mask classifier*, que aplica uma máscara ao objeto. A partir do desenho da máscara de segmentação no objeto pretendido, é possível a classificação dos objetos de acordo com as classes existentes.

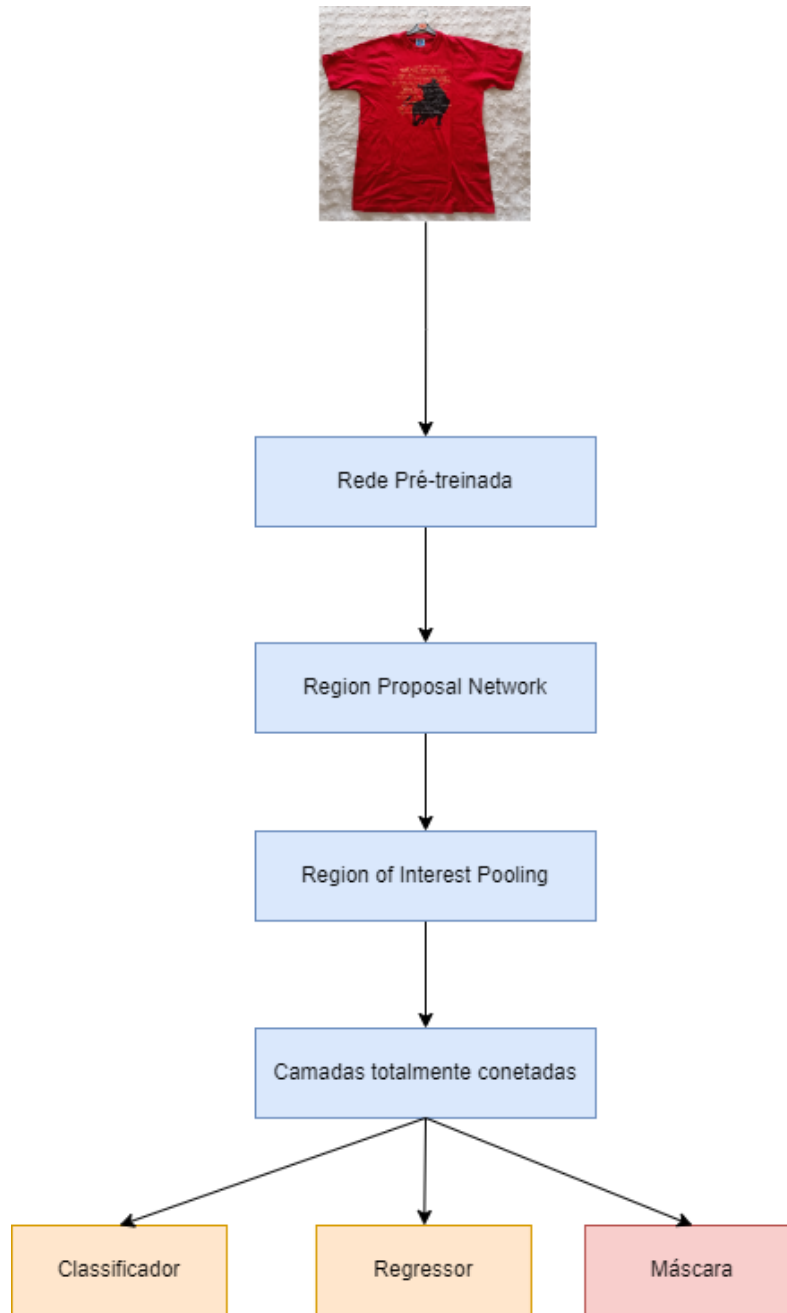


Figura 4 - Arquitetura Mask R-CNN

3.4.2.1. Detectron2

Para a segmentação das imagens, recorreu-se à biblioteca *Detectron2*. Esta plataforma desenvolvida pelo Departamento de Investigação de Inteligência Artificial do *Facebook*, permite a resolução de problemas no âmbito da deteção e segmentação de objetos. Esta plataforma encontra-se implementada em *PyTorch*, tendo incluídas diversas implementações do Estado da

Arte da detecção de objetos, entre as quais a *Faster R-CNN*, *Mask R-CNN* e *DensePose*. Esta implementação inclui diversos modelos pré-treinados, usados para as mais variadas tarefas, fornecendo parâmetros padrão diversos modelos, entre o quais, da *Mask R-CNN*, utilizado na detecção e segmentação das peças de roupa. Tal como abordado anteriormente, é usado o conceito de *Transfer Learning*, permitindo o uso de um modelo pré-treinado em várias tarefas para o treino de um modelo numa nova tarefa, aumentando, assim, a performance da Rede. Neste sentido, os pesos do modelo pré-treinado são usados no treino da nova rede com um conjunto de dados novo.

3.5. Métricas avaliadoras do Modelo

De seguida, encontram-se apresentadas as principais métricas que permitam avaliar o modelo da Rede Neuronal

3.5.1. AP (Precisão Média)

O Precisão Média é usado para a avaliação de modelos de segmentação de imagem, apresentando valores entre 0 e 1, sendo calculada através de diversas métricas, tais como, a *IoU*, *confusion matrix* (TP, FP, TN, FN), *precision* e *recall*. Desta forma, o cálculo da Precisão Média pode ser apresentado seguindo uma linha condutora, tal como apresentada abaixo:

- Cálculo dos TP (*True Positives*), FP (*False Positives*), TN (*True Negatives*), FN (*False Negatives*);
- Cálculo da *precision* e *recall*;
- Cálculo da área a partir da curva de *precision-recall*;
- Medição da Precisão Média.

Relativamente à *confusion matrix*, podem ser obtidos os TP, FP, TN e FN que correspondem às previsões por parte do modelo (Figura 5). Existe um *True Positive* quando o modelo previu a existência de uma caixa delimitadora e essa existência se confirmou. No caso do *False Positive*, corresponde à previsão de uma caixa numa certa posição e esta não se confirmou. Relativamente ao *True Negative*, indica que o modelo não previu a caixa delimitadora numa posição e esta também não existia. Por último, no que se refere à *False Negative*, o modelo não previu uma caixa em certa posição e esta, de facto, existia.

	Positivo	Negativo
Positivo	<p>True Positive Predição Acertada</p>	<p>False Positive Predição Errada</p>
Negativo	<p>True Negative Sem predição Acertada</p>	<p>False Negative Sem predição Errada</p>

Figura 5 - Sumário das previsões do modelo

De seguida, a partir destas últimas variáveis, são calculadas a *precision* e a *recall*. No caso da *recall*, esta indica o quanto o modelo treinado é eficaz nas suas previsões. Assim, pode ser obtida através do cálculo entre o rácio dos *True Positives* e a soma de todas as previsões, ou seja, os *True Positives* e os *False Positives*. No caso da *precision*, corresponde à divisão entre o *True Positive* e a soma do *True Positive* e *False Negative*. Com o cálculo da *recall* e da *precision*, é possível obter a curva da *precision-recall* que corresponde aos valores de *precision/recall* para diferentes

valores de *threshold*, sendo que quanto maior a área sob a curva, maior é o valor da *recall* e da *precision*.

$$Precision = \frac{Predições\ Corretas}{Predições\ Totais} = \frac{TP}{TP + FP}$$

$$Recall = \frac{Predições\ Corretas}{Máscaras\ Totais} = \frac{TP}{TP + FN}$$

A AP corresponde à área da curva de *precision-recall*, tendo como objetivo converter num único valor equivalente à média das *precisions* em cada *threshold*. Para o cálculo da mAP, é efetuado, em primeiro lugar, o cálculo da AP para cada classe e, de seguida, realizada a média dos AP de cada classe.

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k$$

3.5.2. IoU (Intersect Over Union)

Na validação de um qualquer modelo de segmentação de imagem, é necessário verificar se o modelo treinado apresenta resultados dentro do esperado. Deste forma, com as caixas delimitadoras geradas nos objetos sujeitos a análise, é realizada uma comparação com as caixas delimitadoras previstas através da sobreposição das mesmas. Mais concretamente, a IoU é obtida através do cálculo da divisão entre a área de interseção das duas caixas delimitadoras pela área de união (Figura 6). Desta forma, é desejável que se obtenha um valor de *IoU* elevado, indicando, com isto, que os resultados se encontram de acordo com o previsto.

$$IoU = \frac{\text{Área da sobreposição}}{\text{Área da união}} =$$

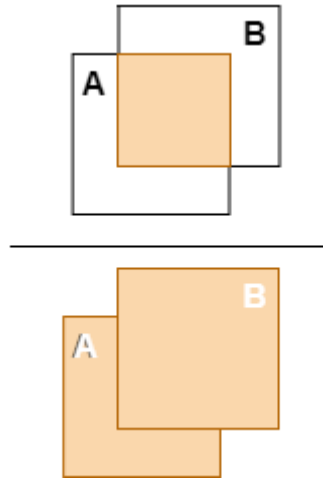


Figura 6 - Cálculo do *IoU*

4. Sistema desenvolvido

Neste capítulo, aborda-se a implementação física e os algoritmos implementados para a obtenção de dados essenciais das peças de roupa. É também apresentado o sistema desenvolvido quer a nível de *hardware*, quer de *software*.

4.1. Visão Geral do Sistema

Tal como referido anteriormente, o principal objetivo do sistema mecatrónico passa pela gestão automatizada de peças de roupa para pessoas cegas. O sistema dividiu-se em duas grandes etapas: o protótipo físico e o *software* de controlo do *hardware* (Figura 7). Encontra-se integrado um leitor NFC que será responsável pela obtenção do código UID de cada peça pedido pelo utilizador. Para facilitar a movimentação das peças de roupa no interior do armário de teste, encontra-se um motor de passo aparafusado ao teto, possibilitando a realização de um percurso circular. Desta forma, as peças realizam o caminho em fila, de modo a que, quando solicitado pelo utilizador, sejam capturadas as respetivas fotos. Em cada cabide, existe um motor responsável pela rotação de cada peça aquando da captura da respetiva foto por parte da câmara englobada no protótipo.

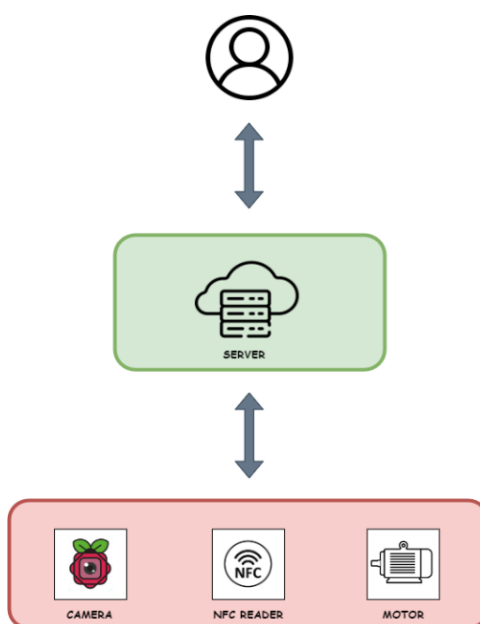


Figura 7 - Visão Geral do Sistema

Deste modo, é capturada uma foto da frente e da traseira da peça da roupa para posterior validação. Todo este sistema é baseado numa placa de desenvolvimento que será responsável pelo processamento dos dados recebidos e, quando pedidos, enviados para o utilizador. Tal como pode ser observado na Figura 7, existe uma interface entre o utilizador e o sistema mecatrónico desenvolvido. Esta integração encontra-se desenvolvida a partir da implementação de um servidor alocado na *Raspberry Pi*, que permite a gestão dos pedidos por parte dos utilizadores. Ao nível do *hardware*, existem três componentes principais: o leitor NFC, o sistema de visão e os motores responsáveis pelo movimento das peças. Assim, de acordo com os pedidos do utilizador, a placa de desenvolvimento aciona os automatismos respetivos, de forma que a foto da peça de roupa seja capturada e retornados ao utilizador os dados necessários, ou seja, tipo de peça e cores predominantes. No que se refere aos algoritmos, foram implementados dois algoritmos que permitem a obtenção de características importantes das peças de roupa, tendo sido utilizado um algoritmo de *Machine Learning* para a segmentação e classificação das peças de roupa e um algoritmo de visão por computador para a obtenção das cores predominantes de cada peça.

4.2. Modelo físico

O objetivo do projeto prende-se com o desenvolvimento de um sistema mecatrónico capaz de identificar diversos tipos de peças de roupa. Para este efeito, foi utilizado um pequeno armário do IKEA [33] com as seguintes dimensões: (50x30x80) cm (Figura 8). Este armário apresenta dimensões reduzidas, sendo, neste sentido, apenas testado com peças de roupa de pequeno tamanho. É constituído por quatro painéis laterais, um superior e outro inferior. Todos estes painéis são placas de madeira, sendo todos baseados num aglomerado de madeira, à exceção do painel lateral traseiro que é produzido a partir de uma fibra de madeira (MDF). Todas estas placas são unidas através de parafusos, permitindo que a estrutura do armário não oscile, servindo de base a todo o projeto.



Figura 8 - Armário usado para teste [33]

No interior do armário, encontra-se toda a eletrónica responsável pelo funcionamento do sistema (Figura 9). Numa das paredes laterais, encontra-se aparafusada a placa de desenvolvimento responsável pela ligação com os sensores e atuadores. Mais acima, encontra-se anexado o motor que permite a rotação da câmara em 180° , para a captura das fotos. Este motor tem acoplado ao seu eixo uma peça em plástico que permitiu o encaixe da câmara, tornando possível a captura de duas fotos com um espaçamento suficientemente grande aquando de uma rotação de 180° . Nas zonas laterais da mesma parede, encontra-se anexada a tira LED responsável pela iluminação do interior do armário. Por último, no teto do armário, encontra-se aparafusado o motor de passo responsável pelo movimento circular das peças de roupa. Ao eixo deste motor, encontra-se acoplada uma pequena plataforma circular responsável pela fixação dos motores anexados a cada cabide.

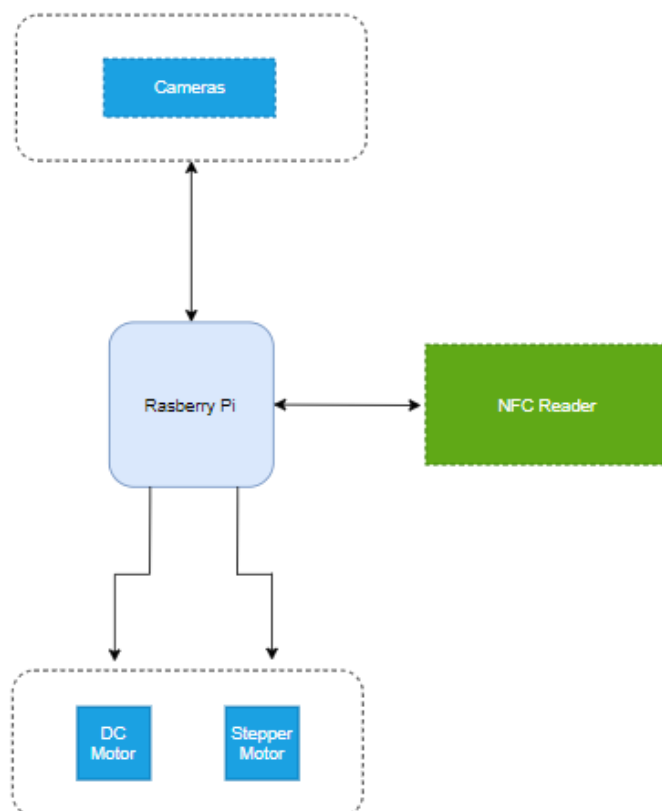


Figura 9 - Arquitetura de *hardware*

4.3. *Hardware* utilizado

Nesta secção, são apresentados, de forma detalhada, os componentes eletrónicos utilizados para o funcionamento do sistema desenvolvido.

4.3.1. Placa de desenvolvimento

Relativamente à placa utilizada como base para o desenvolvimento do sistema, foi escolhida a *Raspberry Pi 3B+* com um processador de 64 *bits quad-core* a uma frequência de 1.4 GHz (Figura 10). Esta placa pode ser utilizada em projetos complexos com uma grande versatilidade e a um baixo preço, pois apresenta um *pinout* com 40 pinos que podem ser utilizados para as mais diversas funções, tais como, I2C (*Inter-Integrated Circuit*), SPI (*Serial Peripheral Interface*) e UART (*Universal Asynchronous Receiver-Transmitter*). Como memória, este “minicomputador” apresenta

1 GB de memória RAM e tem agregado um cartão de memória SD com 16 GB de armazenamento, que permite a instalação do sistema operativo *Raspbian*.



Figura 10 - Raspberry Pi 3B+ [34]

4.3.2. Leitor NFC

De forma a obter os dados relativos as etiquetas anexadas a cada peça de roupa, recorreu-se a um leitor PN532 que permite a leitura das etiquetas NFC (Figura 11). Este módulo baseia-se num *chip* NXP32 que possibilita a conexão I2C, SPI ou UART com a placa de desenvolvimento e permite detetar as etiquetas até 7 cm de distância. Nesta placa, existem pinos com conexão I2C, SPI e UART. Os pinos de I2C e UART partilham o mesmo *pinout*, sendo usado um *on-board switch* para a mudança de interface.



Figura 11 - Leitor PN532 NFC [35]

4.3.3. Atuadores

No que se refere à atuação exercida para o movimento das peças e do cabide, foram utilizados um motor de passo e diversos servo motores, respetivamente. O motor de passo usado é o 28BYJ-48, sendo responsável pelo movimento circular das peças no interior do armário e encontrando-se conectado a um *driver* ULN2003, que permite o controlo da direção e velocidade do motor (Figura 12). Este *chip* possui um conjunto de 7 *drivers* de transístores que permitem o acionamento de cargas indutivas. As 4 entradas IN1, IN2, IN3 e IN4 são compatíveis com sinais TTL e CMOS, com uma tensão máxima de 5 V. Este motor apresenta uma capacidade bastante eficaz e precisa de rotação para o ângulo necessário.

Para a rotação dos cabides, foram escolhidos motores servo MG995. Cada motor encontra-se conectado a cada cabide, permitindo a rotação de 180° aquando da captação das fotos. Estes motores apresentam uma gama controlável de rotação de 0 a 180° com uma excelente linearidade, controlo e precisão.

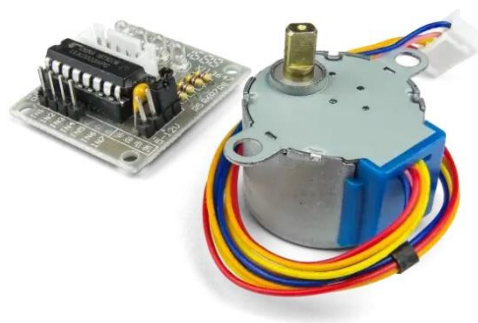


Figura 12 - Motor de Passo 28BYJ-48 e driver ULN2003 [36]

4.3.4. Câmara

Para a captura das fotos das peças de roupa, foi usada uma câmara OV5647 (Figura 13). Esta câmara é integrada com uma placa responsável pela conexão com o processador BCM2835 através da comunicação CSI (*Channel State Information*). A câmara apresenta 5 MP e capta imagens com uma resolução máxima de até 2592x1944 pixels.



Figura 13 - Câmara OV5647 [37]

4.3.5. Iluminação

Uma questão importante relativa à projeção do sistema, prende-se com a iluminação. Trata-se de um fator importante, pois, na captura das fotos, as peças de roupa têm de estar totalmente iluminadas de forma a que não existam quaisquer tipo de sombras. Aqui, a luz apresenta uma importância fulcral, pois permite destacar o elemento ao qual se pretende dar importância, neste caso, a peça de roupa. É aqui que o conceito de iluminação difusa “entra” de forma mais concreta. Trata-se de um tipo de iluminação cuja luz não é muito forte, mas é direta e permite a iluminação de todo um ambiente. Geralmente, este tipo de iluminação difusa é suavizado e difundido pelo espaço em questão, através do uso de um vidro ou acrílico, ou seja, uma espécie de filtro difusor que retém cerca de 20 % da luz, gerando, por isso, menos sombras. Desta forma, foi possível reconhecer que a iluminação teria de apresentar como principais características a cor branca e a capacidade de difusão, de modo que as características das peças fossem evidenciadas. Assim, a escolha recaiu numa tira de LED combinada com calhas de acrílico difusor, permitindo a divisão homogénea da iluminação pela parede. Este tipo de iluminação LED apresenta uma boa flexibilidade, por um preço reduzido, sendo constituída por centenas de *LEDs* que emitem luz a uma temperatura de cerca de 6000 K.

Ao nível da implementação p, surgiram diversos problemas relativos à captura das imagens e ao movimento das peças de roupa dentro do armário. Na captura das imagens, a problemática apresentada relacionava-se com a incapacidade de captura total da peça de roupa. Desta forma,

surgiu a solução de utilizar uma *shield* que permitisse a ligação de duas câmaras OV456 à *Raspberry Pi* a partir do barramento I2C. No entanto, com a conexão da placa, surgiu um problema relativo à inexistência de pinos visto a placa ocupar uma grande parte do *pinout*. Assim, esta ideia foi abandonada e conectou-se um servo motor a uma das paredes do armário com a câmara anexada ao eixo através de uma pequena extensão que, com a rotação de 180° do motor, permitisse a existência de duas posições suficientemente espaçadas de forma a serem capturadas a duas fotos da peça. No que se refere à movimentação das peças, a principal questão colocada prendeu-se com a leitura das etiquetas, tendo sido necessário realocar o leitor para uma das paredes de forma a que peças de roupa não se prendessem no leitor, aquando da leitura das etiquetas

4.4. Arquitetura de software

Neste tópico, é apresentado todo o desenvolvimento do sistema a nível de *software*, tendo sido abordados o fluxo de controlo de *hardware*, o servidor HTTP e os algoritmos de segmentação e deteção de cor.

4.4.1. Software de controlo de hardware

O funcionamento do sistema foi baseado numa máquina de estados com seis estados: READY, STEPPER, DETECTION, CAPTURE, SERVO e STITCH. Quando o utilizador efetua o pedido da peça desejada, o estado inicial de READY altera-se para o de STEPPER. Neste estado, o motor de passo que controla o movimento das peças no interior do armário é acionado e move-se até que seja detetada o UID da etiqueta associada à peça de roupa pedida. Relativamente ao controlo do sistema, foi necessário desenvolver um algoritmo que permitiu o controlo do automatismo do armário (Figura 14), integrando, desta forma, a sensorização existente com os diversos atuadores. Assim, o funcionamento do sistema baseou-se numa máquina de estados detetada a etiqueta respetiva, o estado muda para DETECTION e o UID é apresentado. Logo de seguida, o sistema altera o estado para CAPTURE e duas fotos da frente da peça de roupa (uma da parte de cima e outra da parte de baixo) são capturadas com recurso ao servo motor conectado à câmara que permite a captura de duas fotos espaçadas de 180 °.

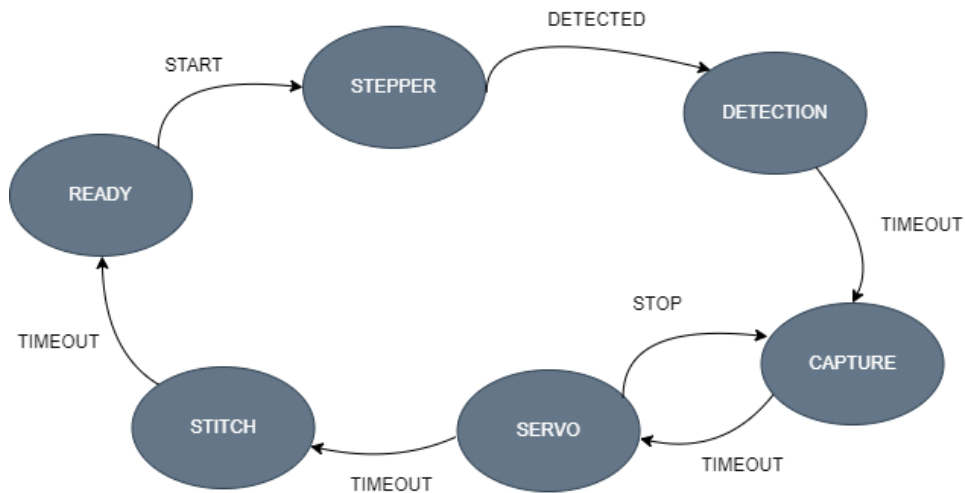


Figura 14 - Fluxo do *software* de controle

Após a primeira captura, o estado é alterado para SERVO e o servo motor anexado ao cabide é acionado, efetuando uma rotação de 180°, voltando novamente ao estado CAPTURE. Aqui, são capturadas de novo duas fotos com recurso ao servo conectado à câmara, sendo estas relativas à parte de trás da peça de roupa. Com as fotos capturadas, o sistema altera o seu estado para STITCH, onde um algoritmo de *stitching* é aplicado, de forma a combinar os dois pares de fotos, originando duas fotos completas, uma da parte da frente e outra da parte de trás da peça de roupa. Este algoritmo é usado de forma a colmatar uma falha relativa à impossibilidade de captura total de cada peça de roupa com apenas uma captura. Assim, este algoritmo que é parte da biblioteca de OpenCV permite a combinação de imagens com base na deteção de pontos chave, transpondo os pontos em comum de cada foto. Assim, o sistema inicia-se com o acionamento do motor de passo que se movimenta até que a etiqueta relativa à peça pedida seja detetada pelo leitor NFC (Figura 15).

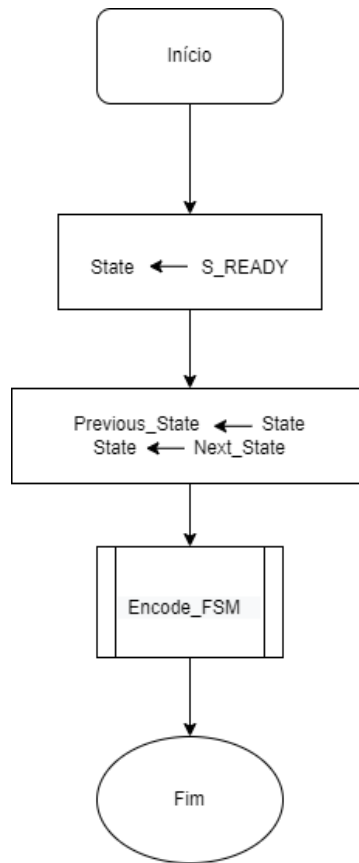


Figura 15 - Função *Encode_State*

Logo que seja detetada, as duas fotos da frente são capturadas e o servo motor do cabide realiza uma rotação de 180° para a captura das duas fotos relativas à parte de trás da peça. Por último, as fotos são submetidas, em pares, ao algoritmo de *stitching*, originando duas fotos completas usadas para posterior validação. Desta forma, a fim de facilitar o desenvolvimento e a codificação de cada estado a implementar no sistema, foram desenvolvidos fluxogramas para cada um dos estados. Numa primeira fase, o fluxo principal do sistema desenvolve-se a partir da função *Encode_FSM* descrita. Numa primeira atribuição, é definido o estado inicial pelo qual se inicia a máquina de estados, sendo, de seguida, realizadas duas atribuições que permitem a atualização do estado da máquina aquando da passagem pelos diversos estados. Por último, é chamada a função principal que possibilita todo este ciclo. Esta função baseia-se no conceito de apontador para funções e trata-se de uma função do tipo *void*, ou seja, não retorna nenhum elemento e requer um parâmetro de entrada numérico que corresponde a cada um dos estados do fluxo. De seguida, o sistema avança do estado READY para o estado SERVO. Neste estado, primeiramente, é efetuada uma inicialização dos pinos a utilizar para a rotação do motor, sendo também usada a função de *SoftPwmCreate* para o ajuste da rotação do motor de acordo com a frequência.

Posteriormente, a chamada a função *Servo_Control* permite a rotação de 180°. Por último, é atribuído o CAPTURE como estado seguinte de acordo com o fluxo originalmente definido. No estado seguinte de STEPPER, inicialmente, são realizadas as inicializações relativas aos pinos a utilizar. Depois, são definidos os valores do ângulo de rotação, da direção e da velocidade do movimento através da função *run*. Por último, encontra-se atribuído o estado seguinte, sendo, neste caso, o DETECTION. Neste próximo estado, o sistema obtém o UID da etiqueta correspondente ao pedido efetuado. Assim, o ciclo inicia-se com a chamada da função *PN532_Get_I2C_UID* que permite a obtenção do valor numérico do UID da etiqueta através do protocolo de comunicação I2C. De seguida, é atribuído o estado seguinte de CAPTURE para a captação das fotos das peças de roupa respetivas. Neste estado, são capturadas duas fotos das peças de roupa, sendo necessária a rotação de 180°, quer dos motores servo conectados ao cabide, quer da câmara. Assim, entre cada foto, o sistema altera o estado para SERVO (para a rotação de 180°) e retorna ao CAPTURE. Desta forma, foi necessário verificar o estado anterior para que o fluxo fosse o correto. Primeiramente, verifica-se se o *previous_state* é o DETECTION ou o SERVO. Caso seja o DETECTION, é chamada a função *Photo_capture* e é atribuído o SERVO ao estado seguinte. Pelo contrário, caso o estado anterior seja o SERVO, é novamente chamada a função *Photo_capture*, sendo, no entanto, atribuído como estado seguinte o STITCH. Por último, no estado STITCH, é chamada a função *stitch* que possibilita o uso do algoritmo de *stitching* para a combinação das duas fotos (Figura 16).

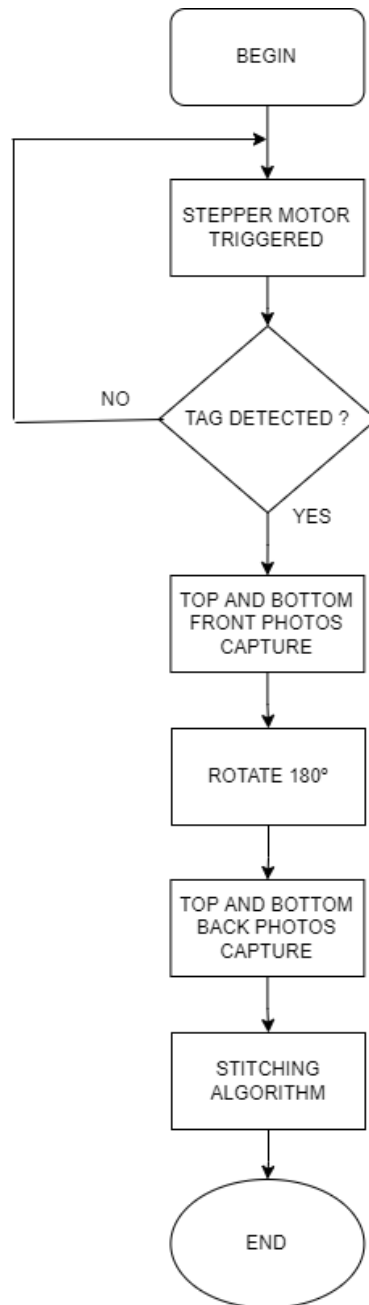


Figura 16 - Fluxograma do Sistema

4.4.2. Servidor HTTP

Para implementar a interface entre o protótipo físico e o utilizador, foi necessário a criar um servidor na *Raspberry Pi* que permitisse a gestão dos pedidos por parte do utilizador (Figura 17).



Figura 17 - Visão Geral do Servidor

Recorreu-se à *Flask framework* que permitiu, através da linguagem *Python*, a codificação de um servidor alojado na placa de desenvolvimento. Numa outra vertente, foi necessário desenvolver um módulo que permitisse o envio de uma foto da peça de roupa e fosse devolvida, quer a classe correspondente, quer as cores predominantes de peça. Este módulo encontra-se englobado pelo modelo da Rede Neuronal usado para a classificação e segmentação e pelo algoritmo de processamento de imagem responsável pela deteção das cores predominantes das peças. Visto não ser possível a instalação do *Detectron2* nativamente na *Raspberry Pi*, foi necessário utilizar um ambiente virtual, tendo-se recorrido à plataforma *Docker*, que possibilita a implementação de aplicações dentro de *containers* virtuais. Assim, foi criado um *container* com a codificação do algoritmo de deteção das cores e instalação de todos os pré-requisitos para o uso do *Detectron2*, tendo sido carregado o *model_final.pth*, correspondente ao modelo usado para a segmentação, tal como se pode observar na Figura 18. De seguida, foi necessário tornar esta funcionalidade acessível de fora do ambiente virtual, tendo sido usado, tal como referido anteriormente, a *Flask Framework*, que recebe as fotos capturadas pelo sistema de visão do protótipo e devolve a classe respetiva e as cores predominantes. Com isto, usando uma outra máquina local a funcionar como cliente, foi possível testar o funcionamento do servidor, com a criação de um menu com diversas escolhas, replicando o menu que estará na aplicação móvel. Foram criadas funções que representam cada uma das escolhas por parte do utilizador, sendo este fluxo representado pelo fluxograma da Figura 19.

```

def prepare_predictor():

    # create config

    cfg = get_cfg()

    # below path applies to current installation location of Detectron2

    cfgFile = "./detectron2/model_zoo/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml"

    cfg.merge_from_file(cfgFile)

    cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5 # set threshold for this model

    cfg.OUTPUT_DIR = "./"

    cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")

    cfg.MODEL.ROI_HEADS.NUM_CLASSES = 6

    cfg.MODEL.DEVICE = "cpu" # we use a CPU Detectron copy

    classes = MetadataCatalog.get("roupa_train").thing_classes = ["calças", "calções", "camisa", "camisola", "polo", "vestido"]

    predictor = DefaultPredictor(cfg)

    print("Predictor has been initialized.")

    return (predictor, classes)

```

Figura 18 – Parâmetros do Modelo treinado

A função *menu_escolher* é chamada quando da escolha pelo utilizador da peça de roupa que pretende, apresentada como parâmetro de entrada, sendo o *id* associado à peça de roupa. No caso da função *menu_adicionar*, é chamada quando o utilizador pretende adicionar uma peça ao guarda-roupa. Relativamente à remoção das peças, foi criada uma função bastante semelhante às anteriores, *menu_remove*, tendo como parâmetro de entrada principal o *id* da peça que o utilizador pretende remover.

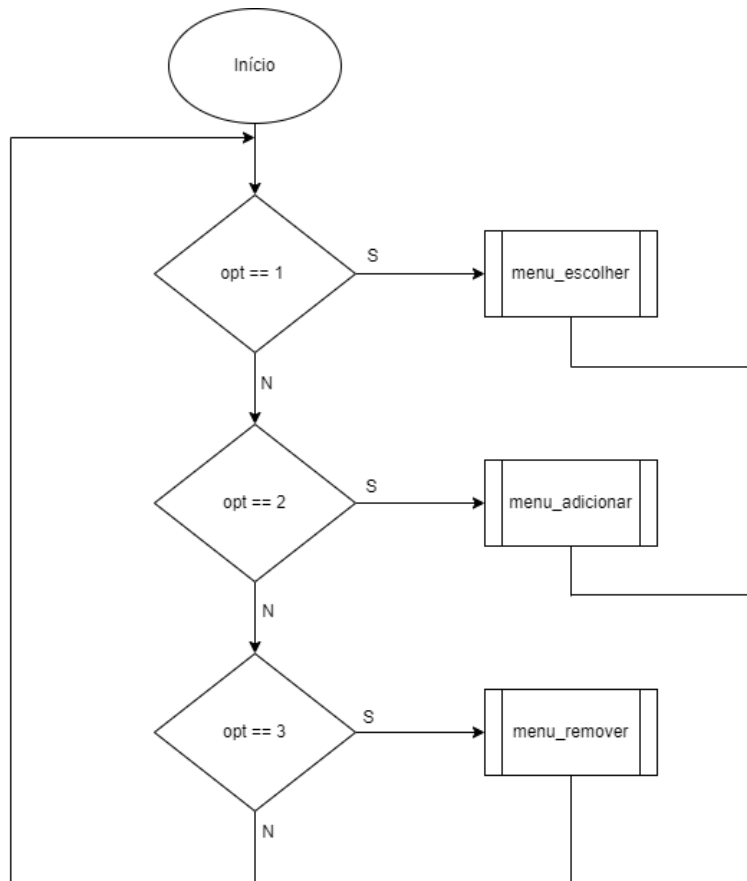


Figura 19 - Fluxo do menu principal

4.4.3. Algoritmos de processamento de imagem

De seguida, encontra-se descrita a metodologia usada para o desenvolvimento e implementação dos algoritmos de remoção de fundo e deteção de cor.

4.4.3.1. Algoritmos de Machine Learning

Para a implementação da remoção do fundo nas imagens capturadas, quer no protótipo físico, quer através da câmara do *smartphone*, foi necessário o desenvolvimento de um algoritmo que permitisse a segmentação das peças de roupa nas fotos. Desta forma, numa primeira fase, foi testado um algoritmo de segmentação através da definição de um certo *threshold*. Com isto, verificou-se que, ao longo de diversos testes, o algoritmo não apresentava resultados satisfatórios,

visto não se conseguir adaptar às diversas cores em contraste com os diversos fundos. Assim sendo, de forma a mitigar este problema, foi necessário alterar o modelo para um algoritmo que tivesse a capacidade de aprender ao longo do tempo, apresentando, por isso, uma capacidade de reconhecer e adaptar-se aos diversos contextos, ou seja, um algoritmo de *Machine Learning*. Foi necessário reconhecer a existência de diversos modelos de inteligência artificial, tendo sido realizada uma análise no sentido de escolher o modelo que melhor se enquadrava para esta aplicação em específico, a segmentação de imagens e objetos. Após o estudo dos modelos, a escolha recaiu na *Mask R-CNN*, sendo um algoritmo específico para segmentação de imagem. Desta forma, a fim de testar o modelo desta rede neuronal, foi necessário escolher uma ferramenta que possibilitasse a existência de uma elevada capacidade de processamento para o treino da rede neuronal, sem que existissem grandes custos associados. A escolha recaiu na ferramenta *Colab* da *Google*, que permite a escrita e a execução de blocos de código. Desta forma, foi possível o treino de modelos computacionais poderosos, a partir de qualquer máquina com ligação à *Internet*. Após a escolha da ferramenta para o treino da rede neuronal, foi necessário proceder à construção do *dataset* com fotos de vários tipos de peças de roupa. Assim, procedeu-se à captura de diversas fotos de 6 tipos diferentes de peças: calças, calções, camisas, camisolas, polos e vestidos. No entanto, foi preciso recorrer a um *dataset* externo, visto ser necessário um elevado número de peças de roupa para cada uma das classes. Este *dataset* encontra-se disponível em [39] com milhares de peças divididas em cerca de 10 classes. Com a extração das peças necessárias para elaborar o *dataset*, este completou-se com 480 fotos de peças de roupa, tendo sido divididas em igual número pelas 6 classes, ou seja, 80 imagens. Desta forma, o *dataset* ficou numericamente equilibrado, tendo sido um fator bastante importante no desenvolvimento do *dataset*, pois o desequilíbrio afeta negativamente os resultados obtidos. Com o *dataset* escolhido, foi necessário efetuar as anotações em cada uma das fotos, neste caso a segmentação no *dataset* de treino e de validação. Para efetuar as anotações, recorreu-se à ferramenta *VGG Annotator* que permite a anotação de imagens e objetos (Figura 20).



Figura 20 - Segmentação e Classificação no *VGG Annotator*

De seguida, procedeu-se à elaboração de código na linguagem *Python* na ferramenta *Colab*, tendo sido dividido em diversas secções. Numa primeira secção, encontram-se as pré-instalações relativas ao modelo utilizado, o *detectron2*, bem como das bibliotecas necessárias, tais como o *OpenCV*, que possibilita o uso de funções relativas ao processamento de imagem. Após estas instalações, foi necessário desenvolver uma função que permitiu a leitura, no formato *json*, e a obtenção das anotações a partir do *dataset* escolhido. Nesta função, é atribuída a cada anotação a respetiva classe entre seis disponíveis (calças, calções, camisa, camisola, polo e vestido) e declaradas através de um *array*. Nesta função, são capturados os dados necessários (*region_attributes*, *shape_attributes* e as coordenadas *x* e *y* da *bbox*) e convertidos para as variáveis do formato *json* (*bbox*, *segmentation*, *category_id* e *annotations*). No caso do *category_id*, o *id* é atribuído de acordo com o número do índice de cada uma das classes definidas no *array* acima. As últimas quatro linhas correspondem ao registo de cada um dos conjuntos de dados (treino e validação) a cada uma das variáveis criadas (*roupa_train* e *roupa_val*) e respetiva atribuição das classes a cada imagem (Figura 21).

```

def get_garments_dicts(img_dir):
    json_file = os.path.join(img_dir, "via_region_data.json")
    with open(json_file) as f:
        imgs_anns = json.load(f)

    classes = {"calças":0, "calções":1, "camisa":2, "camisola":3, "polo":4, "vestido":5,}
    dataset_dicts = []

    for idx, v in enumerate(imgs_anns.values()):
        record = {}
        filename = os.path.join(img_dir, v["filename"])
        height, width = cv2.imread(filename).shape[:2]
        record["file_name"] = filename
        record["image_id"] = idx
        record["height"] = height
        record["width"] = width
        annos = v["regions"]
        objs = []

        for i, anno in enumerate(annos):
            #assert not anno["region_attributes"]
            a = anno["region_attributes"]
            anno = anno["shape_attributes"]
            px = anno["all_points_x"]
            py = anno["all_points_y"]
            poly = [(x + 0.5, y + 0.5) for x, y in zip(px, py)]
            poly = [p for x in poly for p in x]
            obj = {
                "bbox": [np.min(px), np.min(py), np.max(px), np.max(py)],
                "bbox_mode": BoxMode.XYXY_ABS,
                "segmentation": [poly],
                "category_id": classes[a["names"]],
            }
            objs.append(obj)
        record["annotations"] = objs
        dataset_dicts.append(record)
    return dataset_dicts

for d in ["train", "val"]:
    DatasetCatalog.register("roupa_" + d, lambda d=d: get_garments_dicts("/content/gdrive/MyDrive/roupa/" + d))
    MetadataCatalog.get("roupa_" + d).set(thing_classes=["calças", "calções", "camisa", "camisola", "polo", "vestido"])
    garments_metadata = MetadataCatalog.get("roupa_train")

```

Figura 21 - Função *get_garments_dicts*

A secção seguinte corresponde à visualização das anotações no *dataset* de treino, de forma a verificar a correta leitura através da escolha de três imagens aleatórias do conjunto de dados respetivo. Nesta análise, são desenhadas as regiões segmentadas e as respetivas caixas delimitadoras. Relativamente ao módulo seguinte, é onde são ajustados os parâmetros do modelo a treinar (Figura 22). Tal como abordado anteriormente, neste modelo, é usado o conceito de *transfer learning*, que possibilita o uso de um modelo pré-treinado. Assim, é através de *cfg.merge_from_file* que é carregado o modelo *model_zoo*, que é usado para efeitos de *Transfer Learning*. De seguida, é carregado o respetivo *dataset* (*roupa_train*) para o treino. No caso da *ims_per_batch*, trata-se de um parâmetro que indica o número de amostras treináveis inseridas na rede por cada iteração. No que se refere ao *batch_size_per_image*, indica o tamanho do subconjunto retirado a partir da *Region Proposal Network* (que representa possíveis zonas onde podem ser encontradas regiões de interesse). Relativamente à *learning rate*, trata-se de um parâmetro usado para o controlo do ritmo ao qual a rede neuronal atualiza para o parâmetro estimado, ou seja, a velocidade com que o modelo converge para o objetivo definido.

```
cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("roupa_train",)
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml")
cfg.SOLVER.IMS_PER_BATCH = 1
cfg.SOLVER.BASE_LR = 0.0003
cfg.SOLVER.MAX_ITER = 10000
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 64
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 6

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

Figura 22 - Parâmetros do modelo

Desta forma, é necessário existir um *trade-off* na escolha deste valor, pois, para valores baixos, as atualizações são mais lentas e o número de iterações maior; pelo contrário, para valores altos, as

atualizações são mais rápidas, podendo o modelo afastar-se do valor definido. Seguidamente, ainda existem mais dois parâmetros que podem ser ajustados: *max_iter* e *num_classes*. No caso das iterações, trata-se de um valor que foi alterado com vista a maximizar o valor da Precisão Média e evitar o *overfitting*, que indica a existência de resultados muito próximos para um determinado conjunto de dados, não sendo, por isso, suscetível a alterações aos dados de entrada). Relativamente ao *num_classes*, este apresenta o valor 6, pois representa o número de classes em que se encontra dividido o *dataset* sujeito ao treino. Após o treino do modelo, foi necessário desenvolver uma secção que permitisse a visualização dos dados. Numa representação gráfica, recorreu-se ao *tensorboard*, tendo sido representados diversos parâmetros indicadores do desenvolvimento do modelo ao longo do tempo, tais como, *accuracy* e *total_loss*. Para a representação dos resultados, primeiramente, são carregados os pesos obtidos a partir do treino da rede neuronal. O parâmetro *cfg.MODEL.ROI_HEADS.THRESH_TEST* permite ajustar o valor de *threshold* para o modelo. Na visualização dos resultados, foram analisadas três imagens aleatórias do *dataset* de validação com a respetiva segmentação, classificação e desenho das caixas delimitadoras para cada uma das imagens. De seguida, encontram-se duas secções responsáveis pela apresentação dos resultados da Precisão Média e da *Intersect Over Union*, respetivamente. No caso da *Average Precision*, os resultados apresentam-se em formato de tabela com um valor de AP para cada uma das classes definidas. Para a *Intersect Over Union*, em cada imagem, é captada a máscara binária obtida a partir da segmentação e sobreposta com a máscara prevista pelas anotações, sendo obtido o valor da área interseçada e o valor da união para o cálculo da IoU. Para cada um destes cálculos, é incrementado o valor do *threshold* de 0.5 a 0.95. Por último, foi desenvolvida uma função de *segment* que permitiu a segmentação e classificação de imagens inseridas pelo utilizador. Desta forma, a função apresenta, como parâmetro de entrada, a imagem a segmentar, sendo obtidas as dimensões desta, para o cálculo do tamanho da máscara binária. Com o carregamento dos pesos obtidos a partir do treino, a respetiva imagem é segmentada e classificada, sendo geradas três imagens para cada teste: a imagem com a classificação, desenho da segmentação e caixa delimitadora, a imagem final segmentada e a respetiva máscara binária (Figura 23). Estas duas últimas representações, a imagem segmentada e a respetiva máscara binária, apresentam grande importância, pois foram utilizadas como dados de entrada no algoritmo de deteção de cor.

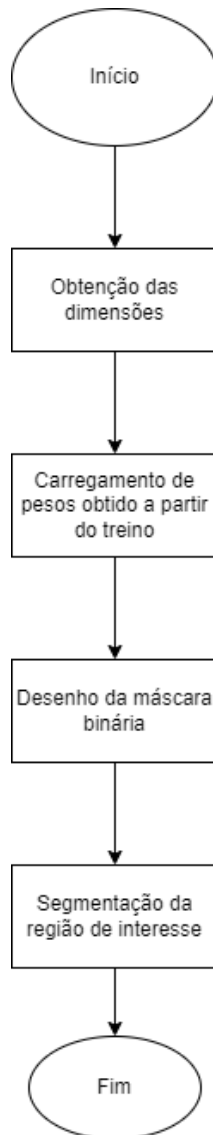


Figura 23 - Fluxo da função

4.4.3.2. Algoritmo de detecção de cor

A fim de possibilitar a obtenção das cores existentes nas peças de roupa captada, quer a partir do protótipo físico, quer a partir da aplicação móvel, foi necessário desenvolver um algoritmo. Desta forma, após a remoção do fundo da peça de roupa, é aplicado um algoritmo de cor que permite a identificação das cores das peças de roupa sujeitas a análise. O algoritmo é baseado no modelo RGB que permite definir cores. Este modelo fornece um método diferenciado com base em

componentes de cores específicos. Assim, o modelo de cores RGB usa as componentes R (*Red*), G (*Green*) e B (*Blue*) para a definição da quantidade de luz vermelha, verde e azul numa determinada cor. Cada componente varia num intervalo de 0 a 255, determinando a cor através da mistura das componentes, permitindo armazenar e exibir uma ampla gama de cores.

No desenvolvimento do algoritmo, primeiramente, foram definidos dois *arrays* de 139 elementos, um com os nomes das cores e outro com os três valores correspondentes ao código RGB de cada cor obtidos a partir de [40]. De seguida, para que o funcionamento do algoritmo esteja de acordo com o esperado, foi necessário obter tanto a foto da peça de roupa com o fundo retirado, como a máscara binária aplicada a partir dos algoritmos de *Machine Learning*. No que se refere à máscara binária, esta é obtida a partir da implementação dos algoritmos de *Machine Learning*, para a tarefa da segmentação de imagem. Esta imagem corresponde a uma representação de zeros e uns em que os uns equivalem à região de interesse e os zeros indicam o fundo retirado (Figura 24).



Figura 24 - Peça de roupa segmentada e respetiva máscara binária

Assim sendo, com a foto da máscara binária, foram obtidas as coordenadas que correspondem aos *pixéis* que se encontram com valor igual a 1, pois são estes que equivalem à região de interesse gerada na foto da peça de roupa original. Como as duas imagens apresentam tamanhos iguais (300x280 pixels), as coordenadas obtidas na máscara binária correspondem às mesmas coordenadas na imagem com fundo retirado. Desta forma, foi sujeita a análise a imagem com o fundo retirado, mais concretamente os *pixéis* obtidos a partir da máscara binária. Com isto, foi desenvolvida uma função *closestColor* que permitiu obter os valores RGB, a partir das coordenadas x e y de cada *pixel*. Esta função verifica a diferença entre os valores RGB obtidos e os valores das

cores do *array* padrão (das 139 cores) definido, devolvendo a cor que corresponde à menor diferença calculada. Com esta função, foram obtidas as cores de cada *pixel* e guardadas num *array*. Após a obtenção das cores de cada *pixel*, foi necessário verificar o número de vezes que cada cor se repetia, tendo sido criado um contador para cada cor. que incrementava sempre que existisse uma ocorrência dessa cor. Estes contadores foram guardados num *array* de forma a que, de seguida, fossem calculadas as percentagens relativas a cada uma das cores presentes na peça de roupa. No que se refere às percentagens, de forma a mitigar possíveis erros na segmentação, para efeitos de análise, foram consideradas apenas as percentagens de cores acima dos 10% (Figura 25).

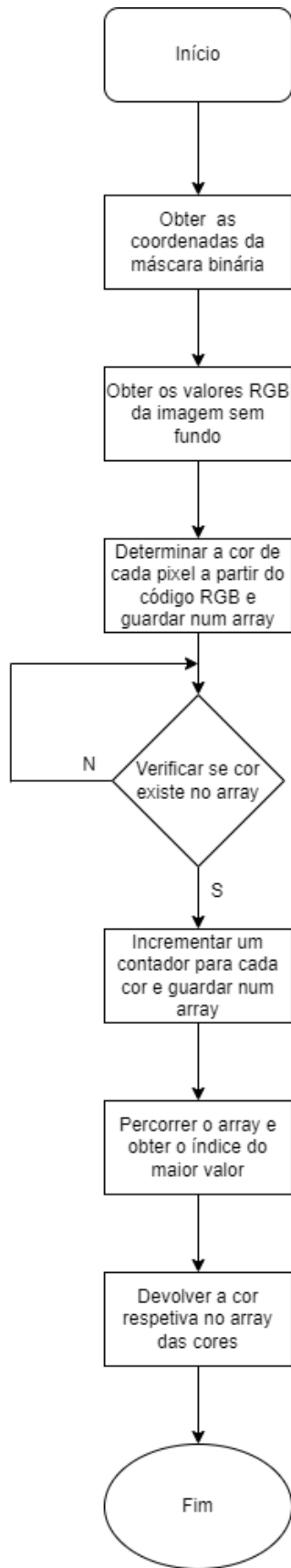


Figura 25 - Algoritmo de Detecção de Cor

5. Testes e resultados

Neste capítulo, são apresentados os diversos testes realizados de forma a validar e verificar o sistema proposto e os resultados obtidos [38]. Foram realizados testes tanto ao nível do *hardware* como do *software*, ou seja, da segmentação de imagem e do algoritmo de deteção de cor.

5.1. Testes Realizados

De modo a garantir a eficiência do sistema, os diferentes componentes foram testados, em primeiro lugar, individualmente, verificando se os sinais correspondem às especificações pretendidas e, posteriormente, integrados no sistema. De seguida, visto o sistema envolver desenvolvimento de código, foi necessário testá-lo em cada um dos estágios que o mesmo engloba, de forma a validar e verificar os requisitos previstos. Relativamente ao *hardware*, à medida que a eletrónica necessária para cada módulo se foi desenvolvendo, esta foi sendo testada e validada de forma individual, de modo a que estivesse de acordo com os requisitos definidos inicialmente. Por fim, de modo a testar e validar o funcionamento do sistema final, todos os módulos foram integrados e o sistema total foi sujeito a diversos testes e validações finais. No que se refere aos testes realizados de forma a validar o conceito a apresentar nesta dissertação, foram divididos em duas secções: validação ao nível do *hardware* físico e ao nível do *software*, ou seja, com os algoritmos de *Machine Learning* para a remoção do fundo das imagens bem como da deteção da cor das peças. Relativamente aos testes que, de certa forma, validam o conceito ao nível do armário físico e garantem a eficiência do sistema, foram, primeiramente, testados os diferentes componentes individualmente, verificando se os valores obtidos estavam de acordo com o pretendido. De seguida, visto o projeto envolver algoritmia e codificação, foi necessário testá-lo com as diversas imagens de peças de roupa que foram recolhidas através da captura da câmara do *smartphone*. Nestas capturas foi necessário ter em conta diversos fatores, entre os quais, a existência de fundos diferentes entre as fotos a fim de verificar a versatilidade e a adaptabilidade do algoritmo a diferentes cenários. Assim, no que se refere aos resultados no protótipo físico, os automatismos responsáveis pelo movimento encontram-se totalmente funcionais, concretamente, o motor de passo, controlado através do *driver* ULN2003, que se apresenta conectado ao eixo

numa plataforma circular que permite o suporte dos vários servos motores responsáveis pela rotação da peça de roupa. Cada servo motor encontra-se conectado a um cabide, possibilitando a rotação de 180° para a captura das fotos. Com isto, e tal como abordado anteriormente, foi desenvolvido um algoritmo de controlo que permite a integração de toda a eletrónica física. Quando se atinge o estado da captura, é usado um algoritmo de *stitching* de forma a captar a peça de forma total, tendo sido realizados diversos testes para verificar a sua fiabilidade (Figuras 26 a 31).



Figura 26 - Foto da zona de cima da peça

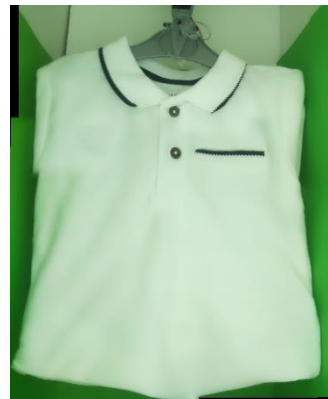
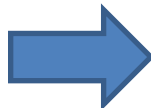


Figura 28 - Foto total da peça



Figura 27 - Foto da zona de baixo da peça



Figura 29 - Foto da zona de cima da peça

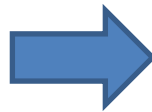


Figura 30 - Foto da zona de baixo da peça



Figura 31 - Foto total da peça

Como se pode observar nos testes efetuados, o algoritmo de *stitching* implementado para mitigar o problema da câmara não conseguir captar de forma total a peça de roupa, apresentou resultados dentro do esperado, tendo este método, de uma forma geral, detetado os diversos pontos comuns entre as duas fotos e combinando-os na foto final. Apesar destes resultados, é necessário ter em conta que o algoritmo apresenta algumas lacunas no que diz respeito aos limites da peça, pois como se pode observar, em ambas as fotos finais, existem zonas em que surgem certas falhas que indicam dificuldade do algoritmo no estabelecimento dos limites. Ainda no que se refere ao *hardware* físico, este protótipo desenvolvido apresenta a capacidade de garantir um ambiente controlado aquando da captura das fotos, de forma a evitar sombras e zonas escuras nas peças de roupa. Para verificar esta evidência, foram capturadas fotos de peças de roupa em três contextos diferentes: uma primeira foto dentro do sistema controlado desenvolvido, uma segunda e terceira fotos captadas fora do sistema controlado, ou seja, com a incidência quer de luz natural, quer de luz artificial e as sombras que a esta luz se encontram associadas (Figuras 32 a 34).



Figura 32 - Fotos captadas fora do ambiente controlado (esquerda e centro) e Foto captada dentro do ambiente controlado (direita)



Figura 33 - Fotos captadas fora do ambiente controlado (esquerda e centro) e Foto captada dentro do ambiente controlado (direita)



Figura 34 - Fotos captadas fora do ambiente controlado (esquerda e centro) e Foto captada dentro do ambiente controlado (direita)

Após o teste e a verificação dos automatismos desenvolvidos, foram realizados testes aos algoritmos de remoção de fundo e deteção de cor. Primeiramente, testaram-se os algoritmos, tendo como ponto de partida a possível integração com uma aplicação móvel, ou seja, o utilizador acede a uma aplicação móvel e captura a foto de uma peça de roupa que pretende, sendo

removido o fundo e obtidas as principais cores da peça sujeita a análise. A fim de proceder à obtenção dos resultados, foram obtidas 7 fotos de peças de roupa de forma aleatória, tendo 4 fotos sido obtidas a partir do *dataset* disponível em [39] e outras 3 capturadas a partir do *smartphone* pessoal. Assim sendo, recorrendo à ferramenta *Colab*, onde foi realizada a modelização da rede neuronal, foi aplicada a segmentação de imagem às respetivas imagens de validação. Com isto, a partir da imagem original foi obtida a máscara binária e a respetiva foto final com o fundo removido (Figura 35).



Figura 35 - Imagem Segmentada

No algoritmo de remoção de fundo, foi utilizada a *Mask R-CNN* como modelo de Rede Neuronal. Desta forma, foram efetuados diversos treinos que permitiram o ajuste de todos os parâmetros com vista à maximização da Precisão Média e da *Intersect Over Union*. Numa primeira parte, foi realizado o treino com um *Batch_Size_Per_Image* igual a 128 e, posteriormente, com um valor igual a 64. Mais concretamente, ao nível do treino, o modelo foi testado com um número de iterações elevado, de forma a verificar o valor no qual este entrava em *overfitting*. Com isto, foi necessário obter os dados de *Perda Total* relativos a cada iteração. O primeiro treino correspondeu a um valor de iterações igual a 5000. Desta forma, verificou-se que o modelo passou a estar em *overfitting* a partir das 3000 iterações, pois o traçado gráfico representado tornou-se praticamente uma linha reta paralela ao eixo das ordenadas (Figura 36).

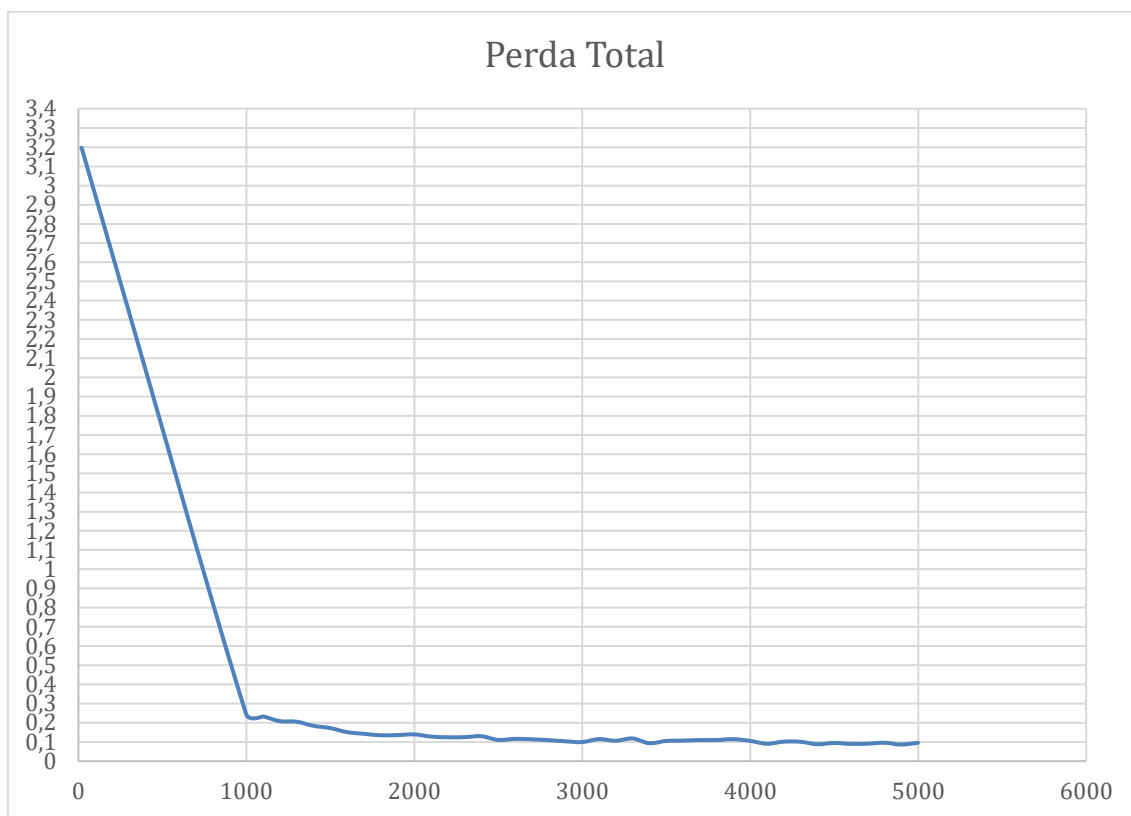


Figura 36 - Perda Total do modelo às 5000 iterações

No que se refere à *Precisão Média*, foi utilizada uma função que permitiu a apresentação dos resultados obtidos em cada uma das classes (Tabela 2).

Tabela 2 - Precisão Média para 5000 iterações

Precisão Média	Calças	Calções	Camisa	Camisola	Polo	Vestido
83.034	91.716	87.129	88.898	84.752	90.908	54.8

Como se pode observar pelos resultados apresentados o valor obtido para a classe do Vestido não está dentro do esperado, tendo o modelo dificuldade em reconhecer da forma mais correta a silhueta do vestido, visto os vestidos do *dataset* não apresentarem uma forma semelhante entre eles. De seguida, ao ajustar o valor das iterações para 3250, os resultados melhoraram, tendo o valor da Precisão Média aumentado para 85.667. Este valor aumentou muito devido à melhoria do valor na classe Vestido (Tabela 3).

Tabela 3 - Precisão Média para 3250 iterações

Precisão Média	Calças	Calções	Camisa	Camisola	Polo	Vestido
85.667	90.565	87.129	91.785	85.941	91.595	66.988

Posteriormente, realizou-se um teste com um valor de iterações um pouco mais baixa, ou seja, 3000 iterações. Neste resultado, o valor global da Precisão Média diminui um pouco, no entanto, o valor obtido na classe Vestido aumentou consideravelmente, tendo-se aproximado dos restantes valores (Tabela 4).

Tabela 4 - Precisão Média para 3000 iterações

Precisão Média	Calças	Calções	Camisa	Camisola	Polo	Vestido
84.094	90.161	87.129	89.81	82.64	81.7	73.124

No gráfico apresentado na Figura 37, verifica-se a descida exponencial do valor da Perda Total com o aumento das iterações. No entanto, entre os valores de 2500 e 3000, existe uma estabilização dos valores, tendo sido realizados outros 2 testes com valores de 2500 e 2750 iterações, de forma a verificar o limite em que o modelo nem esteja em *underfitting* nem em *overfitting*, que se refere a um modelo que não aprendeu o suficiente sobre o conjunto de dados, apresentando um erro elevado ao nível do treino.

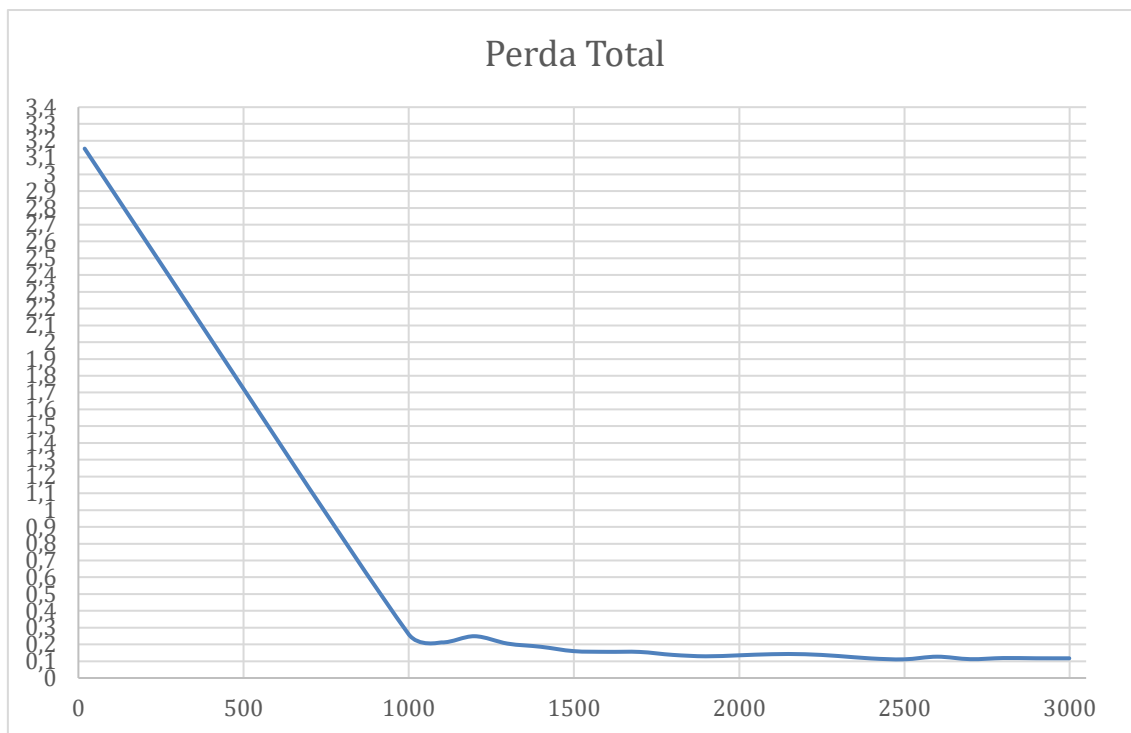


Figura 37 - Perda Total do modelo às 3000 iterações

Desta forma, foram realizados 2 treinos com um número de iterações de 2750 e 2500, tal como apresentado na Tabela 5.

Tabela 5 - Precisão Média para 2500/2750 iterações

Iterações	Precisão Média	Calças	Calções	Camisa	Camisola	Polo	Vestido
2750	85.581	91.011	98.57	93.441	83.564	92.702	54.198
2500	79.637	88.52	92.426	82.028	77.355	80.297	54.198

Nestes resultados apresentados, verifica-se que a partir das 2500 iterações, os resultados tornam-se cada vez menos satisfatórios, pois o modelo entrou em *underfitting*. No caso das 2750 iterações, o valor obtido foi bastante razoável, no entanto, a Precisão Média na classe Vestido encontra-se aquém das expectativas, sendo bastante inferior às outras 5 classes. Além do mais, ainda foram obtidos os valores da *Intersect Over Union* ao longo dos treinos anteriormente realizados, tal como se pode verificar na Tabela 6.

Para a obtenção desta métrica, foi utilizada uma função que permite o cálculo dos valores de *IoU* com um *threshold* variável entre 0.5 e 0.95, espaçados de 0.01. A função percorre as 48 imagens do *dataset* de validação e obtém o respectivo valor do *IoU* para cada uma das fotos, sendo comparada a máscara prevista com a máscara real obtida (a partir da aplicação da Rede Neuronal). Neste cálculo, as duas máscaras são interseçadas, sendo obtidas a área de interseção e de união entre as duas máscaras e calculada a respectiva métrica (Figura 38).

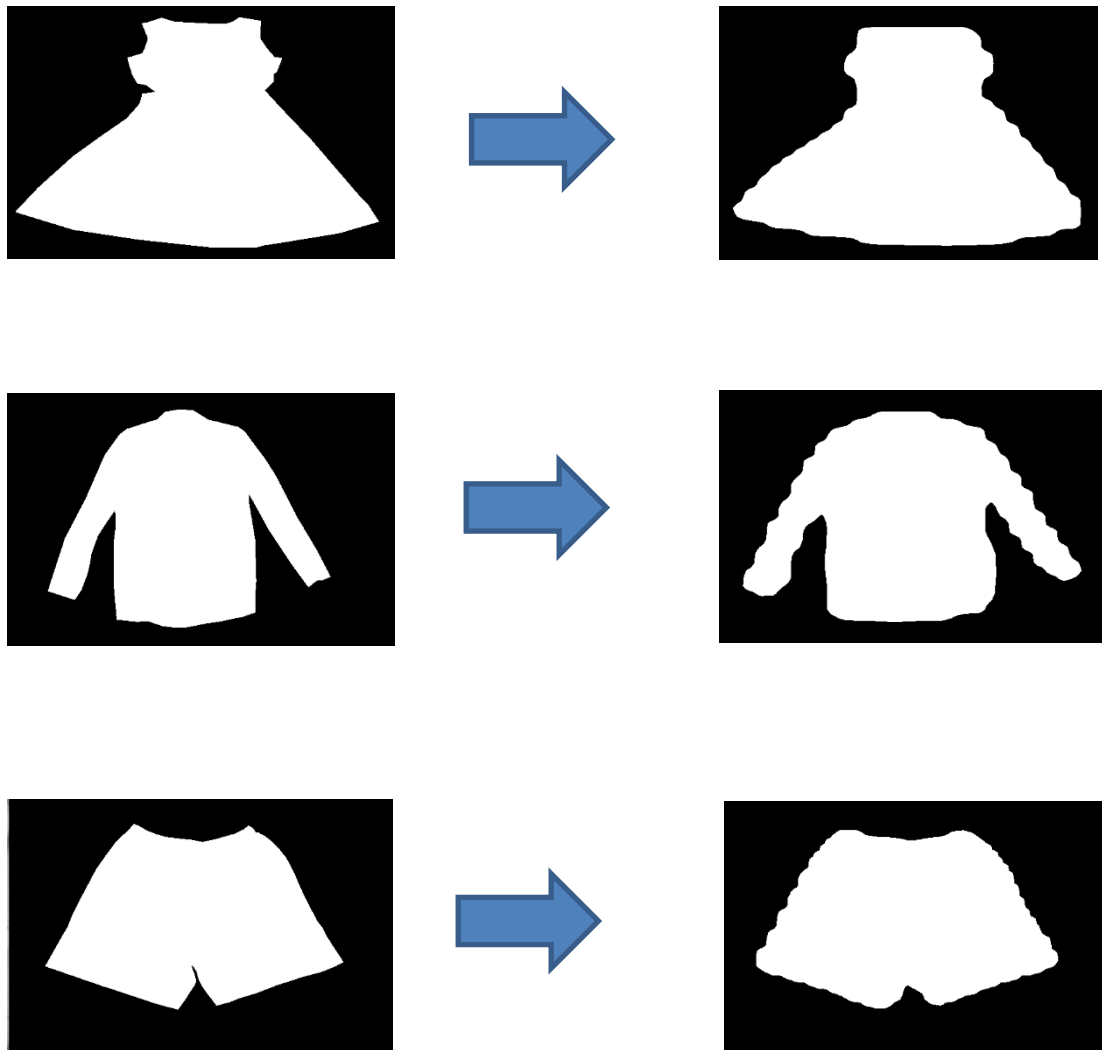


Figura 38 - Máscaras Previstas (à esquerda) e Máscaras obtidas (à direita) para um THR = 0.5

Assim, foram obtidos e apresentados na Tabela 6 os valores para diferentes números de iterações, tendo os resultados estado dentro do esperado, pois para as diversas situações testadas, as máscaras geradas a partir da aplicação do modelo correspondem em grande parte ao previsto.

Tabela 6 - *Intersect Over Union* obtida nos treinos

Iterações	0.5	0.6	0.7	0.8	0.9	Média
5000	0.91	0.92	0.91	0.89	0.74	0.87
3250	0.92	0.93	0.89	0.81	0.6	0.83
3000	0.87	0.90	0.86	0.76	0.63	0.80
2750	0.87	0.88	0.84	0.8	0.53	0.78
2500	0.92	0.91	0.86	0.75	0.67	0.82

Por último, de forma a verificar os resultados obtidos a partir do treino, foi necessário testá-los em diversas imagens de peças de roupa (Figura 39).

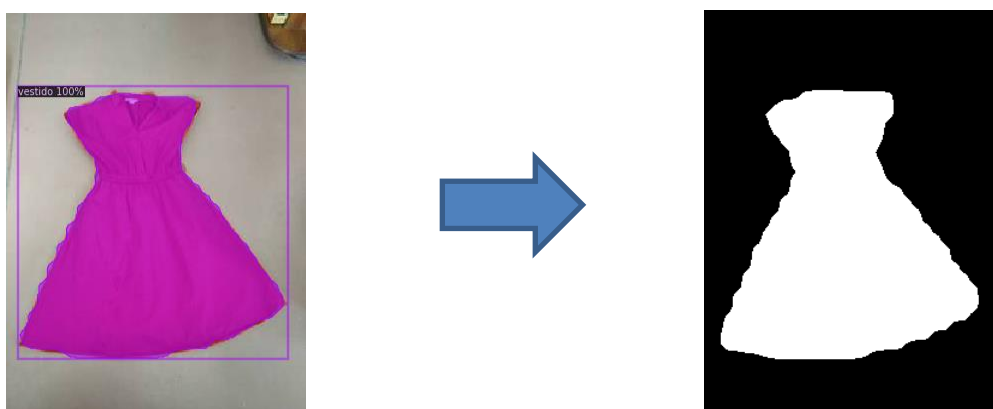


Figura 39 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)

No vestido, a segmentação apresenta uma boa eficácia, tendo o algoritmo de se adaptado e distinguido de forma correta os *pixéis* correspondentes ao fundo e à região de interesse. Relativamente à classificação, a classe gerada correspondeu à classe prevista com uma exatidão de 100%. No que se refere à Figura 40 e 41, os resultados foram bastante semelhantes, tendo a segmentação e a classificação correspondido ao esperado, ou seja, classe Calças na Figura 40 e classe Calções na Figura 41, ambas com uma exatidão de 100%

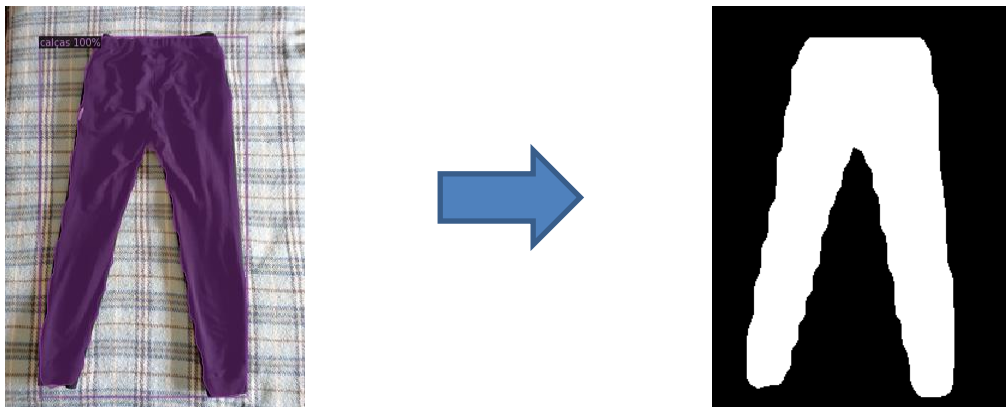


Figura 40 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)

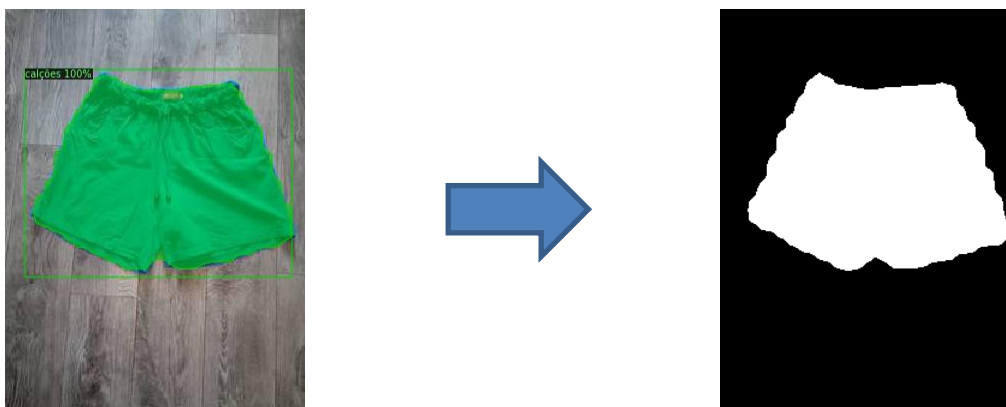


Figura 41 - Imagem classificada e Segmentada (à esquerda); Máscara Binária (à direita)

Na Figura 42, verifica-se uma segmentação com resultados aquém do esperado, visto o algoritmo ter considerado algumas porções relativas ao fundo como regiões de interesse. Esta falha é explicada com o facto do fundo da imagem não apresentar uma cor homogénea, sendo constituído por padrões e desenhos que, de certa forma, “confundem” o algoritmo. Esta falha leva a que o modelo tenha dificuldade em distinguir o que se trata de fundo e o que corresponde a *pixéis* da peça.



Figura 42 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)

Nas Figuras 43 e 44, apresentam-se outras duas imagens que, respetivamente, representam uma camisa e um polo. Para estas duas imagens, o algoritmo apresentou uma exatidão relativamente elevada, tanto na segmentação como na classificação. No caso da segmentação, estes bons resultados podem ser explicados pela existência de um fundo homogéneo, com apenas uma cor (Figura 43) e com um padrão simples, sem grandes desenhos (Figura 44). Já na classificação, visto as silhuetas obtidas serem semelhantes para cada uma das classes, o algoritmo identificou, sem grande dificuldade, as classes relativas cada uma das peças, ou seja, Camisa com 98% de exatidão na Figura 43 e Polo com 97% de exatidão na Figura 44.

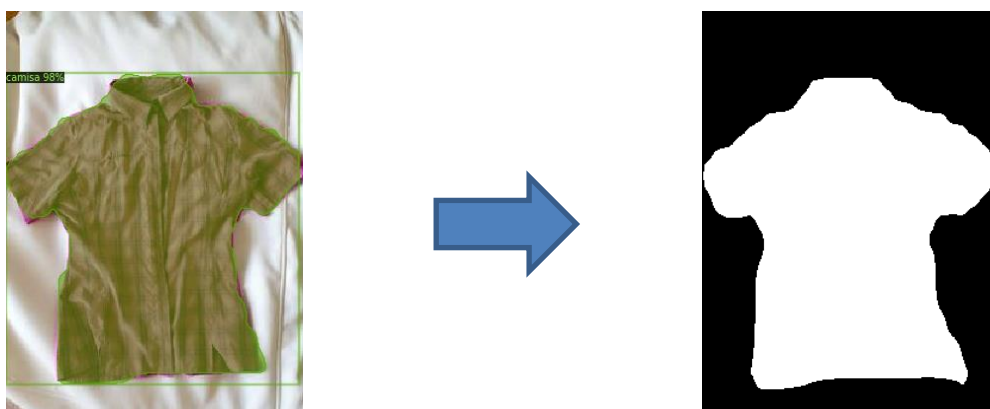


Figura 43 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)

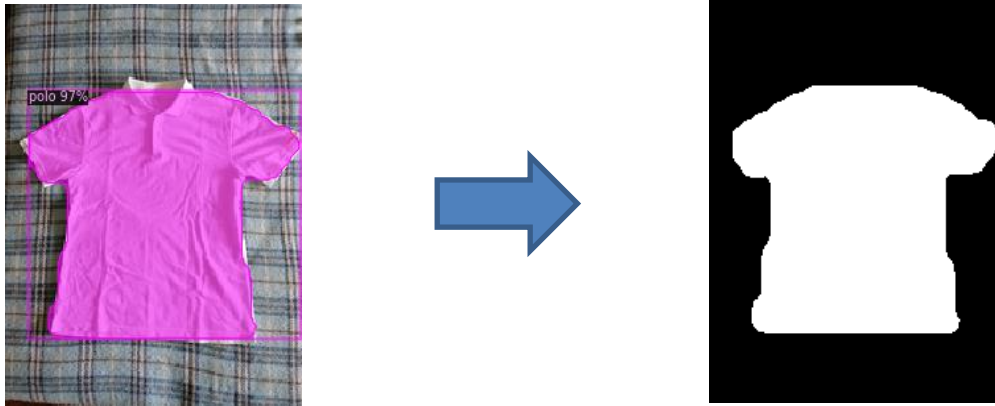


Figura 44 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)

Por último, foi testada uma última imagem capturada a partir da câmara do sistema físico desenvolvido. Visto ser uma peça que se encontra no armário físico, como se pode observar a partir da Figura 45, a peça encontra-se pendurada num cabide. Além do mais, a fim de facilitar a segmentação das peças duas paredes laterais e a parede do fundo foram forradas a cartolina verde, evitando assim fundos com padrões mais diferenciados. Apesar destas medidas, a segmentação da imagem não apresenta resultados ideais, observando-se que, em certas zonas, considerou como região de interesse *pixéis* correspondentes ao fundo, essencialmente na zona do colarinho e do cabide, pois foi considerada uma porção do cabide e do fundo verde como pertencentes à peça de roupa. No caso da classificação, o algoritmo identificou de forma correta a classe, apesar de ter sido com uma exatidão não muito alta, cerca de 75%, pois a silhueta obtida não apresenta parecenças com as silhuetas das camisolas de treino. De seguida, foram realizados os testes ao algoritmo de deteção de cor. Neste algoritmo, aquando da análise da imagem segmentada para a obtenção das percentagens das cores presentes, a existência de erros na segmentação da peça tornaria os resultados enviesados, considerando as cores dos *pixéis*

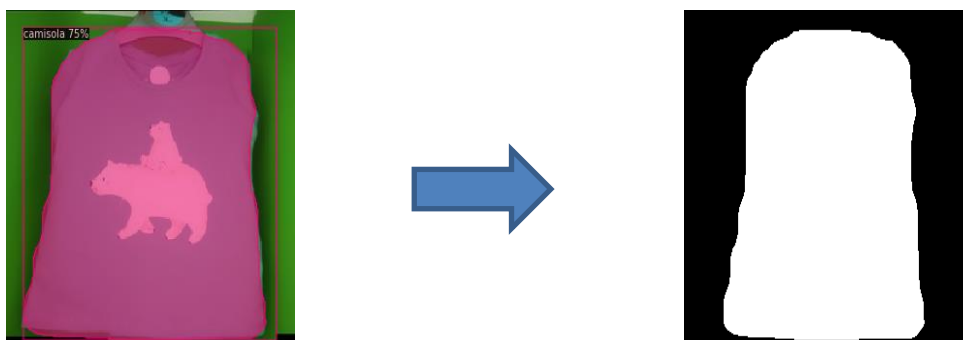


Figura 45 - Imagem Classificada e Segmentada (à esquerda); Máscara Binária (à direita)

correspondentes ao fundo como fazendo parte da peça de roupa. De forma a mitigar este problema, foram apenas consideradas relevantes percentagens acima de 10%, ignorando, assim, valores mais baixos que possam corresponder a possíveis falhas na segmentação. Algo importante a ter em conta nesta percentagem ignorada, é a possível existência de erros, podendo existir cores (pertencentes à peça) com percentagens inferiores a 10%, mas que segundo este algoritmo não são consideradas relevantes e, por isso, ignoradas. Tal como referido na secção de desenvolvimento, para o algoritmo de cor, foi necessária a obtenção da máscara binária gerada a partir da segmentação. A partir da máscara foram obtidas as coordenadas dos *pixéis* de interesse, de forma a que essas mesmas posições fossem percorridas na imagem original. Assim, foram obtidas as percentagens de cada uma das cores existentes nesta peça de roupa.

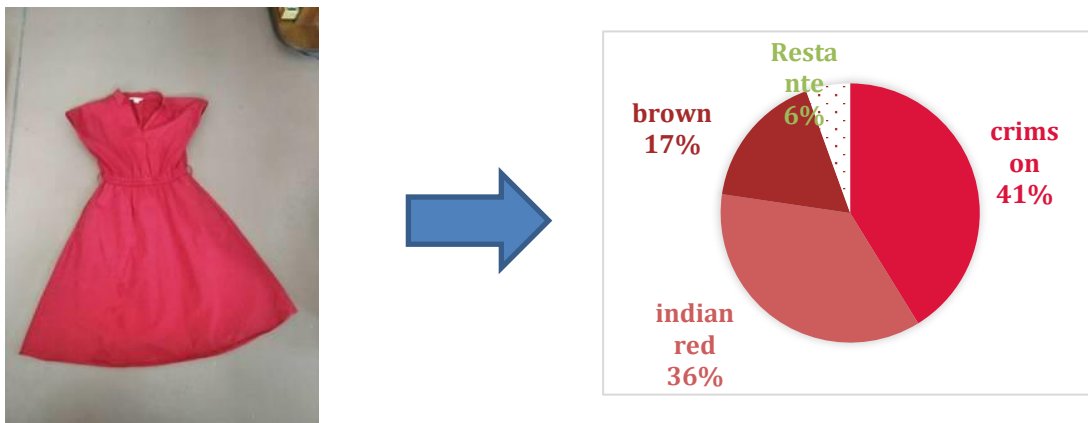


Figura 46 - Distribuição das cores principais na peça de roupa

Ao analisar a distribuição percentual das cores (Figura 46), observa-se a predominância de 3 cores (Castanho, Carmesim e Vermelho indiano) que correspondem a 94% dos *pixéis*, tendo sido ignorados 6%, que dizem respeito a possíveis zonas não relevantes. De seguida, foram realizados testes semelhantes a outras fotos (Figuras 47 a 55).

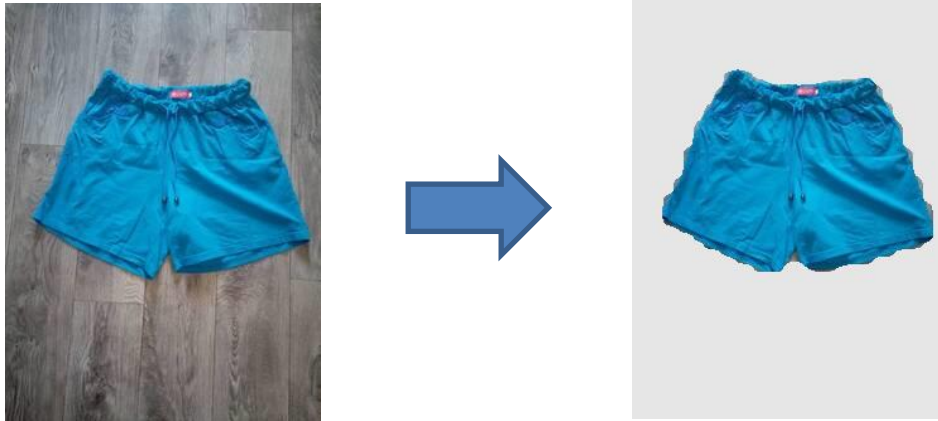


Figura 47 - Imagem Original (à esquerda); Segmentação (à direita)

Neste teste, a percentagem de 22% correspondente às cores desprezáveis (Figura 48) apresenta um valor superior ao anterior, pois, como se pode observar nos limites da peça de roupa, na segmentação foi considerado um maior número de *pixéis* do fundo.

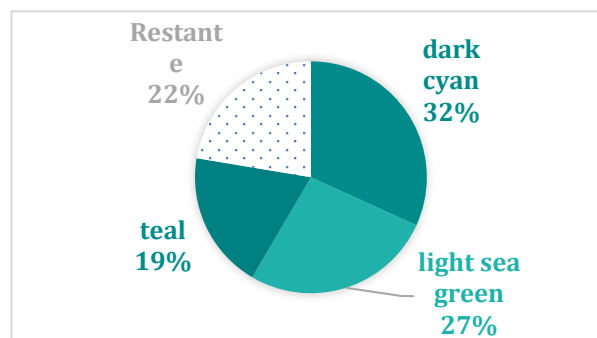


Figura 48 - Distribuição das cores

No teste seguinte (Figura 49), a imagem sujeita a análise correspondia a uma camisa com um padrão axadrezado. Assim, foi realizada a segmentação, gerada a respetiva máscara binária e obtidas as percentagens das cores (Figura 50).



Figura 49 - Imagem Original (à esquerda); Segmentação (à direita)

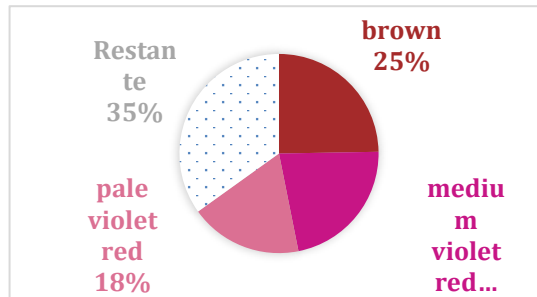


Figura 50 - Distribuição das cores

Ao analisar as percentagens das cores da peça de roupa, verifica-se de imediato um valor relativamente alto da percentagem correspondente aos *pixels* ignorados. Este valor elevado é explicado pelo facto da peça de roupa apresentar um padrão axadrezado e listado, apresentando, por isso, diversas cores com uma percentagem baixa. Visto estas percentagens serem baixas, o algoritmo não considera as cores como relevantes, tornando a análise um pouco enviesada. Desta forma, pode-se concluir que para este algoritmo em específico, com imagens com cores em pouca quantidade, os resultados podem não ser os mais fiáveis. No teste seguinte (Figura 51), foi utilizada uma imagem de um vestido estendido sobre um fundo com padrão floreado com diversas cores. Deste modo, verificou-se que a rede neuronal não se conseguiu ajustar ao fundo e a segmentação não foi realizada da forma mais correta.



Figura 51 - Imagem Original (à esquerda); Segmentação (à direita)

Desta forma, visto a segmentação não estar dentro dos padrões exigidos, a detecção de cor também apresentou resultado incorretos, pois analisou *pixéis* correspondentes ao fundo como pertencentes à peça de roupa. Com isto, a percentagem correspondente às cores rejeitadas apresenta um valor alto, tornando os resultados enviesados (Figura 52). Neste caso, a solução para mitigar este problema não está diretamente relacionada com a detecção de cor, mas com a modelização do algoritmo de segmentação de imagem, nomeadamente, com o aumento de imagens do *dataset*.

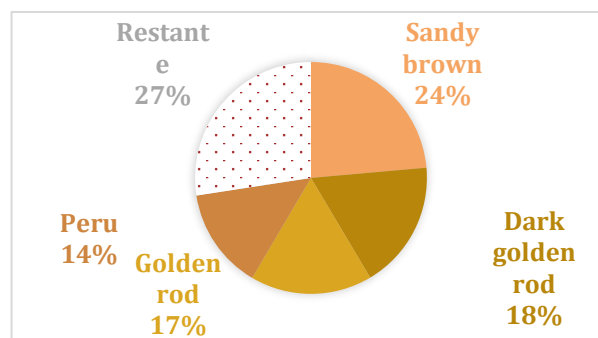


Figura 52 - Distribuição das cores

De seguida, foram realizados outros testes com duas fotos capturadas a partir da câmara do *smartphone* pessoal (Figuras 53 a 54). Desta forma, as duas fotos foram submetidas à remoção de fundo e, posteriormente, à detecção de cor, tendo a Figura 53 representado um polo com predominância de prata. No caso da Figura 54, representa umas calças com uma percentagem de cerca de 65% da cor preta.

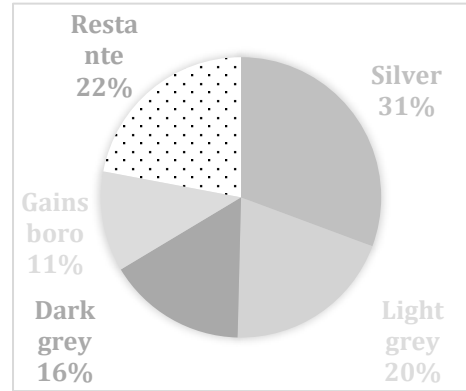
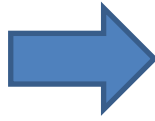


Figura 53 - Imagem Original; Distribuição de cores

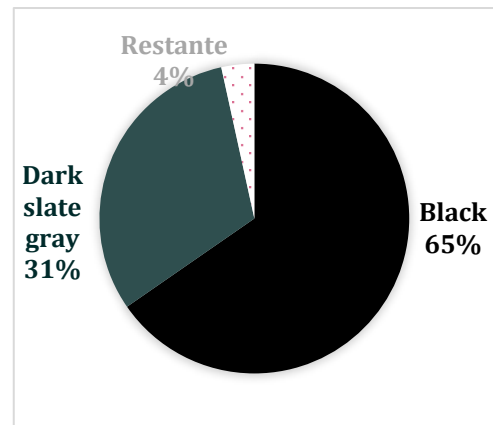
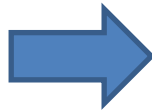


Figura 54 - Imagem Original; Distribuição de cores

Por último, foi realizado um teste com uma imagem de uma peça de roupa obtida a partir da câmara do *hardware* físico desenvolvido (Figura 55), tendo passado pelo processo total de processamento de imagem, ou seja, *stitching*, remoção de fundo e deteção de cor. Com isto, a análise percentual das cores indicou a distribuição representada na Figura 55, apresentando como cor predominante o azul cadete com uma percentagem de 40%. No entanto, existe uma percentagem de cor verde que corresponde a falhas da segmentação no limite da peça, mais concretamente uma pequena parte no cimo da peça, correspondente à separação entre o cabide e a região de interesse. Deste modo, é necessário ter em consideração que esta falha na segmentação altera um pouco os resultados, tornando-os enviesados.

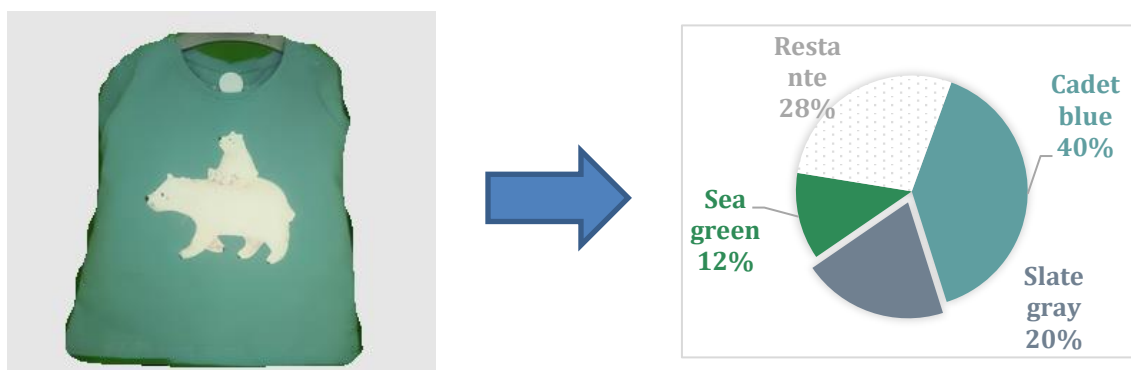


Figura 55 - Segmentação; Distribuição de cores

Deste modo, como conclusão, é possível dizer que as cores detetadas correspondem em grande parte às cores reais de cada em cada peça. No entanto, pode-se referir que o teste envolvendo o Vestido amarelo apresenta resultados enviesados, visto que a segmentação não decorreu dentro do esperado. Ainda existem, noutros testes, alguns resultados que indiciam erros que se encontram associados as falhas na deteção dos limites entre a região de interesse e o fundo. Nesse caso, uma possível solução poderá passar por um maior aperfeiçoamento do modelo da Rede Neuronal e um aumento considerável do *dataset*. Estas sugestões poderão levar a uma maior exatidão na segmentação, melhorando, deste modo, os resultados da deteção da cor.

5.2. Validação preliminar

Tal como referido anteriormente, o objetivo principal deste projeto era o desenvolvimento de um sistema mecatrónico e posterior validação deste junto da ACAPO. Deste modo, foi realizada uma validação na sede da ACAPO em Braga, com uma pessoa cega, tendo como base uma entrevista que permitisse a abordagem de questões essenciais neste tipo de sistemas. Visto ainda não existir um sistema com integração com a aplicação móvel desenvolvida e adaptada às necessidades das pessoas cegas, não foi possível uma validação completa do sistema. Com isto, numa primeira fase, foi apresentada uma visão geral do sistema desenvolvido de modo que o entrevistado tivesse conhecimento de todos pormenores do sistema bem como o seu enquadramento. De seguida, procedeu-se à realização de um questionário que permitiu reconhecer algumas dificuldades existentes no âmbito da escolha de peças de roupa bem como algumas sugestões em relação ao protótipo desenvolvido. Nesse sentido, umas das principais dificuldades mencionadas relaciona-se com a distinção não do tipo de peça de roupa, mas principalmente na distinção das cores de

cada uma das peças. No âmbito da deteção de cor, foi evidenciado pelo entrevistado que a obtenção da percentagem das cores predominantes não é algo que considere relevante, referindo, neste caso, que a indicação da localização das cores (costas, peito, braços, etc.) seria algo a ter mais em conta e mais adequado para o utilizador. Ao nível da classificação das peças de roupa, foram apresentados os tipos de peças de roupa que o sistema tem a capacidade de distinguir, tendo sido sugerido a adição de pelo menos outro tipo de peça, neste caso, a classe Casacos aos já existentes (camisa, camisola, polo, vestido, calças e calções), visto ser, na opinião do entrevistado, uma peça fundamental no dia a dia das pessoas. Já no âmbito da integração do sistema físico com a aplicação móvel, mais concretamente, no modo de mãos-livres, ou seja, quando o utilizador utiliza a câmara de *smartphone* para a captura das fotos, foi referido que existe uma grande dificuldade na captura da imagem visto ser difícil reconhecer os limites quer da peça que da câmara. Devido a este obstáculo, a tarefa terá de ser com recurso à tentativa e erro, tornando a tarefa bastante demorada. Assim, foi sugerido que para este modo de mãos-livres, fosse utilizado um tripé no *smartphone* que permitisse a fixação deste, delimitando o raio de ação da câmara até um certo ponto, facilitando, assim, a captura. Voltando ao sistema físico, foi questionada a forma como a peça escolhida deveria ser “entregue” ao utilizador, tendo sido referido que faria sentido que o sistema devolvesse a peça de roupa na mão do utilizador e não a “deixando” cair no chão por força da gravidade, visto não apresentar qualquer tipo de ajuda para o utilizador. Outra questão prendeu-se com a existência ou não de um *feedback* sonoro por parte do sistema quando o processo de escolha se encontra em andamento, tendo sido, prontamente referido que poderia existir um som crescente gradual em *bip* durante 5 a 10 segundos (caso o processo durasse esse tempo) ou um sistema de voz com indicação de início e fim do processo. Por último, tendo como objetivo a conclusão do questionário, procurou-se saber até que ponto este sistema poderia ajudar na gestão das peças. Tal como evidenciado no início, seria um sistema bastante útil para ajudar na distinção das cores das peças, pois na escolha do tipo de peça não seria algo fundamental. Relativamente à integração aplicação móvel/protótipo físico, algo importante a reter foi a indicação de que o protótipo mecatrónico e a aplicação móvel deveriam ser módulos independentes, sendo útil ao sistema se o protótipo fosse apenas um complemento à aplicação móvel, usando o modo de *kit* de mãos-livres para quem quisesse adquirir o protótipo físico. Caso não adquirisse o sistema físico, o utilizador apenas usava a aplicação móvel. Deste modo, com este questionário, foi possível, de certa forma, efetuar uma validação preliminar a fim de reconhecer certas ideias, dificuldades e sugestões por parte do entrevistado, no que se refere

a este tipo de soluções. Assim, foi evidenciado que este sistema pode ajudar numa tarefa bastante importante do cotidiano, em específico na distinção cores, tonalidades e combinações. Este sistema visa a redução do tempo na escolha, o aumento da autonomia das pessoas e diminuição da dependência de familiares ou amigos.

6. Conclusões e Trabalho Futuro

O objetivo principal do trabalho desenvolvido no âmbito desta dissertação, auxiliar a comunidade cega no seu quotidiano, foi alcançado mediante o desenvolvimento do protótipo de um sistema mecatrónico destinado a auxiliar cegos na escolha do seu vestuário. O *design* do protótipo foi a primeira parte do projeto a ser desenvolvida, tendo sido necessário encontrar no mercado um armário que possibilitasse realizar a prova de conceito num ambiente doméstico. Para o desenvolvimento do sistema foi preciso estudar os requisitos necessários ao seu funcionamento, nomeadamente no que diz respeito à iluminação, atuação e sensorização. Depois da escolha dos componentes, procedeu-se ao desenvolvimento, em módulos, das diversas zonas constituintes do sistema, nomeadamente, a leitura das etiquetas, a rotação e movimentação das peças e a captura de imagens. Para o funcionamento destes módulos, foi necessário desenvolver um algoritmo que permitisse controlar cada um dos elementos eletrónicos existentes. Com a validação e teste individual de cada um dos módulos, procedeu-se à sua integração num único sistema controlado a partir de uma interface gráfica de linha de comandos. Para obter as características das peças de roupa que são necessárias para a identificação automática foram implementados algoritmos de deteção de cor e de remoção de fundo, este último recorrendo a uma Rede Neuronal. Após a implementação dos algoritmos e tendo sido testado com diversas peças de roupa, o sistema apresentou uma taxa de exatidão entre 75% e 100% no reconhecimento dessas peças. O protótipo apresentado faz parte de um projeto de maior dimensão, em desenvolvimento, estando prevista a sua futura integração com uma aplicação móvel em parte do sistema *MyEyes*. Nesse sentido, já se encontra implementado um servidor para a gestão dos pedidos por parte do utilizador. Esta integração permitirá a replicação do guarda-roupa físico num telemóvel, possibilitando a escolha das peças de roupa a partir de um clique. Visto o protótipo desenvolvido tratar-se de uma prova de conceito, o sistema apresenta algumas limitações, nomeadamente o tamanho reduzido do armário. Esta característica permitiu apenas o uso de peças de roupa de pequeno tamanho e de um número reduzido de peças dentro do armário ao mesmo tempo. Toda a eletrónica desenvolvida foi pensada para o uso específico neste protótipo de reduzidas dimensões, tendo, por isso, que ser redimensionada e realocada aquando do desenvolvimento de um protótipo com peças de roupa em maior escala. Quando se fizer a integração do protótipo físico de maiores dimensões com o *MyEyes* [16] e [17], proceder-se-á a uma validação mais completa junto da comunidade

cega, permitindo assim o teste do sistema desenvolvido tendo em vista uma possível comercialização.

Referências Bibliográficas

1. A. Perry, "Consumers' acceptance of smart virtual closets," *J. Retail. Consum. Serv.*, vol. 33, pp. 171–177, 2016.
2. H.Q. Nguyen, A.H.L. Duong, M.D. Vu, T.Q. Dinh and H.T. Ngo "Smart Blind Stick for Visually Impaired People" in 2022 8th International Conference on the Development of Biomedical Engineering (BME8), 2022, Volume 85, pp. 145–165.
3. F.S. Apu, F.I. Joyti, M.A.U. Anik, M.W.U. Zobayer, A.K. Dey and S. Sakhawat "Text and voice braille translator for blind people" in 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 2021.
4. B. Badawi, T.N.M. Aris, M. Zolkepli, N. A. Husin and N.C. Pa "Indoor navigation system for blind people using qr code" in 2021 Journal of Theoretical and Applied Information Technology, 2021, 99 (19), pp. 4497–4510.
5. K. Tateno, N. Takagi, K. Sawai, H. Masuta and T. Motoyoshi "Method for generating captions from clothing images to support visually impaired people" in 2020 Joint 11th International Conference on Soft Computing and Intelligent Systems and 21st International Symposium on Advanced Intelligent Systems (SCIS-ISIS), 2020.
6. H. Yang, Q. Peng, Q. Gao and P.-L. Patrick Rau "Design of a clothing shopping guide website for visually impaired people" in 2015 17th International Conference on Human-Computer Interaction, 2015.
7. S. Sakthimani and C. Hemalatha "Assistive clothing pattern recognition for visually impaired people" in 2015 International Journal of Applied Engineering Research, 2015, 10 (13), pp. 11125-11132.
8. R. Alabduljabbar, "An IoT smart clothing system for the visually impaired using NFC technology," *International Journal of Sensor Networks*, vol. 38, no. 1, pp. 46–57, 2022.
9. S. J. V. Gatis Filho, J. de Assumpção Macedo, M. M. Saraiva, J. E. A. Souza, F. B. Breyer, and J. Kelner, My best shirt with the right pants: improving the outfits of visually impaired people with QR codes and NFC tags, vol. 10919 LNCS. 2018.
10. Y. Tian and S. Yuan, Clothes matching for blind and color blind people, vol. 6180 LNCS, no. PART 2. 2010.
11. Yang, S. Yuan, and Y. Tian, "Assistive Clothing Pattern Recognition for Visually Impaired People," *IEEE Trans. Human-Machine Syst. Human-Machine Syst. IEEE Trans. on, IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 2, pp. 234–243, 2014
12. J. Jarin Joe Rini and B. Thilagavathi, "Recognizing clothes patterns and colours for blind people using neural network," in *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, 2015.
13. M. Allam, I. ElShaarawy, and S. A. Farghal, Recognizing Clothing Patterns and Colors for BVI People Using Different Techniques, vol. 224. 2022.

14. X. Yang, S. Yuan, and Y. L. Tian, "Recognizing clothes patterns for blind people by confidence margin based feature combination," in MM'11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops, 2011, pp. 1097–1100.
15. A. J. Medeiros, L. Stearns, L. Findlater, C. Chen, and J. E. Froehlich, "Recognizing clothing colors and visual Textures using a finger-mounted camera: An initial investigation," in ASSETS 2017 - Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, 2017, pp. 393–394.
16. D. Rocha, V. Carvalho, E. Oliveira, J. Goncalves, and F. Azevedo, "MyEyes-automatic combination system of clothing parts to blind people: First insights," 2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH), Serious Games and Applications for Health (SeGAH), 2017 IEEE 5th International Conference on. pp. 1–5, 2017
17. D. Rocha, V. Carvalho, and E. Oliveira, "'MyEyes - Automatic Combination System of Clothing Parts to Blind People: Prototype Validation". (2017) SENSORDEVICES' 2017 Conference. Rome, Italy 10 – 14 September 2017," 2017.
18. Rocha, D., Carvalho, V., Soares, F., & Oliveira, E. (2019a). Design of a mechatronic system to combine garments for blind people: first insights. Conference: HealthyIoT 2019 - 6th EAI International Conference on IoT Technologies for HealthCareAt: Braga.
19. Website oficial do produto: <https://www.aipoly.com> (Acedido em maio de 2022)
20. Website oficial do produto: <https://www.eastersealstech.com/2013/08/13/coloritest-2000/> (Acedido em maio de 2022)
21. Website oficial do produto: <https://www.eastersealstech.com/2016/07/05/colorinos-color-identifier-light-detector/> [Acedido em maio de 2022]
22. Website oficial do produto:
<https://www.amazon.com/gp/help/customer/display.html?nodeId=202120810>
(Acedido em novembro de 2021)
23. Website oficial do produto: <https://fashiontasteapi.com/our-technology/> (Acedido em novembro de 2021)
24. Website oficial do produto: <https://smartcloset.me/> (Acedido em novembro de 2021)
25. Website oficial do produto <http://www.tailortags.com/> (Acedido em novembro de 2021)
26. K.N. Goh, Y.Y. Chen, and E. S. Lin, "Developing a smart wardrobe system," in 2011 IEEE Consumer Communications and Networking Conference (CCNC), 2011, pp. 303–307
27. Yingheng, X. U., & Yueqi, Z. (2020). Multiple Attention Mechanism Neural Network in Garment Image Segmentation. Proceedings - 2020 International Conference on Computational Science and Computational Intelligence, CSCI 2020, 1677–1683.
28. Khurana, T., Mahajan, K., Arora, C., & Rai, A. (2018). Exploiting Texture Cues for Clothing Parsing in Fashion Images. Proceedings - International Conference on Image Processing, ICIP, 2102–2106.

29. Ge, Y., Zhang, R., Wang, X., Tang, X., & Luo, P. (2019). Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June, 5332–5340.
30. Wang, J., Wan, X., Li, L., & Wang, J. (2021). An Improved DeepLab Model for Clothing Image Segmentation. 2021 IEEE 4th International Conference on Electronics and Communication Engineering, ICECE 2021, 49–54.
31. Zhang, X., Song, C., Yang, Y., Zhang, Z., Zhang, X., Wang, P., & Zou, Q. (2020). Deep learning based human body segmentation for clothing fashion classification. Proceedings - 2020 Chinese Automation Congress, CAC 2020, 7544–7549.
32. Yang, T., Shi, Y., Huang, H., & Du, F. (2021). Recognize the Silhouette Attributes of T-shirts based on Mask R-CNN. IAEAC 2021 - IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference, 106–110.
33. <https://www.ikea.com/pt/pt/p/baggebo-armario-c-porta-branco-60481204/> (Acedido em maio de 2022)
34. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
35. <https://components101.com/wireless/pn532-nfc-rfid-module>
36. <https://medium.com/jungletronics/uln2003-28bjj-48-stepper-motor-f1cc5357eff>
37. <https://www.arducam.com/product/arducam-ov5647-standard-raspberry-pi-camera-b0033/>
38. Silva, L., Rocha, D., Carvalho, V., Sena Esteves, J., & Soares, F. Automatic Wardrobe for Blind People. Conference: EAI HealthyIoT 2022 - 9th EAI International Conference on IoT Technologies for HealthCareAt: Braga.
39. <https://github.com/alexeygrigorev/clothing-dataset> (Acedido em maio 2022)
40. https://www.rapidtables.com/web/color/RGB_Color.html (Acedido em maio de 2022)