



Visão computacional para deteção de hábitos alimentares

ANA CATARINA LOPES ANTELO

outubro de 2022



Visão computacional para deteção de hábitos alimentares

Ana Catarina Lopes Antelo

Aluno nº: 1150379

**Dissertação para obtenção do Grau de
Mestre em Engenharia de Inteligência Artificial**

Orientador: Doutora Maria Goreti Carvalho Marreiros

Co-orientador: Mestre Diogo Emanuel Pereira Martinho

Júri

Presidente:

Doutor Carlos Fernando da Silva Ramos, Professor Coordenador Principal do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Vogais:

Doutora Paula Maria Marques Moura Gomes Viana, Professora Coordenadora do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Doutora Maria Goreti Carvalho Marreiros, Professora Coordenadora com Agregação do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Porto, outubro de 2022

Resumo

O excesso de peso e a obesidade são fatores comportamentais que têm vindo a causar um aumento substancial de mortes em Portugal. Estes fatores podem trazer complicações musculoesqueléticas, efeitos metabólicos como diabetes, riscos cardiovasculares, efeitos sobre a saúde mental e o aparecimento e/ou agravamento de cancro. Seguir uma dieta saudável é importante não apenas para controlar os níveis de açúcar, mas também o perfil lipídico, a tensão arterial, minimizando assim o risco cardiovascular e de complicações microvasculares. Torna-se, portanto, crucial a implementação de soluções que orientem os utilizadores a optar por opções alimentares mais benéficas à sua saúde, para que os indivíduos previnam o aparecimento de outras doenças ou exacerbações das doenças que já possam possuir. Estas soluções podem ser manuais como a contagem manual de hidratos de carbono ou digitais como as várias aplicações móveis existentes no mercado que permitem a monitorização de doenças e o controlo nutricional. Atualmente, grande parte da sociedade possui um dispositivo móvel com capacidade de tirar fotografias e cada vez mais os telemóveis são usados como assistentes pessoais, ajudando o ser humano a ser mais eficaz nas suas tarefas diárias. Estes dispositivos representam um recurso computacional versátil, com grande capacidade de deteção e inferência. As técnicas de *machine learning* aplicadas nas câmaras dos telemóveis permitem a estabilização de imagem, tradução de texto automática, deteção de objetos, reconhecimento de rostos, entre outros. Os próprios sensores dos telemóveis são cada vez mais complexos e podem ser usados para detetar movimentos e padrões, inferir níveis de stress e emocionais do utilizador, reconhecimento de lugares, estimativa de profundidade dos elementos numa fotografia, e assim por diante. Estes sensores possibilitam a extração de dados sem que o utilizador tenha de realizar uma tarefa específica. O objetivo desta tese foi implementar e estudar sistemas inovadores que, através de visão computacional, auxiliem na tarefa de controlo nutricional e que permitam a monitorização de doenças. Neste âmbito, desenvolveu-se um sistema de reconhecimento de alimentos utilizando Detectron2 com o modelo PointRend que, com o auxílio de um modelo de Regressão Linear capaz de prever uma estimativa do peso dos alimentos presentes em uma imagem, permitiu que o controlo nutricional se tornasse em uma tarefa simples. A solução proposta nesta dissertação permitirá que o utilizador poupe tempo e esforço, e que realize decisões alimentares mais conscientes. Além disso, esta solução também estará preparada para auxiliar pacientes diabéticos, indicando, por exemplo, as unidades de insulina que deve injetar, tendo em conta a refeição que irá ingerir.

Palavras-chave: Visão Computacional, Reconhecimento de alimentos, Estimativa de peso, Saúde, Diabetes.

Abstract

Overweight and obesity are behavioral factors that have been causing a substantial increase in deaths in Portugal. These factors can bring musculoskeletal complications, metabolic effects such as diabetes, cardiovascular risks, effects on mental health, and the onset and/or worsening of cancer. Following a healthy diet is important not only to control sugar levels but also the lipid profile, and blood pressure, thus minimizing the risk of cardiovascular and microvascular complications. It is therefore crucial to implement solutions that guide users to choose food options that are more beneficial to their health so that individuals prevent the onset of other diseases or exacerbations of diseases they may already have. These solutions can be manual, such as the manual counting of carbohydrates, or digital, such as the multiple mobile applications on the market that allow disease monitoring and nutritional control. Currently, a large part of society has a mobile device capable of taking pictures and mobile phones are increasingly used as personal assistants, helping human beings to be more effective in their daily tasks. These devices represent a versatile computing resource, with great detection and inference capabilities. Machine learning techniques applied to mobile phone cameras allow image stabilization, automatic text translation, object detection, and face recognition, among others. Mobile phone sensors themselves are increasingly complex and can be used to detect movements and patterns, infer user stress and emotional levels, place recognition, estimate the depth of elements in a photograph, and so on. These sensors make it possible to extract data without the user having to perform a specific task. This thesis objective was to implement and study innovative systems that, through computer vision, help in nutritional control and allow disease monitoring. In this context, a food recognition system was developed using Detectron2 with the PointRend model which, with the aid of a Linear Regression model capable of predicting an estimate of the weight of the food present in an image, allowed nutritional control to become a simple task. The solution proposed in this dissertation will allow the user to save time and effort, and to make more conscious food decisions. In addition, this solution will also be prepared to help diabetic patients, indicating, for example, the units of insulin that must be injected, considering the meal that will be ingested.

Keywords: Computer Vision, Food recognition, Weight estimation, Health, Diabetes.

Agradecimentos

Em primeiro lugar, quero agradecer a todos os que contribuíram para que fosse possível realizar este trabalho.

Aos meus orientadores, Professora Doutora Goreti Marreiros e Professor Diogo Martinho, pela força e motivação, pelas suas orientações e apoio que me prestaram durante o desenvolvimento deste projeto.

Um agradecimento especial aos meus amigos pelas suas ideias e sugestões, pela paciência infindável e carinho imensurável.

À minha mãe, pilar da minha vida, que sempre me apoiou e incentivou em todos os desafios a que me propus. Pelo tempo que não lhe dediquei, pela formação que me permitiu adquirir e por toda a ajuda incondicional.

Ao meu namorado, que sofreu com as minhas indisponibilidades e impaciências, pelo seu carinho, companheirismo e compreensão.

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Problema	3
1.3	Objetivos	6
1.4	Resultados Esperados	7
1.5	Estrutura do Documento	7
2	Contextualização	9
2.1	Projeto FoodFriend	9
2.2	Visão Computacional	11
3	Revisão Literária	13
3.1	Sistema de Monitorização	13
3.1.1	Glucose Buddy	13
3.1.2	Diabetes:M	14
3.1.3	Diabetes Connect	15
3.1.4	MySugr	15
3.1.5	DiabTrend	15
3.1.6	MyFitnessPal	16
3.1.7	Calorie Mama	17
3.1.8	SnapCalorie	18
3.2	Abordagens Inteligentes	19
3.2.1	Discussão	19
3.2.2	Metodologia	22
3.2.3	Sistemas de Monitorização da Diabetes	25
3.2.4	Sistemas de Reconhecimento e Estimativa de Peso de Alimentos	26
4	Enquadramento Teórico	39
4.1	Conceitos	39
4.1.1	Machine Learning	39
4.1.2	Dados	40
4.1.3	Tipos de Aprendizagem	41
4.1.4	Algoritmos e Modelos	42
4.1.5	Ferramentas	55
4.1.6	Métodos	57
5	Abordagens Consideradas	59
5.1	SVM, CNN e MobileNetV2	59
5.1.1	Conjunto de Dados	59
5.1.2	Aplicação dos Algoritmos de Aprendizagem	61
5.1.3	Resultados	63

5.2	Reconstrução 3D	67
5.2.1	Aplicação da Reconstrução 3D	68
5.3	Detectron2	70
5.3.1	Conjunto de Dados	70
5.3.2	Aplicação do Detectron2	70
5.3.3	Resultados.....	74
5.4	Regressão Linear	76
5.4.1	Conjunto de Dados	76
5.4.2	Aplicação da Regressão Linear	76
5.4.3	Resultados.....	77
5.5	Discussão das Abordagens.....	79
6	Modelação da Solução	85
6.1	Análise de Requisitos	85
6.1.1	Atores Principais	85
6.1.2	Requisitos Funcionais	85
6.2	Design.....	87
6.3	Arquitetura.....	89
7	Implementação da Solução	91
8	Experimentação e Avaliação	95
8.1	1º Caso de Estudo: Emagrecimento e Tonificação do Corpo	95
8.1.1	Recolha de dados.....	96
8.1.2	Avaliação da solução.....	96
8.1.3	Discussão de resultados	97
8.2	2º Caso de Estudo: Paciente Diabético.....	98
8.2.1	Recolha de dados.....	99
8.2.2	Avaliação da solução.....	99
8.2.3	Discussão de resultados	101
9	Conclusões	103
9.1	Síntese	103
9.2	Objetivos Realizados	104
9.3	Resultados Alcançados	105
9.4	Trabalho Futuro	105

Lista de Figuras

Figura 1 - Fatores de risco comportamentais e ambientais geradores de mortes	3
Figura 2 - Nutri-score	11
Figura 3 - Descodificador de rótulos	11
Figura 4 - Etapas de um sistema de Visão Computacional típico	12
Figura 5 - Ecrãs da aplicação Glucose Buddy	14
Figura 6 - Ecrãs da aplicação DiabTrend	16
Figura 7 - Reconhecimento de alimentos MyFitnessPal	17
Figura 8 - Ecrãs da aplicação Calorie Mama.....	18
Figura 9 - Ecrãs da aplicação SnapCalorie	19
Figura 10 - Processo de seleção de artigos	25
Figura 11 - Recomendação de refeição com calorias e a quantidade da porção	27
Figura 12 - Testes ao modelo de reconhecimento de alimentos para pratos locais e continentais.....	27
Figura 13 - Reconhecimento de alimentos usando objeto de referência.....	28
Figura 14 - Técnicas de extração de pontos-chave: detetor SIFT, amostragem aleatória e amostragem densa.....	29
Figura 15 - <i>Labeling</i> utilizando a técnica <i>pixel-dense labeling a)</i> e segmentação das classes de objetos alimentares b)	30
Figura 16 - Detecção da classe de alimentos e obtenção das calorias de cada objeto	30
Figura 17 - Resultados da deteção da segmentação de alimentos e cálculo de calorias	31
Figura 18 - Etapas do processamento da imagem: imagem original, deteção de placa, segmentação de alimentos, reconhecimento de alimentos e reconstrução 3D	32
Figura 19 - Diagrama em blocos do reconhecedor de alimentos	33
Figura 20 - Análise de imagens e máscara após o pré-processamento	34
Figura 21 - Output da máscara.....	34
Figura 22 - Imagem antes e depois da máscara R-CNN e previsão de calorias	35
Figura 23 - Reconstrução 3D de pares de imagens.....	36
Figura 24 - Processo de reconstrução 3D.....	36
Figura 25 - Projeção do centro ótico, imagem e objeto no sistema de coordenadas xyz	37
Figura 26 – Identificação do hiperplano ideal e dos vetores de suporte.....	43
Figura 27 - Exemplos da influência do C nos limites de decisão.....	43
Figura 28 - Exemplos da influência do <i>kernel</i> nos limites de decisão	44
Figura 29 - Exemplos da influência do <i>gamma</i> nos limites de decisão.....	44
Figura 30 - Representação do funcionamento de um nó da <i>Neural Network</i>	45
Figura 31 - Representação do funcionamento de uma NN com 2 camadas ocultas	46
Figura 32 - Camadas que compõe as CNN	46
Figura 33 - Operação de convolução.....	47
Figura 34 - Modo como um computador vê uma imagem	47
Figura 35 - Arquitetura da MobileNetV2	49
Figura 36 - Algumas funcionalidades do Detectron2.....	50

Figura 37 – Segmentação de instância com PointRend	50
Figura 38 - Fluxo de processo do modelo PointRend.....	51
Figura 39 - Exemplos de resultados do Objectron	52
Figura 40 - Imagem e mapa de profundidade correspondente obtido pelo MiDaS	53
Figura 41 – Point Cloud	53
Figura 42 – Categorias utilizadas no caso de estudo.....	60
Figura 43 - Demonstração gráfica que com 500 componentes capta 90% da variância	61
Figura 44 – Exemplos de imagens e a sua reconstrução a partir dos 500 componentes	61
Figura 45 – Matriz de confusão SVM.....	64
Figura 46 – Matriz de confusão CNN.....	64
Figura 47 - Matriz de confusão MobileNetV2	65
Figura 48 - Exemplos de imagens com o mapa de profundidade correspondente	68
Figura 49 - Imagem com escala de cinzas e imagem de profundidade.....	69
Figura 50 - Point Cloud	69
Figura 51 - Formato COCO.....	71
Figura 52 - Treino do modelo	72
Figura 53 – Alguns resultados da aplicação do modelo PointRend	73
Figura 54 – Métricas disponibilizadas pelo COCOEvaluator	74
Figura 55 - Resultados da avaliação do modelo Detectron2.....	75
Figura 56 - Exemplo da aplicação do método get_dummies	77
Figura 57 – Exemplos de imagens com respetiva previsão CNN.....	81
Figura 58 - Imagem real para posterior reconstrução 3D	82
Figura 59 - Point Cloud obtida após a Reconstrução 3D	82
Figura 60 - Perceção de imagem	84
Figura 61 - Diagrama de casos de uso	86
Figura 62 - Diagrama de sequência simplificado do sistema	87
Figura 63 - Diagrama de sequência do sistema.....	88
Figura 64 - Diagrama de componentes	89
Figura 65 - Arquitetura da solução final.....	90
Figura 66 - Segmentação de instância resultante da aplicação do PointRend	91
Figura 67 - Alimentos pesados e o seu peso real	92
Figura 68 - Exemplo do ficheiro json com informação nutricional	92
Figura 69 - Valores nutricionais dos alimentos baseados no peso estimado	93
Figura 70 - Cálculo da quantidade de insulina	93
Figura 71 - Resultados do cálculo da insulina.....	93
Figura 72 - Interface gráfica da aplicação desenvolvida	94
Figura 73 - Refeição do indivíduo do 1º caso de estudo e feedback do sistema	96
Figura 74 - Resultados aplicação MyFitnessPal.....	97
Figura 75 - Primeira refeição do 2º caso de estudo e valores nutricionais.....	99
Figura 76 - Segunda refeição do 2º caso de estudo e valores nutricionais.....	100
Figura 77 - Valor de insulina para uma refeição de arroz com frango	100
Figura 78 - Valor de insulina para uma refeição de massa à bolonhesa	101
Figura 79 - Aplicação com o feedback das duas refeições	101

Figura 80 - Reconhecimento de alimentos da aplicação DiabTrend..... 102

Lista de Tabelas

Tabela 1 - Comparação entre os sistemas de monitorização	19
Tabela 2 - Comparação entre os sistemas de reconhecimento de alimentos	21
Tabela 3 - Questões de pesquisa.....	23
Tabela 4 - Bases de dados eletrónicas.....	23
Tabela 5 - Palavras de pesquisa.....	23
Tabela 6 - Critérios de inclusão	24
Tabela 7 - Critérios de exclusão.....	24
Tabela 8 – Resultado do processo de otimização de hiperparâmetros SVM.....	62
Tabela 9 - Resultado do processo de otimização de hiperparâmetros CNN.....	62
Tabela 10 - Valores usados na criação do modelo MobileNetV2	62
Tabela 11 - Valores usados na compilação do modelo MobileNetV2.....	63
Tabela 12 – Resultados da matriz de confusão SVM	66
Tabela 13 – Resultados da matriz de confusão CNN.....	66
Tabela 14 – Resultados da matriz de confusão MobileNetV2	67
Tabela 15 - Resultados da avaliação da segmentação	75
Tabela 16 - Resultados da segmentação por categoria	75
Tabela 17 - Relação entre o peso estimado o peso real dos alimentos.....	92
Tabela 18 – Relação entre valores reais e estimados	94
Tabela 19 - Peso dos alimentos do 1º caso de estudo	97
Tabela 20 – Peso dos alimentos do 2º caso de estudo	100
Tabela 21 – Objetivos realizados.....	104

Acrónimos e Símbolos

Lista de Acrónimos

AP	Average Precision
BoF	<i>Bag-of-Features</i>
CNN	<i>Convolutional Neural Networks</i>
GECAD	Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e o Desenvolvimento
HC	Hidratos de carbono
IG	Índice Glicémico
IoU	<i>Intersection over Union</i>
ISEP	<i>Instituto Superior de Engenharia do Porto</i>
KNN	<i>K-Nearest Neighbor</i>
MAE	Erro Médio Absoluto
mAP	<i>Mean Average Precision</i>
ML	<i>Machine Learning</i>
MSE	Erro Quadrático Médio
NN	<i>Neural Network</i>
PCA	<i>Principal Component Analysis</i>
RF	<i>Random Forest</i>
RMSE	Raiz do Erro Quadrático Médio
SVM	<i>Support Vector Machines</i>
UC	Caso de Uso

Lista de Símbolos

γ	Gamma
----------	-------

1 Introdução

Neste capítulo é feita uma breve apresentação do âmbito do presente trabalho, descrevendo o contexto e problema, enunciando, de seguida, os seus objetivos e resultados esperados. É também descrita a estrutura do presente documento.

1.1 Enquadramento

O conceito de qualidade de vida é muito abrangente, compreendendo não só a saúde física como o estado psicológico, o nível de independência, as relações sociais em casa, na escola e no trabalho, e até a relação com o meio ambiente. A qualidade de vida pode ser definida como a satisfação do indivíduo em relação ao seu dia a dia. A saúde e qualidade de vida são termos indissociáveis, uma vez que a saúde contribui para melhorar a qualidade de vida dos indivíduos e esta é fundamental para que um indivíduo ou comunidade tenha saúde.

Pequenas mudanças de hábitos podem estimular e conduzir à felicidade e, ao mesmo tempo, melhorar a condição de saúde de um indivíduo. Ter uma boa alimentação e praticar exercício físico são fatores essenciais que podem melhorar exponencialmente a vida de uma pessoa e evitar que problemas de saúde se agravem. A ingestão desequilibrada de alimentos é um fator relevante em muitos problemas de saúde, que vão desde a desnutrição e obesidade, diabetes e cancro.

Reduzir os riscos de doenças relacionadas à dieta é uma tarefa difícil devido às diferentes necessidades nutricionais de cada paciente. Por essa razão, torna-se vital a adoção de estratégias que permitam aos pacientes gerir, de uma forma mais autónoma, o seu estado de saúde, de modo a prevenir ou, pelo menos, controlar determinadas doenças [1–4]. Nesse âmbito, é fundamental a utilização dos avanços tecnológicos e de dispositivos que estão cada vez mais presentes no dia a dia de cada um, para automatizar processos do quotidiano e assim, dar apoio ao utilizador nessa gestão.

Os telemóveis são progressivamente usados como assistentes pessoais, ajudando o ser humano a ser mais eficaz nas suas tarefas do dia-a-dia. Estes dispositivos são capazes de processar grandes quantidades de dados e utilizam técnicas de *machine learning* capazes de detetar e reconhecer emoções humanas, objetos, rostos e alimentos, de traduzir e transcrever textos em tempo real, de classificar áudio, e muito mais [5].

Os *smartphones* têm mostrado uma evolução notável, integrando uma diversa quantidade de sensores para diferentes aplicações que, conjuntamente, estão a permitir o desenvolvimento de novas aplicações numa grande variedade de domínios como a saúde, as redes sociais, a segurança e os transportes [6, 7].

Estes dispositivos possuem sensores de localização, movimento, visuais e áudio, para deteção de campos magnéticos, luz, proximidade, pressão atmosférica, temperatura e humidade. Os sensores de algumas câmaras dos telemóveis emitem um feixe de luz infravermelha, invisível aos olhos do ser humano, para medir distâncias. Esta conceção de distância entre planos é normalmente chamada de campo de profundidade e é usado em fotografia como o popular *Modo Retrato*, mas também é aplicável noutras situações como o reconhecimento de objetos, animais, rostos e comida. Com a perceção de distância e profundidade, o telemóvel tem a capacidade de construir uma imagem tridimensional mais precisa do item/pessoa/alimento para o sistema disponibilize feedback mais confiável.

Têm surgido várias soluções com o objetivo de inovar e melhorar processos, como a monitorização e controlo de doenças, como o projeto FoodFriend, apresentado na secção 2.1. O projeto FoodFriend consiste no desenvolvimento de uma aplicação móvel para o acompanhamento e monitorização remota de pacientes que requerem alimentação por sonda e cuidado nutricional para doenças crónicas, entre as quais se destaca a doença que está por base do trabalho aqui proposto, a diabetes. Para além da aplicação móvel, o sistema proposto pelo FoodFriend é composto por diversos módulos (apresentados detalhadamente na secção 2.1), que disponibilizam funcionalidades como medição automática da ingestão de alimentos, disponibilização de feedback personalizado com base na nutrição, entre outras.

Sendo a diabetes uma das áreas de atuação do projeto FoodFriend, este visa naturalmente o desenvolvimento de tecnologias capazes de:

- Melhorar a qualidade de vida dos pacientes diabéticos;
- Controlar a doença (diabetes);
- Criação e validação de tecnologias para *self-monitoring* e gestão de pacientes com diabetes tipo 2 (com base na ingestão alimentar e monitorização do estilo de vida);

O Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e o Desenvolvimento (GECAD) é um dos membros que compõem o consórcio do FoodFriend. O GECAD é uma unidade de I&D com sede no Instituto Superior de Engenharia do Instituto Politécnico do Porto (ISEP-IPP) que se dedica à promoção e desenvolvimento de investigação

científica no domínio da Inteligência Artificial, Sistemas baseados em Conhecimento e Sistemas de Apoio à Decisão. Esta unidade é coordenada pela Doutora Maria Goreti Carvalho Marreiros, integra 50 investigadores, sendo 30 doutorados. Constitui a maior unidade de I&D de todo o subsistema do Ensino Superior Politécnico, sendo reconhecida pela Fundação para a Ciência e a Tecnologia [8].

O GECAD atua em áreas como gestão de emergências, sistemas de produção, sistemas de educação e tutoria inteligente, sistemas inclusivos, genética, saúde, mercados e sistemas de energia. São abordadas outras áreas e paradigmas como a web semântica, inteligência ambiental, sistemas ciberfísicos, redes e cidades inteligentes.

A contribuição da pesquisa do GECAD para o estado da arte nas áreas em que atua pode ser organizada em três vertentes: parte do trabalho é dedicada ao desenvolvimento de metodologias e técnicas básicas que podem ser aplicadas em um amplo conjunto de campos; parte do trabalho é dedicada a campos de finalidade específica; e, mais recentemente, alguns trabalhos foram dedicados a aspetos como transferência de tecnologia e inovação.

As metodologias e tecnologias básicas utilizadas pelo GECAD abrangem aspetos que vão desde o hardware até ao software inteligente útil para o desenvolvimento de Sistemas Inteligentes de Engenharia. Este hardware inclui todos os dispositivos de automação úteis para deteção, monitorização e controlo de Sistemas Inteligentes de Engenharia. No entanto, o GECAD concentra a atenção no uso inteligente desse hardware, portanto, a pesquisa em sistemas inteligentes, sistemas de suporte à decisão e sistemas baseados em conhecimento são o foco principal do GECAD. O GECAD é reconhecido internacionalmente, alcançando pontuações notáveis em revistas científicas de alto nível.

1.2 Problema

Segundo a Comissão Europeia, o perfil de saúde de Portugal no ano 2021 mostrou que um terço das mortes registadas em 2019 podem ser atribuídas a fatores de risco comportamentais, nomeadamente ao tabagismo, aos riscos alimentares, ao consumo de álcool, e ao baixo nível de exercício físico [9].



Figura 1 - Fatores de risco comportamentais e ambientais geradores de mortes [9]

O excesso de peso e a obesidade constituem um problema de saúde pública cada vez maior nos adultos e jovens. A Organização Mundial da Saúde publicou, em maio de 2022, o relatório sobre o estado da obesidade na região europeia [10]. O excesso de peso (incluindo a obesidade) causam mais de 1.2 milhões de mortes em toda a região europeia todos os anos, correspondendo a mais de 13% do total de mortes. Além disso, são os principais fatores comportamentais que aumentam o risco de deficiência. Em Portugal, o excesso de peso e a obesidade contribuíram para cerca de 9% da mortalidade no ano de 2019.

A obesidade tem efeitos desfavoráveis na saúde provenientes do aumento de peso, como complicações musculoesqueléticas, efeitos metabólicos como diabetes tipo 2, riscos cardiovasculares, e efeitos sobre a saúde mental. A obesidade também é considerada a causa de, pelo menos, 13 tipos diferentes de cancro.

A Organização Mundial da Saúde [11] menciona que a diabetes é uma das principais causas de morte no mundo. Cerca de 422 milhões de pessoas em todo o mundo têm diabetes e 1.5 milhões de mortes são atribuídas diretamente a esta doença todos os anos. Tanto o número de casos quanto a prevalência de diabetes têm aumentado constantemente nas últimas décadas. A diabetes é uma doença metabólica crónica caracterizada por níveis elevados de açúcar no sangue, o que leva, ao longo do tempo, a complicações no organismo que podem afetar o coração, vasos sanguíneos, olhos, rins e nervos.

Existem quatro tipos principais de diabetes [12, 13]:

- Diabetes tipo 1: é o tipo menos comum e afeta principalmente crianças e jovens. É considerada uma doença autoimune, pois o próprio sistema ataca e destrói as células do pâncreas, onde é produzida a insulina. Como a insulina não é produzida, a glucose não é transportada para as células e acaba por se acumular no sangue. Pacientes com diabetes tipo 1 necessitam de tratamento com insulina;
- Diabetes tipo 2: é o tipo mais comum e acontece quando o corpo cria uma resistência à insulina, normalmente devido a maus hábitos alimentares. Esta resistência diminui a ação da insulina no corpo, fazendo com que a glucose se acumule no sangue. Os fatores de risco deste tipo são a hereditariedade, obesidade, sedentarismo e a predisposição genética;
- Diabetes gestacional: como o nome indica, apenas pode acontecer durante a gravidez e está relacionado com a produção, pela placenta, de outros hormônios que bloqueiam a ação da insulina;
- Pré-diabetes: ocorre quando o nível de açúcar no sangue está mais alto do que o normal, mas ainda não está alto o suficiente para um diagnóstico de diabetes tipo 2.

A diabetes ainda pode ser dividida noutros tipos mais raros, mas que não serão aqui abordados. Os sintomas mais comuns desta doença são a sensação de sede exagerada, o aumento da fome, vontade frequente para urinar, boca seca, cansaço fácil, perda de peso súbita e alterações da visão. Além destes sintomas, os homens com diabetes podem ter uma diminuição do desejo

sexual, disfunção erétil e pouca força muscular, já as mulheres com diabetes podem ter sintomas como infecções urinárias, infecções fúngicas, pele seca e com comichão [12].

Apesar de não ter cura, esta doença pode ser controlada através de [14]:

- Dieta equilibrada e personalizada;
- Prática de atividade física moderada, mas regular;
- Perda de peso;
- Injeções de insulina (diabetes tipo 1);
- Injeções de insulina e/ou outros medicamentos (diabetes tipo 2).

Quando a diabetes não é tratada adequadamente, os níveis de açúcar no sangue podem ficar elevados por muito tempo e podem causar danos em vários órgãos. As principais complicações são o aparecimento de doenças cardiovasculares, neuropatia, retinopatia, surdez, depressão, disfunção sexual, doenças de foro dentário e problemas nos pés, como úlceras, deformações ou perda de sensibilidade[12, 13].

Esta doença é impactante na qualidade de vida dos doentes, sendo que o acesso a tratamento acessível, incluindo insulina, é fundamental para sua sobrevivência. De modo a ser possível controlar a doença, a utilização de aplicações de acompanhamento e monitorização de sinais vitais têm vindo a ser utilizadas mais frequentemente.

Sendo a diabetes uma doença metabólica crónica caracterizada pela alteração da regulação da glicemia, e sendo os HC presentes nos alimentos um dos nutrientes essenciais na alimentação e que mais influenciam a glicemia, a terapia nutricional assume particular importância na gestão desta doença [15].

Diversos pacientes com diabetes utilizam o método de contagem de HC, que foca os HC como o principal nutriente que afeta a resposta glicémica pós-prandial. A quantidade e o tipo ou a fonte de HC ingeridos são os maiores determinantes dos níveis de glicose pós-prandial. Diferentes fontes de HC podem originar diferentes respostas glicémicas, no entanto, considera-se que o total de HC ingeridos têm prioridade na avaliação do impacto destes na glicemia e na quantidade de insulina a administrar para metabolizar adequadamente a glicose proveniente desses HC [15].

O método de contagem de HC baseia-se na estimativa da quantidade de carboidratos presente numa determinada refeição. No entanto, esta estimativa é efetuada por métodos indiretos pelo que, mesmo sendo realizada de forma rigorosa, o cálculo não é exato. Além disso, tanto os lípidos como as proteínas podem ter impacto no controlo glicémico, o que dificulta o controlo nutricional para os pacientes, pois ainda não existem evidências sobre um algoritmo ideal que agregue estas particularidades.

Outro fator alimentar que deve ser considerado é o Índice Glicémico (IG), que permite diferenciar os alimentos fornecedores de HC de acordo com o seu efeito na glicemia pós-prandial. Alguns estudos demonstram que o consumo de alimentos de baixo índice glicémico contribuem para o controlo glicémico, redução de marcadores inflamatórios e prevenção de diabetes tipo 2 [16–21].

O índice glicémico demonstra a velocidade com a qual o açúcar presente nos alimentos chega à corrente sanguínea e altera os níveis de glicemia. Quanto maior o índice glicémico de um alimento, menos a sua qualidade do ponto de vista nutricional, dado que os hidratos de carbono simples são facilmente digeríveis e transformam-se em açúcar mais rapidamente no sangue, reduzindo o tempo de saciedade [22–24].

O índice glicémico está dividido em três categorias:

- Baixo IG: valor menor ou igual a 55;
- Médio IG: valores entre 56 e 69;
- Alto IG: valor maior ou igual a 70;

Sendo o controlo nutricional uma das ferramentas de autogestão da doença e tendo em conta que este processo exige um esforço inicial de aprendizagem e adaptação, conclui-se que é imperativa a criação de soluções que sejam capazes de agilizar este método. A partir de soluções que detetem hábitos alimentares e disponibilizem informações nutricionais relevantes para os pacientes diabéticos, os utilizadores terão maior flexibilidade na escolha de alimentos, nas quantidades, nos horários e número de refeições [25–28].

1.3 Objetivos

O trabalho proposto insere-se no projeto FoodFriend que está ligado à área da saúde no tópico do acompanhamento de pacientes diabéticos no seu dia-a-dia através da análise dos hábitos diários (nomeadamente a alimentação e atividade física) e na promoção de melhores comportamentos de forma a permitir uma gestão mais controlada e correta da doença.

Este projeto consiste no desenvolvimento de um módulo de reconhecimento de hábitos alimentares de pacientes diabéticos, que serão incluídos no projeto FoodFriend, como mencionado anteriormente.

Este módulo disponibiliza aos seus utilizadores:

- Reconhecimento de hábitos alimentares;
- Estimativa do peso dos alimentos identificados;
- Cálculo nutricional de itens alimentares;

- Cálculo de insulina rápida (caso o utilizador seja paciente diabético);
- Visualização de indicadores que demonstrem a qualidade das refeições ingeridas;
- Disponibilização do feedback da aplicação através de uma interface gráfica.

1.4 Resultados Esperados

Com o desenvolvimento do trabalho proposto é esperado que os objetivos definidos na secção 1.3 sejam cumpridos, ou seja, é expectável que seja implementado um sistema que incorpore técnicas de visão computacional no que diz respeito à gestão nutricional com foco na doença da diabetes.

Com este sistema, pretende-se que a tarefa de controlo nutricional com que os pacientes diabéticos têm de lidar diariamente, se torne numa ação simples, rápida e mais intuitiva. Com a disponibilização de indicadores indiquem se determinadas refeições são ou não adequadas para o utilizador, e através do cálculo nutricional a partir do reconhecimento de alimentos, o utilizador poderá ter uma vida mais orientada e regrada.

1.5 Estrutura do Documento

O presente documento está organizado em oito capítulos. O documento é iniciado por este capítulo introdutório, onde é feita uma breve apresentação do contexto e, de seguida, são expostos os objetivos a concretizar e os respetivos resultados esperados.

No segundo capítulo, Contextualização, é aprofundado o âmbito do trabalho proposto, apresentando o projeto FoodFriend onde o projeto desta dissertação será incluído, expondo, de seguida, os conceitos abrangidos pelo presente trabalho.

No terceiro capítulo, Revisão Literária, são exploradas as várias soluções de sistemas de monitorização e reconhecimento de alimentos, e abordagens inteligentes presentes na literatura e no mercado.

No quarto capítulo, Enquadramento Teórico, são abordados alguns conceitos fundamentais para a implementação da solução final, são apresentadas as diferentes tecnologias de *machine learning*, as ferramentas e métodos utilizados durante o desenvolvimento deste projeto.

No quinto capítulo, Abordagens Consideradas, é descrita a implementação das diferentes abordagens adotadas, bem como o conjunto de dados utilizado para cada alternativa e os resultados obtidos. Por fim, é realizada uma discussão sobre as tecnologias utilizadas e a justificação das opções tomadas.

No sexto capítulo, Modelação da Solução, é realizada uma análise de requisitos, onde são apresentados os atores principais e os requisitos funcionais da solução proposta. Depois, são demonstrados o design e a arquitetura do sistema proposto.

No sétimo capítulo, Implementação da solução, é descrito o processo de aplicação das tecnologias adotadas e são mostrados os resultados obtidos.

No oitavo capítulo, Experimentação e Avaliação, é demonstrado o processo utilizado para a avaliação da solução final. Nesse âmbito, são apresentados dois casos de estudo com indivíduos reais e que facilmente poderiam ser utilizadores da solução. De seguida, os resultados são avaliados e é realizada uma discussão desses resultados.

No nono capítulo, Conclusões, é efetuada uma síntese desta dissertação, são apresentados os resultados obtidos em contraste com os objetivos inicialmente definidos, e é definido o trabalho futuro a ser realizado.

2 Contextualização

No presente capítulo será enquadrado o trabalho proposto, apresentando o projeto onde este está incluído, seguindo-se de uma exposição de dois conceitos associados ao trabalho: FoodFiend e visão computacional.

2.1 Projeto FoodFriend

O FoodFriend é um projeto enquadrado na área de saúde móvel, financiado pelo Itea3 [29], que conta com um consórcio de quatro parceiros nacionais, sendo estes:

- Faculdade de Medicina da Universidade do Porto (FMUP);
- Instituto Superior de Engenharia do Porto (ISEP), através do GECAD;
- FYI Digital Innovation;
- CLOO Behavioral Insights Unit.

De modo a potenciar o sucesso deste projeto, o consórcio possui experiência nas áreas abrangidas por este, como medicina (FMUP), tecnologias de informação (GECAD), desenvolvimento de produtos digitais (FYI) e definição de caso de uso de diabetes e definição de requisitos focada principalmente em fornecer dados baseados em evidências sobre os processos psicológicos, preconceitos cognitivos e comportamentos dos utilizadores (CLOO).

FoodFriend está focado em dois aspetos de doenças relacionadas à dieta: prevenção da desnutrição para pacientes que requerem alimentação por sonda e cuidado nutricional para doenças crónicas, como a diabetes. Este projeto tem como principal objetivo melhorar a qualidade de vida e realizar o controlo da diabetes pelo paciente diabético.

A diabetes é uma doença metabólica crónica caracterizada pelo aumento dos níveis de açúcar/glucose no sangue. Isto acontece porque o hormônio insulina que transporta a glucose

do sangue para o interior das células, não é produzida ou não funciona corretamente, fazendo com que o açúcar se vá acumulando no sangue ao invés de ser gasto nas células do corpo [13].

Como referido na secção 1.2, a diabetes é uma doença que tem vindo a aumentar drasticamente. Os pacientes diabéticos necessitam de dietas específicas para controlar o seu estado de saúde e a hospitalização destes pacientes pode ser desencadeada por descompensação da doença e/ou comorbilidades associadas.

Existem fatores de risco que são modificáveis, como a obesidade. Para que seja possível uma mudança, é necessário controlo da ingestão de alimentos e uma participação ativa dos pacientes no processo de gestão de doenças. Nesse âmbito, o FoodFriend pretende:

- Medir automaticamente a ingestão de alimentos;
- Transformá-lo em feedback personalizado e acionável, com base para a nutrição;
- Mínima necessidade de inserção de dados;
- Análises em tempo real;
- Fornecer feedback que combine técnicas de recomendação e visualização;
- Disponibilizar aplicações móveis interativas e fáceis de utilizar e *smartwatches* direcionados para implementações de soluções tecnológicas.

Com este projeto, é expectável a obtenção de uma solução de *coaching* baseada em evidências para auto monitorização e gestão da diabetes tipo 2, com base na ingestão de alimentos e monitorização do estilo de vida. Para atingir estes objetivos, foram definidas algumas metas a atingir, como:

- Estimular o envolvimento contínuo do utilizador com a aplicação, para que haja um fornecimento regular sobre métricas de saúde do utilizador, permitindo que tanto o indivíduo quanto o profissional de saúde acompanhem o progresso;
- Incentivar a mudança de comportamento no estilo de vida, tanto na dieta quando na prática de exercício físico, para aumentar o bem-estar do indivíduo.

O conjunto de ferramentas utilizadas combina hardware na forma de sensores e software no formato de aplicação ou de um website. Este projeto é complementado por um conjunto de análises e técnicas de fusão de dados (transformar dados dos sensores em informações sobre o comportamento da ingestão de alimentos) que são posteriormente disponibilizados na plataforma online, no formato de relatórios de ingestão de alimentos, análises e relatórios de feedback personalizados.

Além do que já foi referido, a aplicação prestará auxílio no ato de compra de alimentos. Possuirá um leitor de código de barras que forneça a informação nutricional do alimento e que forneça

um semáforo nutricional relativamente ao mesmo. Existem alguns produtos no mercado que já possuem este nutri-score na embalagem.



Figura 2 - Nutri-score

Adicionalmente, a aplicação também terá incorporação do descodificador de rótulos da Direção Geral de Saúde e material educativo.

DESCODIFICADOR DE RÓTULOS ALIMENTOS por 100g					DESCODIFICADOR DE RÓTULOS BEBIDAS por 100ml				
	GORDURA (Lípidos)	GORDURA SATURADA	AÇÚCARES	SAL		GORDURA (Lípidos)	GORDURA SATURADA	AÇÚCARES	SAL
ALTO	mais de 17,5g	mais de 5g	mais de 22,5g	mais de 1,5g	ALTO	mais de 8,75g	mais de 2,5g	mais de 11,25g	mais de 0,75g
MÉDIO	entre 3 e 17,5g	entre 1,5 e 5g	entre 5 e 22,5g	entre 0,3 e 1,5g	MÉDIO	entre 1,5 e 8,75g	entre 0,75 e 2,5g	entre 2,5 e 11,25g	entre 0,3 e 0,75g
BAIXO	3g ou menos	1,5g ou menos	5g ou menos	0,3g ou menos	BAIXO	1,5g ou menos	0,75g ou menos	2,5g ou menos	0,3g ou menos

Mais informações consulte www.alimentacaosaudavel.dgs.pt

Figura 3 - Descodificador de rótulos

Será possível fazer a inserção de alimentos ingeridos, inserção de código de barras, facilitação da quantificação da porção do alimento e geração de relatórios semanais. Foram tidas em consideração as bases de dados: TCAP – INSA, Food Central Data, FatSecret, EuroFir e GoogleFit.

2.2 Visão Computacional

Como referido anteriormente, as escolhas alimentares têm bastante importância no controlo da diabetes. Este controlo alimentar e identificação dos nutrientes que são ingeridos é um processo muito moroso, levando a pessoa a saber a quantidade dos alimentos e a comparar com tabelas nutricionais, onde muitas das vezes a mesma não possui acesso ou conhecimento deste tipo de recurso [30].

Torna-se essencial a implementação de processos e métodos que ajudem as pessoas nesta verificação, tornando esta tarefa mais simples e acessível a todos. Nesse âmbito, a integração de tecnologias de inteligência artificial visual permite que imagens sejam processadas e analisadas mais rapidamente, com mais precisão e mais eficientemente.

Neste contexto, destaca-se o poder da visão computacional. Do reconhecimento facial ao processamento de ações de um jogo de futebol ao vivo, a visão computacional supera as capacidades da visão humana em diversas áreas.

A visão computacional funciona em cinco etapas básicas:

- **Aquisição:** as imagens podem ser adquiridas em tempo real através de vídeo, fotografia ou tecnologia 3D para análise. Nesta etapa, são identificados detalhes como intensidade da luz, cores ou profundidade por meio da interceção de pixéis;
- **Pré-processamento:** a imagem é processada, sendo realizados ajustes, como redução de ruídos, cor e luminosidade, para garantir a qualidade da imagem;
- **Extração de caraterísticas:** utilização de algoritmos para detetar e isolar porções desejadas ou formas de uma imagem digitalizada ou vídeo;
- **Segmentação:** dividir a imagem em regiões ou objetos distintos, a partir de caraterísticas do objeto. Esta etapa permite agrupar os pixéis de uma imagem em regiões homogéneas de acordo com o valor de uma determinada caraterística;
- **Reconhecimento de padrões:** para realizar este reconhecimento, um sistema necessita de uma base de conhecimento dos objetos a serem reconhecidos. Esta base de conhecimento pode ser aprendida a partir de um conjunto de amostras dos objetos a serem reconhecidos utilizando técnicas de *machine learning*.

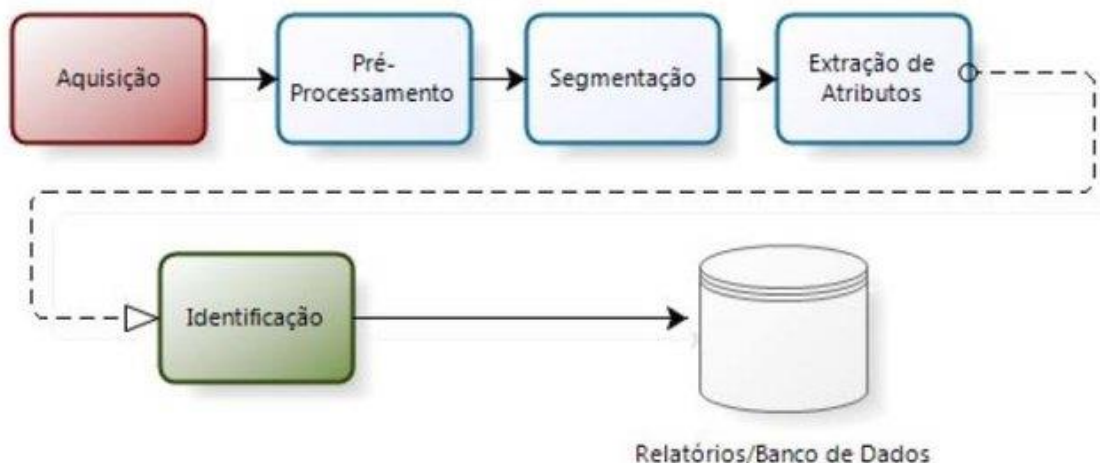


Figura 4 - Etapas de um sistema de Visão Computacional típico [31]

3 Revisão Literária

No presente capítulo serão explorados alguns sistemas de monitorização de pacientes diabéticos presentes no mercado, de acordo com o contexto do projeto que está a ser desenvolvido. Além disso, foi efetuado um estudo relativo às abordagens inteligentes utilizadas para a realização do acompanhamento personalizado dos pacientes, bem como técnicas utilizadas para o reconhecimento de alimentos e estimativa do seu peso. Este estudo é complementado pela realização de uma discussão relativa aos sistemas e abordagens inteligentes exploradas, avaliando a adequação dos métodos utilizados ao contexto do trabalho proposto.

3.1 Sistema de Monitorização

Nesta secção são apresentados alguns sistemas desenvolvidos para o acompanhamento e monitorização de pacientes diabéticos.

3.1.1 Glucose Buddy

O Glucose Buddy¹ é uma aplicação desenvolvida para todos os indivíduos portadores de diabetes tipo 1 ou tipo 2 e profissionais de saúde como enfermeiros e médicos. Desenvolvida pela empresa Azumio Inc, nos Estados Unidos, esta aplicação permite que os utilizadores registem manualmente os níveis de glucose no sangue, doses de insulina, outros medicamentos, alimentos ingeridos (em gramas), e atividade física (em minutos).

¹ <https://www.glucosebuddy.com/>

Esta aplicação possibilita manter e analisar registos dos níveis de glucose no sangue, da ingestão de carboidratos, de doses de medicamentos e da atividade física realizada pelo utilizador, ilustrado na Figura 5.

Além disso, este sistema permite a definição de lembretes de glucose e medicamentos, bem como a impressão e/ou envio de relatórios ao profissional de saúde do utilizador, e a visualização de gráficos para monitorizar tendências.

O Glucose Buddy faz o reconhecimento de alimentos através de uma imagem capturada pela câmara de telemóvel do utilizador.

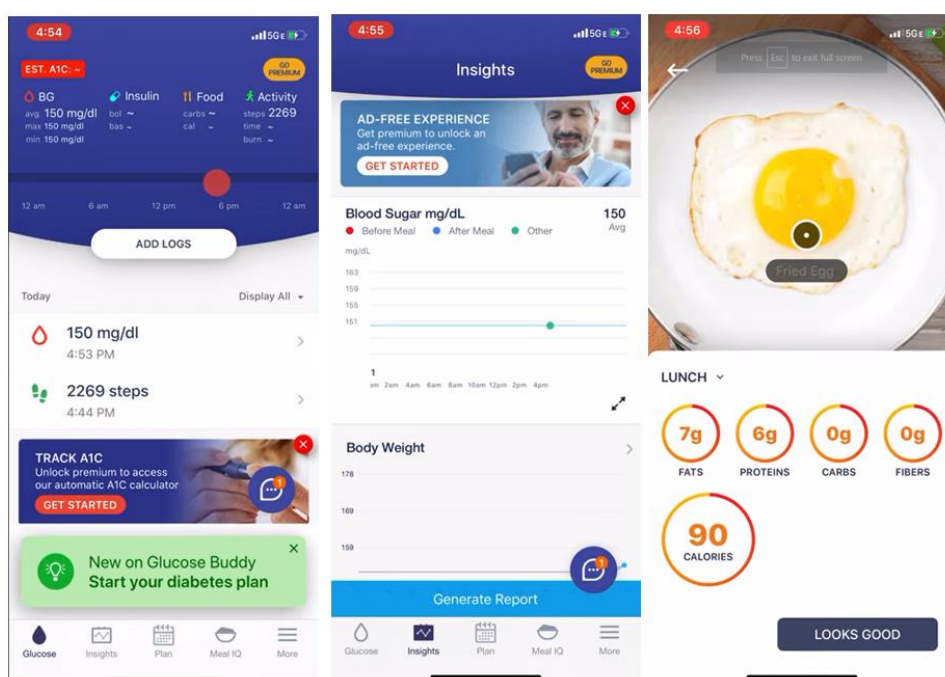


Figura 5 - Ecrãs da aplicação Glucose Buddy [32]

3.1.2 Diabetes:M

A aplicação Diabetes:M², desenvolvida pela empresa Sirma Medical Systems, na Bulgária, tem como objetivo auxiliar todos os indivíduos portadores de diabetes do tipo 1 ou tipo 2, através da monitorização da doença.

A Diabetes:M faz o registo de níveis de glucose no sangue, quantidade de carboidratos (tendo em conta uma lista categorizada de alimentos), peso, cetonas, HbA1c, colesterol, pressão arterial, pulso e atividades físicas.

A aplicação realiza o cálculo da insulina com base na ingestão de carboidratos, gordura e proteína. Além de possibilitar que o utilizador faça uma análise gráfica dos dados registados, a

² <https://www.diabetes-m.com/>

Diabetes:M emite alertas quando deteta a necessidade de ingestão de carboidratos adicionais ou para atrasar a próxima refeição devido ao alto nível de açúcar no sangue. Este sistema também permite a geração de relatórios, importação e exportação de dados, e a criação de lembretes.

3.1.3 Diabetes Connect

A aplicação Diabetes Connect³, tal como as aplicações previamente apresentadas, pretende realizar uma monitorização da diabetes e assim, melhorar a vida de todos os indivíduos portadores de diabetes tipo 1 ou tipo 2. Esta aplicação foi desenvolvida pela empresa SquareMed Software GmbH, na Alemanha.

A Diabetes Connect monitoriza o peso, açúcar no sangue, refeições, exercícios, injeções de insulina, pressão arterial, frequência cardíaca e medicamentos. A aplicação possibilita uma análise gráfica dos níveis de glucose, dados de refeições, entre outras informações úteis. Permite sincronizar informações com outros dispositivos e exportar e/ou importar dados.

3.1.4 MySugr

A aplicação MySugr⁴ é mais uma aplicação de monitorização da diabetes que promete tornar a doença mais simples de lidar. Foi criada pela empresa mySugr GmbH, na Áustria, e permite, tal como as anteriores, fazer o registo de níveis de glucose no sangue, refeições através de imagem, atividade física, insulina, entre outros. O MySugr permite a conexão com o medidor de glucose do utilizador, de modo a registar automaticamente as leituras de açúcar no sangue.

Esta aplicação faz o cálculo da quantidade de insulina que o utilizador precisa tomar. Adicionalmente, o MySugr mostra gráficos com dados relevantes até 90 dias, e permite a impressão e partilha digital de dados.

3.1.5 DiabTrend

A aplicação DiabTrend⁵ surgiu em 2015 e providencia monitorização personalizada da diabetes com o auxílio de inteligência artificial. DiabTrend faz o reconhecimento de alimentos através de uma imagem capturada em tempo real, estima a quantidade de porções e realiza o cálculo das calorias e dos hidratos de carbono. Além disso, esta aplicação prevê o valor de glucose no sangue nas próximas 4 horas e prevê eventos de hipo e hiperglicemia.

³ <http://www.diabetesconnect.de/en/>

⁴ <https://www.mysugr.com/en/>

⁵ <https://diabtrend.com/>



Figura 6 - Ecrãs da aplicação DiabTrend

Esta aplicação faz o reconhecimento de alimentos 10 vezes mais rápido, não precisa de internet, disponibiliza receitas ao utilizador e permite o registo dos níveis de glucose no sangue, insulina ou outro medicamento, refeições, atividade física, sono, peso, pressão sanguínea, batimentos cardíacos, HbA1c, vacinação, entre outros. É compatível com apple health, google fit e reconhecimento de voz.

3.1.6 MyFitnessPal

A aplicação MyFitnessPal⁶ foi desenvolvida em São Francisco e promete não ser apenas um sistema de monitorização de refeições ou de contagem de calorias, permitindo que o utilizador conheça os seus hábitos, aprenda a fazer escolhas mais inteligentes e ganhe motivação para se tornar mais saudável e atingir os seus objetivos.

Nesse âmbito, o MyFitnessPal pode ser utilizada por todos os indivíduos, permitindo que o utilizador faça um registo manual de itens alimentares, através do leitor de código de barras ou através da captura de uma imagem. Através da fotografia capturada, o sistema faz o reconhecimento dos alimentos e, baseando-se em quantidades por defeito, apresenta a sua informação nutricional.

⁶ <https://www.myfitnesspal.com/pt/>

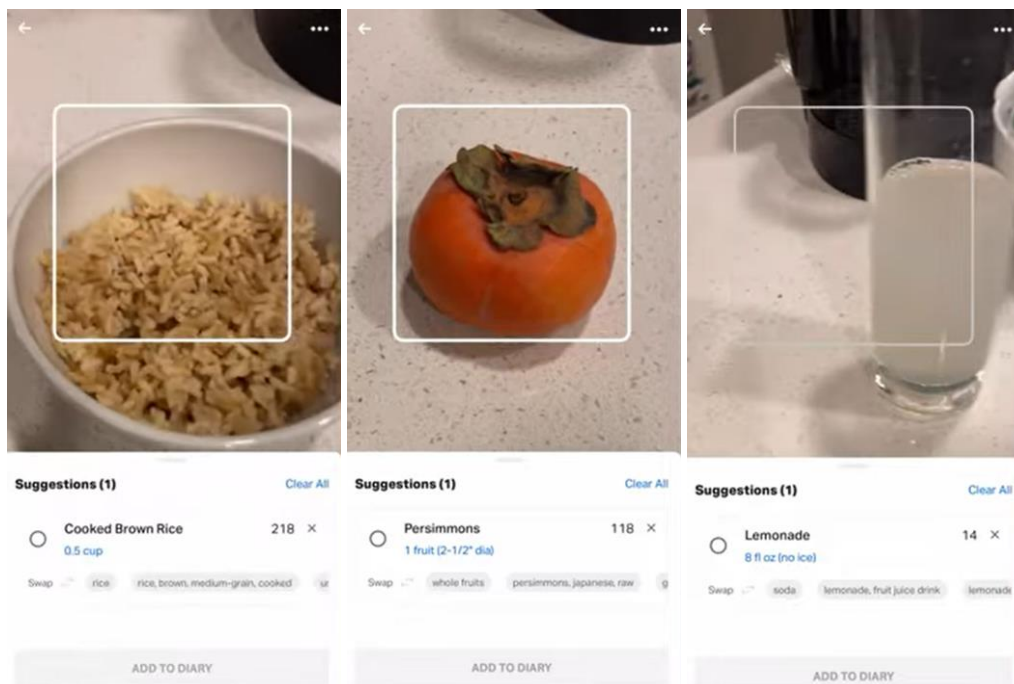


Figura 7 - Reconhecimento de alimentos MyFitnessPal [33]

3.1.7 Calorie Mama

A aplicação Calorie Mama⁷ foi desenvolvida pela Azumio e tem a possibilidade de integrar com a aplicação Glucose Buddy, apresentada anteriormente. Esta aplicação faz o reconhecimento de alimentos a partir de uma foto de uma refeição, ajuda a compreender os hábitos do utilizador e a aumentar a probabilidade de atingir os seus objetivos.

A Calorie Mama permite um registo rápido de alimentos, através de código de barras, bem como a visualização de macro e micronutrientes essenciais, e planos alimentares baseados em preferências. Adicionalmente, a aplicação disponibiliza uma representação gráfica de exercícios físicos e instruções escritas.

⁷ <https://www.caloriemama.ai/>



Figura 8 - Ecrãs da aplicação Calorie Mama [34]

3.1.8 SnapCalorie

A aplicação SnapCalorie⁸ foi fundada em 2021 e é a primeira aplicação que permite que o utilizador capte uma fotografia de uma refeição e rapidamente disponibiliza uma contagem de calorias bastante precisa, tendo em conta o volume dos alimentos. Por ser uma aplicação ainda recente, apresenta algumas limitações, como a necessidade de indicar o tamanho do prato onde estão os alimentos e não consegue fazer o reconhecimento de alimentos se estes não estiverem num prato.

⁸ <https://www.snapcalorie.com/>

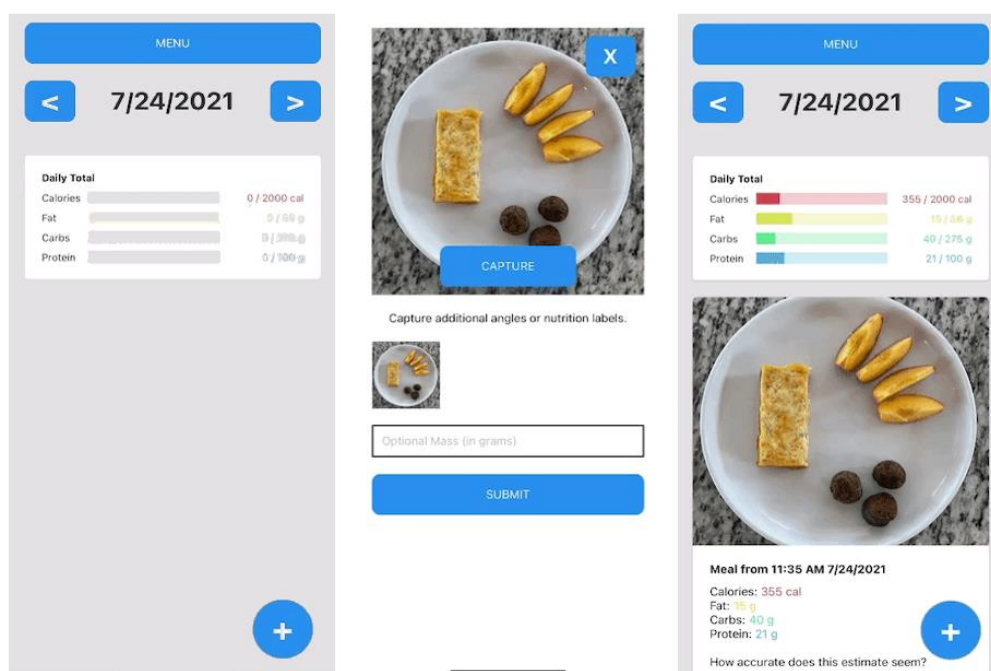


Figura 9 - Ecrãs da aplicação SnapCalorie

O SnapCalorie ajuda a compreender os hábitos do utilizador e a aumentar a probabilidade de este atingir os seus objetivos. Além disso, esta aplicação permite a visualização de calorias, carboidratos e proteínas contidos no prato. Por fim, é possível salvar refeições.

3.2 Abordagens Inteligentes

Nesta secção são identificadas e descritas algumas abordagens inteligentes utilizadas na literatura para a realização de reconhecimento de alimentos que serão ingeridos pelo utilizador do sistema.

3.2.1 Discussão

A Tabela 1 apresenta a comparação realizada entre os sistemas de monitorização apresentados anteriormente, analisando os pontos fortes e fracos destes.

Tabela 1 - Comparação entre os sistemas de monitorização

Nome	Pontos fortes	Pontos fracos
Glucose Buddy	Fácil visualização de tendências. Integrável com a aplicação Calorie Mama. Exportação de dados. Possui monitor de exercícios.	Demora um pouco para incluir informações nutricionais. A versão gratuita vários anúncios pop-up.

Diabetes:M	Monitorização remota. Integrável com vários dispositivos. Suporta vários utilizadores. Exportação de dados.	Muitos recursos apenas estão disponíveis na versão paga. Não faz o reconhecimento de alimentos. Não estima peso dos alimentos. A versão gratuita vários anúncios pop-up. Muitos recursos apenas estão disponíveis na versão paga. Não possui monitorização de exercícios. Não faz o reconhecimento de alimentos. Não estima peso dos alimentos.
Diabetes Connect	Interface mais simples. Boa opção para quem não entende de telemóveis. Baixo custo. Exportação de dados.	Opções de suporte limitadas. Não possui monitorização de exercícios. Não faz o reconhecimento de alimentos. Não estima peso dos alimentos.
MySugr	<i>Coaching</i> personalizado. Gráficos com dados relevantes até 90 dias. Estimativa dos valores HbA1c. Exportação de dados.	Não possui monitorização de exercícios. A calculadora de bolus de insulina não está disponível em todos os países. Não faz o reconhecimento de alimentos. Não estima peso dos alimentos.
DiabTrend	Reconhecimento de alimentos. Estimativa do peso dos alimentos. Previsão de glucose no sangue. Permite registo de valores. Calcula calorias, hidratos de carbono e quantidade de insulina. Exportação de dados.	Só permite o registo de refeições reconhecidas com o plano pago. O peso estimado varia consoante o ângulo e proximidade da câmara. Peso estimado em excesso.

Como podemos verificar na Tabela 1, só existe uma aplicação de monitorização da diabetes no mercado que permita fazer o registo de refeições através de reconhecimento de alimentos. É de realçar que apenas permite registar as refeições identificadas caso o utilizador possua a versão paga, caso contrário tem de registar manualmente como nas restantes aplicações. Algumas permitem retirar uma fotografia à refeição para ficar registado visualmente.

Apesar do Glucose Buddy ser a aplicação mais abrangente e mais utilizada, apresenta alguns pontos fracos. Apesar de não possuir sistema de reconhecimento de alimentos, esta aplicação dá a possibilidade de fazer integração com a aplicação Calorie Mama, desenvolvida pela mesma empresa, para que o utilizador possa fazer o registo das suas refeições mais facilmente.

A aplicação Diabetes:M é a melhor aplicação para quem precisa de monitorizar remotamente um membro da família ou para clínicas e profissionais de saúde que precisam de monitorizar vários pacientes. É integrável com vários dispositivos, como medidores de glucose, dispositivos Fitbit, entre outros, através de *Bluetooth*.

Para quem prefere uma interface mais simples de utilizar, a aplicação Diabetes Connect é a melhor solução. É bastante intuitiva, muito fácil de usar e é de baixo custo. Apesar de ser possível fazer o registo nutricional e de vários sinais vitais, não se consegue fazer o registo de atividade física. De salientar que esta aplicação possui várias unidades para fazer registos, como mg/dl e mmol/l, e item alimentar e unidades de carboidratos.

O MySugr é a melhor solução para conteúdos personalizados. Possui ferramentas que dão acesso a mensagens individuais com um treinador de diabetes certificado. É possível rastrear os níveis de açúcar de sangue em gráficos visualmente apelativos, que abrangem até 90 dias, e é disponibilizado o valor estimado de HbA1c a qualquer momento.

A aplicação DiabTrend é a mais completa de todas as mencionadas na Tabela 1, utilizando inteligência artificial para realizar o reconhecimento de alimentos e previsões de glucose. Contudo, a versão gratuita desta aplicação não permite o registo direto das refeições previamente identificadas pelo modelo de reconhecimento de alimentos, sendo necessário registar manualmente a refeição. Além disso, após alguns testes, verificou-se que esta aplicação possui alguma dificuldade em reconhecer alimentos (por exemplo, identificou que feijão preto eram mirtilos) e relativamente à estimativa de peso dos alimentos também não foi satisfatório, pois sempre que se mudava de ângulo ou se colocava a câmara mais próxima da refeição, esta alterava bruscamente de peso. Além disso, como era conhecido o peso real dos alimentos, o peso estimado pela aplicação estava sempre em excesso.

A Tabela 2 apresenta a comparação realizada entre os sistemas de reconhecimento de alimentos apresentados anteriormente.

Tabela 2 - Comparação entre os sistemas de reconhecimento de alimentos

Nome	Pontos fortes	Pontos fracos
MyFitnessPal	Monitorização de calorias. Possui <i>scanner</i> de refeição. Leitor de código de barras. Base de dados de alimentos extensa. Sincronização com outros dispositivos. Relatórios de progresso. Possui monitor de exercícios. Definição de metas. Monitorização de peso.	Não consegue estimar o volume dos alimentos. O scanner de refeição apenas está disponível na versão paga.
Calorie Mama	Integração com a aplicação Glucose Buddy.	Não consegue estimar o volume dos alimentos.

	<p>Leitor de código de barras. Possui <i>scanner</i> de refeição. Possui monitor de exercícios. Definição de desafios e metas. Monitorização de peso. Planos alimentares.</p>	<p>Não possui uma base de dados de alimentos muito extensa.</p>
SnapCalorie	<p>Faz o cálculo nutricional tendo em conta o volume dos itens alimentares. Possui <i>scanner</i> de refeição.</p>	<p>Não mostra a identificação dos alimentos da refeição. Os valores nutricionais apresentados são referentes à refeição completa. Não possui monitor de exercícios. Não é possível definir desafios nem metas. Não dá para monitorizar o peso.</p>

A aplicação MyFitnessPal é o melhor *tracker* de alimentos, pois possui uma base de dados enorme que inclui vários itens de restaurantes e lojas populares. É possível fazer o registo de refeições manualmente, através de leitura do código de barras dos alimentos e pelo scanner de alimentos onde, a partir de uma fotografia da refeição, os alimentos são reconhecidos e o utilizador apenas necessita de alterar quantidades (apresentadas com valores por defeito).

O Calorie Mama ajuda a compreender os hábitos do utilizador e a aumentar a probabilidade de atingir os seus objetivos. Esta aplicação permite o registo rápido de alimentos, através de código de barras e/ou através do *scanner* de refeição. Contudo, não possui automatização da inserção do peso dos alimentos, sendo necessário realizar registos manualmente. Além disso, esta solução disponibiliza planos alimentares baseados em preferências, bem como a representação gráfica de exercícios físicos e instruções escritas.

O SnapCalorie é a única aplicação que faz o cálculo nutricional tendo em conta o volume dos itens alimentares. Contudo, esta aplicação foca-se principalmente no cálculo nutricional a partir de uma fotografia capturada pelo utilizador, pelo que muitas das funcionalidades disponibilizadas nas restantes aplicações não poderão ser encontradas nesta solução.

Como é possível verificar, apenas existe uma aplicação no mercado que tem em consideração as necessidades dos pacientes diabéticos e que utilize inteligência artificial para agilizar o dia a dia do paciente. Contudo, esta solução apresenta algumas funcionalidades bloqueadas na versão gratuita e a estimativa do peso dos alimentos não é fidedigna, sendo sempre identificada em excesso e variando de acordo com a posição da câmara face à refeição.

3.2.2 Metodologia

Para a revisão sistemática, seguiu-se a metodologia PRISMA [35] onde se definiu as questões de pesquisa e estratégia de busca, e, por fim, fez-se uma revisão dos resultados obtidos para obter os artigos que mais se enquadravam no contexto do projeto desenvolvido.

Primeiramente, definiu-se o intuito da pesquisa: conhecer o que existe atualmente na literatura que permita o monitoramento de pacientes diabéticos, dando ênfase ao reconhecimento de alimentos através de visão por computador. Nesse âmbito, também importou analisar documentos que apresentassem técnicas que permitissem fazer uma estimativa do volume de itens alimentares. Assim, estabeleceu-se três questões de pesquisa, que podem ser encontradas na Tabela 3.

Tabela 3 - Questões de pesquisa

Identificador	Questões de Pesquisa
QP1	Quais são os métodos utilizados na monitorização de pacientes diabéticos?
QP2	Quais são as técnicas utilizadas no reconhecimento de alimentos através de visão computacional presentes nos projetos já existentes?
QP3	Quais são as técnicas utilizadas na estimativa de volume de alimentos presentes nos projetos já existentes?

Na primeira questão analisa quais as principais características tecnológicas utilizadas para identificar qual o tipo de software e hardware é mais frequentemente considerado no desenvolvimento de sistemas de monitorização de diabetes. A segunda questão diz respeito às técnicas e métodos mais utilizados para o reconhecimento de alimentos através de visão por computadas. Por fim, a questão três aborda as metodologias empregues para a estimativa de volume de itens alimentares já presentes na literatura. Depois de definidas as questões de pesquisa, foram identificadas e definidas quais as fontes de dados seriam utilizadas. Para este projeto, foram consideradas três bases de dados representadas na Tabela 4.

Tabela 4 - Bases de dados eletrónicas

Identificador	Base de Dados	URL
BD1	PubMed	https://www.ncbi.nlm.nih.gov/pubmed/
BD3	IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp
BD4	B-On	https://www.b-on.pt

De seguida, foram definidas estratégias de pesquisa através de palavras de busca que refletissem as questões de pesquisa previamente definidas. Por esse motivo, identificou-se três palavras de busca representadas na Tabela 5. É importante salientar que os conjuntos de pesquisa foram pesquisados em inglês, de modo a abranger estudos publicados mundialmente.

Tabela 5 - Palavras de pesquisa

Identificador	Base de Dados	URL
S1	Público-alvo	Diabet* AND
S2	Tarefa	(Recogni* OR Monitor* OR Enhanc* OR Manage*) AND
S3	Item	Food

É importante salientar que as palavras de pesquisa S1 e S3 foram aplicadas apenas ao título dos artigos e o conjunto das palavras de pesquisa S2 foi aplicado ao todo o documento. Apenas foi realizado esta alteração no filtro para restringir a pesquisa.

O processo de seleção de estudos e extração de dados incluiu a definição da metodologia de seleção e refinamento para filtrar resultados indesejáveis e, em seguida, recuperar resultados relevantes que possam responder às questões de pesquisa definidas.

Para esse processo, foram definidos critérios de inclusão e exclusão que podem ser consultados na Tabela 6 e na Tabela 7. Foram considerados os estudos que atenderam a pelo menos um ou mais critérios de inclusão, enquanto os estudos que atenderam a um ou mais critérios de exclusão foram excluídos.

Tabela 6 - Critérios de inclusão

Identificador	Critérios de inclusão
CI1	O artigo foca a utilização de sistemas de monitorização aplicados a pacientes diabéticos.
CI2	O artigo aborda a utilização de técnicas de reconhecimento de alimentos.
CI3	O artigo mostra a importância da utilização de técnicas de monitorização e de reconhecimento de alimentos.

Tabela 7 - Critérios de exclusão

Identificador	Critérios de exclusão
CE1	Artigos publicados antes de 2014.
CE2	Artigos que não estão redigidos em inglês.
CE3	Artigos não são artigos de jornal ou materiais de conferência.
CE4	Artigos que são revisões sistemáticas ou casos de estudo.
CE5	Artigos duplicados.

No processo de seleção e extração de dados dos estudos, foi realizado um processo dividido em quatro fases: identificação, triagem, elegibilidade e inclusão.

Como é possível verificar na Figura 10, foram identificados 1490 artigos das bases de dados previamente selecionadas. A base de dados B-On foi a que devolveu mais resultados, contudo na fase de triagem muitos resultados foram excluídos. Assim, 786 documentos foram removidos através do critério de exclusão CE5. Depois do título e o resumo dos documentos serem revistos, foram removidos 373 registos. Na fase da elegibilidade, foram analisados os restantes artigos de acordo com os critérios de inclusão e exclusão. Apenas 3 artigos foram elegíveis para a fase de inclusão.

A pequena quantidade de artigos elegíveis deve-se ao facto de a *query* de pesquisa ser muito restrita, isto é, com a pesquisa era expectável que os documentos incidissem no reconhecimento ou monitorização de alimentos e que se focasse nos pacientes diabéticos. Com isto, pode-se constatar que, nos últimos dez anos, este não tem sido um tema de grande foco.

Por esta razão, foi necessária a inclusão de artigos considerados relevantes que, devido à filtragem da *query*, tinham sido removidos. Deste modo, foram recuperados 8 artigos que abordam não só o reconhecimento de alimentos como falam de técnicas para a estimativa do volume de itens alimentares (o que pode ser bastante útil no desenvolvimento deste projeto), totalizando 11 registos elegíveis.

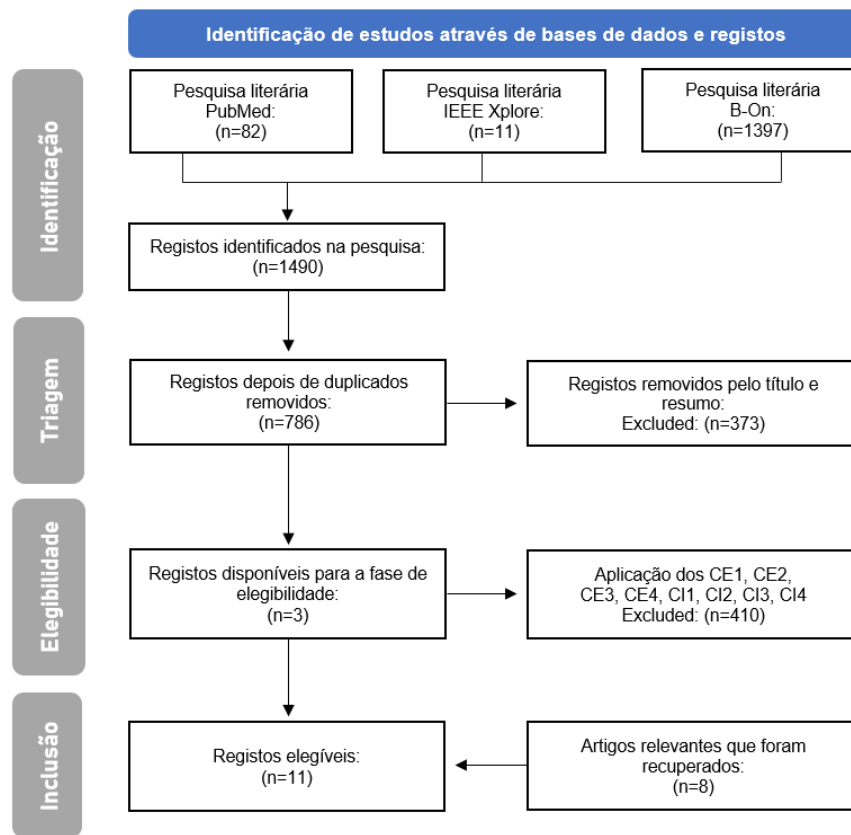


Figura 10 - Processo de seleção de artigos

3.2.3 Sistemas de Monitorização da Diabetes

Após o processo de pesquisa, foram estudados os diversos recursos tecnológicos que têm vindo a ser frequentemente utilizados na monitorização da diabetes. Os artigos mostram que a sua preocupação não está apenas voltada para a monitorização de sinais vitais, mas também para controlar e gerir a doença e facilitar a vida dos diabéticos.

No estudo [36] de Zahra Darabi *et al.*, é apresentado um sistema multiagente para auxiliar pacientes diabéticos do tipo 2. Este estudo apresenta um protótipo de solução que disponibiliza

recomendações de dietas ao paciente a partir dos seus dados de registo na aplicação. Foram desenvolvidos três agentes – agente utilizador, agente dietético e agente medidor de glucose.

O primeiro agente é considerado o agente principal do sistema, interage com todos os agentes do sistema e envia para eles as informações necessárias. O segundo agente recebe informações dos utilizadores, como a quantidade de atividade física praticada, a sua pressão arterial, níveis de colesterol, entre outros, e calcula o Índice de Massa Corporal, a Taxa Metabólica Basal e a quantidade de carboidratos, proteínas, gorduras, fibras e sódio ingeridos. O terceiro agente usa o modelo *Ackerman* e *Random Walk* [37] [38] para medir a glicemia do paciente durante o dia e durante a noite. As três secções de *Ackerman* calculam a glucose no sangue nas principais refeições durante o dia, enquanto as três fases do *Random Walk* analisam a glucose no sangue à noite.

3.2.4 Sistemas de Reconhecimento e Estimativa de Peso de Alimentos

Existem vários estudos que utilizam visão computacional para fazer o reconhecimento de alimentos numa refeição. Nesse contexto, foram recolhidos artigos que fizessem uma análise da aplicação de diferentes técnicas no reconhecimento de alimentos e de alguns métodos utilizados para fazer a estimativa do volume de itens alimentares, para disponibilizar ao utilizar valores nutricionais mais próximos da realidade.

No artigo [39], Robert A. Sowah *et al.* apresentam um sistema de software para fazer a gestão da diabetes através de uma abordagem de *machine learning*. Este artigo expõe um sistema de recomendações de refeições com reconhecimento de alimentos, que tem como foco a geração de planos de alimentação diários e personalizados para os utilizadores, de acordo com suas necessidades nutricionais e preferências alimentares anteriores.

Os autores desse estudo [39] construíram um modelo de rede neuronal Tensorflow para a classificação de alimentos, que permite aos utilizadores fazer o carregamento de imagens de refeições e assim, determinar se a refeição é recomendada ou não para consumo. Foi implementado um algoritmo *K-Nearest Neighbor* (KNN) que trata do sistema de recomendações de refeições e foram usadas ciências cognitivas para construir um chatbot de perguntas e respostas sobre diabetes.

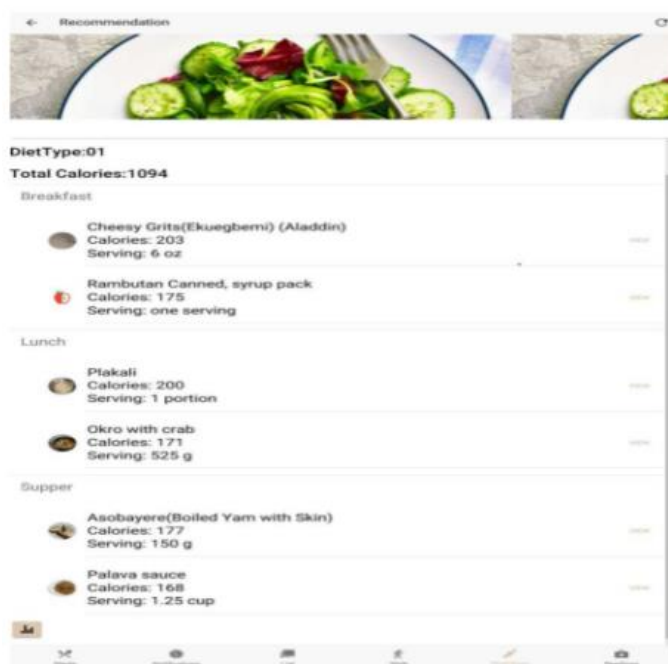


Figura 11 - Recomendação de refeição com calorias e a quantidade da porção [39]

Além disso, este sistema rastreia a atividade e a geolocalização do utilizador, permite a geração de PDF's de leituras de açúcar no sangue previamente registadas. Na Figura 12 é possível constatar alguns testes realizados pelos autores, ao módulo de reconhecimento de alimentos com pratos.

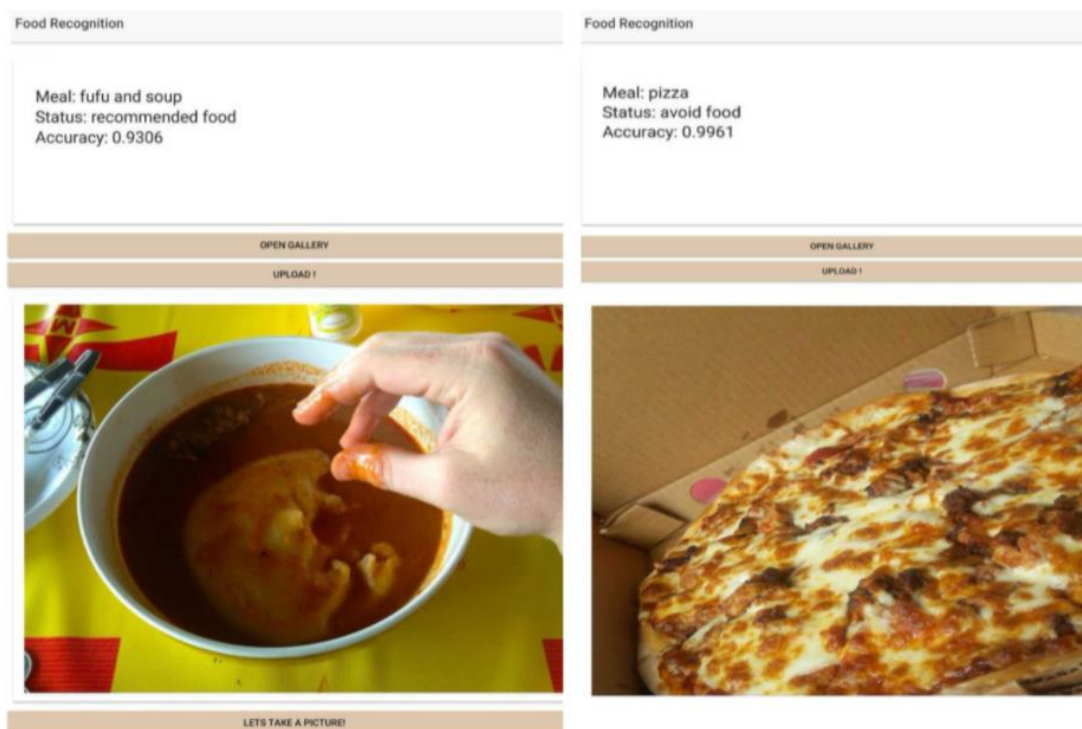


Figura 12 - Testes ao modelo de reconhecimento de alimentos para pratos locais e continentais [39]

O estudo [40] de Geeta Shroff *et al.* propõe um sistema de monitorização de calorias para dispositivos móveis chamado DiaWear. Este sistema pretende melhorar a qualidade de vida de pacientes com diabetes e de indivíduos que necessitam de uma gestão de nutrição. Para isso, a aplicação faz um reconhecimento semiautomático de alimentos através de redes neuronais para identificar itens alimentares a partir de uma imagem capturada do alimento com um objeto de referência ao seu lado (Figura 13).

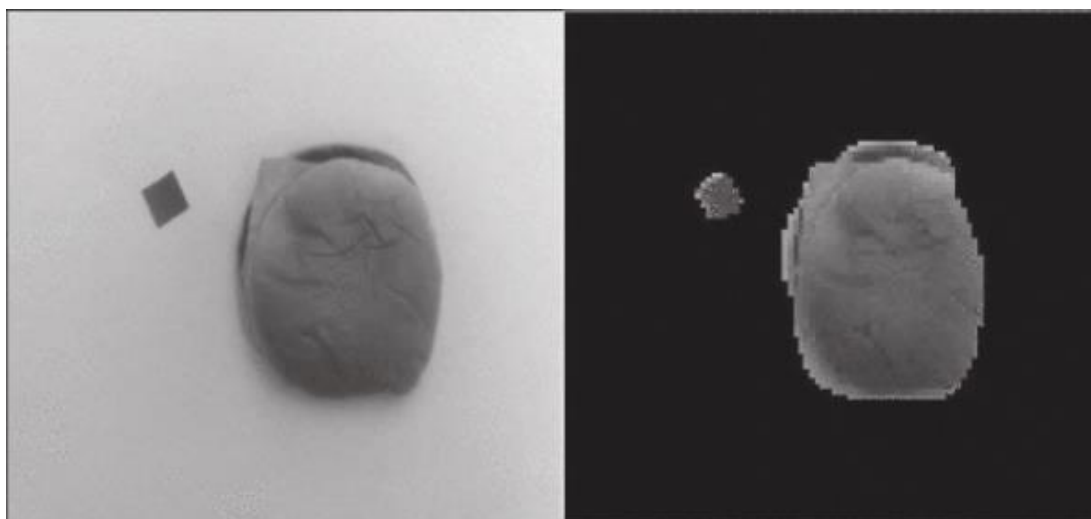


Figura 13 - Reconhecimento de alimentos usando objeto de referência [40]

Após o pré-processamento, o fundo é removido e são comparadas estatísticas de segmentação, cor e tamanho do objeto de referência e do item alimentar para determinar cores, tamanho, forma e textura. O vetor de recursos é o input para o algoritmo de *back propagation* de uma rede neuronal *feed-forward* de duas camadas. A camada de saída consiste em quatro neurónios que representam quatro classes de alimentos – hambúrguer, batatas fritas, nugget de frango e tarte de maçã.

Os autores deste estudo [40] classificam um alimento como válido quando é encontrada a menor distância euclidiana de um vetor de saída da rede neuronal abaixo do limite calculado experimentalmente. Para cada item alimentar foi calculado um peso contextual e foi aplicada a Lei da Probabilidade Total para fundir o contexto e probabilidades não contextuais para cada uma dessas classes. Assim, é obtida a faixa de calorias para o alimento e é exibida ao utilizador.

Alguns trabalhos propõem metodologias baseadas no modelo *bag-of-features* (BoF). Neste estudo [41], Marios M. Anthimopoulos *et al.* utilizaram um *dataset* com 4868 imagens de comida caseira da Europa Central que foram organizados em 11 classes de alimentos diferentes. Cada classe foi descrita através do modelo BoF, passando por quatro fases: extração de pontos-chave, descrição do recurso local, aprender o dicionário visual e quantização do descritor.

Assim, foram extraídos três pontos-chave - detetores de pontos de interesse, amostragem aleatória e amostragem densa (Figura 14) -, catorze descrições de imagens, dois métodos de *clustering* – *k-means* e *hk-means* - para a criação do dicionário visual e foram testados seis

classificadores – *Support Vector Machine* (SVM_{linear} , SVM_{RBF} e SVM_{x^2}), *Artificial Neural Network* (ANN_{nh} e ANN_{wh}) e *Random Forests* (RF).

Foram realizadas cinco experiências relativas aos efeitos de vários parâmetros como o número de pontos-chave extraídos, o tamanho do descritor e outras características do descritor, comparação de algoritmos de cluster para criação do dicionário visual e de classificadores baseados em *machine learning*. O sistema otimizado alcançou uma precisão de 78%.



Figura 14 - Técnicas de extração de pontos-chave: detetor SIFT, amostragem aleatória e amostragem densa [41]

Reza Dea Yogaswara *et al.* publicaram o estudo [42] onde apresentam um sistema que calcula o conteúdo calórico de itens alimentares com base no volume dos alimentos usando o algoritmo *Mask Region-based Convolutional Neural Network* (R-CNN). Para isso, é necessário que o utilizador capture uma imagem da sua refeição através da câmara de um smartphone. Foi utilizado o Mask R-CNN, pois após alguns testes, verificaram que este método superou outros dois algoritmos: *Multi-task Network Cascade* e *Fully Convolutional Instance-aware Segmentation*. Começaram por determinar as classes de objetos a serem calculados para o número de calorias numa porção, onde foi obtida a segmentação de imagens, a altura e a densidade de cada classe.

Para a aquisição de dados, foram recolhidos dados de imagens com combinações de alimentos de diferentes classes de objetos, foi medida a altura, a área e a massa dos itens alimentares. Foi realizado o *labeling* de imagens, de seguida o conjunto de dados foi dividido em três partes (treino, desenvolvimento e teste), e depois realizaram o treino e validação do modelo de segmentação.

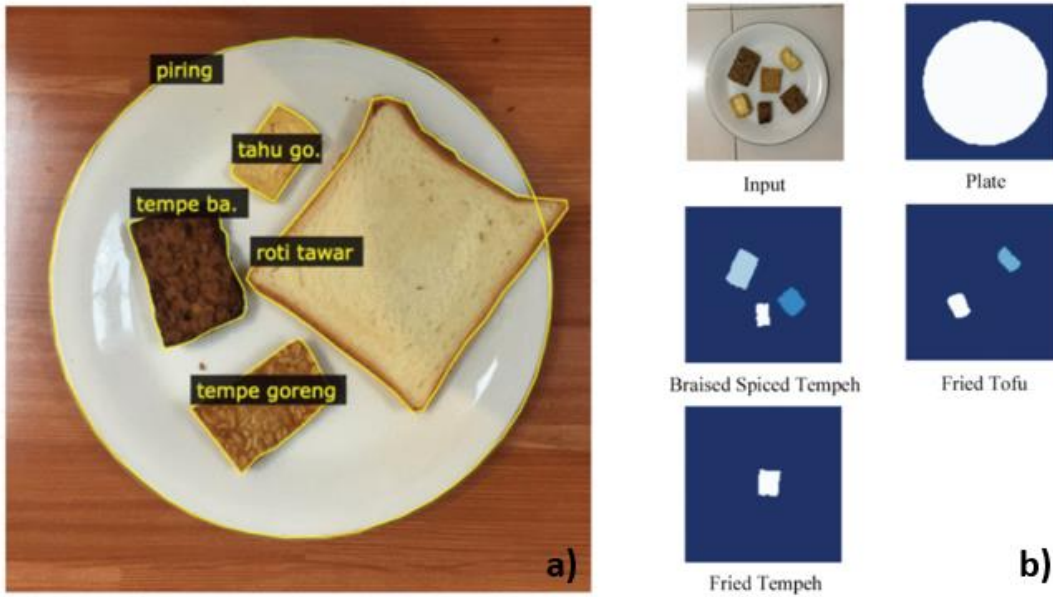


Figura 15 - Labeling utilizando a técnica *pixel-dense labeling* a) e segmentação das classes de objetos alimentares b) [42]

Posteriormente, testaram o modelo de segmentação de alimentos, calcularam o volume do objeto detectado, mostraram os resultados da segmentação e o cálculo de calorias. Este sistema apresentou ser preciso no cálculo de calorias, obtendo uma precisão média de 97.48%.

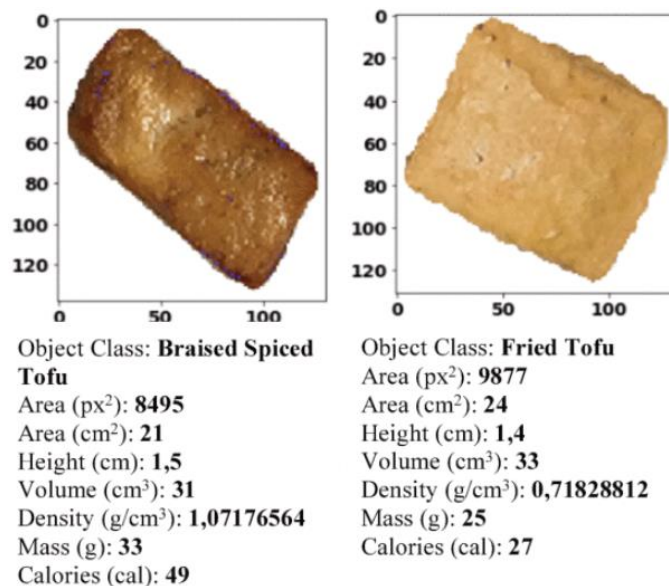


Figura 16 - Detecção da classe de alimentos e obtenção das calorias de cada objeto [42]

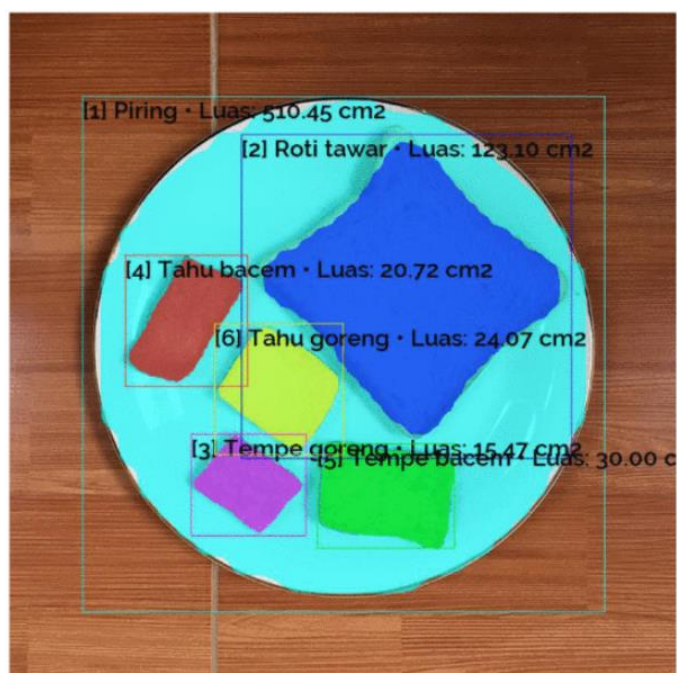


Figura 17 - Resultados da detecção da segmentação de alimentos e cálculo de calorias [42]

A aplicação móvel GlucoseML abordada no estudo [43] de Fotis Konstantakopoulos *et al.*, faz o reconhecimento de alimentos e uma estimativa de nutrientes, de modo a auxiliar pacientes com diabetes do tipo 1. Este sistema pretende gerir a diabetes do tipo 1 baseando-se em análises preditivas a curto prazo a partir da trajetória da glicose. O sistema faz aquisição de dados biológicos, fisiológicos e comportamentais do utilizador em tempo real através de sensores e fornece *feedback* personalizado.

Para este sistema, utilizaram uma base de dados com 11 categorias de alimentos, sendo que cada imagem foi anotada com a categoria a que pertence, o nome exato do alimento, o peso, o volume e os seus macro e micronutrientes.

O sistema começa por detetar o prato, deteta e separa os diferentes itens alimentares dentro do prato, e, de seguida, faz a descrição dos alimentos a partir dos recursos cor e textura (Figura 18). Foram testados alguns algoritmos de extração de pontos-chave como o *scaled-invariant feature transform*, o *denseSIFT* e o *speed up robust feature*. Um descritor dividido em duas categorias – cor e textura – é implementado em torno de cada ponto-chave, gerando vetores de características que são posteriormente agrupados para criar palavras visuais e consequentemente, através do algoritmo *k-means* são criados os dicionários visuais. Os alimentos são classificados através de métodos de *machine learning*.



Figura 18 - Etapas do processamento da imagem: imagem original, detecção de placa, segmentação de alimentos, reconhecimento de alimentos e reconstrução 3D [43]

Para a estimativa do volume dos alimentos, foram necessárias duas imagens, de diferentes ângulos, do prato com um cartão de referência colocado próximo do mesmo. Foi realizada a estimativa da posição relativa, reconstrução densa e, por fim, o volume foi estimado. Os autores apresentaram também a estimativa da composição nutricional dos alimentos, que resultou do cálculo exato dos nutrientes dos alimentos com base na saída do reconhecimento de alimentos e da estimativa de volume.

O documento [44] redigido por Muhammad Usman *et al.*, apresenta uma forma automática de fazer o controlo da glicose e da ingestão alimentar de pacientes diabéticos utilizando sensores e algoritmos de *machine learning*. Este sistema promete auxiliar os médicos a analisar o impacto da ingestão de alimentos na glicose no sangue em tempo real. A leitura contínua da glicose foi realizada através do sensor FreeStyle Libre usado pelo paciente e para a identificação dos alimentos foi necessária a captura de uma imagem, a partir da câmara de um smartphone.

O *dataset* utilizado foi recolhido manualmente, tendo 7530 imagens de 38 categorias diferentes de alimentos do Médio Oriente. Para o reconhecimento de alimentos, utilizaram uma metodologia composta por extração de recursos, classificação e fusão (Figura 19).

A extração de recursos foi realizada através de modelos pré-treinados como AlexNet, VggNet, GoogleNet e ResNet, e as SVM foram treinadas nos recursos extraídos. Foi realizada a fusão precoce (concatenação simples de vetores de características) e três métodos de fusão tardia (*Particle Swarm based Optimization*, Algoritmo Genético e média simples das pontuações obtidas em modelos individuais).

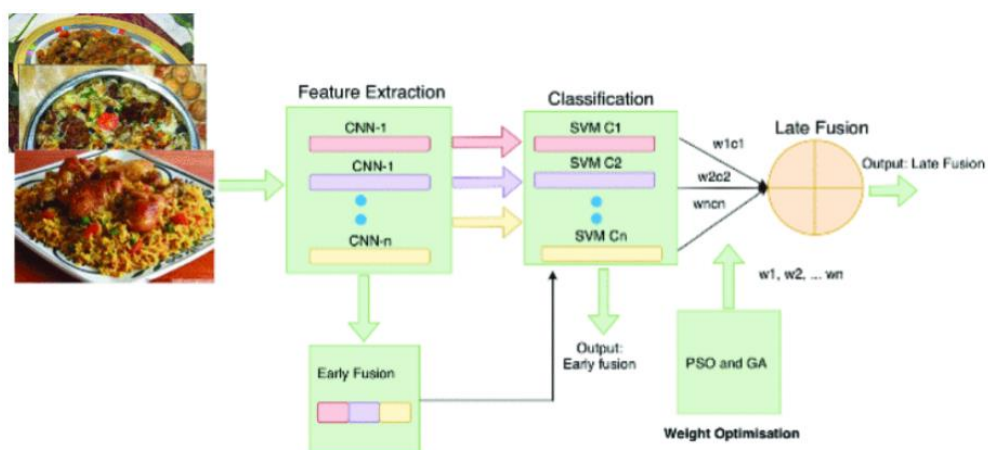


Figura 19 - Diagrama em blocos do reconhecedor de alimentos [44]

A aplicação apresentada recolhe dados de sensores CGM por Bluetooth, constrói um gráfico de tempo com esses dados, reconhece a refeição a partir de uma fotografia capturada pelo utilizador e mapeia-a no gráfico. Além disso, este sistema informa o utilizador caso o nível de glicose no sangue esteja alto, recomenda alguma atividade física, alerta o médico e familiares imediatos se o nível de glicose estiver muito alto e recomenda o utilizador a não comer alimentos ricos em carboidratos.

No estudo [45] de Narayana Darapaneni *et al.*, é proposto um modelo que prevê as calorias consumidas pelo utilizador com ajuda de uma imagem da porção do alimento que será ingerido. Este modelo utiliza uma abordagem em camadas em que cada camada executa tarefas diferentes.

A primeira camada recebe a imagem capturada pelo utilizador e prepara a mesma para alimentar a segunda camada, esta realiza a identificação de alimentos através do método Mask-RCNN juntamente com a *bounding box* e a geração da máscara; a terceira camada calcula a área de superfície coberta pelos itens alimentares e a quarta camada prevê as calorias consumidas pelo utilizador com base na área de superfície coberta pelo alimento presente na imagem fornecida.

Foi utilizado um *dataset* de 638 imagens divididas em 6 categorias diferentes de alimentos. Como já referido, foi utilizado o método Mask R-CNN para realizar a segmentação. Na Figura 21 é possível verificar a *bounding box* criada em torno do item alimentar identificado e a *label* que lhe foi atribuída.

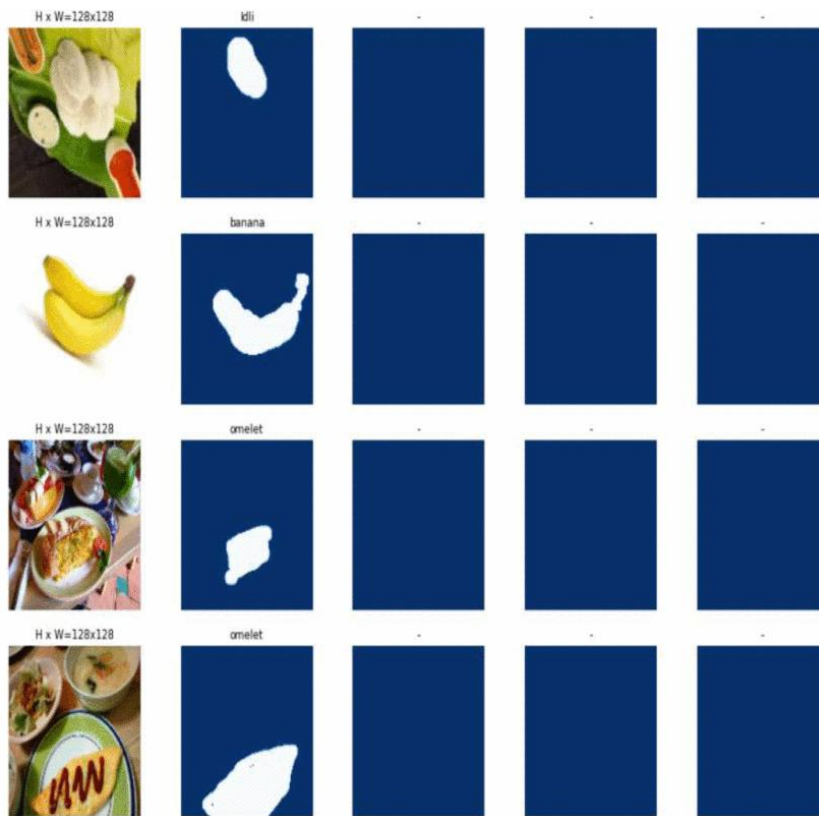


Figura 20 - Análise de imagens e máscara após o pré-processamento [45]



Figura 21 - Output da máscara [45]

Para fazer o treino do modelo, eram necessárias muitas imagens e como o *dataset* que os autores tinham não era grande o suficiente, utilizaram uma abordagem de transferência de

aprendizagem do repositório *matterport*. Esta abordagem permite que, em vez de treinar um modelo do zero, inicia-se com um arquivo de pesos que foi previamente treinado no conjunto de dados COCO. O conjunto de dados COCO contém muitas imagens, então os pesos do modelo treinado já aprendem muitos dos recursos comuns em imagens naturais.

Como uma imagem de câmara *pin-hole* não fornece os tamanhos reais dos objetos, os autores adotaram um método de aproximação de proporções. Nesta abordagem, a caloria por máscara da classe de alimento é considerada como referência para prever as calorias da imagem fornecida.

Por fim, cada imagem foi dimensionada para o tamanho 128x128 e posteriormente, a área segmentada da classe de alimentos foi calculada e multiplicada pelos valores presentes na coluna “Calorie_per_mask”.

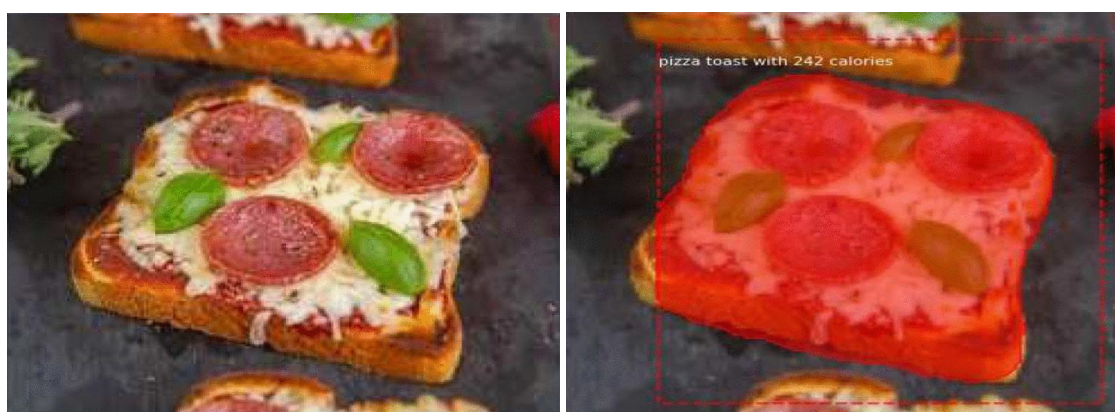


Figura 22 - Imagem antes e depois da máscara R-CNN e previsão de calorias [45]

Foram poucas as abordagens encontradas que fizessem a estimativa do volume de diferentes tipos de comida, contudo destacam-se dois estudos [46] e [47] que são descritos de seguida.

O artigo [46] apresentado por Joachim Dehais *et al.*, expõe um sistema computacional para estimar o volume de diferentes tipos de comida. Este sistema faz uma reconstrução 3D da refeição, tendo em conta duas imagens recolhidas, e estima o volume dos alimentos contidos num prato com base na geometria de duas vistas. Este sistema foi concebido para pacientes diabéticos do tipo 1, de modo a ser realizada uma estimativa do teor de carboidratos da refeição que será ingerida pelo mesmo.

Para a reconstrução do modelo 3D, foram utilizados dois detetores – *Scale Invariant Feature Transform* e *Speed Up Robust Features* – que extraem pontos chave, enquanto a correspondência de pontos foi realizada com uma *kD-Tree*. As correspondências de pontos foram filtradas para *outliers* usando o paradigma *LO-RANSAC* combinado com testes *Tdd*, e para discriminar *inliers* de *outliers*, utilizaram o algoritmo *Levenberg-Marquard* para calcular, de forma otimizada, uma distância epipolar simétrica adaptativa.

Para a estimativa do volume foi utilizado um objeto com dimensões conhecidas ao lado do prato com a refeição para fornecer uma referência de escala. O sistema possui um tempo computacional abaixo de 15 segundos por par de imagens e apresentou uma taxa de falhas baixa em casos experimentais.

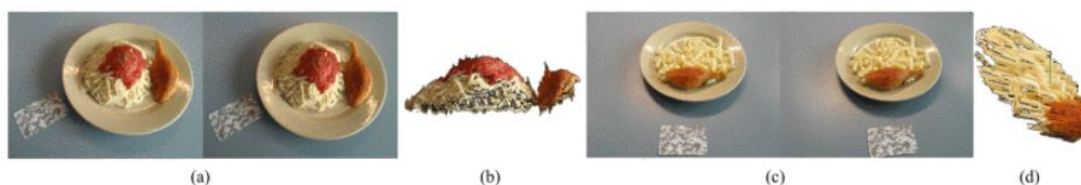


Figura 23 - Reconstrução 3D de pares de imagens [46]

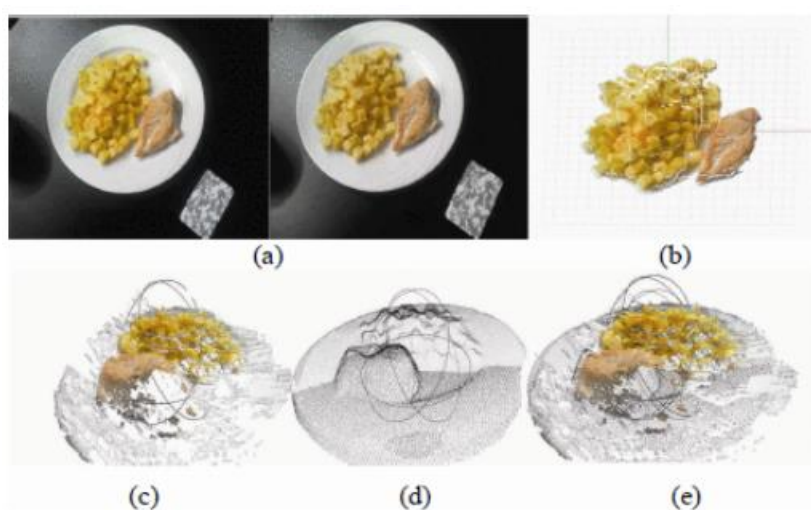


Figura 24 - Processo de reconstrução 3D [46]

O documento [47] de Yaofeng Yue *et al.*, apresenta uma abordagem para o cálculo do volume de itens alimentares de várias dimensões, a partir de uma única imagem 2-D. Esta abordagem requer que o alimento esteja num prato, tigela ou xícara com tamanho conhecido e que a imagem seja captada assumindo que a câmara esteja nivelada e inclinada em direção ao objeto com o item alimentar. Foi utilizado o modelo de câmara *pin-hole* que fornece uma projeção em perspectiva do plano do objeto para o plano da imagem.

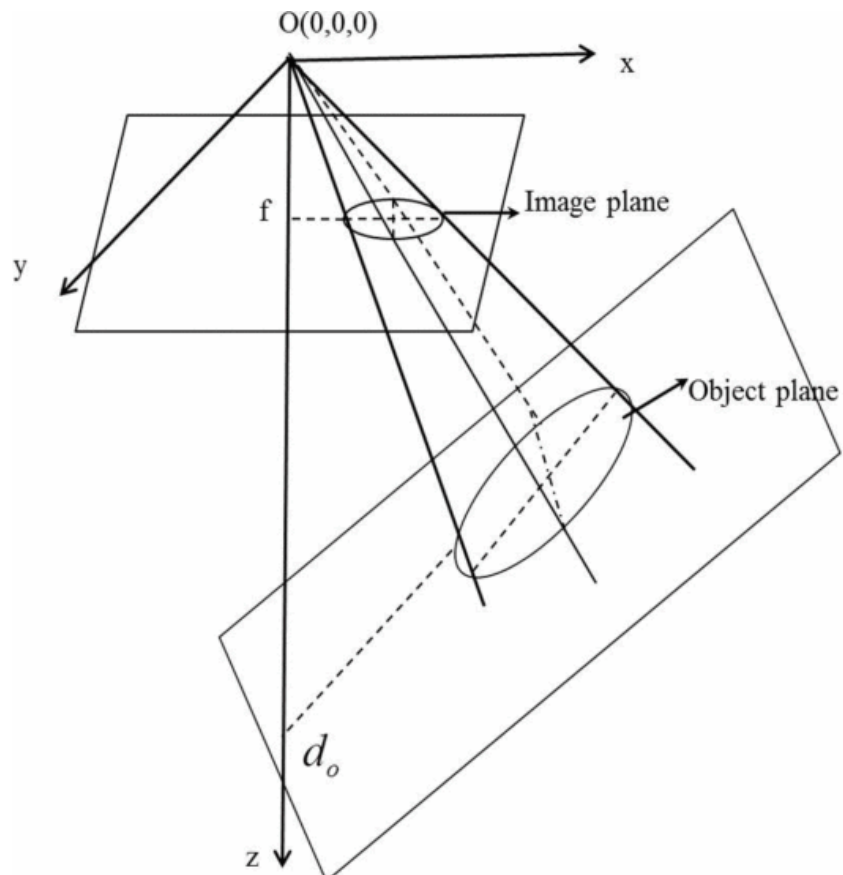


Figura 25 - Projeção do centro óptico, imagem e objeto no sistema de coordenadas xyz [47]

Primeiramente, o modelo é estabelecido para um sistema de coordenadas xyz, mas para simplificar a derivação, o sistema é transformado linearmente para um sistema de coordenadas $x'y'z'$ para que o plano do objeto fosse paralelo ao plano $x'y'$. De seguida, resolveram a orientação do plano do objeto através de uma equação circular e, por fim, considerando que as dimensões do objeto circular são fornecidas, que as principais projeções de perspectiva entre o objeto e os planos de imagem foram estabelecidas, foi possível apresentar uma estimativa do volume dos alimentos.

4 Enquadramento Teórico

O presente capítulo tem como objetivo a apresentação de conceitos fundamentais associados ao trabalho proposto.

4.1 Conceitos

Nesta secção serão abordados os conceitos explorados no desenrolar do caso de estudo e que são a base teórica fundamental para entender a execução do mesmo.

4.1.1 Machine Learning

As técnicas de ML permitem aprender informação e detetar padrões relevantes de forma automática, a partir de dados fornecidos. A parte de aprendizagem dos algoritmos permitem minimizar o erro e maximizar as previsões corretas. Desse modo, existem termos de ML que são bastante relevantes, como:

- **Dados:** Itens usados para treino e/ou avaliação;
- **Features:** conjunto de atributos que, na maioria das vezes, é representado como um vetor associado a um dado;
- **Labels:** valores associados aos dados;
- **Hiperparâmetros:** parâmetros ajustáveis usados pelos algoritmos de aprendizagem;
- **Training Sample:** dados usados para treinar o algoritmo;

- *Test Sample*: dados usados para testar o algoritmo de aprendizagem, devem ser usados dados diferentes dos usados para treino para o resultado não ser “enganador”;
- *Loss Function*: função que mede a diferença entre o dado previsto e a sua verdadeira *label*.

4.1.2 Dados

Apesar dos algoritmos de *Machine Learning* serem utilizados para extrair conhecimento de um dado conjunto de dados, o seu desempenho depende grande parte da qualidade dos dados [48]. Além disso, é importante considerar se o conjunto de dados contém atributos numéricos ou simbólicos, bem como se os dados possuem ruído e imperfeições, como valores incorretos, inconsistentes ou duplicados, pois são variáveis que também influenciam o desempenho dos algoritmos.

Caraterização dos Dados

Os atributos de um conjunto de dados podem ser caraterizados de acordo com o seu tipo e a sua escala. O tipo de um atributo define a representação deste, e pode ser:

- Quantitativo contínuo: pode assumir um número infinito de valores, que são medidos, mas não são contados (por exemplo, a altura ou peso de uma pessoa);
- Quantitativo discreto: podem assumir um número finito ou infinito contável de valores (por exemplo, o número de identificação fiscal);
- Qualitativo: simbólico (ou categórico), característico de atributos que podem ser associados a categorias (por exemplo, o nome de disciplinas).

Os atributos quantitativos podem assumir valores binários, inteiros ou reais, enquanto os atributos qualitativos são geralmente representados por uma quantidade finita de símbolos ou nomes.

Pré-Processamento dos Dados

O pré-processamento de dados tem o objetivo de melhorar a qualidade de um conjunto de dados através da minimização ou eliminação das irregularidades associadas aos dados [49]. Para facilitar a utilização de certos tipos de algoritmos de aprendizagem, bem como aumentar o desempenho dos mesmos relativamente à construção de modelos mais eficientes [49], são algumas técnicas como:

- *Eliminação Manual de Atributos*: remoção de dados que contém atributos irrelevantes para o problema em questão;
- *Integração de Dados*: quando os dados estão armazenados em locais diferentes, é necessário passar por uma fase de integração dos vários dados para uniformizar os mesmos;

- Limpeza de dados: o conjunto de dados pode ser melhorado através da minimização e eliminação de dados que prejudicam e/ou dificultam o desempenho dos algoritmos, podendo ser:
 - Dados incompletos: dados que ausentam valores de atributos cruciais para o desempenho do algoritmo, sendo necessário eliminar estes dados ou então completar os atributos que estão ausentes;
 - Dados inconsistentes: dados que contém os mesmos valores de atributos, mas o resultado esperado é diferente, por isso é necessário eliminar estes dados;
 - Dados redundantes: quando vários dados têm os mesmos valores em todos os seus atributos, são considerados dados redundantes. É aconselhável eliminar todos os dados redundantes;
 - Dados com ruído: dados que, numa primeira análise, não pertencem à distribuição que gerou os dados.
- Transformação de Dados: alguns tipos de algoritmos de ML apresentam limitações relativamente à manipulação de valores de um determinado tipo, bem como uma dependência no intervalo de variação dos valores numéricos para um bom desempenho. Por isso, existem técnicas que podem ajudar, como:
 - Conversão Simbólico-Numérico: conversão de valores simbólicos para numéricos;
 - Conversão Numérico-Simbólico: conversão de valores numéricos para valores simbólicos;
 - Normalização: processo para normalizar um conjunto de dados, alterando a amplitude ou a distribuição.
- Redução de dimensionalidade: dados com um número de atributos significativamente grandes podem causar problemas, pois a maioria dos algoritmos são sensíveis à dimensionalidade da amostra. Por essa razão, é possível realizar duas abordagens:
 - Agregação: combinação de atributos utilizando funções lineares e não lineares;
 - Seleção de atributos: seleção de atributos, descartando os restantes.

4.1.3 Tipos de Aprendizagem

Em *Machine Learning* são caracterizados quatro tipos diferentes:

- Aprendizagem Supervisionada (*Supervised Learning*): o algoritmo precisa que os dados sejam caracterizados, ou seja, cada dado possui um resultado esperado, *label*, mapeando assim ambos através de uma função. Esta aprendizagem divide-se em classificação e regressão;
 - Classificação: utilizam-se dados previstos ou dados categóricos e é necessário ter tantas instâncias quanto possível que já estejam classificadas;

- Regressão: utilizado quando é necessário prever valores contínuos, normalmente valores inteiros e decimais.
- Aprendizagem Não Supervisionada (*Unsupervised Learning*): o algoritmo agrupa o conjunto de dados de entrada, não necessitando de mapear com o resultado esperado. Esta aprendizagem divide-se em *clustering*, redução de dimensionalidade e regras de associação;
 - *Clustering*: organizar os dados em grupo de acordo com a semelhança das características;
 - Redução da dimensionalidade: generalizar e utilizar apenas as características mais determinantes;
 - Regras de associação: identificar determinadas sequências;
- Aprendizagem Semi Supervisionada (*Semi Supervised Learning*): o algoritmo usa uma combinação de dados de entrada em que alguns estão caracterizados (ou seja, têm um resultado esperado associado) e dados não caracterizados;
- Aprendizagem com base em reforço (*Reinforcement Learning*): o algoritmo vai-se ajustando à medida que vão existindo novas interações com o ambiente.

4.1.4 Algoritmos e Modelos

Os algoritmos de *Machine Learning* têm de ser selecionados com base na finalidade do problema e na informação que existe para o mesmo. Existem algoritmos mais adequados para determinadas tarefas, pelo que é possível agrupar os algoritmos de ML de acordo com o tipo de aprendizagem necessária. De seguida explicam-se alguns dos algoritmos e modelos de *Machine Learning* que foram estudados e aplicados para obter a solução desejada.

Support Vector Machines

As *Support Vector Machines* (SVM) são técnicas supervisionadas que permitem fazer classificação linear ou não linear, regressão, ou deteção de *outliers*. Tem como objetivo a separação de uma grande quantidade de dados utilizando o conceito de margens e distanciamento. Esta técnica, com base no princípio da maximização da margem dos hiperplanos, foi apresentada em 1992 por Bernhard E. Boser, Isabelle M. Guyon e Vladimir N. Vapnik [50].

Os vetores de suporte são pontos no conjunto de dados que têm a menor distância até ao hiperplano (ou plano da superfície), pelo que são os casos mais difíceis de classificar. Os vetores de suporte têm influência direta na localização ótima da superfície de decisão[50].

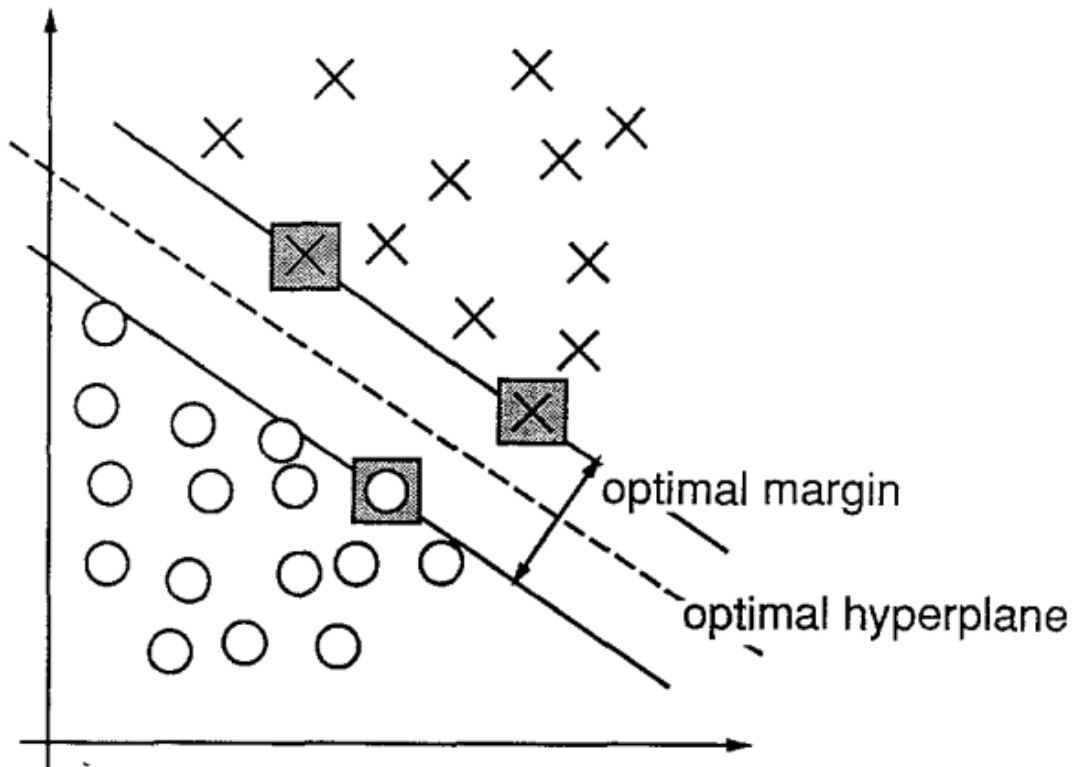


Figura 26 – Identificação do hiperplano ideal e dos vetores de suporte [50]

O desempenho das SVM depende muito dos hiperparâmetros que devem ser ajustados conforme o tipo de dados que se está a lidar. Assim sendo, destacam-se os seguintes:

- C: é um parâmetro de regulação, controlando o tamanho da margem da superfície de decisão. Quanto maior for o valor de C, menor será o tamanho da margem, sendo necessário ter atenção ao *overfitting*. Por outro lado, quanto menor for o valor de C, maior será o tamanho da margem, podendo causar *underfitting*;
 - *Overfitting*: quando o algoritmo não consegue generalizar para novos dados, ou seja, tem uma performance boa nos dados de treino, mas uma performance fraca com dados nunca vistos;
 - *Underfitting*: quando o algoritmo tem dificuldade tanto com os dados de treino, bem como a generalizar para novos dados.

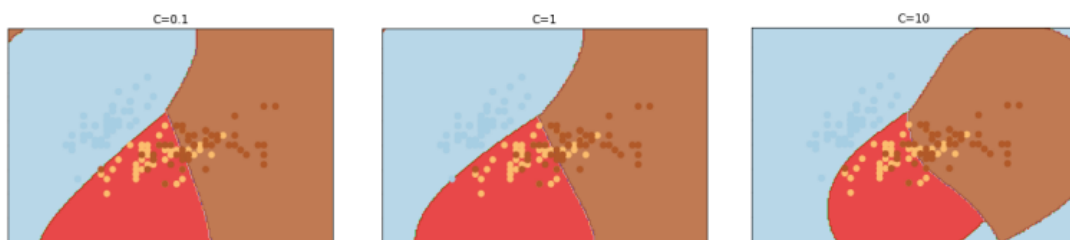


Figura 27 - Exemplos da influência do C nos limites de decisão [51]

- Kernel: função utilizada para transformar o conjunto de dados que define a similaridade dos mesmos. Pode ser definido como linear, polu, rbf, sigmoid, precomputed ou fornecer o próprio *callable*.

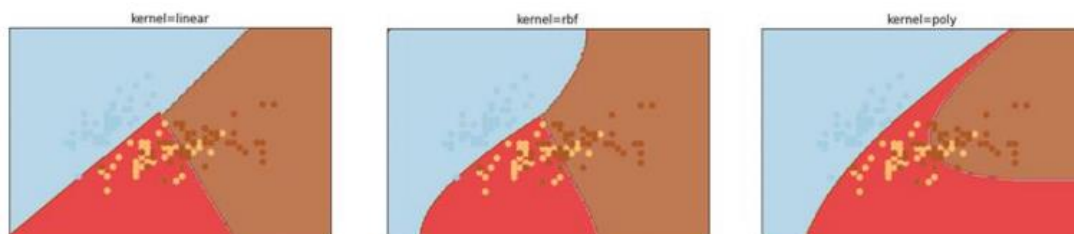


Figura 28 - Exemplos da influência do *kernel* nos limites de decisão [51]

- Gamma: define o peso da influência de um dado de treino, pelo que valores baixos de γ correspondem a uma influência “alargada” e valores altos a uma influência “próxima”. Este parâmetro aplica-se às funções *kernel* RBF (*Radial Basis Function*), sigmoide e polinomial.

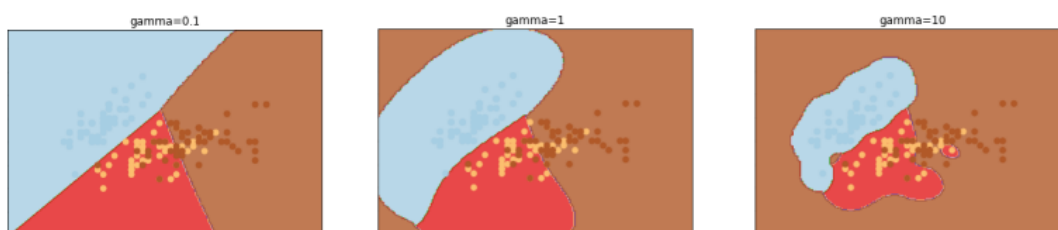


Figura 29 - Exemplos da influência do *gamma* nos limites de decisão [51]

As SVM são eficazes em espaços de grande dimensionalidade e em termos de memória, pois apenas utilizam os vetores de suporte para determinar os limites de decisão e não o conjunto de dados inteiro.

Neural Networks

Uma *Neural Network* (NN) é um conjunto interligado de elementos de processamento simples, com o nome de unidades ou nós, cuja funcionalidade tem inspiração no conhecimento atual do sistema nervoso animal, embora não tenha representação detalhadamente realística [52]. A capacidade de processamento da rede é armazenada na força das ligações entre as unidades, também conhecida por pesos. Estes pesos são obtidos através de um processo de adaptação ou aprendizagem de um conjunto de dados [53].

As unidades (ou neurónios artificiais) da NN representam a unidade de processamento fundamental de uma *Neural Network*, apresentando um modelo simples do neurónio artificial. Cada terminal de entrada e os valores recebidos com um determinado peso são combinados por uma função matemática que equivale ao processamento realizado pela soma. O valor de saída corresponde à resposta do neurónio aos valores de entrada, pelo que várias funções

podem ser utilizadas. A combinação dos valores de entrada pode ser descrita pela seguinte equação [52]:

$$y = \sum_{i=1}^n x_i w_i$$

, onde y representa a entrada total recebida pelo neurónio, x_i representa o valor presente na ligação de entrada i , w_i representa o valor do peso presente na ligação de entrada i e n representa o número de entradas no neurónio.

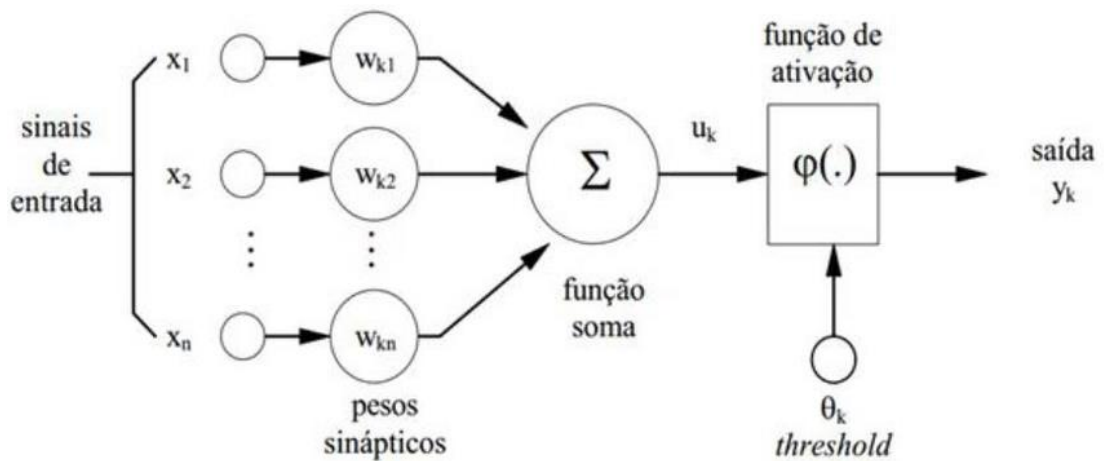


Figura 30 - Representação do funcionamento de um nó da *Neural Network* [54]

As *Neural Networks* podem apresentar várias camadas com diferentes tipos de ligações, pelo que podem ser utilizadas para problemas de regressão e classificação. Uma *NN* com várias camadas apresenta uma camada de entrada que recebe os valores de entrada para processar, pelo menos uma camada invisível de processamento responsável por efetuar o processamento dos dados e uma camada de *output*, que converte o processamento da rede no formato de saída pretendido [52].

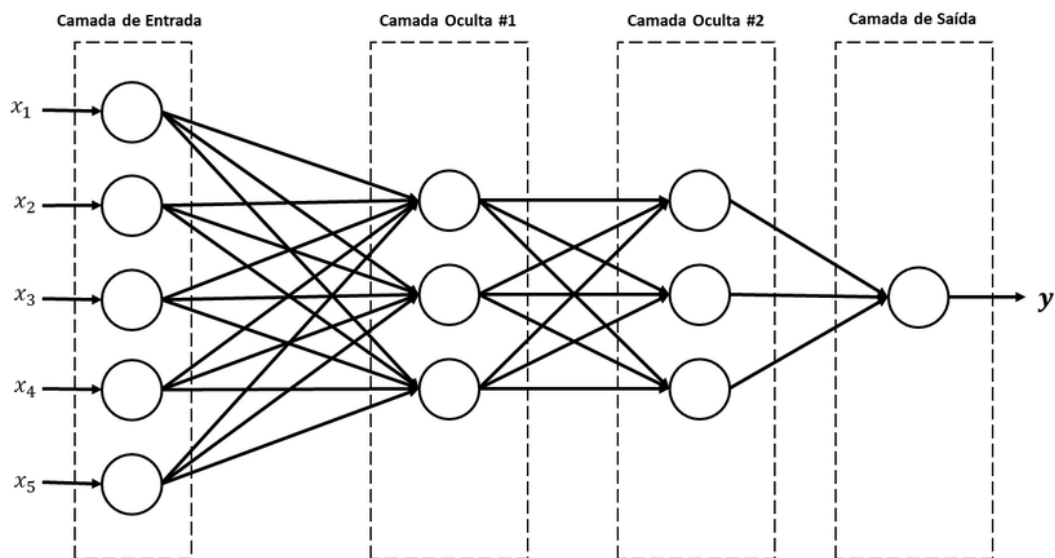


Figura 31 - Representação do funcionamento de uma NN com 2 camadas ocultas [55]

A vantagem da utilização de uma *Neural Network* com várias camadas é o facto de ser capaz de aprender modelos não lineares e também conseguir efetuar aprendizagem em tempo real. No entanto, podem exigir a configuração de uma quantidade maior de hiperparâmetros.

Convolutional Neural Networks

Convolutional Neural Networks (CNN) é um algoritmo de *Deep Learning* amplamente utilizado no reconhecimento de imagem porque são mais precisos que o julgamento humano [56]. Estas redes são uma construção matemática composta por três tipos de camadas: *convolution*, *pooling* e *fully connected layers*. As duas primeiras, *convolution* e *pooling*, desempenham a extração de recursos, e a terceira, *fully connected layer*, mapeia os recursos extraídos na saída final, como classificação.

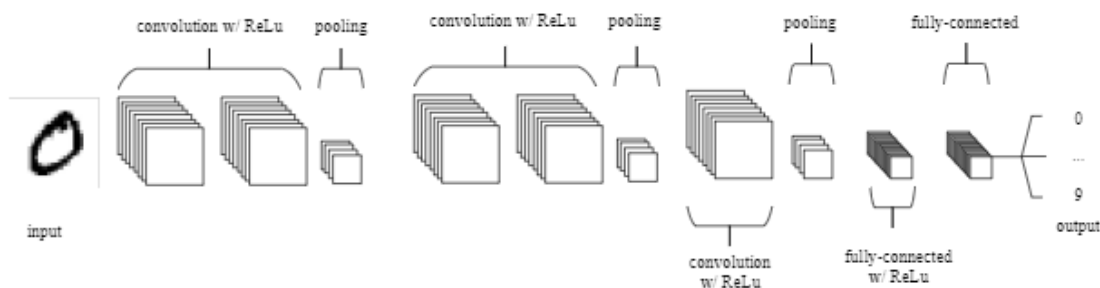


Figura 32 - Camadas que compõe as CNN [57]

Em imagens digitais, a *convolution layer* armazena os valores dos pixéis em um *array* de números e uma matriz de números chamada kernel é aplicada em cada posição da imagem, o que torna as CNN's altamente eficientes no processamento de imagens. O *output* de uma camada torna-se o *input* da próxima camada, levando a que os recursos extraídos sejam cada

vez mais complexos [58]. Dois hiperparâmetros chave que definem a operação de convolução são o tamanho e o número de kernels. O primeiro é tipicamente 3×3 , mas pode ser 5×5 ou 7×7 . O último é arbitrário e determina a profundidade dos mapas de recursos de saída.

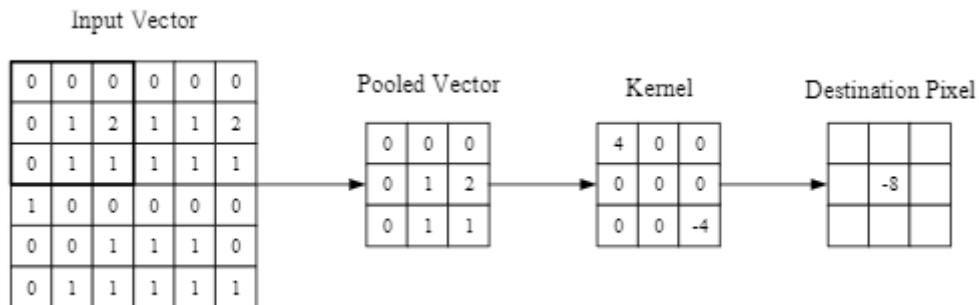


Figura 33 - Operação de convolução [57]

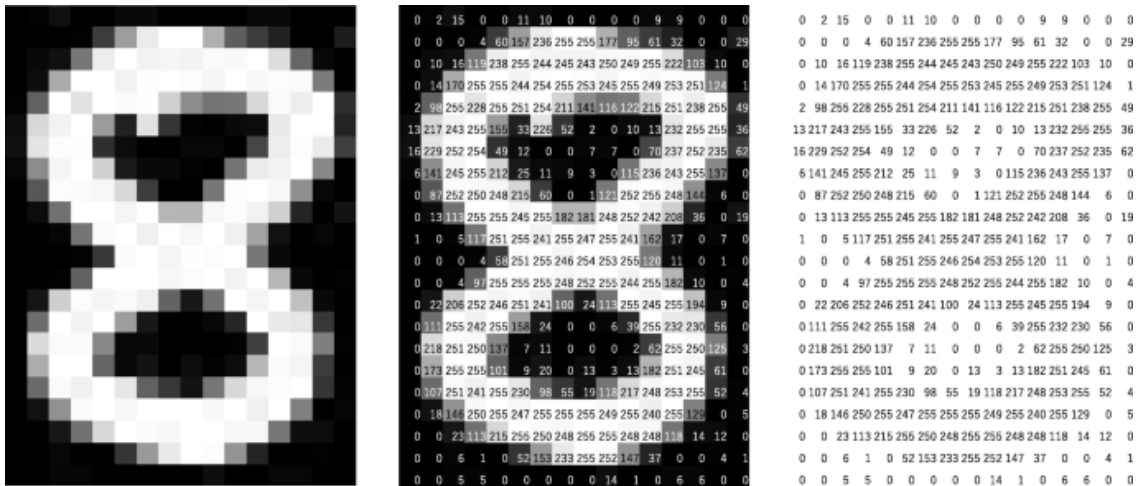


Figura 34 - Modo como um computador vê uma imagem [58]

Como se verifica na Figura 34, um computador vê uma imagem como um *array* de números. A matriz representada à direita contém números entre 0 e 255, cada um dos quais corresponde ao brilho do pixel na imagem da esquerda. Ambos são sobrepostos na imagem do meio [58]. No caso de imagens coloridas, passa a ser uma matriz de 3 dimensões, em que é dado o valor de cada pixel para o R, G e B.

A *pooling layer* fornece uma operação típica de *downsampling* que reduz a dimensionalidade dos mapas de recursos, de modo a diminuir o poder computacional necessário para o processamento de dados. Além disso, é útil para extrair características dominantes que são invariantes rotacionais e posicionais, mantendo assim o processo efetivo de treino do modelo.

Existem dois tipos de *pooling*: *max pooling* e *average pooling*. *Max pooling* retorna o valor máximo da parte da imagem coberta pelo *kernel*, e o *average pooling* retorna a média de todos os valores da parte da imagem coberta pelo *kernel*. O *max pooling* funciona como supressor de ruído, descartando completamente ativações ruidosas e também realiza redução de ruído

juntamente com redução de dimensionalidade. O *average pooling* realiza redução de dimensionalidade como um mecanismo de supressão de ruído. Portanto, o *max pooling* tem um desempenho muito melhor que o *average pooling* [57].

Após a imagem de entrada ser convertida de forma adequada para o *multi-level perceptron*, é necessário achatar a imagem em um vetor coluna, que vai alimentar uma rede neuronal *feed-forward* e *backpropagation* aplicada a cada iteração do treino. Ao longo de uma série de épocas, o modelo é capaz de distinguir características em imagens e classificá-las usando a técnica de classificação Softmax.

MobileNetV2

Em 2017, a Google apresentou o MobileNetV1, uma família de redes neurais de visão computacional projetada com dispositivos móveis para suportar classificação, detecção, entre outros. Um ano mais tarde, 2018, a Google anunciou o MobileNetV2 que pretendia impulsionar a próxima geração de aplicações móveis. Esta nova rede neuronal mostrou ser uma melhoria significativa em relação ao MobileNetV1, destacando-se no reconhecimento visual móvel, incluindo classificação detecção de objetos e segmentação semântica [59].

MobileNetV2 utiliza um modelo de arquitetura de CNN para classificar imagens e é especial porque usa muito menos poder de computação para ser executado. Este facto torna-o perfeito para dispositivos móveis com recursos de computação limitados, sistemas incorporados e computadores para serem executados sem GPUs. Este modelo é treinado usando o conjunto de dados ImageNet. A arquitetura MobileNetV2 utiliza uma estrutura residual invertida onde a entrada e a saída dos blocos residuais são camadas *bottleneck* finas.

Estas camadas codificam as entradas e saídas intermédias do modelo, enquanto a camada interna encapsula a capacidade do modelo de transformar conceitos de nível inferior como pixéis em descritores de nível superior com categorias de imagem. Como nas conexões residuais tradicionais, são adicionados atalhos (*shortcuts*) que permitem que o treino seja mais rápido e mais preciso.

MobileNetV2 também usa convoluções leves para filtrar recursos na camada de expansão. Finalmente, remove não linearidades nas camadas estreitas [60].

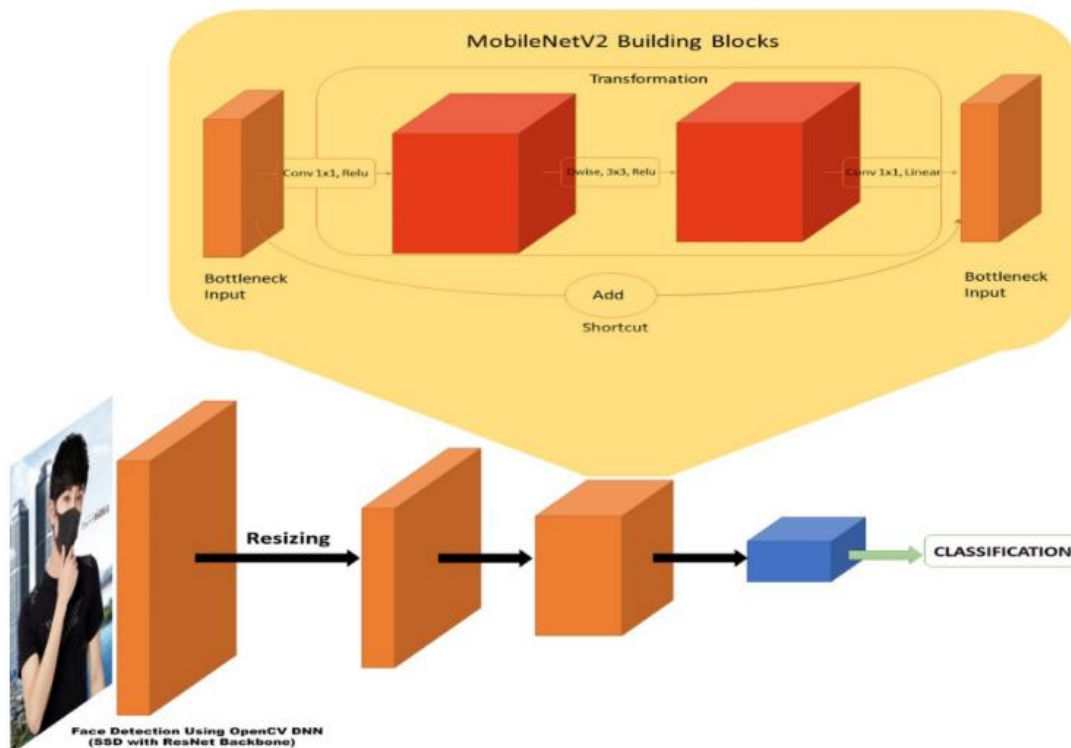


Figura 35 - Arquitetura da MobileNetV2 [61]

A arquitetura da MobileNetV2 contém uma camada inicial *fully convolution* com 32 filtros, seguida por 19 camadas *residual bottleneck*. A arquitetura foi adaptada para diferentes pontos de desempenho, usando a resolução da imagem de entrada o multiplicador de largura como hiperparâmetros ajustáveis, que podem ser ajustados dependendo da precisão desejada ou das compensações de desempenho [61].

O MobileNetV2 supera o MobileNetV1 com maior precisão e menor latência. Além disso, é um extrator de recursos muito eficaz para detecção e segmentação de objetos. No caso de detecção, quando emparelhado com o Single Shot Detector Lite, o MobileNetV2 é cerca de 35% mais rápido com a mesma precisão que o MobileNetV1.

Detectron2

Detectron2 é o pacote de detecção de objetos, segmentação e estimativa de pose de código aberto da Meta AI. É o sucessor do Detectron e maskrcnn-benchmark [62]. Esta biblioteca inclui novos recursos, como segmentação panóptica, Densepose, Cascade R-CNN, *rotated bounding boxes*, PointRend, DeepLab, entre outros.

O Detectron2 possui um design modular e extensível, e fornece um treino mais rápido em um ou vários servidores GPU. Já se encontra implementado em PyTorch, o que facilita a iteração no design e experiências nos modelos. É possível modificar o código existente, usá-lo para fins privados, científicos e até comerciais.

Dada uma imagem de entrada, a biblioteca Detectron2 pode retornar rótulos, caixas delimitadoras, pontuações de confiança, máscaras e esqueletos de objetos.

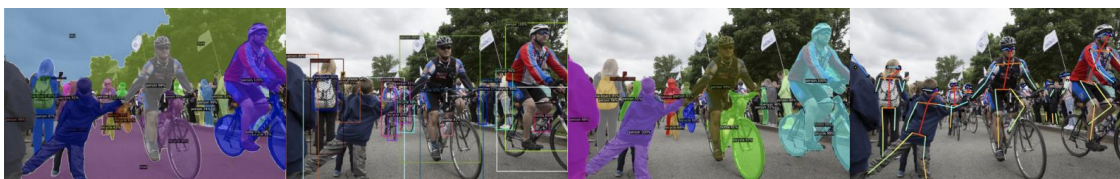


Figura 36 - Algumas funcionalidades do Detectron2 [63]

Oferece o *model zoo* com implementações como Mask R-CNN e R-CNNs em geral, juntamente com RetinaNet.

PointRend

PointRend (*Point-based Rendering*) trata a segmentação como um problema de renderização, usando uma estratégia de segmentação para selecionar de forma adaptativa um conjunto de pontos não uniformes e calcular rótulos. O PointRend pode ser incorporado de forma flexível em meta-arquiteturas de segmentação de instância comuns, como Mask R-CNN, e meta-arquiteturas de segmentação semântica, como Fully Convolutional Networks [64].

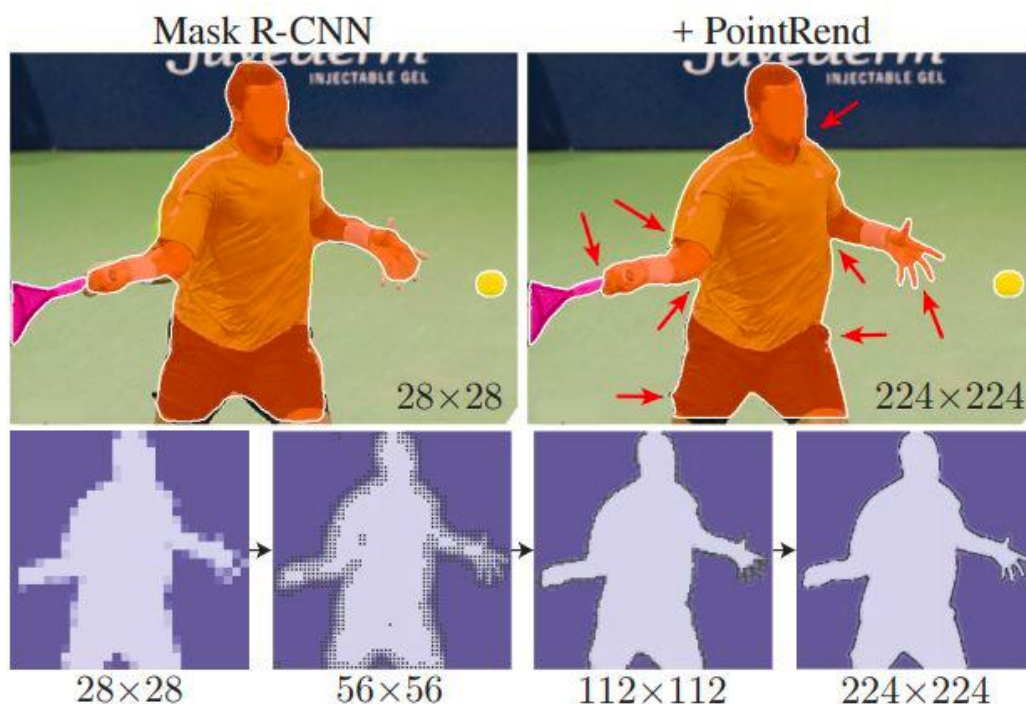


Figura 37 – Segmentação de instância com PointRend [64]

Ao calcular mapas de segmentação de alta resolução, a estratégia de subdivisão do PointRend é uma ordem de magnitude mais eficiente em operações de ponto flutuante em comparação

com a computação direta. O PointRend não realiza previsões em todos os pontos da grade de saída, apenas em pontos cuidadosamente selecionados.

PointRend atinge maior nitidez em limites de objetos complicados, como dedos, e pode ser adicionado na segmentação semântica e de instância. PointRend supera o Mask R-CNN em tarefas de segmentação de imagem nos conjuntos de dados de referência COCO e Cityscapes. A eficiência deste modelo permite resoluções de saída que são impraticáveis em termos de memória ou computação em comparação com as abordagens existentes.

É composto por três componentes principais:

- *Point Selection Strategy* - uma estratégia de seleção de pontos seleciona um pequeno número de pontos de valor real para fazer previsões, evitando computação excessiva para todos os pixels na grade de saída de alta resolução;
- *Point-Wise feature Representation* - para cada ponto selecionado, é extraída uma representação de recurso pontual.
- Point head - uma pequena rede neural treinada para prever um rótulo a partir dessa representação de recurso pontual, independentemente para cada ponto.

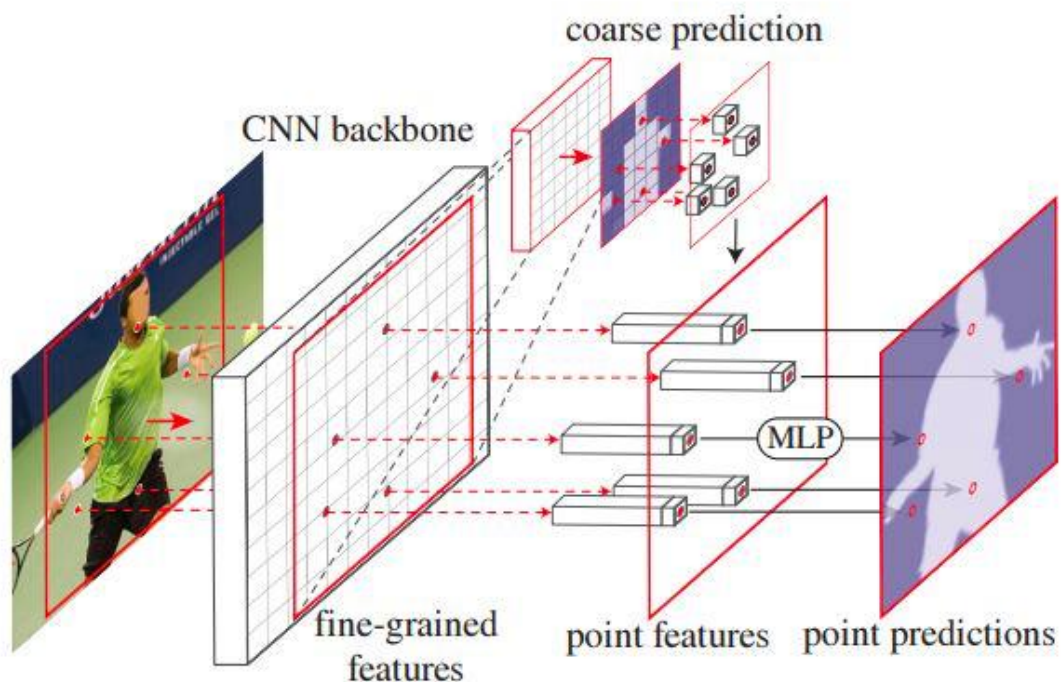


Figura 38 - Fluxo de processo do modelo PointRend [64]

Uma imagem de entrada é enviada para a rede CNN para obter uma saída da *coarse prediction*. Esta saída tende a ser de baixa resolução. O vetor gerado pela *coarse prediction* é enviado como entrada para o modelo PointRend, onde é aumentado duas vezes usando interpolação bilinear e N pontos incertos são gerados usando a estratégia de seleção de pontos PointRend. São feitas

novas previsões nesses pontos utilizando uma estratégia de previsão pontual com um *multi-layer perceptron* e este processo é repetido até ser atingida a resolução de saída desejada.

MediaPipe Objectron

MediaPipe Objectron é uma solução móvel desenvolvida pela Google para detecção de objetos 3D em tempo real para objetos do dia a dia. Ele detecta objetos em imagens 2D e estima a sua posição através de um modelo de ML, treinada em um conjunto de dados Objectron [65].

Este conjunto de dados é uma coleção de cliques de vídeo curtos centrados em objetos, que são acompanhados por metadados de sessão AR que incluem poses de câmera, nuvens de pontos esparsas e caracterização das superfícies planas no ambiente circundante. Em cada vídeo, a câmera move-se em redor do objeto, de modo a captar vários ângulos. Os dados possuem caixas delimitadoras (*bounding box*) 3D anotadas manualmente para cada objeto, que descrevem a posição, orientação e dimensões do objeto.

O conjunto de dados Objectron possui quinze mil vídeos anotados e mais de quatro milhões de imagens anotadas para as seguintes categorias: *bikes, books, bottles, cameras, cereal boxes, chairs, cups, laptops* e *shoes*. A solução de detecção de objetos 3D apenas está disponível para as categorias *shoes, chairs, cups* e *cameras*.



Figura 39 - Exemplos de resultados do Objectron

Este modelo possui uma arquitetura codificador-decodificador baseada em MobileNetV2, as redes neurais do Google para classificação e detecção visual. Usando essas redes, a projeção 2D da imagem e um algoritmo de estimativa 3D, o modelo pode processar uma saída 3D do referido objeto.

MiDaS

MiDaS é um modelo de ML que calcula a profundidade inversa relativa de uma imagem. Este modelo foi treinado em dez *datasets* (ReDWeb, DIML, Movies, MegaDepth, WSVD, TartanAir, HRWSI, ApolloScape, BlendedMVS, IRS) com otimização multiobjetivo.

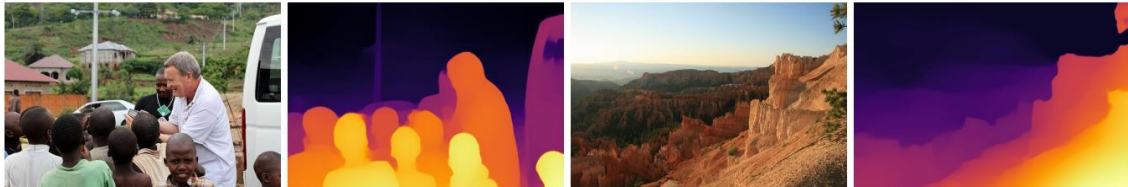


Figura 40 - Imagem e mapa de profundidade correspondente obtido pelo MiDaS [66]

O mapa de profundidade é uma imagem em que cada pixel contém informações de profundidade ao invés de RGB, e normalmente é apresentado como uma imagem em escala de cinza. As informações de profundidade significam a distância da superfície dos objetos da cena a partir de um ponto de vista.

Os mapas de profundidade podem ser usados em várias situações, como mapeamento de sombras, renderização 3D, reconstrução 3D, entre outros.

Point Cloud

Point Cloud é um conjunto de pontos no espaço, que permite representar formas ou objetos 3D. Cada ponto possui o seu conjunto de coordenadas cartesianas (x , y , z). Point Cloud é um tipo de geometria que permite armazenar grandes quantidades de dados, normalmente recolhidos de aplicações LiDAR. O LiDAR permite uma recolha rápida e precisa de dados.

A geometria do Point Cloud possibilita o processamento rápido e eficiente de uma grande coleção de vértices no espaço 3D que representam as superfícies externas de objetos. Quando se unem estes vértices, forma-se um modelo capaz de se transformar e visualizar.



Figura 41 – Point Cloud [67]

O Point Cloud possui operações como afinamento, divisão e combinação, de modo a produzir um conjunto de vértices mais utilizável. A cada vértice estão associadas várias componentes que contêm um valor descritivo do ponto. Esses valores das componentes podem ser usados para classificar diferentes seções do conjunto de pontos contido na geometria da *point cloud*.

Point Cloud tornaram-se em um dos formatos de dados mais significativos para representação 3D e estão a ganhar cada vez mais popularidade devido à disponibilidade de dispositivos de aquisição, bem como ao aumento da aplicação em áreas como robótica, direção autônoma, realidade aumentada e virtual.

Regressão Linear

A regressão é uma das técnicas estatísticas mais potentes e de utilização mais frequente. A análise da regressão estuda a relação entre uma variável chamada variável independente e outras variáveis independentes. Esta relação é representada por um modelo matemático que pode ser designado por modelo de regressão linear simples ou modelo de regressão linear múltipla.

Designa-se por modelo de regressão linear simples caso defina uma relação linear entre a variável dependente e uma variável independente. Se em vez de uma, forem incorporadas várias variáveis independentes, o modelo passa a denominar-se modelo de regressão linear múltipla [68].

A fórmula de uma regressão linear simples é representada por:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

, sendo y a variável dependente, x é a variável independente, β_0 é a interseção na origem da reta, e β_1 é o declive da reta. ε é um termo aleatório que capta a influência de outros pequenos fatores que influenciam y , para além do x .

O modelo de regressão linear múltipla é definido pela expressão:

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \varepsilon$$

, sendo x_1, \dots, x_k as variáveis explicativas ou independentes medidas sem erro (não aleatórias), ε é uma variável aleatória residual na qual se procuram incluir todas as influências no comportamento da variável y que não podem ser explicadas linearmente pelo comportamento das variáveis x_1, \dots, x_k e os possíveis erros de medição, β_0, \dots, β_k são parâmetros desconhecidos do modelo, e y é a variável dependente.

A análise de correlação está intimamente ligada à análise de regressão. A análise de correlação é usada para medir o grau de associação (linear) entre as variáveis quantitativas, e é dada pela raiz quadrada de R^2 . Pode designar-se por coeficiente de correlação simples se apenas estiver envolvida uma variável independente, ou por coeficiente de correlação múltiplo caso estejam envolvidas pelo menos duas variáveis independentes.

4.1.5 Ferramentas

Nesta secção são abordadas as ferramentas utilizadas na implementação das abordagens estudadas nesta dissertação.

Python

O Python foi desenvolvido por Guido van Rossum em 1991 [69], com o objetivo de otimizar a leitura de códigos e estimular a produtividade de quem os cria. É considerada uma linguagem de programação de alto nível, dinâmica, interpretada, modular, multiplataforma e orientada a objetos [69]. O Python possui um grande número de bibliotecas, nativas e de terceiros, a sua utilização e distribuição é gratuita, e é possível estudar e efetuar alterações no código fonte da linguagem.

O Python não é uma linguagem fortemente tipificada, permitindo a utilização de variáveis sem a declaração destas e determinando o tipo implicitamente. Também utiliza a indentação como estrutura de controlo, pelo que não se utiliza chavetas para definir blocos de código. A linguagem de programação Python também não obriga à definição de classes, apesar de ser possível definir quando é necessário ou conveniente defini-las [69].

A escolha desta linguagem de programação para o desenvolvimento deste projeto teve por base o facto de conter uma biblioteca específica para a utilização de algoritmos de aprendizagem, com o nome de scikit-learn [70]. Outro fator de seleção é o facto de o Python ser uma das linguagens mais utilizadas na área científica devido à facilidade de aprendizagem e utilização da mesma [71].

Scikit-Learn

O scikit-learn é uma ferramenta gratuita, simples e eficiente para análise preditiva de dados[70]. Esta biblioteca fornece muitos algoritmos de aprendizagem supervisionada e não supervisionada, bem como técnicas para análise de dados, desde pré-processamento até à validação cruzada.

Esta biblioteca é acessível a todos e é reutilizável em vários contextos. Foi contruída em NumPy, SciPy e Matplotlib, e é *open source* [70].

Tkinter

Tkinter é uma *framework* de *Graphical User Interface* embutido na biblioteca padrão do Python. O Python disponibiliza outras bibliotecas para criar interfaces gráficas como o Kivy, o Python Qt e o wxPython, mas o Tkinter é o mais utilizado de todos. Esta biblioteca dispõe de diversos controlos como menus, botões, caixas de texto, entre outros, que podem ser utilizados na interface gráfica. Estes controlos são chamados de *widgets*.

A interface gráfica criada com esta biblioteca é alvo de algumas críticas por não parecer moderna e atrativa. No entanto, o Tkinter é relativamente leve e fácil de utilizar em comparação com outras *frameworks*.

Como o objetivo principal desta dissertação não era a implementação de uma aplicação móvel, deu-se prioridade aos restantes objetivos definidos e por isso, esta biblioteca revelou-se uma boa opção para a criação de uma interface funcional e multiplataforma onde fosse possível disponibilizar o feedback do sistema para o utilizador.

Open3D

O Open3D é uma biblioteca *open source* que suporta desenvolvimento rápido de software para processamento de dados 3D, como reconstrução de cenas, visualização e *machine learning* 3D. Pode ser configurado em diferentes plataformas e compilado facilmente a partir da fonte.

O código é limpo, com estilo consistente e mantido por meio de um mecanismo de revisão de código claro. O Open3D tem sido usado em vários projetos de pesquisa publicados e é implementado ativamente na nuvem [72].

Esta biblioteca possui uma estrutura de dados para imagens que suporta várias funções como `read_image`, `write_image`, `filter_image` e `draw_geometries`. Uma imagem Open3D pode ser convertida diretamente de/para um array numpy. Uma Open3D RGBDImage é composta por duas imagens, `RGBDImage.depth` e `RGBDImage.color` [73].

MeshLab

O MeshLab é uma ferramenta *open source* de processamento, visualização e edição de *meshes* triangulares 3D. Fornece um conjunto de ferramentas para edição, limpeza, correção, inspeção, renderização, texturização e conversão de *meshes*. Esta ferramenta oferece recursos para processamento de dados produzidos por ferramentas/dispositivos de digitalização 3D e para preparação de modelos para impressão 3D.

O MeshLab é baseado na biblioteca VCGlib, e está disponível para Windows, macOS e Linux.

Labelme

Labelme é uma ferramenta de anotação *open source escrita* em Python utilizada como suporte na anotação manual de imagens para deteção, classificação e segmentação de objetos. Esta ferramenta permite criar várias formas incluindo polígonos, círculos, retângulos, linhas e pontos. As anotações realizadas nesta ferramenta podem ser gravadas como ficheiros json diretamente da ferramenta.

O repositório Labelme oferece um script python que permite converter anotações em PASCAL VOL, contudo não suporta formatos como YOLO e COCO.

4.1.6 Métodos

Nesta secção são abordados alguns métodos utilizados na implementação das diversas abordagens estudadas para obter a solução desejada para esta dissertação.

Análise de Componentes Principais

Grandes conjuntos de dados são cada vez mais comuns e, muitas vezes, mais difíceis de interpretar. Como referido anteriormente, o conjunto de dados utilizado possuía uma grande quantidade de dados, pelo que mesmo após a redução da sua dimensionalidade, ainda surgiram problemas de memória. Para ultrapassar este obstáculo, utilizou-se o processo de análise de componentes principais (PCA).

O PCA reduz a dimensionalidade de um conjunto de dados, preservando o máximo da variância (ou seja, informações estatísticas) possível. Esta técnica consiste numa transformação linear de dados para um novo sistema de coordenadas, de forma que a maior variância por qualquer projeção de dados fique captada ao longo da primeira coordenada, a segunda maior variância ao longo da segunda coordenada e assim sucessivamente [74].

A redução de variáveis de um conjunto de dados leva naturalmente a uma redução de precisão, contudo conjuntos de dados menores são mais fáceis de explorar e visualizar, e tornam a análise de dados muito mais fácil e rápida para algoritmos de ML [74].

GridSearch

Grid search é a técnica de realizar o ajuste de hiperparâmetros para determinar os valores ideais para um determinado modelo. O desempenho de um modelo depende significativamente do valor dos hiperparâmetros. Como não existe forma de saber antecipadamente os melhores valores para os hiperparâmetros, é crucial tentar todos os valores possíveis para conhecer quais os valores ideais [75].

Realizar este processo manualmente consome uma quantidade considerável de tempo e recursos, pelo que foi utilizada a função GridSearchCV para automatizar o ajuste de hiperparâmetros. Esta função é disponibilizada pela biblioteca scikit-Learn e ajuda a percorrer os hiperparâmetros predefinidos a assim, ajustar o modelo no conjunto de dados de treino.

O GridSearchCV faz uma pesquisa exaustiva com todas as combinações de parâmetros e valores. Deste modo, é possível selecionar a melhor combinação de parâmetros, ou seja, a que tem melhor desempenho é usada para fazer as previsões. De salientar que o tempo de treino e a

avaliação de desempenho cresce exponencialmente com o número de parâmetros e valores dos mesmos.

5 Abordagens Consideradas

No decorrer deste capítulo apresentam-se as várias abordagens aplicadas na implementação do projeto, é realizada uma comparação entre as mesmas e justifica-se a escolha realizada. Para cada abordagem, referem-se os métodos de recolha de dados utilizados, bem como a aplicação dessas abordagens e os resultados obtidos dessas implementações.

5.1 SVM, CNN e MobileNetV2

Para realizar o reconhecimento de alimentos a partir de uma imagem, começou-se por analisar três abordagens frequentemente utilizadas neste contexto: *Support Vector Machine*, *Convolutional Neural Network* e *MobileNetV2*. O estudo destas abordagens foi apresentado na Conferência Europeia de Inteligência Artificial EPIA realizada em setembro de 2022 e é possível aceder ao artigo em [25].

Nesta secção é apresentado o conjunto de dados utilizado e as três técnicas são analisadas, de modo a obter a técnica com melhor desempenho para ser utilizada posteriormente na implementação da solução apresentada nesta dissertação.

5.1.1 Conjunto de Dados

Como já foi referido anteriormente, pretendia-se que o sistema desenvolvido fosse capaz de reconhecer refeições tipicamente portuguesas. Após uma extensa pesquisa por conjuntos de dados que possuíssem imagens de refeições portuguesas, verificou-se que, até à data, não existia nenhum repositório público com estas características. Por essa razão, construiu-se um *dataset* manualmente e, quando foi realizado este estudo, este *dataset* possuía apenas 11

categorias de refeições, totalizando 5528 imagens. Este *dataset* foi publicado no Kaggle [76] e, neste momento, tem 23 categorias, com 6726 imagens. Pretende-se que este repositório seja alimentado com regularidade, para que futuramente seja a fonte de informação de muitos projetos.

bacalhau_bras	batatas_fritas	bolo_chocolate
cachorro	donuts	esparguete_bolonhesa
francesinha	gelado	hamburger
pizza	waffles	

Figura 42 – Categorias utilizadas no caso de estudo

Algumas categorias presentes neste *dataset* como batatas fritas, bolo de chocolate, cachorro, donut, gelado, hambúrguer, pizza e waffles foram recolhidas do conjunto de imagens disponibilizado em [77, 78]. Este *dataset* contém uma vasta variedade de imagens de alimentos, estando dividido em 101 categorias e cada uma possui 1000 imagens.

As restantes categorias – bacalhau à brás, esparguete à bolonhesa e francesinha – foram reunidas a partir de pesquisas no Google Images e em aplicações de recomendações de restaurantes, como Zomato, The Fork e TripAdvisor.

Pré-Processamento dos dados

Devido à existência de variados tamanhos de imagens, foi necessário passar por um processo de normalização do tamanho de todas elas. Para isso, foi calculado o tamanho médio de todas as imagens e, em seguida, todas foram redimensionadas para a metade desse mesmo tamanho, pois caso não fosse metade do tamanho médio continuavam a surgir erros de memória como citados anteriormente.

O resultado foram imagens 286x258 pixels, ao qual se manteve as cores, pois é um fator de grande variância entre os diferentes alimentos. Assim sendo, como os computadores leem as cores como uma série de três valores – vermelho, verde e azul -, e como cada pixel tem, na verdade, três valores para o computador, resultou em um vetor de 286x258x3, ou seja, 221364 atributos para cada imagem.

O conjunto de dados está dividido em subpastas por classe, portanto foi contruído um algoritmo que percorreu todas as subpastas, converteu todas as imagens para o tamanho 286x258x3 e, em seguida, foram convertidas para um *array* do tipo *numpy* e foi armazenada a sua devida *label*. De seguida, os índices foram baralhados para não ficarem as imagens da mesma classe em sequência. Posteriormente, o *array* de imagens e de *labels* foram divididos em 85% para treino e 15% para teste.

De forma a agilizar e reduzir a complexidade do processo de treino e previsão, foi usado o processo de análise de componentes principais (PCA), explicado na secção 4.1.6. Para manter um bom desempenho e tentar perder o mínimo de informação possível, decidiu-se captar pelo

menos 90% da variância. Para isso, foram usadas 500 componentes principais, como se pode verificar na Figura 43. A partir das 500 componentes, o ganho de variância por componente é residual, pelo que não compensa estar a adicionar mais componentes.

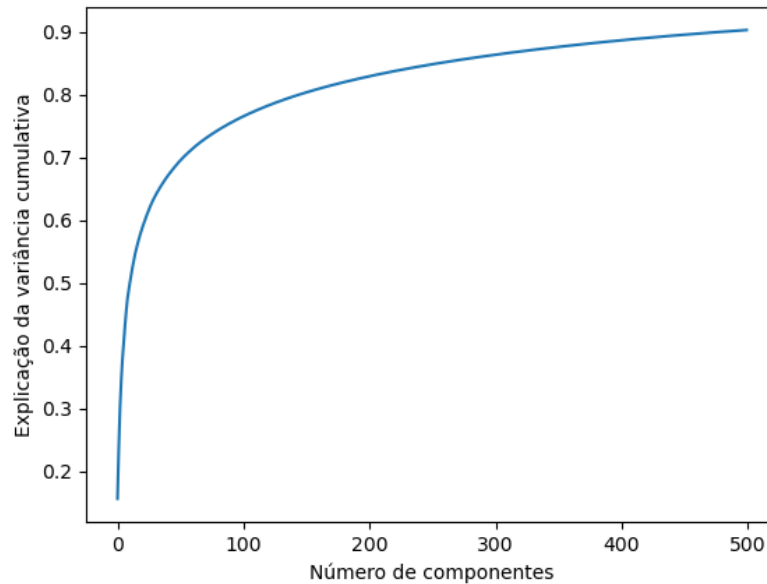


Figura 43 - Demonstração gráfica que com 500 componentes capta 90% da variância

Na Figura 44 é possível observar as imagens originais dos alimentos e as imagens reconstruídas a partir dos 500 componentes extraídos.

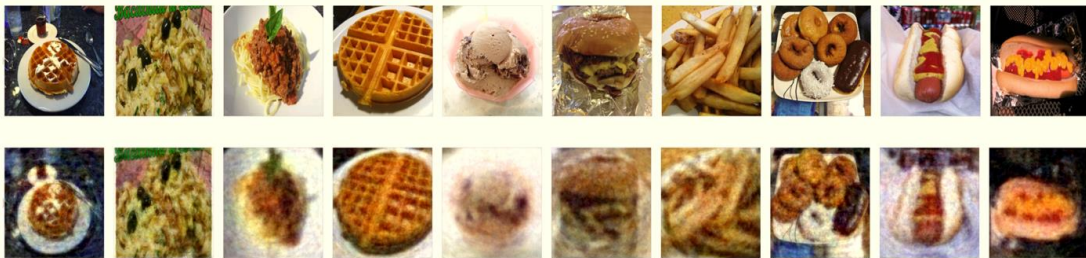


Figura 44 – Exemplos de imagens e a sua reconstrução a partir dos 500 componentes

Verifica-se que, apesar da redução de atributos, as imagens continuam a conter as partes essenciais para a definição de cada um dos alimentos.

5.1.2 Aplicação dos Algoritmos de Aprendizagem

Para a criação de um modelo preditivo capaz de identificar os itens alimentares, foram implementados os algoritmos SVM e CNN, utilizando a biblioteca do scikit-learn.

Nas *Support Vector Machines* foram testados os seguintes hiperparâmetros: função *Kernel*, valor do coeficiente de *C* e valor de *Gamma*.

Para obter os melhores parâmetros, foi utilizado o método GridSearch, explicado na secção 4.1.6, com validação cruzada de 5 *folds*. Com isto, cada combinação existente na grelha foi treinada e testada, de forma a encontrar a melhor combinação de parâmetros. Foram testadas 192 combinações, estando na Tabela 8 demonstrados os melhores resultados obtidos no processo de otimização dos hiperparâmetros.

Tabela 8 – Resultado do processo de otimização de hiperparâmetros SVM

Hiperparâmetro	Valor
Kernel	RBF
C	10
Gamma	0.0001

Nas *Convolutional Neural Networks*, testaram-se os seguintes hiperparâmetros: *Hidden Layer Size*, *Activation*, *Solver*, *Alpha* e *Max_iter*. Também foi utilizado o método GridSearch com validação cruzada, testando 144 combinações, sendo que a melhor combinação está demonstrada na Tabela 9.

Tabela 9 - Resultado do processo de otimização de hiperparâmetros CNN

Hiperparâmetro	Valor
Hidden Layer Size	(500, 500, 500, 500)
Activation	relu
Solver	lbfgs
Alpha	1
Max_iter	750

Para avaliar o MobileNetV2, foram criados geradores utilizando ImageDataGenerator que permite aumentar a quantidade de dados em tempo real enquanto o modelo está a ser treinado. Basicamente, um lote de imagens de entrada é apresentado ao ImageDataGenerator e este transforma cada imagem do lote através de uma série aleatória de translações, rotações, entre outros. O lote transformado é então retornado à função de chamada.

Após os dados estarem prontos para alimentar o MobileNetV2, definiram-se alguns argumentos para criar o modelo, como é possível averiguar na Tabela 10.

Tabela 10 - Valores usados na criação do modelo MobileNetV2

Argumento	Valor
input_shape	(224, 224, 3)
include_top	False
weights	imagenet

pooling

avg

A saída do modelo pré-treinado passou por duas camadas densas com 128 neurónios e ativação de relu. Em seguida, foi definida a camada de classificação final, que armazena os valores reais de probabilidade para cada classe. Esta camada foi definida com 11 classes e com ativação softmax. O resultado desta camada são 11 probabilidades e a maior será a classificação. Após a criação do modelo com as entradas e saídas definidas, iniciou-se a fase de treino com a compilação do modelo. O modelo foi configurado com os valores presentes na Tabela 11.

Tabela 11 - Valores usados na compilação do modelo MobileNetV2

Argumento	Valor
optimizer	adam
loss	categorical_crossentropy
metrics	['accuracy']

A função de perda foi configurada como entropia cruzada categórica ao invés de entropia cruzada categórica esparsa, pois quando são utilizados ImageDataGenerators, as classes são codificadas como vetores, então ao invés de serem passados inteiros para a classe, são passados vetores. De seguida, o modelo foi treinado para 100 *epochs*.

5.1.3 Resultados

Sendo este caso um problema de classificação, foram aplicadas métricas de classificação para avaliar a performance dos algoritmos. Assim sendo, foram criadas matrizes de confusão que resumem os resultados do teste do classificador. A matriz de confusão mostra o número de previsões corretas e incorretas feitas por um modelo em comparação com os resultados reais.

Na matriz de confusão representada na

Figura 45, verifica-se que algumas categorias como cachorro, donuts, pizza e waffles, possuem poucas previsões corretas. No caso dos waffles, o modelo previu corretamente 12 imagens, contudo previu que outras 12 imagens da categoria waffles eram cachorros. Não existe qualquer semelhança entre estas duas categorias, e é importante ressaltar que não existem imagens no conjunto de dados com ambos os alimentos na mesma imagem.

Label Prevista	bacalhau_bras	batatas_fritas	bolo_chocolate	cachorro	donuts	esparquete_bolonhesa	francesinha	gelado	hamburguer	pizza	waffles
bacalhau_bras	87	14	2	2	3	3	1	2	3	2	4
batatas_fritas	15	65	0	18	13	9	13	6	21	25	11
bolo_chocolate	2	0	79	5	14	5	0	13	10	6	11
cachorro	10	11	3	33	12	17	8	10	15	13	15
donuts	4	8	14	10	29	7	6	10	8	14	18
esparquete_bolonhesa	5	9	10	11	11	65	6	7	10	8	28
francesinha	3	16	0	13	5	14	92	4	13	7	8
gelado	12	10	11	20	31	9	6	85	16	23	19
hamburguer	9	11	11	11	6	10	13	12	44	5	10
pizza	2	11	5	10	11	7	4	9	10	38	13
waffles	5	9	6	12	5	3	4	3	4	8	12
Label Verdadeira	bacalhau_bras	batatas_fritas	bolo_chocolate	cachorro	donuts	esparquete_bolonhesa	francesinha	gelado	hamburguer	pizza	waffles

Figura 45 – Matriz de confusão SVM

A Figura 46 mostra a matriz de confusão criada para a CNN. Houve uma ligeira melhoria das previsões quando comparadas com as previsões obtidas pela SVM. A categoria waffles também não se destacou nas previsões resultantes da aplicação da CNN, como já tinha acontecido com o algoritmo SVM. Ainda se verificam muitos falsos negativos e falsos positivos.

Label Prevista	bacalhau_bras	batatas_fritas	bolo_chocolate	cachorro	donuts	esparquete_bolonhesa	francesinha	gelado	hamburguer	pizza	waffles
bacalhau_bras	94	19	4	6	3	3	3	3	7	1	1
batatas_fritas	10	54	0	16	11	10	9	5	11	21	6
bolo_chocolate	1	1	82	7	14	6	0	12	16	5	11
cachorro	9	15	1	36	11	19	7	13	7	8	14
donuts	6	8	14	11	40	5	3	14	9	12	18
esparquete_bolonhesa	1	5	7	10	9	61	5	4	5	13	20
francesinha	7	24	0	12	2	14	98	2	11	11	9
gelado	8	11	11	15	29	7	3	86	11	13	19
hamburguer	6	9	8	10	6	9	12	11	58	7	12
pizza	3	14	8	9	9	6	10	7	14	50	23
waffles	9	4	6	13	6	9	3	4	5	8	16
Label Verdadeira	bacalhau_bras	batatas_fritas	bolo_chocolate	cachorro	donuts	esparquete_bolonhesa	francesinha	gelado	hamburguer	pizza	waffles

Figura 46 – Matriz de confusão CNN

A matriz de confusão obtida para o MobileNetV2 demonstra uma melhoria substancial das previsões efetuadas. Verifica-se que praticamente todas as previsões estão dispostas diagonalmente. Em 3284 imagens, apenas 44 imagens foram previstas incorretamente.

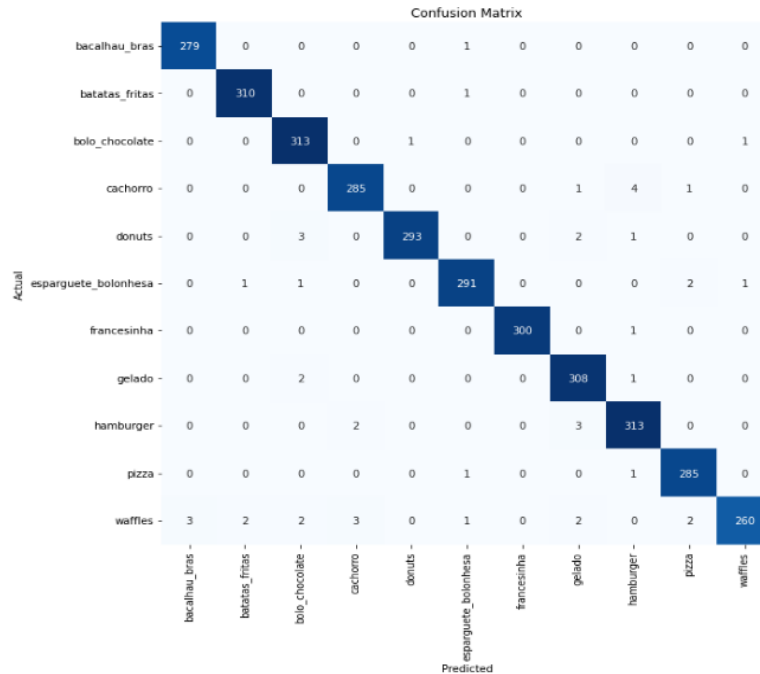


Figura 47 - Matriz de confusão MobileNetV2

Com os dados das matrizes de confusão, foi possível calcular vários valores que são bastante úteis para ajudar a avaliar o desempenho dos algoritmos. Deste modo, foi calculada a precisão, o *recall* e o *f1-score*.

- **Precisão:** relação entre as previsões positivas realizadas corretamente e todas as previsões positivas;

$$Precision = \frac{VP}{VP + FP} \quad (1)$$

, onde VP são os verdadeiros positivos e FP são os falsos positivos.

- **Recall:** relação entre os resultados previstos e os resultados previstos corretamente;

$$Recall = \frac{VP}{VP + FN} \quad (2)$$

, onde VP são os verdadeiros positivos e FN são os falsos negativos.

- **F1-Score:** ponderação entre métricas de precisão e de *recall*;

$$f_1 - score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \quad (3)$$

A métrica de precisão é um bom indicativo para mostrar a performance em relação à detecção de falsos positivos. Quanto maior for o resultado, melhor é a performance do algoritmo em relação a evitar falsos positivos. O *Recall* permite saber quantos resultados o algoritmo conseguiu efetivamente prever, por isso quanto maior for o resultado, melhor é a performance em evitar falsos negativos. Por fim, o *F1-Score* tenta encontrar o balanceamento entre a precisão e o *recall*, pelo que os falsos positivos e os falsos negativos têm um custo similar.

Tabela 12 – Resultados da matriz de confusão SVM

Label	Precision	Recall	F1-Score	Support
bacalhau_bras	0.71	0.56	0.63	154
batatas_fritas	0.33	0.40	0.36	164
bolo_chocolate	0.54	0.56	0.55	141
cachorro	0.22	0.23	0.23	145
donuts	0.23	0.21	0.22	140
esparguete_bolonhesa	0.38	0.44	0.41	149
francesinha	0.53	0.60	0.56	153
gelado	0.35	0.53	0.42	161
hamburguer	0.31	0.29	0.30	154
pizza	0.32	0.26	0.28	149
waffles	0.17	0.08	0.11	149
Accuracy	0.38	0.38	0.38	1659
Macro-average	0.37	0.38	0.37	1659
Weighted average	0.37	0.38	0.37	1659

Tabela 13 – Resultados da matriz de confusão CNN

Label	Precision	Recall	F1-Score	Support
bacalhau_bras	0.65	0.61	0.63	154
batatas_fritas	0.35	0.33	0.34	164
bolo_chocolate	0.53	0.58	0.55	141
cachorro	0.26	0.25	0.25	145
donuts	0.29	0.29	0.29	140
esparguete_bolonhesa	0.44	0.41	0.42	149
francesinha	0.52	0.64	0.57	153
gelado	0.40	0.53	0.46	161
hamburger	0.39	0.38	0.38	154
pizza	0.33	0.34	0.33	149

waffles	0.19	0.11	0.14	149
Accuracy	0.41	0.41	0.41	1659
Macro-average	0.39	0.41	0.40	1659
Weighted average	0.40	0.41	0.40	1659

Tabela 14 – Resultados da matriz de confusão MobileNetV2

Label	Precision	Recall	F1-Score	Support
bacalhau_bras	1.00	1.00	1.00	280
batatas_fritas	0.99	0.99	0.99	311
bolo_chocolate	0.97	0.99	0.98	315
cachorro	0.98	0.99	0.98	291
donuts	0.99	0.98	0.98	299
esparquete_bolonhesa	0.99	0.99	0.99	296
francesinha	1.00	1.00	1.00	301
gelado	0.97	0.99	0.98	311
hamburger	0.99	0.98	0.98	318
pizza	0.99	0.99	0.99	287
waffles	0.97	0.96	0.97	275
Accuracy	0.99	0.99	0.99	3284
Macro-average	0.99	0.99	0.99	3284
Weighted average	0.99	0.99	0.99	3284

Como o objetivo é minimizar ao máximo tanto os falsos negativos como os falsos positivos, consideram-se que a melhor métrica para avaliar o desempenho do algoritmo será o *F1-Score*.

O algoritmo CNN obteve melhor desempenho que o SVM, contudo nada se iguala à performance do modelo MobileNetV2. O SVM apenas obteve uma precisão de 38%, a CNN apresentou uma performance ligeiramente melhor que o SVM, com 41%, e, por fim, o MobileNetV2 destacou-se com uma performance de 99% nas métricas *precision*, *recall* e *f1-score* e uma precisão do modelo de 99%.

5.2 Reconstrução 3D

Nesta seção é apresentada a implementação da reconstrução 3D, realizada com o intuito de obter uma estimativa das refeições previamente detetadas através do modelo de reconhecimento de alimentos.

5.2.1 Aplicação da Reconstrução 3D

Nesta subseção é apresentado todo o processo de implementação da reconstrução 3D.

Conversão de Imagem para Mapa de Profundidade

Antes de mais, foi necessário o timm que é uma dependência do MiDaS. De seguida, utilizou-se o modelo DPT_Large do MiDaS v3, que apresenta uma maior precisão e uma velocidade de inferência mais lenta que os restantes modelos do MiDaS (DPT_Hybrid e MiDaS_small). O modelo foi então carregado e, caso o GPU estivesse disponível, o modelo seria então movido para o GPU, senão permanecia no CPU.

Após serem carregadas as transformações para redimensionar e normalizar a imagem, foi carregada uma imagem e foram aplicadas as transformações sobre a mesma. Depois, previu-se e redimensionou-se a imagem para a sua resolução original.

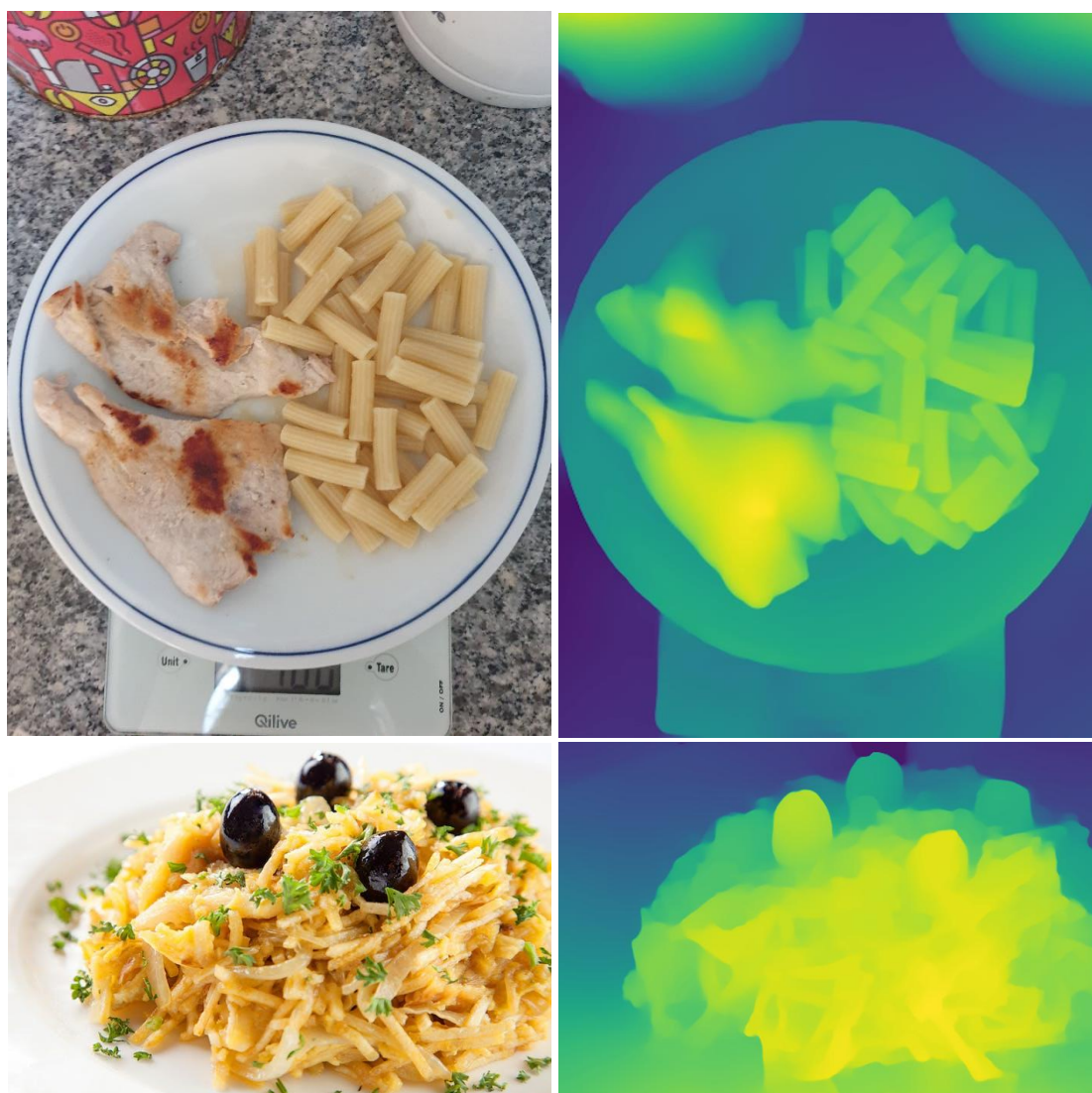


Figura 48 - Exemplos de imagens com o mapa de profundidade correspondente

Conversão do Mapa de Profundidade para Point Cloud

Para realizar a conversão de *depth map* para *point cloud*, utilizou-se a biblioteca Open3D. Começou-se por ler a imagem real e imagem de profundidade gerada no passo anterior, invocou-se a função `create_rgbd_image_from_color_and_depth` com as duas imagens como argumentos e assim criou-se a `RGBDImage`. Esta função converteu a imagem real numa imagem de escala de cinzas e armazenou-a em um *float* variando entre 0 e 1, e a imagem de profundidade foi armazenada em um *float*, representando o valor da profundidade em metros.

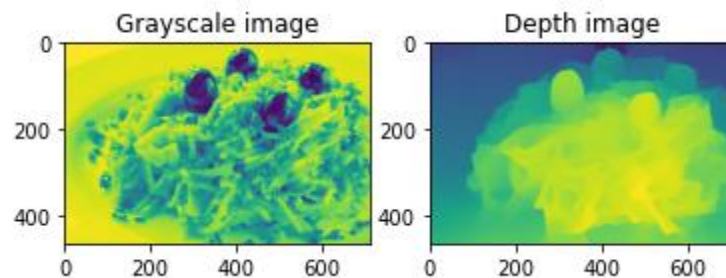


Figura 49 - Imagem com escala de cinzas e imagem de profundidade

De seguida, converteu-se a `RGBDImage` em *point cloud*. Para isso, foi necessário saber parâmetros intrínsecos da câmara, como o comprimento focal e o centro ótico da câmara, para ser possível projetar os pontos em uma *point cloud*. Utilizou-se a função `create_point_cloud_from_rgbd_image` para criar a *point cloud* a partir da `RGBDImage` e dos parâmetros da câmara.

Para os parâmetros da câmara foi necessário criar a estrutura de dados `PinholeCameraIntrinsic`. Como a imagem real foi recolhida do Google Images, não se sabia quais eram os parâmetros da câmara e, por isso, utilizaram-se parâmetros intrínsecos padrão da câmara, dados pelo comando `PinholeCameraIntrinsicParameters.PrimeSenseDefault` disponibilizado pelo Open3D. Os parâmetros padrão são resolução de imagem 640x480, distância focal $(f_x, f_y) = (525, 0, 525, 0)$ e centro ótico $(c_x, c_y) = (319, 5, 239, 5)$.

Realizou-se uma transformação *flip* para que a *point cloud* não estivesse virada do avesso e permitisse uma melhor visualização. Por fim, passou-se uma lista de geometrias como argumento para a função `draw_geometries` e visualizou-se a *point cloud* obtida.

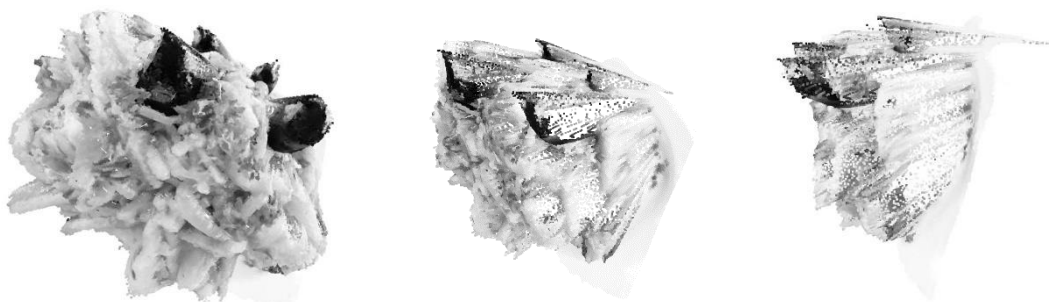


Figura 50 - Point Cloud

5.3 Detectron2

Nesta seção é apresentado o conjunto de dados utilizados na implementação desta abordagem e é explicado todo o processo realizado para estudar esta técnica. Por fim, são mostrados alguns resultados e é realizada uma discussão dos mesmos.

5.3.1 Conjunto de Dados

Para analisar esta abordagem, não se utilizou o *dataset* previamente construído, que serviu como base de conhecimento na aplicação dos modelos SVM, CNN e MobileNetV2. Pretendia-se realizar testes neste âmbito com um conjunto de dados mais pequeno, para que permitisse entender a performance do modelo e a viabilidade de optar por esta abordagem, tendo em conta que o modelo MobileNetV2 já tinha obtido uma precisão extremamente satisfatória e demonstrava ser um bom modelo a utilizar na implementação desta solução.

Assim sendo, foi necessário recolher algumas imagens para construir o conjunto de dados. Algumas imagens da categoria bolonhesa do *dataset* [76] foram reutilizadas para fazer alguns testes com esta *framework*, e as restantes imagens foram recolhidas manualmente através da câmara de dispositivos móveis e do *download* a partir da internet. Foram apenas utilizadas 8 categorias: *1euro*, *beans*, *bolonhese*, *chicken*, *egg*, *fries*, *rice* e *salad*. De modo que o modelo construído fosse robusto, utilizaram-se imagens com backgrounds diferentes, condições de luminosidade variáveis e alguns objetos aleatórios de fundo.

A categoria *1euro* foi criada a pensar no próximo passo da solução: após o reconhecimento de alimentos, pretendia-se que o sistema fosse capaz de obter o peso dos alimentos e consequentemente obter os seus valores nutricionais o mais próximo possível da realidade. Para isso, precisou-se de um objeto de referência que permita fazer a escala com os alimentos presentes na imagem de entrada.

O *dataset* possui atualmente 414 imagens.

5.3.2 Aplicação do Detectron2

Nesta subseção é apresentado todo o processo de implementação do modelo Detectron2.

Anotação dos Dados

De seguida, realizou-se o *labeling* de imagens para que o modelo soubesse o que precisava de aprender. Para isso, utilizou-se um software de *labeling* chamado Labelme. Esta ferramenta é de fácil instalação e utilização.

Após este processo, os dados foram divididos em duas pastas – uma pasta de treino e uma de teste – para que mais tarde fosse possível obter uma estimativa razoável do quão bom seria o

modelo construído. Cada imagem ficou acompanhada por um ficheiro json com as suas anotações.

Conversão do Dataset para o Formato COCO

O COCO possui cinco tipos de anotação: *object detection*, *keypoint detection*, *stuff segmentation*, *panoptic segmentation*, e *image captioning*. O formato COCO assemelha-se ao seguinte:

```
annotation {
  "id": int,
  "image_id": int,
  "category_id": int,
  "segmentation": RLE or [polygon],
  "area": float,
  "bbox": [x,y,width,height],
  "iscrowd": 0 or 1,
}
categories [{
  "id": int,
  "name": str,
  "supercategory": str,
}]
```

Figura 51 - Formato COCO

Foi necessário converter as anotações para um formato que o Detectron2 percebesse. Os ficheiros json criados a partir do Labelme foram convertidos para o formato COCO através do ficheiro `labelme2coco.py` desenvolvido por Tony607 [79]. Ao utilizar o `labelme2coco`, foi criado um único ficheiro `train.json` e um ficheiro `test.json`.

Registo do Dataset

Instalaram-se algumas dependências como o `pyyaml` e o `detectron2`. Como o conjunto de dados estava no formato COCO, todo o processo tornou-se mais simples. Apenas foi necessário invocar o método `register_coco_instances` e passar como parâmetros o nome do conjunto de dados, o caminho para o arquivo json e o caminho para o diretório de imagens. Este passo foi realizado para os dados de treino e de teste.

Configuração do Modelo

Para verificar se o *dataset* estava com o formato correto, visualizamos algumas anotações de imagens aleatórias do conjunto de dados de treino. Para isso, importaram-se algumas classes e funções do `detectron2`, como o `model zoo`, `DefaultPredictor`, `get_cfg`, `Visualizer`, `MetadaCatalog` e `DatasetCatalog`. A classe `DefaultPredictor` é utilizada para construir o detetor, passando um objeto de configuração que o configura. O `Visualizer` oferece suporte para visualização de resultados. `MetadaCatalog` e `DatasetCatalog` pertencem à API de dados do

Detectron2 e oferecem informações sobre o conjunto de dados integrados, bem como os seus metadados.

Treino do Modelo

Iniciou-se com a função `get_cfg()` que inicializa uma configuração vazia, que será preenchida mais tarde com as definições desejadas. O método `merge_from_file()` permite então definir a configuração do modelo, passando como argumento uma string com o caminho para o ficheiro local ou um arquivo de configuração `model_zoo.get_config_file()`. Aqui, especificou-se o modelo que se pretendia executar a partir do `model_zoo`: o `PointRend`. Depois, carregou-se de forma semelhante os pesos do modelo, fornecendo uma string para o modelo zoo.

Definiu-se o número de *workers* e é importante referir que quanto maior o número de *workers*, mais rápido os dados serão carregados, contudo sobrecarregará ainda mais o cpu. Por essa razão, apenas se definiu dois *workers*. Definiram-se duas imagens por batch, uma taxa de aprendizagem de 0.00025, o máximo de iterações foi de 1000, o número de classes do modelo (neste caso, são 8 classes), o dispositivo onde se pretende treinar o modelo (cpu), e o local onde o modelo será gravado.

```
[09/25 20:09:09 d2.data.datasets.coco]: Loaded 241 images in COCO format from train.json
[09/25 20:09:09 d2.data.build]: Removed 0 images with no usable annotations. 241 images left.
[09/25 20:09:09 d2.data.build]: Distribution of instances among all 8 categories:
| category | #instances | category | #instances | category | #instances |
|-----|-----|-----|-----|-----|-----|
| leuro    | 144       | beans    | 22        | bolonhese | 87       |
| chicken  | 144       | egg      | 8         | fries     | 20       |
| rice     | 83        | salad    | 28        |           |         |
| total   | 536       |          |           |          |         |
[09/25 20:09:09 d2.data.dataset_mapper]: [DatasetMapper] Augmentations used in training: [ResizeShortestEdge(short_edge_length=(640, 672, 704, 736, 768, 800), max_size=1333, sample_style='choice'), RandomFlip()]
[09/25 20:09:09 d2.data.build]: Using training sampler TrainingSampler
[09/25 20:09:09 d2.data.common]: Serializing 241 elements to byte tensors and concatenating them all ...
[09/25 20:09:09 d2.data.common]: Serialized dataset takes 0.28 MiB
Weight format of PointRend models have changed! Applying automatic conversion now ...
The checkpoint state_dict contains keys that are not used by the model:
  pixel_mean
  pixel_std
[09/25 20:09:10 d2.engine.train_loop]: Starting training from iteration 0
C:\Users\CATARINA\anaconda3\envs\detectron2\site-packages\torch\functional.py:678: UserWarning: torch.meshgrid: in an upcoming release, it will be required to pass the indexing argument. (Triggered internally at C:\pyb\
return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
[09/25 20:22:49 d2.utils.events]: eta: 11:35:29 iter: 19 total_loss: 2.319 loss_cls: 0.8795 loss_box_reg: 0.89218 loss_mask: 0.4863 loss_mask_point: 0.6999 loss_rpn_cls: 0.01807 loss_rpn_loc: 0.01494 time: 40.7960 data
[09/25 20:34:07 d2.utils.events]: eta: 9:17:47 iter: 39 total_loss: 2.162 loss_cls: 0.7413 loss_box_reg: 0.89302 loss_mask: 0.4569 loss_mask_point: 0.488 loss_rpn_cls: 0.006458 loss_rpn_loc: 0.01082 time: 37.1440 data
[09/25 20:44:43 d2.utils.events]: eta: 8:41:47 iter: 59 total_loss: 1.993 loss_cls: 0.6997 loss_box_reg: 0.1169 loss_mask: 0.3594 loss_mask_point: 0.5284 loss_rpn_cls: 0.01451 loss_rpn_loc: 0.01454 time: 35.2840 data_1
[09/25 20:55:03 d2.utils.events]: eta: 8:28:59 iter: 79 total_loss: 1.841 loss_cls: 0.8066 loss_box_reg: 0.104 loss_mask: 0.2382 loss_mask_point: 0.462 loss_rpn_cls: 0.01022 loss_rpn_loc: 0.01035 time: 34.2037 data_ti
[09/25 21:06:05 d2.utils.events]: eta: 8:17:24 iter: 99 total_loss: 1.694 loss_cls: 0.818 loss_box_reg: 0.1072 loss_mask: 0.2168 loss_mask_point: 0.3838 loss_rpn_cls: 0.007432 loss_rpn_loc: 0.01099 time: 33.9748 data_1
[09/25 21:17:37 d2.utils.events]: eta: 8:07:24 iter: 119 total_loss: 1.498 loss_cls: 0.702 loss_box_reg: 0.1321 loss_mask: 0.2092 loss_mask_point: 0.4042 loss_rpn_cls: 0.009432 loss_rpn_loc: 0.01237 time: 34.0786 data
[09/25 21:29:29 d2.utils.events]: eta: 7:56:46 iter: 139 total_loss: 1.429 loss_cls: 0.5655 loss_box_reg: 0.1321 loss_mask: 0.2092 loss_mask_point: 0.4185 loss_rpn_cls: 0.005086 loss_rpn_loc: 0.01019 time: 34.5009 data
[09/25 21:41:46 d2.utils.events]: eta: 7:46:24 iter: 159 total_loss: 1.129 loss_cls: 0.4502 loss_box_reg: 0.08603 loss_mask: 0.1887 loss_mask_point: 0.3682 loss_rpn_cls: 0.004268 loss_rpn_loc: 0.006905 time: 34.6252 di
[09/25 21:51:18 d2.utils.events]: eta: 7:35:18 iter: 179 total_loss: 1.149 loss_cls: 0.448 loss_box_reg: 0.1201 loss_mask: 0.1831 loss_mask_point: 0.3663 loss_rpn_cls: 0.006532 loss_rpn_loc: 0.009578 time: 34.6192 data
[09/25 22:04:24 d2.utils.events]: eta: 7:23:38 iter: 199 total_loss: 0.7465 loss_cls: 0.1963 loss_box_reg: 0.05509 loss_mask: 0.164 loss_mask_point: 0.314 loss_rpn_cls: 0.001953 loss_rpn_loc: 0.004932 time: 34.4857 dat
[09/25 22:15:57 d2.utils.events]: eta: 7:13:05 iter: 219 total_loss: 0.9065 loss_cls: 0.2299 loss_box_reg: 0.08534 loss_mask: 0.1717 loss_mask_point: 0.3338 loss_rpn_cls: 0.003901 loss_rpn_loc: 0.008019 time: 34.5011 r
[09/25 22:27:16 d2.utils.events]: eta: 7:01:44 iter: 239 total_loss: 0.8174 loss_cls: 0.1933 loss_box_reg: 0.07354 loss_mask: 0.1582 loss_mask_point: 0.3573 loss_rpn_cls: 0.00463 time: 34.4540 di
[09/25 22:38:40 d2.utils.events]: eta: 6:51:38 iter: 259 total_loss: 0.7218 loss_cls: 0.2313 loss_box_reg: 0.06446 loss_mask: 0.1382 loss_mask_point: 0.3165 loss_rpn_cls: 0.003023 loss_rpn_loc: 0.007609 time: 34.4664 i
[09/25 22:49:54 d2.utils.events]: eta: 6:41:21 iter: 279 total_loss: 0.641 loss_cls: 0.1426 loss_box_reg: 0.05529 loss_mask: 0.1194 loss_mask_point: 0.3077 loss_rpn_cls: 0.003967 loss_rpn_loc: 0.005115 time: 34.3810 di
[09/25 23:01:01 d2.utils.events]: eta: 6:28:20 iter: 299 total_loss: 0.7787 loss_cls: 0.1586 loss_box_reg: 0.09561 loss_mask: 0.1563 loss_mask_point: 0.3492 loss_rpn_cls: 0.005268 loss_rpn_loc: 0.005268 time: 34.3118 r
[09/25 23:11:10 d2.utils.events]: eta: 6:19:03 iter: 319 total_loss: 0.7303 loss_cls: 0.14 loss_box_reg: 0.0567 loss_mask: 0.1452 loss_mask_point: 0.3555 loss_rpn_cls: 0.00318 loss_rpn_loc: 0.004288 time: 34.4635 data
```

Figura 52 - Treino do modelo

Inferência e Avaliação

Finalmente, construiu-se o preditor com este `cfg` e executou-se o preditor nas entradas. A classe `Visualizer` foi utilizada para desenhar previsões na imagem, neste caso instâncias segmentadas, classes e caixas delimitadoras.



Figura 53 – Alguns resultados da aplicação do modelo PointRend

Como é possível verificar na Figura 53, o modelo tem uma boa performance, apesar de possuir poucos dados.

5.3.3 Resultados

Depois do modelo ser treinado, foi avaliado o seu desempenho ou precisão em relação aos dados de teste. Existem várias métricas que podem ser usadas para avaliar os modelos, contudo a sua utilização vai depender da declaração e da aplicação do problema. As métricas comumente utilizadas em problemas de classificação de imagem são *Precision*, *Recall*, Matriz de Confusão, *PR-curve*, entre outros. Na segmentação de imagem é usado frequentemente o *Mean Intersection over Union*, mais conhecido como mIoU, e na detecção de objetos é utilizada a *Mean Average Precision* (mAP).

Como na detecção de objetos o objetivo não é apenas classificar corretamente o(s) objeto(s) presentes na imagem, mas também descobrir onde esse(s) objeto(s) se localizam na imagem, não se pode simplesmente usar as métricas de classificação de imagem. Por essa razão, nesta subseção será analisado o mAP utilizando o COCOEvaluator.

O modelo foi avaliado com o COCOEvaluator que avalia a precisão média da detecção das caixas delimitadoras, da segmentação de instância e da detecção de pontos-chave. O COCOEvaluator foi inicializado com o conjunto de dados de validação e, de seguida, foi invocado o `build_detection_test_loader` com a configuração do modelo e conjunto de dados fornecido como parâmetros. O `build_detection_test_loader` retorna um *DataLoader*. Por fim, o modelo avaliado e o *DataLoader* foram utilizados como argumentos da função `inference_on_dataset`, que executa o modelo no conjunto de dados e avalia a métrica com o *Evaluator*.

O COCOEvaluator disponibiliza 12 métricas para caracterizar o desempenho de um detetor de objetos:

```
Average Precision (AP):
AP                % AP at IoU=.50:.05:.95 (primary challenge metric)
APIoU=.50        % AP at IoU=.50 (PASCAL VOC metric)
APIoU=.75        % AP at IoU=.75 (strict metric)
AP Across Scales:
APsmall          % AP for small objects: area < 322
APmedium        % AP for medium objects: 322 < area < 962
APlarge         % AP for large objects: area > 962
Average Recall (AR):
ARmax=1         % AR given 1 detection per image
ARmax=10        % AR given 10 detections per image
ARmax=100       % AR given 100 detections per image
AR Across Scales:
ARsmall         % AR for small objects: area < 322
ARmedium       % AR for medium objects: 322 < area < 962
ARlarge         % AR for large objects: area > 962
```

Figura 54 – Métricas disponibilizadas pelo COCOEvaluator [80]

No contexto COCO, não há diferença entre AP e mAP. COCO cita que AP é a média de todas as categorias, e que, tradicionalmente, isso é chamado de precisão *Mean Average Precision* (mAP). COCO não faz distinção entre AP e mAP e assumem que a diferença é clara a partir do contexto [80].

```

Loading and preparing results...
DONE (t=0.01s)
creating index...
index created!
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.638
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.798
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.710
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.481
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.648
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.642
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.679
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.679
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.506
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.685

```

Figura 55 - Resultados da avaliação do modelo Detectron2

Como se pode verificar na Figura 55, as métricas chave são:

- *Intersection over Union* (IoU): dita como os objetos reais e as deteções previstas se sobrepõem. Um IoU de 0 significa que não há sobreposição de um IoU de 1 significa que ocorreu uma deteção perfeita;
- *Average Precision* (AP): determina se uma deteção de um objeto é considerada correta com base em um limite de IoU especificado. Então, na Figura 55, o AP50 e AP75, por exemplo, denotam uma precisão média com limites IoU de 0.5 e 0.75, respetivamente. O objetivo seria atingir um AP de 100%, mas é algo muito difícil de alcançar.

Tabela 15 - Resultados da avaliação da segmentação

AP	AP50	AP75	APs	APm	API
63.785	79.782	70.970	0.0	48.093	64.822

Tabela 16 - Resultados da segmentação por categoria

Categoria	AP	Categoria	AP	Categoria	AP
1euro	74.715	beans	76.050	bolonhese	93.68
chicken	59.261	egg	0.0	fries	76.799
rice	78.774	salad	50.990		

O modelo obteve uma *Mean Average Precision* de 63.8%, o que não é mau se se tiver em conta que a precisão média da classe egg (em português, ovo) foi de 0%. Este facto pode ter acontecido porque o conjunto de dados fornecido continha apenas duas imagens com ovo.

5.4 Regressão Linear

Nesta seção é apresentada a aplicação do modelo de regressão linear, bem como os seus resultados.

5.4.1 Conjunto de Dados

O conjunto de dados utilizado para fazer o teste deste modelo consiste em um ficheiro excel composto por cinco colunas: nome da imagem, categoria, área do alimento, peso do alimento e área da moeda. Por outras palavras, cada linha deste ficheiro possui a informação da imagem de entrada no modelo de deteção de objetos (Detectron2), uma das categorias detetadas pelo modelo, a área da máscara do alimento detetado em centímetros quadrados, o peso real do alimento em questão, e, por fim, a área da máscara da moeda.

Para o treino do modelo de regressão linear, apenas foram registados os valores das máscaras das imagens captadas às refeições acompanhadas por uma moeda de um euro. Como estas imagens foram capturadas manualmente, foi registado também o peso dos alimentos.

Atualmente, o conjunto de dados apenas possui 387 registos.

5.4.2 Aplicação da Regressão Linear

Para fazer a análise deste modelo, foram utilizadas as máscaras obtidas na deteção de alimentos com Detectron2. Nesta subseção serão descritos todos os passos realizados para obter o peso de cada alimento presente na imagem de entrada fornecida pelo utilizador.

Cálculo da Área da Máscara

Após obter as máscaras, extraiu-se o contorno de cada máscara prevista e guardou-se numa lista. Os contornos foram extraídos através da função `findContours` disponível no OpenCV. Esta função deteta alterações na cor da imagem e marca essas alterações como contornos. Os parâmetros usados nesta função foram: `RETR_TREE` que recupera todos os contornos e reconstrói uma hierarquia completa de contornos misturados, e `CHAIN_APPROX_SIMPLE` como método de aproximação do contorno, que compacta segmentos horizontais, verticais e diagonais, e deixa apenas os seus pontos finais.

Iterou-se a lista de contorno e a cada contorno aplicou-se a função `countourArea` do OpenCV, de modo a obter a área do contorno. Sabendo que 78 pixéis equivalem a 1 cm, foi possível converter a área da máscara de cada alimento em centímetros quadrados.

Previsão do Peso dos Alimentos

Várias imagens pertencentes ao *dataset* foram recolhidas manualmente. Aquando dessa recolha, os alimentos foram pesados e fotografados juntamente com uma moeda de um euro,

que serviu como objeto de referência. Esta moeda torna todo este processo mais viável, pois trata-se de um objeto que possui medidas fixas (pesa 7.50 gramas) e mesmo que a imagem seja capturada de ângulos e distâncias diferentes, a moeda será sempre a referência que permitirá fazer a escala com os alimentos presentes na imagem de entrada.

A informação do peso dos alimentos presentes nas imagens foi registada num ficheiro excel com quatro colunas: categoria, área, peso e moeda. Esta informação foi importada através da biblioteca pandas e definiu-se que *x* eram as colunas categoria, área e moeda, e *y* era a coluna do peso. De seguida, utilizou-se a função `get_dummies` que converte informação categórica em variáveis numéricas. As variáveis *dummy* podem assumir apenas dois valores: 0 ou 1, sendo que 1 indica a presença de uma determinada categoria e 0 indica a ausência da categoria.

	area	moeda	Categoria_bolonhese	Categoria_chicken	Categoria_rice
0	134.54	6.97	0	1	0
1	103.18	5.97	0	1	0
2	141.77	8.10	0	1	0
3	217.01	8.10	0	0	1
4	116.99	6.35	0	1	0
..
101	158.07	2.04	1	0	0
102	144.80	1.96	1	0	0
103	126.20	1.91	1	0	0
104	211.47	4.99	1	0	0
105	110.16	1.52	1	0	0

Figura 56 - Exemplo da aplicação do método `get_dummies`

Muitos algoritmos de ML, como a regressão linear e a regressão logística, exigem estritamente dados de entrada numéricos. Nestes casos, se forem usados dados categóricos baseados em *string*, será despoletado um erro. Por esta razão, foi preciso codificar os dados categóricos como variáveis fictícias numéricas.

Após a conversão para variáveis *dummy*, separou-se o conjunto de dados para treino e teste através do método `train_test_split` do `sklearn`. Criou-se o modelo `LinearRegression` do `sklearn`, treinou-se o modelo com os dados de treino e, por fim, foi possível fazer previsões do peso dos alimentos numa determinada imagem.

5.4.3 Resultados

Para avaliar os resultados do modelo, utilizaram-se quatro métricas: coeficiente de determinação, coeficiente de determinação ajustado, erro quadrático médio (MSE), raiz do erro quadrático médio (RMSE) e erro médio absoluto (MAE).

O coeficiente de determinação, também chamado de R^2 , é uma métrica que visa expressar a quantidade da variância dos dados que é explicada pelo modelo linear e, portanto, diz o quão

próximas estão as medidas reais do modelo. O valor do coeficiente de determinação varia de 0 a 1 e geralmente é apresentado em percentagem. Quanto maior for o R^2 , mais explicativo é o modelo linear, ou seja, melhor se ajusta à amostra.

O coeficiente de determinação é dado pela fórmula:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

, na qual n é o número de observações, y_i é o valor real, \hat{y}_i é o valor estimado de y_i e \bar{y} é a média das observações.

O modelo de regressão linear implementado obteve um $R^2 = 0.67$, o que significa que o modelo linear explica 66,6% da variância da variável dependente a partir das variáveis independentes incluídas no modelo construído. O valor do coeficiente podia ser melhorado com o aumento do conjunto de dados usados para o treino do modelo.

Este coeficiente pode ser utilizado como uma medida de confiança depositada na equação de regressão como instrumento de previsão. Sendo o coeficiente próximo de 1, pode-se considerar que o modelo linear é bastante adequado.

Quanto ao coeficiente de determinação ajustado, este ajuda a medir a proporção de variação descrita apenas por variáveis independentes com uma influência real sobre a variável dependente. Se forem adicionadas variáveis independentes que não afetam a variável dependente, o modelo é penalizado.

O R^2 ajustado é calculado pela seguinte equação:

$$R^2_{ajustado} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

, sendo n o tamanho total da amostra e p o número de preditores.

O R^2 ajustado é mais eficiente que o R^2 , pois após a adição de uma variável independente ao modelo do R^2 , o valor aumentará mesmo nos casos em que a variável independente é insignificante. O valor do R^2 ajustado aumentará somente se a variável independente for significativa e caso afete a variável dependente. O R^2 ajustado obteve 62% para esta regressão, o que significa que foram adicionadas variáveis com pouco poder explicativo.

O MSE consiste na média do erro das previsões ao quadrado, pelo que quanto maior for este número, pior é o modelo. Esta métrica apresenta um valor mínimo de 0, sem valor máximo, e pode ser descrito pela fórmula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

, onde \hat{y}_i é o valor real da variável dependente, y_i é o valor previsto e n é o tamanho total da amostra. Esta métrica obteve 1726.13.

Relativamente ao RMSE, que calcula o quão próximo os pontos de dados reais estão dos valores previstos pelo modelo, o modelo obteve 41.55g (a unidade do RMSE é a mesma da variável dependente).

O RMSE é usado para medir o desvio padrão dos resíduos e é calculado utilizando a seguinte equação:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$$

, sendo \hat{y}_i é o valor real da variável dependente, y_i é o valor previsto e n é o tamanho total da amostra.

Quanto menor o valor de RMSE, melhor, pois é indicação de um bom ajuste. O RMSE ajuda a saber o quão bem o modelo é capaz de prever a resposta.

Por fim, utilizou-se a métrica MAE. Esta métrica consiste na média das distâncias entre valores previstos e reais. Contrariamente ao MSE e RMSE, esta métrica não penaliza tão severamente os *outliers* do modelo. Apresenta valor mínimo de 0 e não possui valor máximo.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

O modelo apresentou um MAE de 24.02.

5.5 Discussão das Abordagens

Com o intuito de solucionar o problema apresentado nesta dissertação, três algoritmos bastante utilizados no estado de arte foram analisados: Support Vector Machine, Convolutional Neural Network e MobileNetV2. Toda a implementação e os resultados obtidos são descritos detalhadamente na seção 5.1.

Através da análise de resultados, percebe-se que os algoritmos SVM e CNN não tiveram uma ótima precisão. Isto pode ter acontecido por várias razões desde:

- Quantidade de dados: devido aos problemas de escassez de memória, foi utilizado um conjunto de dados exponencialmente mais pequeno comparativamente ao *dataset* inicial;
- Dimensionalidade de imagens: como as imagens do conjunto de dados eram muito grandes, surgiram, mais uma vez, os problemas de memória, pelo que as imagens foram redimensionadas para metade da média do tamanho de todas as imagens do *dataset*.

Com esta alteração de tamanho, os modelos podem não ter conseguido captar recursos distintos que ajudariam a fazer o reconhecimento dos alimentos;

- **Diversidade de alimentos:** trabalhar com um conjunto de dados onde cada classe possui uma vasta diversidade, pode ter levado a que os modelos não tenham sido tão precisos como o esperado. Isto é, a classe de bolo de chocolate, por exemplo, agrega variados tipos e formas de bolos de chocolate, desde bolos apenas de chocolate, com recheio, com enfeites, com cobertura, podem ser apenas uma fatia, como várias fatias, bolos inteiros circulares ou com a forma de um quadrilátero, entre outros. Com uma variedade de bolo de chocolate tão grande e dada a quantidade de dados que se utilizou, os modelos podem ter sentido dificuldades em obter atributos suficientes que permitissem uma boa previsão.

De modo a perceber o que possa ter causado estes valores de precisão, foi gerada a Figura 57 a partir das previsões do algoritmo CNN. É possível detetar, a título de exemplo, que a segunda imagem da primeira coluna que, para o ser humano, é facilmente identificada como sendo pizza, foi prevista como esparguete à bolonhesa.

Apesar de não ser possível identificar concretamente quais as razões que levaram a esta falha na previsão, pode dever-se ao facto de as cores normalmente associadas à refeição esparguete à bolonhesa (amarelo e vermelho) serem as mesmas da imagem referida. A situação inversa acontece com a terceira imagem da penúltima coluna, onde a imagem é esparguete à bolonhesa, mas foi prevista como sendo pizza. Contudo, esta é apenas uma possibilidade, mas seria necessário realizar mais testes e utilizar mais métodos que permitissem fazer um reconhecimento de alimentos com mais exatidão.

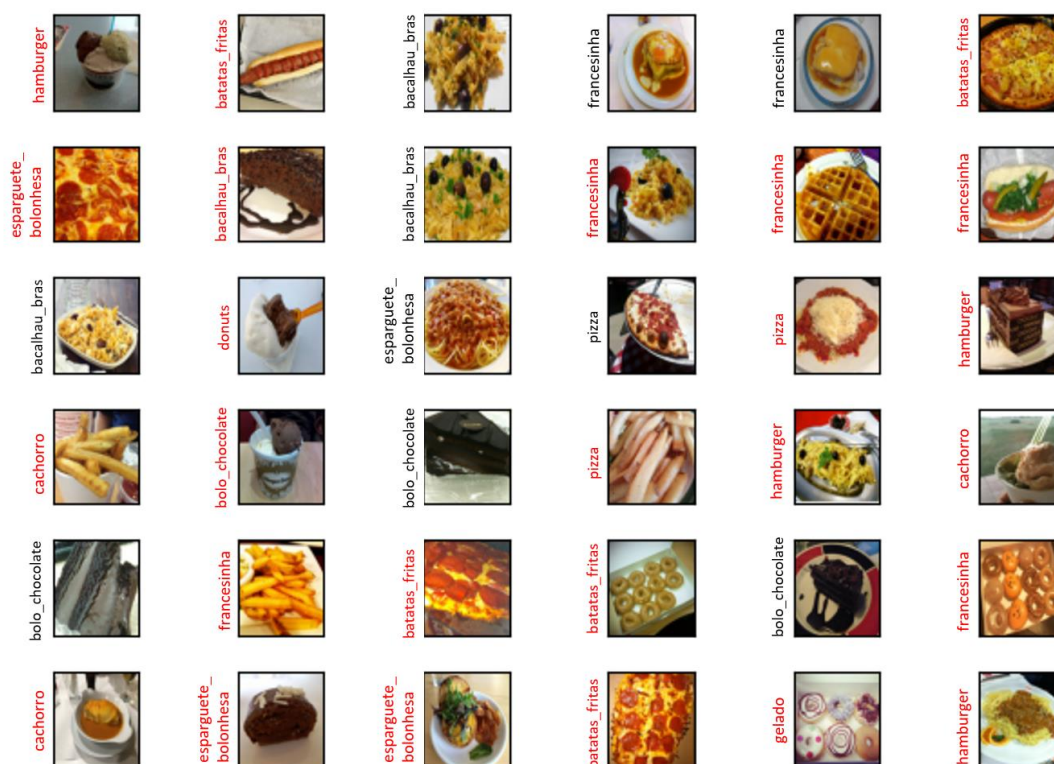


Figura 57 – Exemplos de imagens com respetiva previsão CNN

Apesar das CNN possuírem uma precisão ligeiramente melhor que as SVM, ambos apresentaram uma precisão insatisfatória. A precisão da CNN foi de 41% e a do SVM foi de 38%. Por essa razão, analisou-se o MobileNetV2, que se destacou com uma performance de 99%. Além de ser um modelo bastante simples de utilizar, o seu treino é rápido e não exerce computação excessiva.

Seria de esperar que o modelo escolhido para realizar o reconhecimento de alimentos desta dissertação fosse o MobileNetV2. Contudo, pretendia-se que este sistema fizesse, para além do reconhecimento de alimentos a partir de uma imagem, uma estimativa do peso dos alimentos e posterior cálculo nutricional dos mesmos, para ser apresentado ao utilizador.

Após uma análise de vários artigos, acreditou-se que a melhor abordagem a seguir para realizar a estimativa do peso dos alimentos seria através da reconstrução 3D. Ao obter uma reconstrução 3D das refeições, saber-se-iam facilmente algumas coordenadas que permitiriam realizar cálculos para a obtenção do volume dos alimentos.

Porém, quando se utilizou a ferramenta MeshLab para visualizar a point cloud obtida verificou-se que a reconstrução 3D alcançada não permitia saber os limites da refeição presente na imagem. O ângulo da imagem, a área fotografada, a luminosidade, entre outros fatores, influenciaram negativamente o desempenho da reconstrução 3D.

Como é possível verificar na Figura 59, muita informação da Figura 58 foi perdida. Não é possível determinar a perspetiva da imagem, pelo que não se consegue determinar a altura, largura e

profundidade da refeição. Sem a noção destes limites, tornou-se impraticável a obtenção do volume dos alimentos.



Figura 58 - Imagem real para posterior reconstrução 3D

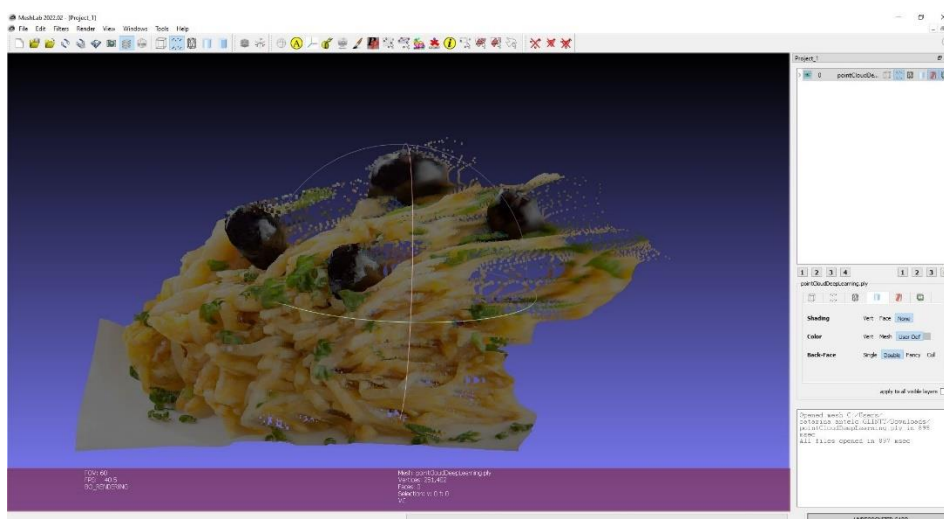


Figura 59 - Point Cloud obtida após a Reconstrução 3D

Caso o utilizador disponibilizasse mais que uma imagem de entrada, captada de vários ângulos, acredita-se que esta teria sido uma ótima abordagem a seguir. Contudo, neste sistema pretende-se que haja praticidade e que permita que o utilizador obtenha resultados rápidos, sem grande esforço. Assim sendo, a reconstrução 3D tornou-se em uma opção inexecutável, pelo que foi necessário ponderar outras opções.

Uma das abordagens que podia resolver o problema de estimativa de volume seria utilizando o MediaPipe Objectron, apresentado anteriormente. Mas após analisar melhor a utilização desta solução da Google, chegou-se à conclusão de que esta também era uma alternativa inadequada.

Além desta solução apenas estar preparada para realizar a detecção de objetos 3D nas categorias *shoes, chairs, cups* e *cameras*, mesmo que o sistema detetasse a refeição, o volume obtido seria sempre em excesso, pois tratar-se-ia do volume da caixa delimitadora 3D que seria construída sob a refeição. E este facto pode ser perigoso quando se trata de um utilizador diabético que pretende saber quantas unidades de insulina deve fazer, podendo atingir a hipoglicemia pelo excesso de insulina.

Posteriormente, optou-se por uma abordagem preditiva, através de um modelo de regressão linear. Sabendo o peso real dos alimentos e o espaço que os mesmos ocupam na imagem, e aplicando estes dados a um modelo regressivo, seria possível prever uma estimativa do peso de futuras imagens de refeições.

A utilização do modelo MobileNetV2 deixou de ser benéfico para a solução, pois apesar da sua ótima performance, apenas permitia detetar a refeição presente na figura. Após uma análise exaustiva, chegou-se à conclusão de que o melhor seria realizar a segmentação da imagem, para que a partir da máscara dos alimentos fosse possível obter os contornos dessas máscaras e calcular a área dos alimentos. Foi deste modo que surgiu a implementação do Detectron2 com o modelo PointRend (ver subsecção 5.3.2), que mostrou ser bastante preciso, mesmo dispondo de poucos dados.

Porém, o espaço que os alimentos ocupam na imagem é influenciado pelo modo como a imagem foi capturada. O ângulo e a proximidade são fatores que afetam severamente a perceção do utilizador.

A Figura 60 mostra a mesma refeição fotografada de posições diferentes. Se se tivesse apenas em consideração a área que a refeição ocupa na imagem, era expectável que o peso da imagem da direita pesasse mais que a imagem da esquerda. Todavia, trata-se de imagens da mesma refeição, com o mesmo peso, apenas capturadas em instantes e ângulos diferentes.



Figura 60 - Percepção de imagem

Por essa razão, foi necessário incluir um objeto de referência com tamanho e peso conhecido, que permitisse criar uma escala para melhor estimativa do peso dos alimentos. Assim, tornou-se crucial que as imagens disponibilizadas pelo utilizador tivessem a refeição acompanhada por uma moeda de um euro, pois esta moeda é que irá permitir fornecer dados nutricionais mais próximos da realidade.

A utilização do Detectron2 com o modelo PointRend para o reconhecimento e segmentação dos alimentos permitiu não só identificar mais que um alimento na imagem como também possibilitou a obtenção das áreas das máscaras dos alimentos e da moeda da imagem. Com esses dados, o modelo de regressão linear conseguiu prever eficientemente o peso dos alimentos, apresentando um coeficiente de determinação de 67%. Este coeficiente pode ser facilmente melhorado com o aumento do conjunto de dados utilizados para o treino do modelo.

6 Modelação da Solução

Neste capítulo será apresentada a solução proposta no presente trabalho. O capítulo começa por uma análise dos requisitos funcionais identificados para o trabalho proposto, é apresentado o modelo de domínio da solução e é realizada a especificação da realização dos requisitos funcionais identificados.

6.1 Análise de Requisitos

Nesta seção são identificados os atores principais da solução, seguindo-se de uma análise de requisitos funcionais.

6.1.1 Atores Principais

Foram identificados dois atores principais:

- Utilizador – entidade que utiliza a aplicação, fornece uma imagem de uma refeição acompanhada de uma moeda de um euro ao sistema;
- Sistema – entidade encarregue de realizar o reconhecimento e previsão do peso dos alimentos presentes na imagem e disponibilização dos valores nutricionais da refeição.

6.1.2 Requisitos Funcionais

Na Figura 61 apresentam-se os casos de uso da solução proposta.

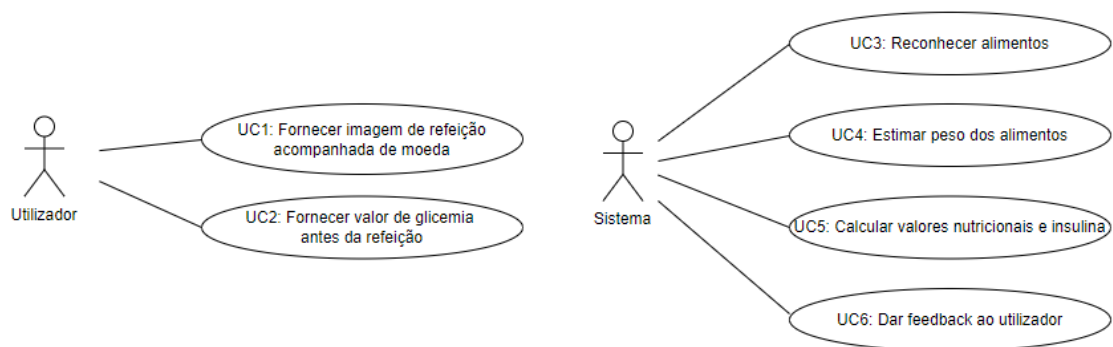


Figura 61 - Diagrama de casos de uso

UC1: Fornecer imagem de refeição acompanhada de moeda

O presente caso de uso consiste no fornecimento de uma imagem ao sistema. O utilizador capta uma fotografia da sua refeição acompanhada por uma moeda de um euro. Esta imagem é o *input* do sistema.

UC2: Fornecer valor de glicemia antes da refeição

Este caso de uso é uma função complementar que apenas é solicitada aos utilizadores diabéticos. Com este valor, será possível realizar os cálculos de insulina a injetar. Caso não sejam facultados pelo utilizador, não serão apresentados os valores de insulina.

UC3: Reconhecer alimentos

O caso de uso 3 consiste na receção da imagem fornecida pelo utilizador e no reconhecimento de alimentos presentes na imagem.

UC4: Estimar peso dos alimentos

O presente caso de uso consiste na estimativa de peso dos alimentos presentes na imagem. Após o sistema detetar os alimentos através do UC3, é adquirida a área de cada alimento na imagem e, com essa área, é realizada uma estimativa do peso em gramas.

UC5: Calcular valores nutricionais e insulina

Este caso de uso compreende os cálculos dos valores nutricionais dos alimentos previamente identificados e, caso o paciente seja diabético, são efetuados os cálculos para obter a quantidade de insulina a injetar.

UC6: Dar feedback ao utilizador

O caso de uso 6 agrega as funções dos restantes casos de uso. O utilizador facultar uma imagem da refeição acompanhada por uma moeda de um euro e, caso seja diabético, o utilizador também fornece o valor da sua glicemia antes da refeição. De seguida, o sistema realiza o

reconhecimento dos alimentos, faz a estimativa do peso desses alimentos, calcula os valores nutricionais da refeição e, se o paciente for diabético, calcula também a quantidade de insulina que o utilizador deve injetar. Por fim, o sistema disponibiliza essa informação ao utilizador.

A Figura 62 apresenta o diagrama de sequência simplificado do sistema.

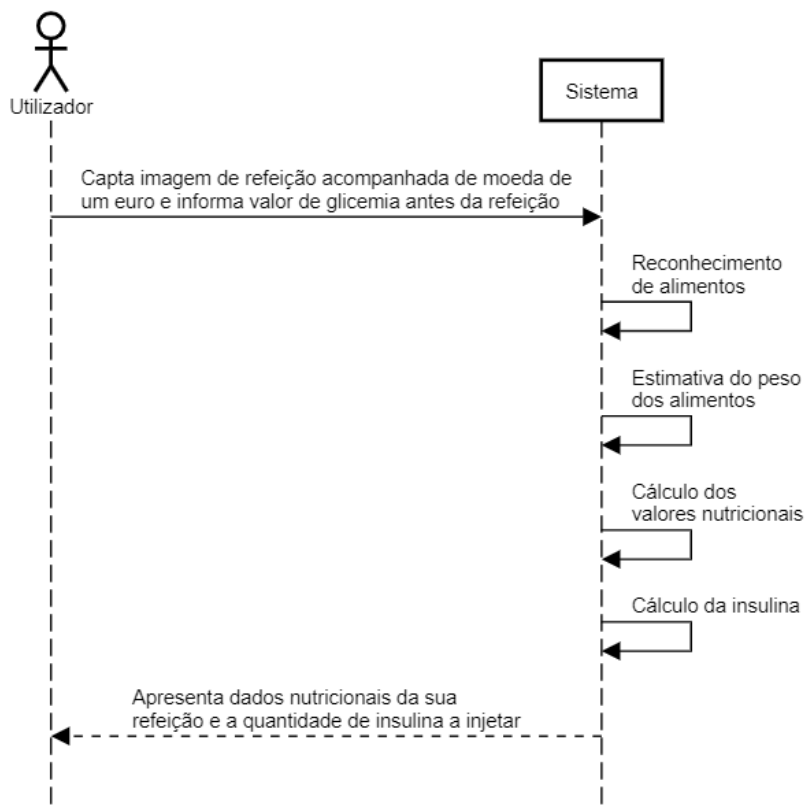


Figura 62 - Diagrama de sequência simplificado do sistema

6.2 Design

Nesta seção é especificada a realização do UC6 apresentado na subseção 6.1.2, com recurso a diagrama de sequência. Apenas é descrito este caso de uso porque é o mais completo, agregando todas as funcionalidades dos restantes casos de uso e descrevendo, de certa forma, todo o sistema.

Como mencionado anteriormente, este caso de uso divide-se em seis funcionalidades:

- Visualização da informação fornecida pelo utilizador;
- Reconhecimento dos alimentos presentes na imagem;
- Estimativa do peso dos alimentos previamente reconhecidos;
- Cálculos nutricionais;

- Cálculos da insulina;
- Fornecimento de informação sobre os valores nutricionais da refeição e da quantidade de insulina que deve injetar.

Assim sendo, este caso de uso foi especificado através do diagrama da Figura 63.

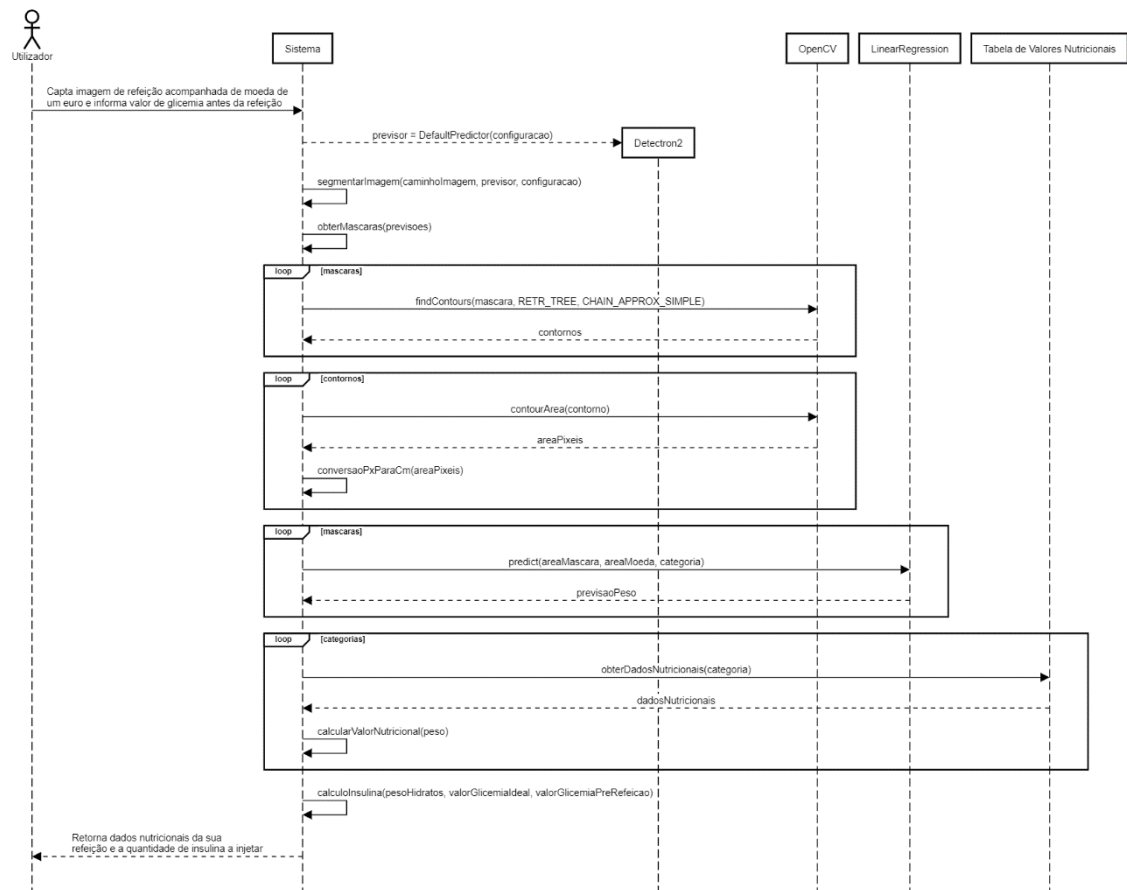


Figura 63 - Diagrama de seqüência do sistema

Como se pode verificar no diagrama anterior, o Utilizador fornece uma imagem da refeição que irá ingerir acompanhada por uma moeda de um euro. Como se trata de utilizador portador da doença diabetes, faculta também o valor da sua glicemia antes da refeição.

Neste diagrama, o Sistema pretende realizar:

- Reconhecimento de alimentos;
- Estimativa de peso dos alimentos;
- Cálculos nutricionais e de glicemia.

Relativamente ao reconhecimento de alimentos, foi utilizado Detectron2 com o modelo PointRend (ver seção 5.5). Após a configuração prévia do modelo, o Sistema cria o previsor com essa configuração. Com o previsor criador, o Sistema realiza a segmentação e deteção da

imagem, realizando assim o reconhecimento dos alimentos e desenhando máscaras nos mesmos.

De modo a prever uma estimativa do peso dos alimentos presentes na imagem, são adquiridas as máscaras resultantes do reconhecimento de alimentos e, para cada máscara, são obtidos os seus contornos. De seguida, é obtida a área de cada contorno em pixéis e é realizada a conversão de pixéis para centímetros quadrados. Sabendo as categorias dos alimentos identificados e as respetivas áreas que ocupam, para cada categoria, o modelo de regressão linear realiza uma previsão, tendo em conta a área da máscara dessa categoria, a área da moeda de um euro e a categoria. O modelo de regressão linear prevê assim o peso de cada categoria.

De seguida, para cada categoria, o Sistema acede uma tabela previamente preenchida com valores nutricionais de vários de alimentos e retorna os dados nutricionais respetivos. É assim realizado o cálculo nutricional de cada alimento, tendo em conta o peso estimado.

Por fim, o Sistema realiza o cálculo de insulina. Para isso, o Sistema possui informação do valor de glicemia ideal do Utilizador e do valor de glicemia que o Utilizador possui antes da refeição. Sabendo que cada porção de hidratos (açúcares) equivale a 12g, e sabendo o peso de hidratos (açúcares) de cada alimento obtido previamente, o Sistema realiza o cálculo de insulina necessária para o Utilizador.

Tanto o valor nutricional da refeição como o valor de insulina são fornecidos ao Utilizador.

6.3 Arquitetura

Como ilustra a Figura 64, a arquitetura deste sistema é composta por quatro componentes: Classificação, Reconhecimento Alimentos, Estimativa Peso e Cálculos.

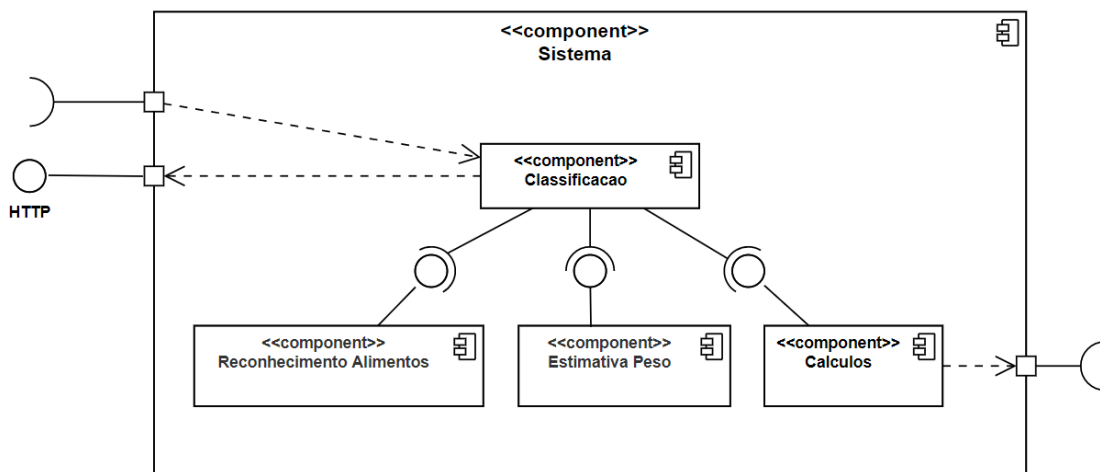


Figura 64 - Diagrama de componentes

A componente Classificação está encarregue da interação entre os atores principais e com as restantes componentes que compõe o sistema. Esta componente recebe uma imagem por

parte do utilizador e, através de interações com os restantes componentes, faz o processamento da informação recebida e disponibiliza essa informação ao utilizador.

A componente Reconhecimento Alimentos é responsável pelo reconhecimento e segmentação dos alimentos. Através da segmentação da imagem, são obtidas as máscaras dos alimentos e é realizada a conversão da área das máscaras de píxeis para centímetros quadrados. Esta componente fornece essa informação à componente Classificação.

A componente Estimativa Peso recebe a informação das máscaras dos alimentos identificados previamente e realiza a previsão do peso dos alimentos. Esta informação é então fornecida à componente Classificação.

A componente Cálculos realiza os cálculos dos valores nutricionais tendo em conta o peso previsto dos alimentos e, caso o utilizador seja diabético, calcula a quantidade de insulina que o utilizador deve injetar. Posteriormente, faculta essa informação à componente Classificação.

A Figura 65 ilustra a arquitetura da solução final:

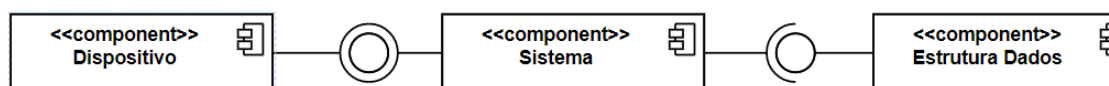


Figura 65 - Arquitetura da solução final

A solução final é composta pelo módulo Sistema que agrega as várias componentes presentes na Figura 64. Este módulo recebe uma imagem fornecida pelo utilizador através de pedidos HTTP. Esta interface é utilizada para fazer o reconhecimento dos alimentos, estimativa do peso dos alimentos e, por fim, disponibiliza dados nutricionais e valor da insulina.

7 Implementação da Solução

Como já foi referido na seção 5.5, optou-se pela aplicação do Detectron2 com o modelo PointRend na detecção, segmentação e classificação de alimentos. A implementação deste modelo pode ser analisada mais detalhadamente na subseção 5.3.2, onde é abordado todo o processo de implementação do Detectron2.

Após a aplicação do Detectron2 com o modelo PointRend, os alimentos foram corretamente detetados (Figura 66). De seguida, foram obtidas as áreas em centímetros quadrados das máscaras de cada alimento e moeda previamente detetados. As áreas foram agrupadas por categoria para que não houvesse mais que uma área por categoria.



Figura 66 - Segmentação de instância resultante da aplicação do PointRend

Estas áreas foram posteriormente enviadas para o modelo de regressão linear (explicado sucintamente na subseção 5.4.2), que através do processo descrito anteriormente, retornaram o peso de cada alimento presente na imagem de entrada. Verifica-se na Tabela 17 uma comparação entre o peso estimado e o peso real dos alimentos identificados na imagem de entrada no modelo. Os valores são bastante próximos da realidade, pelo que com um conjunto de dados maior seria possível realizar previsões cada vez mais fidedignas.



Figura 67 - Alimentos pesados e o seu peso real

Tabela 17 - Relação entre o peso estimado o peso real dos alimentos

Categoria	Peso Real (gramas)	Peso Estimado (gramas)
Beans	90	90.83
Chicken	93	119.78
Rice	85	85.68

Com o peso de cada alimento, foi possível aceder ao conjunto de dados recolhidos sobre a informação nutricional dos alimentos (Figura 68) e fazer os devidos cálculos, que serão apresentados ao utilizador para que este tenha a possibilidade de fazer escolhas mais saudáveis, fazer o registo da sua dieta sem a necessidade de ter uma balança por perto ou, até mesmo, em caso de ser diabético, saber quantas insulinas deve injetar, tendo em conta a refeição que está a ingerir.

A informação nutricional dos alimentos foi recolhida em [81–83].

```
"rice": {
  "calorias": 130,
  "gorduras_totais": 0.3,
  "gorduras_saturadas": 0.1,
  "colesterol": 0,
  "sodio": 0.001,
  "potassio": 0.035,
  "carboidratos": 28.5,
  "fibra_alimentar": 0.4,
  "acucar": 0.1,
  "proteinas": 2.7,
  "vitamina_a": 0,
  "vitamina_c": 0,
  "ferro": 1.14,
  "calcio": 0.76,
  "indice_glicemico": 70
},
```

Figura 68 - Exemplo do ficheiro json com informação nutricional

Tendo em consideração os alimentos detetados na imagem inicial, a estimativa do peso desses alimentos e o seu valor nutricional por 100 gramas, foram realizados os devidos cálculos nutricionais da refeição. Os valores resultantes são posteriormente apresentados ao utilizador.

ALIMENTO: rice	ALIMENTO: beans	ALIMENTO: chicken
PESO: 85.68g	PESO: 90.83g	PESO: 119.78g
- calorias : 111.38 kcal	- calorias : 119.9 kcal	- calorias : 263.52 kcal
- gorduras_totais : 0.26 g	- gorduras_totais : 0.45 g	- gorduras_totais : 15.45 g
- gorduras_saturadas : 0.09 g	- gorduras_saturadas : 0.09 g	- gorduras_saturadas : 4.31 g
- colesterol : 0.0 g	- colesterol : 0.0 g	- colesterol : 0.11 g
- sodio : 0.0 g	- sodio : 0.01 g	- sodio : 0.09 g
- potassio : 0.03 g	- potassio : 0.32 g	- potassio : 0.24 g
- carbohidratos : 24.42 g	- carbohidratos : 21.62 g	- carbohidratos : 0.0 g
- fibra_alimentar : 0.34 g	- fibra_alimentar : 7.9 g	- fibra_alimentar : 0.0 g
- acucar : 0.09 g	- acucar : 0.32 g	- acucar : 0.0 g
- proteinas : 2.31 g	- proteinas : 7.9 g	- proteinas : 28.15 g
- vitamina_a : 0.0 %	- vitamina_a : 10.68 %	- vitamina_a : 0.14 %
- vitamina_c : 0.0 %	- vitamina_c : 0.64 %	- vitamina_c : 0.0 %
- ferro : 0.98 %	- ferro : 10.56 %	- ferro : 10.71 %
- calcio : 0.65 %	- calcio : 1.9 %	- calcio : 1.13 %
- indice_glicemico : 70	- indice_glicemico : 86	- indice_glicemico : 1

Figura 69 - Valores nutricionais dos alimentos baseados no peso estimado

É de notar que o índice glicémico dos alimentos também é apresentado ao utilizador. O índice glicémico é utilizado na elaboração de um plano alimentar para auxiliar no controlo glicémico, e no controlo ou perda de peso.

Alimentos com IG baixo demoram mais para serem digeridos e para haver a quebra das moléculas de glicose, que vai permanecer mais tempo na corrente sanguínea, aumentando a sensação de saciedade. Por outro lado, alimentos com IG alto libertam a glucose rapidamente, e quando há uma descarga de açúcar muito grande no sangue, o corpo armazena essa energia na forma de glicogénio e gordura. Por isso, o índice glicémico é do interesse geral.

Caso o utilizador da aplicação seja diabético, é-lhe apresentada também a quantidade de insulina necessária tendo em conta a quantidade de hidratos de carbono (açúcares) que vai ingerir, o valor da sua glicemia ideal e antes da refeição, e as unidades de insulina por cada 50 mg/dL acima da glicemia ideal indicada.



Figura 70 - Cálculo da quantidade de insulina [84]

```

GLICEMIA IDEAL: 100mg/dL
GLICEMIA PRE REFEICAO: 250mg/dL
1 insulina por cada 50mg/dL de glicemia acima dos 100mg/dL
-----
TOTAL:
- hidratos de carbono: 9.08 g
- porções: 3.84
- insulina rápida antes da refeição: 6.8 unidade(s)

```

Figura 71 - Resultados do cálculo da insulina

Adicionalmente, foi implementada uma *Graphical User Interface* simples com o auxílio da biblioteca Tkinter. Começou-se por criar a base da janela da aplicação e adicionou-se nela

diferentes elementos como o título do programa, uma *frame*, um botão para carregar uma imagem e um botão para fazer a classificação da imagem. Para que ao clicar nos botões executasse a sua função, foram criadas duas funções: `carregarImagem()` e `classificarImagem()`.

A função `carregarImagem()` começa por limpar a *frame* no caso de haver algum widget adicionado previamente à *frame*, permite-se que o utilizador escolha a imagem e por fim, redimensiona-se e mostra-se a imagem na *frame* da janela. A função `classificarImagem()` aplica os modelos criados (Detectron2 com PointRend e Regressão Linear) e apresenta a imagem segmentada, o nome da refeição e a quantidade estimada, os valores nutricionais e a quantidade de insulina rápida necessária para a ingestão dessa refeição.

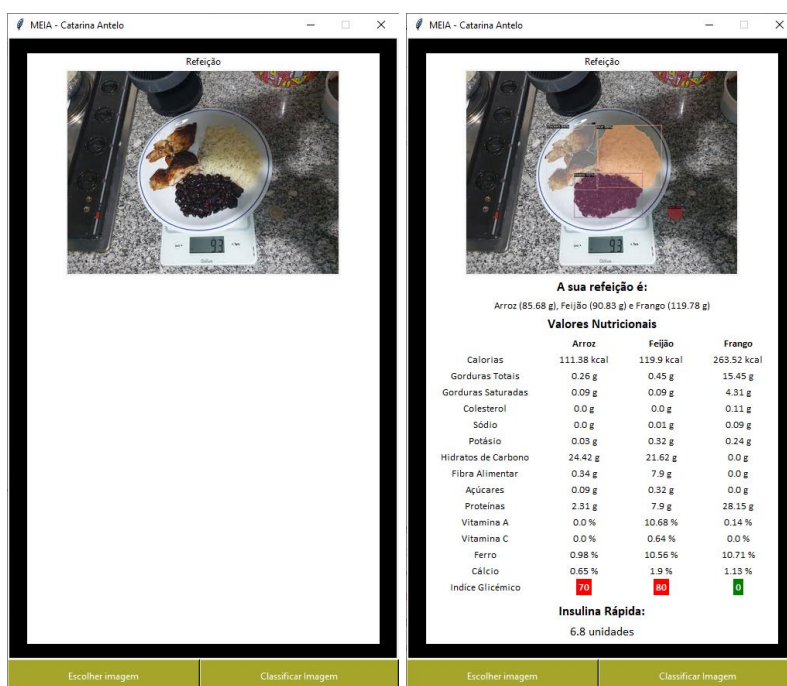


Figura 72 - Interface gráfica da aplicação desenvolvida

Se se comparar com o valor de insulina rápida que o utilizador devia fazer antes da refeição sem utilizar a solução proposta, os resultados seriam:

Tabela 18 – Relação entre valores reais e estimados

Categoria	Real		Estimado	
	Peso (g)	Insulina	Peso (g)	Insulina
Beans	90		90.83	
Chicken	93	6.8	119.78	6.8
Rice	85		85.68	

Como se pode verificar, neste caso a quantidade de insulina seria exatamente a mesma. Por isso, conclui-se que a solução proposta seria vantajosa para o utilizador.

8 Experimentação e Avaliação

Nesta seção pretende-se recriar duas situações onde existe a necessidade de obter informações nutricionais que permitam auxiliar escolhas alimentares. Para isso, foi necessário realizar o reconhecimento de alimentos a partir de uma imagem e posteriormente estimar o peso dos itens alimentares para serem disponibilizados valores nutricionais aproximados da realidade.

De modo a realizar uma avaliação correta da solução final, foi necessário definir especificamente o que vai ser avaliado. Para isso, foram definidos dois casos de estudo onde foi aplicada e a avaliada a solução proposta. Os casos de estudo foram aplicados a dois tipos de utilizadores que facilmente poderiam ser utilizadores interessados no sistema desenvolvido. Ambos, pretendem controlar a sua ingestão alimentar, mas para fins diferentes: emagrecer/tonificar o corpo e monitorizar quantidades de hidratos de carbono para calcular a quantidade de insulina rápida.

8.1 1º Caso de Estudo: Emagrecimento e Tonificação do Corpo

Este caso de estudo foi efetuado em um indivíduo do sexo feminino com 25 anos de idade. O indivíduo trabalha 8 horas por dia sentado, pratica exercício físico diariamente e tem uma dieta equilibrada. Este indivíduo está a passar por um período de constante stress no trabalho, pelo que tem compulsões alimentares com frequência. Atualmente, está a ser acompanhado por um profissional de desporto, que lhe recomendou uma dieta com cerca de 1400kcal diárias.

Para conseguir seguir o plano alimentar definido pelo seu *personal trainer*, o indivíduo utiliza a aplicação MyFitnessPal para fazer o registo das suas refeições. Quando o indivíduo trabalha por casa, pesa todas as suas refeições e faz o registo na aplicação. Contudo, sempre que se desloca para o seu local de trabalho ou para atividades de lazer, o indivíduo precisa de se acompanhar das suas refeições previamente pesadas em casa, para que não falhe a sua dieta.

Já existem muitos estabelecimentos com opções saudáveis, mas nenhum indica o valor nutricional das refeições, a menos que estas sejam previamente embaladas e rotuladas. Um dos grandes desafios de quem começa a seguir uma dieta balanceada com foco no emagrecimento é comer fora de casa e, por isso, muitos indivíduos nesta situação, tendem a privar-se de momentos descontraídos e sociáveis com os amigos e família.

Portanto, a solução é adotar estratégias para conseguir comer fora sem comprometer a dieta. Combinando as escolhas alimentares do utilizador com a solução desenvolvida no âmbito desta dissertação, torna-se mais simples obter a qualidade nutricional das refeições escolhidas. O utilizador consegue fazer o registo das suas refeições sabendo uma estimativa aproximada das quantidades que efetivamente vai ingerir.

8.1.1 Recolha de dados

Para se avaliar a solução, pediu-se ao indivíduo deste caso de estudo que fotografasse as suas refeições acompanhadas por uma moeda de um euro e que registasse o peso dessas mesmas refeições. O indivíduo prontificou-se a realizar este processo durante 2 dias, o que permitiu reunir 10 refeições.

8.1.2 Avaliação da solução

Pretende-se que a solução faça o reconhecimento dos alimentos das refeições, realize uma estimativa do peso dos alimentos e, por fim, disponibilize a informação nutricional da refeição e o peso estimado de cada alimento para facilitar o registo na aplicação MyFitnessPal, utilizada pelo indivíduo deste caso de estudo.

Apesar deste processo ter sido realizado a todas as refeições disponibilizadas pelo utilizador, apenas será apresentado o resultado obtido para uma refeição. Na Figura 73 é possível observar uma das refeições disponibilizadas pelo utilizador e os resultados obtidos do reconhecimento e estimativa de peso dos alimentos.

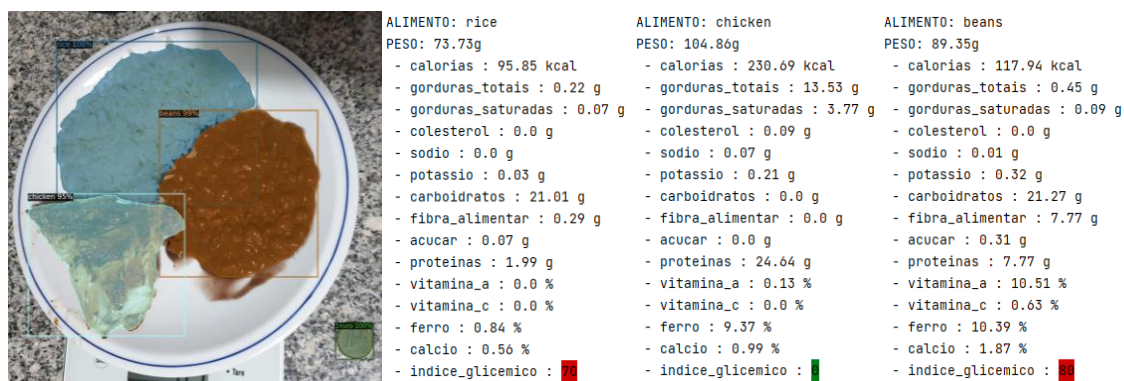


Figura 73 - Refeição do indivíduo do 1º caso de estudo e feedback do sistema

Como se verifica na Figura 73, a refeição composta por frango, arroz e feijão foi devidamente identificada, apresentando as máscaras obtidas dos alimentos. As máscaras são bastante precisas, não identificam zonas em excesso, estando os alimentos e a moeda de um euro perfeitamente contornados.

Com a área que cada alimento ocupa na imagem, o sistema foi capaz de prever o peso dos alimentos previamente identificados. Como é possível verificar na Tabela 19, o peso estimado está bastante próximo da realidade.

Tabela 19 - Peso dos alimentos do 1º caso de estudo

Categoria	Peso Real (gramas)	Peso Estimado (gramas)
Beans	94	89.35
Chicken	82	104.86
Rice	80	73.73

A categoria Chicken (frango) apresenta um erro maior que as restantes deste exemplo. Acredita-se que, com o aumento de dados de treino, o modelo consiga ter uma melhor performance e consequentemente diminuir o erro na estimativa do peso dos alimentos.

8.1.3 Discussão de resultados

Como neste caso de estudo o indivíduo utiliza a aplicação MyFitnessPal e sendo esta uma aplicação que possui o reconhecimento de alimentos na versão paga, decidiu-se verificar o que a aplicação identificava. Em vários casos, a aplicação MyFitnessPal reconheceu diversos alimentos que não tinham semelhanças com a refeição do utilizador, mas retirando as associações que não faziam sentido, para a mesma imagem apresentada na subsecção 8.1.2, os resultados foram:

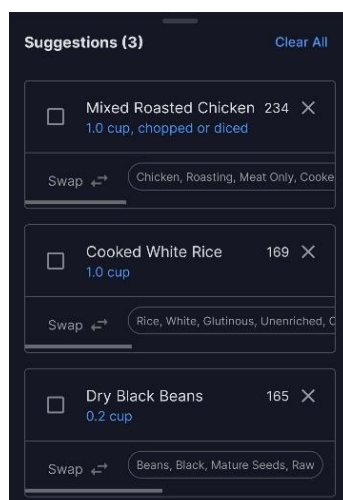


Figura 74 - Resultados aplicação MyFitnessPal

Como se verifica na Figura 74, a aplicação não realiza uma estimativa de peso dos alimentos. A aplicação define um valor padrão por alimento (por exemplo, o frango identificado tem o valor padrão de 1 *cup* que equivale a 100g), mas o utilizador precisa de alterar as quantidades para as que efetivamente está a ingerir.

Assim sendo, o utilizador acaba por ter o mesmo trabalho, tendo de pesar todas as refeições que consome para conseguir fazer um registo da sua dieta alimentar. Ou simplesmente não altera as quantidades e assume estes valores padrão que a aplicação define. E, deste modo, a aplicação acaba por transmitir um sentimento agridoce, pois por um lado os utilizadores acreditam que com este registo padronizado estão a ter uma alimentação saudável e até a cumprirem objetivos delineados, mas por outro lado, estão a enganar-se a si mesmo.

A sociedade não aceita amistosamente o facto de estarem em incumprimento de algo a que se propõem. Apesar da antiga expressão que “errar é humano”, o erro sempre foi assumido na sociedade com conotação negativa. E a verdade é que este acontece muitas vezes por ignorância, muitas outras por puro excesso de confiança.

Se uma pessoa que precisa de ter atenção à sua alimentação não for plenamente sincera consigo própria, admitindo e aprendendo com os seus erros, neste caso, alimentares, nunca estará ao alcance dos seus objetivos. O simples ato de comer cinco bolachas e registar apenas uma porque se trata do valor padrão definido na aplicação, pode trazer impacto na sua saúde.

Como já foi abordado anteriormente, o excesso de peso e a obesidade são fatores que levam ao aparecimento de algumas doenças, por isso se as pessoas usam aplicações que auxiliam o controlo alimentar, é esperado que sejam corretos quando o fazem. Senão, a melhor solução é utilizar a solução proposta nesta dissertação, que além de fazer o reconhecimento de alimentos, estima também o seu peso e disponibiliza toda a informação nutricional das refeições.

O indivíduo deste caso de estudo referiu que esta solução o tornaria mais relaxado nas suas escolhas alimentares quando tivesse de comer fora de casa, pois lhe proporcionaria valores nutricionais muito próximos da realidade sem ter de se preocupar em pesar as refeições.

8.2 2º Caso de Estudo: Paciente Diabético

Este caso de estudo foi efetuado em um indivíduo do sexo masculino, com 27 anos de idade e é portador da doença diabetes há pouco mais de 1 ano. Pratica exercício físico diariamente e faz uma dieta saudável.

A equipa de saúde que o acompanha, informou-o que o seu valor ideal de glicemia antes da refeição é de 100 mg/dL, que deve fazer 1 unidade de insulina “rápida” por cada 50 mg/dL de

glicemia acima dos 100 mg/dL e que deve fazer 1 unidade de insulina “rápida” por cada porção de hidratos de carbono⁹.

No caso das pessoas que fazem insulina dita “rápida” (insulina prandial) antes das refeições, é útil poder calcular a dose exata de insulina a administrar antes das refeições, para que o valor de glicemia depois da refeição não seja nem demasiado alto, nem demasiado baixo.

Numa pessoa com diabetes, o nível de glicemia após uma refeição depende da quantidade de hidratos de carbono (açúcares) ingerida, o valor da glicemia que se tinha antes da refeição e da insulina “rápida” administrada. Para se encontrar a dose certa de insulina “rápida” antes da refeição, é necessário saber a quantidade de porções de hidratos de carbono que vai ser ingerida na refeição e a glicemia antes da refeição (através da realização da “picada” no dedo).

8.2.1 Recolha de dados

De modo a avaliar a solução, pediu-se ao indivíduo deste caso de estudo que facultasse imagens das suas refeições, bem como o peso das mesmas e o valor da sua glicemia antes de ingerir a refeição. O indivíduo disponibilizou 2 dias de registos.

8.2.2 Avaliação da solução

Tal como o caso de estudo anterior, o sistema foi aplicado em todos os dados disponibilizados pelo utilizador, contudo apenas serão apresentadas 2 refeições.

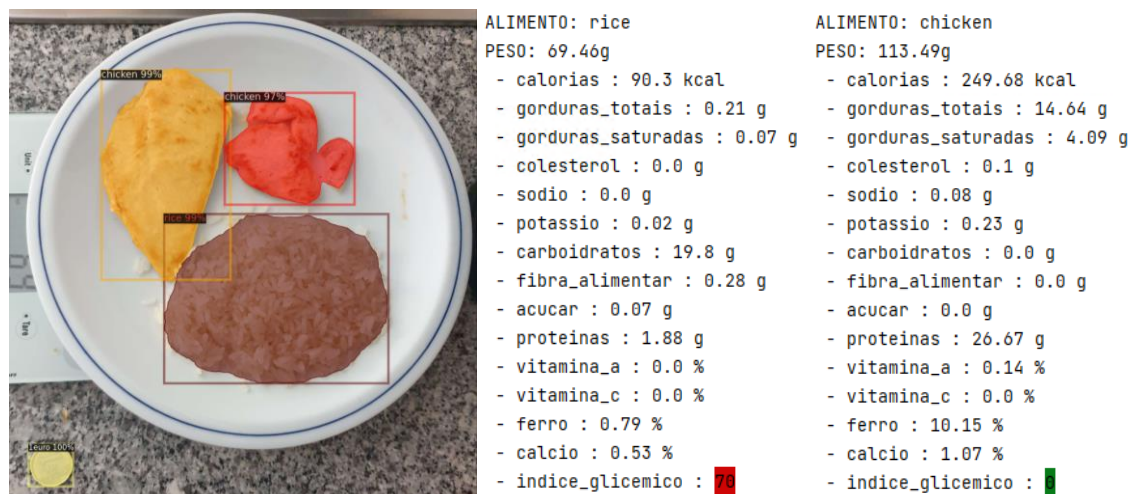


Figura 75 - Primeira refeição do 2º caso de estudo e valores nutricionais

⁹ Uma porção de hidratos de carbono é uma quantidade fixa em gramas de hidrato de carbono (habitualmente 1 porção = 12 gramas)



Figura 76 - Segunda refeição do 2º caso de estudo e valores nutricionais

Com estas imagens e com os valores da Tabela 20, é possível concluir que a detecção e classificação dos alimentos foi realizada corretamente pela solução e que os pesos estimados dos alimentos se encontram próximos da realidade.

Tabela 20 – Peso dos alimentos do 2º caso de estudo

Categoria	Peso Real (gramas)	Peso Estimado (gramas)
Chicken	78	113.49
Rice	69	69.46
Bolonhese	493	428.53

Após serem disponibilizados os valores nutricionais, tratando-se de um utilizador que possui a doença diabetes, o sistema realiza os devidos cálculos para mostrar a quantidade de insulina rápida necessária a ser injetada.

GLICEMIA IDEAL: 100mg/dL
 GLICEMIA PRE REFEICAO: 250mg/dL
 1 insulina por cada 50mg/dL de glicemia acima dos 100mg/dL

 TOTAL:
 - hidratos de carbono: 9.08 g
 - porções: 1.65
 - insulina rápida antes da refeição: 4.7 unidade(s)

Figura 77 - Valor de insulina para uma refeição de arroz com frango

GLICEMIA IDEAL: 100mg/dL
 GLICEMIA PRE REFEICAO: 250mg/dL
 1 insulina por cada 50mg/dL de glicemia acima dos 100mg/dL

 TOTAL:
 - hidratos de carbono: 9.08 g
 - porções: 0.76
 - insulina rápida antes da refeição: 3.8 unidade(s)

Figura 78 - Valor de insulina para uma refeição de massa à bolonhesa

A representação gráfica destes resultados é apresentada na Figura 79.

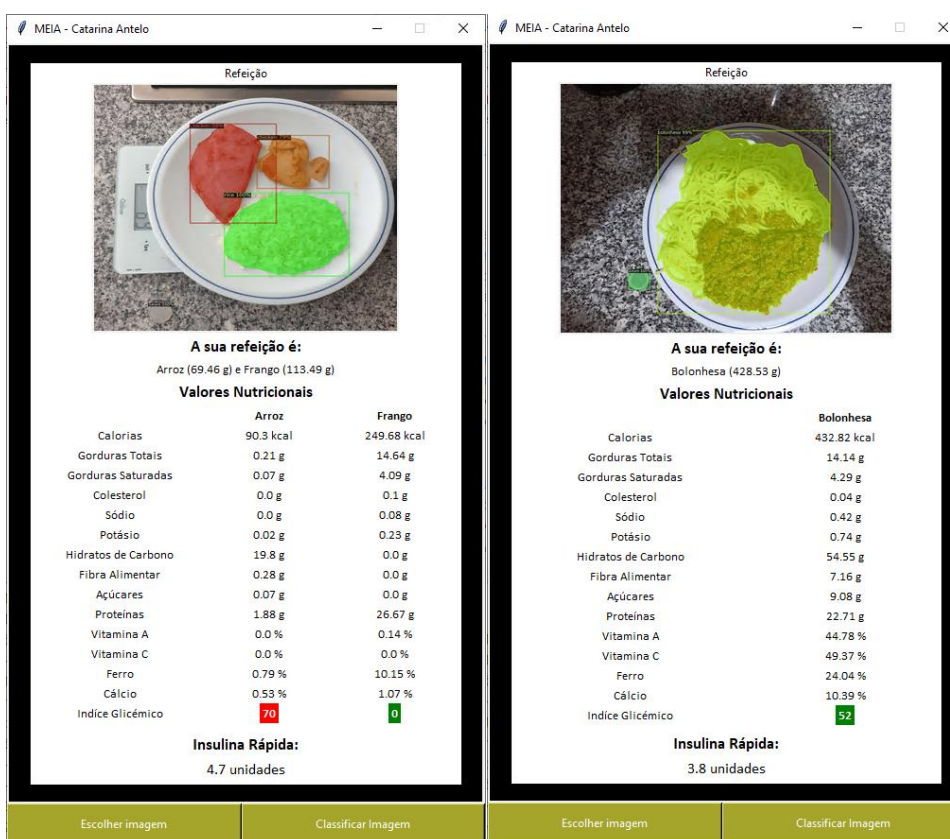


Figura 79 - Aplicação com o feedback das duas refeições

8.2.3 Discussão de resultados

Na subsecção 3.2.4 referiu-se uma aplicação existente no mercado chamada DiabTend, que além de fazer o reconhecimento de alimentos, disponibiliza o cálculo nutricional e o valor da insulina. Assim, acredita-se que neste caso de estudo seria benéfico fazer uma comparação entre a solução descrita nesta dissertação e a aplicação DiabTrend.

A aplicação DiabTrend mostrou-se imprecisa no reconhecimento de alimentos e na estimativa do peso do seu peso. A Figura 80 mostram que, com ambas as refeições aplicadas na subseção anterior, a aplicação DiabTrend identificou incorretamente a comida e quando se tentava obter o peso, verificou-se que o peso dos alimentos variava consoante a posição da câmara. Quando a câmara estava muito próxima do prato de comida, alguns alimentos que pesavam menos de 100 gramas, eram identificados como tendo quase 200 gramas.

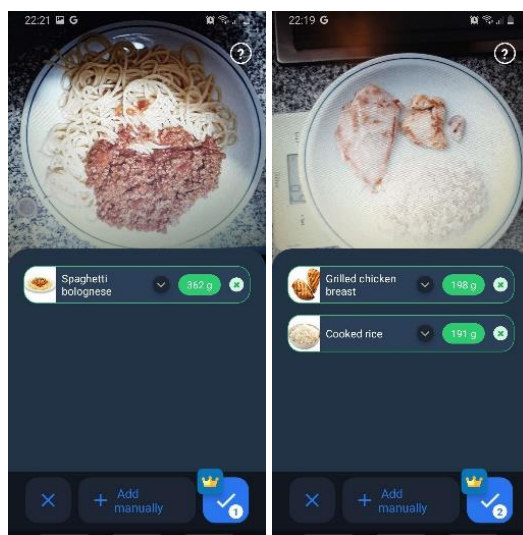


Figura 80 - Reconhecimento de alimentos da aplicação DiabTrend

Estes valores podem levar a que, após os cálculos nutricionais, a aplicação recomende ao utilizador que faça determinada quantidade de insulina rápida devido à quantidade de hidratos de carbono que o sistema identificou na sua refeição, e essa quantidade, estando em excesso, pode aumentar os níveis de glicemia excessivamente e ocorrer hiperglicemia.

Esta situação pode ser evitada com a solução abordada nesta dissertação. As mesmas refeições apresentaram valores de peso e nutricionais bastantes próximos da realidade e os cálculos da quantidade de insulina foram de encontro ao que o indivíduo deste caso de estudo faria caso voltasse a ingerir estas refeições.

Além disso, é importante salientar que o facto de a quantidade de insulina ser apresentado com décimas foi propositado. Quando foi apresentada esta solução ao indivíduo deste caso de estudo, ele sugeriu que apresentasse a quantidade de insulina com décimas. Caso o utilizador esteja a usar uma caneta que vai de uma em uma unidade, é necessário que ele arredonde para a unidade mais próxima (por exemplo, o valor da insulina foi de 4.6u, ficaria 5u).

Com os dados nutricionais e o índice glicémico disponibilizados pela solução apresentada nesta dissertação, o utilizador conseguirá tomar decisões alimentares mais equilibradas, saberá quais são os alimentos da sua refeição que o saciarão por mais tempo e o seu controlo glicémico será mais claro e descomplicado.

9 Conclusões

Neste capítulo são apresentadas as conclusões retiradas do desenvolvimento do trabalho proposto. O capítulo é iniciado pela apresentação da síntese do trabalho realizado, seguido da apresentação dos objetivos concretizados e, por fim, é descrito o trabalho futuro a realizar.

9.1 Síntese

A obesidade e excesso de peso são fatores comportamentais que geram diversas mortes anualmente e provocam o aparecimento e/ou exacerbação de outras doenças, das quais se destacam as doenças cardiovasculares, diabetes e cancro. A diabetes está intimamente relacionada com a obesidade. Decisões alimentares mais saudáveis e conscientes, bem como a prática de exercício físico, podem controlar e evitar a progressão dessas doenças. A adoção de soluções que auxiliem escolhas alimentares, fornecendo dados nutricionais e, mais especificamente para utilizadores diabéticos, apresentem a quantidade de insulina que devem injetar, são essenciais para uma vida melhor e menos despreocupada.

O projeto FoodFriend, na qual o presente trabalho será incluído, consiste no desenvolvimento de uma solução de *coaching* baseada em evidências para auto monitorização e gestão da diabetes tipo 2, com base na ingestão de alimentos e monitorização do estilo de vida. Com a solução proposta nesta dissertação, a monitorização de alimentos ingeridos pelo utilizador será realizada de forma mais rápida e simples. O presente trabalho teve como objetivos:

- Reconhecimento de hábitos alimentares;
- Cálculo nutricional de itens alimentares;
- Disponibilização do histórico de estado de saúde do utilizador;
- Visualização de indicadores que demonstrem a qualidade das refeições ingeridas.

Relativamente ao estado de arte, verificou-se que já existem no mercado várias soluções que ajudem a monitorizar a diabetes e a controlar refeições ingeridas, contudo as aplicações que integram estas duas componentes não conseguem estimar o peso dos alimentos, pelo que o utilizador tem de registar manualmente as quantidades que está a ingerir.

O processo de ter de pesar todas as refeições que se ingere é algo custoso e que nem sempre é praticável, pois ninguém transporta uma balança para todo o lado que se desloca. A solução proposta acaba com este problema. Tanto os alimentos presentes na refeição como o peso dos mesmos, são identificados a partir de uma única imagem disponibilizada pelo utilizador.

Nesta dissertação foram abordados diversos termos fundamentais que permitem compreender o sistema que foi desenvolvido, bem como a definição das ferramentas e métodos que foram utilizadas para o desenvolvimento do sistema. Foram descritas as várias abordagens utilizadas, como estas foram aplicadas e quais resultados obtidas.

Este documento explora tecnologias como Support Vector Machine, Convolutional Neural Network e MobileNetV2 [25], e Detectron2 com o modelo PointRend, para executarem a tarefa de reconhecimento de alimentos. A Reconstrução 3D e um Modelo de Regressão foram analisadas e aplicadas com o objetivo de realizarem uma estimativa do peso dos alimentos previamente identificados.

De modo a avaliar a solução, definiram-se dois casos de estudo:

- Utilizadora com cerca de 25 anos, do sexo feminino, que pretende perder massa gorda e aumentar a sua massa muscular. Pratica exercício físico diariamente, possui uma dieta equilibrada ingerindo cerca de 1400kcal por dia, trabalha 8h/dia sentada.
- Utilizador diabético com cerca de 27 anos, do sexo masculino, que pretende controlar a quantidade de hidratos de carbono que ingere e saber a quantidade de insulina que deve fazer, tendo em conta o que ingeriu.

A solução permitiu obter bons resultados tanto no reconhecimento de alimentos como na estimativa do peso da refeição, disponibilizando informações nutricionais viáveis e próximas da realidade.

9.2 Objetivos Realizados

De forma geral, a solução desenvolvida cumpre os objetivos definidos na seção 1.3. A Tabela 21 apresenta o grau de realização de cada objetivo identificado.

Tabela 21 – Objetivos realizados

Objetivo	Grau de realização
Reconhecimento de hábitos alimentares	Implementado

Estimativa do peso dos alimentos identificados	Implementado
Cálculo nutricional de itens alimentares	Implementado
Cálculo de insulina rápida (caso o utilizador seja paciente diabético)	Implementado
Visualização de indicadores que demonstrem a qualidade das refeições ingeridas	Implementado
Disponibilização do feedback da aplicação através de uma interface gráfica	Implementado

A solução desenvolvida é composta por todas as componentes inicialmente definidas, sendo capaz de, através de uma única imagem captada em tempo real pelo utilizador, disponibilizar indicadores nutricionais sobre a sua refeição que permitem saber a qualidade nutricional dos alimentos ingeridos e, mais direcionado aos pacientes diabéticos, é disponibilizado também as unidades de insulina necessárias em relação à quantidade de hidratos de carbono ingeridos.

Estes dados são exibidos através de uma interface gráfica simples de utilizar, onde o utilizador pode carregar uma imagem a partir do seu dispositivo móvel e de seguida, solicitar a classificação da refeição, onde são apresentados todos os indicadores previamente referidos.

9.3 Resultados Alcançados

Durante o desenvolvimento desta solução, foram utilizadas várias tecnologias de modo a avaliar o seu desempenho quando inseridas na arquitetura deste projeto. As primeiras abordagens utilizadas para este trabalho foram as Support Vector Machines, Convolutional Neural Networks e MobileNetV2. O estudo destas abordagens resultou em um artigo que foi publicado e apresentado na Conferência Europeia de Inteligência Artificial EPIA realizada em setembro de 2022. É possível aceder a este artigo em [25].

9.4 Trabalho Futuro

Embora a solução desenvolvida apresente as funcionalidades fundamentais para o cumprimento dos objetivos inicialmente definidos, esta deve ser melhorada de forma a contemplar todos os requisitos propostos pelo projeto FoodFriend.

Apesar do Detectron2 com o modelo PointRend apresentarem uma boa performance, o conjunto de dados utilizado necessita de ser melhorado. Futuramente, pretende-se que o *dataset* [76] construído no âmbito desta dissertação, seja utilizado para treino do modelo. Assim sendo, será necessário realizar a anotação de todas as imagens disponibilizadas no *dataset*, o que apesar de ser um processo moroso, acredita-se que trará melhorias substanciais

para a solução. Além de reconhecer uma maior quantidade de alimentos, a sua performance pode melhorar bastante com o aumento de dados do *dataset*.

Para que a performance do modelo de Regressão Linear melhore, também será necessário aumentar o conjunto de dados. Contudo, os dados necessários para este modelo terão de ser recolhidos manualmente, pois não existe nenhum repositório público com as características definidas para o treino deste modelo. Este *dataset* será também disponibilizado no Kaggle para auxiliar a comunidade na criação de futuros projetos inovadores, que dependam deste tipo de dados.

Posteriormente, pretende-se efetuar algumas melhorias na aplicação móvel desenvolvida. Será implementada uma interface mais *user-friendly*, permitindo que o utilizador se registre, adicione informação ao seu perfil e consulte o seu histórico de estado de saúde. Pretende-se que a solução permita o registo de dados importantes para o utilizador como o peso, prática de atividade física, calorias consumidas, valores de glucose no sangue, entre outros. Atualmente a aplicação apenas permite carregar uma imagem a partir do dispositivo móvel, por isso será também disponibilizada a possibilidade de capturar uma imagem em tempo real.

Referências

1. Martinho, D., Vieira, A., Carneiro, J., Martins, C., Almeida, A., Marreiros, G.: A Definition of a Coaching Plan to Guide Patients with Chronic Obstructive Respiratory Diseases. *Advances in Intelligent Systems and Computing*. 1161 AISC, 54–64 (2020). https://doi.org/10.1007/978-3-030-45697-9_6/COVER.
2. Martinho, D., Carneiro, J., Corchado, J.M., Marreiros, G.: A systematic review of gamification techniques applied to elderly care. *Artif Intell Rev*. 53, 4863–4901 (2020). <https://doi.org/10.1007/S10462-020-09809-6/TABLES/12>.
3. Martinho, D., Carneiro, J., Neves, J., Novais, P., Corchado, J., Marreiros, G.: A Reinforcement Learning Approach to Improve User Achievement of Health-Related Goals. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 12981 LNAI, 266–277 (2021). https://doi.org/10.1007/978-3-030-86230-5_21/COVER.
4. Martinho, D., Freitas, A., Sá-Sousa, A., Vieira, A., Meira, J., Martins, C., Marreiros, G.: A Hybrid Model to Classify Patients with Chronic Obstructive Respiratory Diseases. *J Med Syst*. 45, (2021). <https://doi.org/10.1007/S10916-020-01704-5>.
5. Martins, C.L., Martinho, D., Marreiros, G., Conceição, L., Faria, L., Simões de Almeida, R.: Artificial Intelligence in Digital Mental Health. <https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-7998-8634-1.ch010>. 201–225 (1)AD. <https://doi.org/10.4018/978-1-7998-8634-1.CH010>.
6. Castel-Branco, J., Martins, S.: A Internet das Coisas em Serviços de Saúde, <https://repositorio.ucp.pt/handle/10400.14/28405>, (2019).
7. João, A., Fernandes, P., Doutora, S.O., Cristina Da Costa, A., Alves, O., Rui, D., Gomes, J.R.: Mestrado em Informática e Sistemas, <https://comum.rcaap.pt/bitstream/10400.26/29858/1/Joao-Pedro-Fernandes-simoes.pdf>.
8. Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e o Desenvolvimento: GECAD, <http://www.gecad.isep.ipp.pt/GECAD/Pages/Presentation/Description.aspx>, last accessed 2022/10/10.
9. Commission, E.: Estado da Saúde na UE.
10. World Health Organization: WHO EUROPEAN REGIONAL OBESITY REPORT 2022. (2022).
11. World Health Organization: Diabetes, <https://www.who.int/news-room/fact-sheets/detail/diabetes>, last accessed 2021/11/10.

12. Harris, M.I.: Diabetes in America - Google Livros, https://books.google.pt/books?hl=pt-PT&lr=&id=hcRrAAAAMAAJ&oi=fnd&pg=PA1&dq=diabetes&ots=gDK9kDXWcS&sig=ikh oLR6aropkPT3tOnTKUUqq_E4&redir_esc=y#v=onepage&q=diabetes&f=false, last accessed 2022/02/08.
13. Portugal, A.P. dos D. de: ABC da Diabetes | Portal da Diabetes, <https://apdp.pt/diabetes/abc-da-diabetes/>, last accessed 2022/01/29.
14. Walker, K.Z., O’Dea, K., Gomez, M., Girgis, S., Colagiuri, R.: Diet and exercise in the prevention of diabetes. *Journal of Human Nutrition and Dietetics*. 23, 344–352 (2010). <https://doi.org/10.1111/J.1365-277X.2010.01061.X>.
15. Nutrição, A.P. de: Lançamento do Manual de Contagem de Hidratos de Carbono na Diabetes Mellitus, <https://www.apn.org.pt/noticia.php?id=330>, last accessed 2022/03/28.
16. Brand Miller, J.C.: Importance of glycemic index in diabetes. *Am J Clin Nutr*. 59, 747S-752S (1994). <https://doi.org/10.1093/AJCN/59.3.747S>.
17. Willett, W., Manson, J., Liu, S.: Glycemic index, glycemic load, and risk of type 2 diabetes. *Am J Clin Nutr*. 76, 274S-280S (2002). <https://doi.org/10.1093/AJCN/76/1.274S>.
18. Riccardi, G., Rivellese, A.A., Giacco, R.: Role of glycemic index and glycemic load in the healthy state, in prediabetes, and in diabetes. *Am J Clin Nutr*. 87, 269S-274S (2008). <https://doi.org/10.1093/AJCN/87.1.269S>.
19. Hodge, A.M., English, D.R., O’Dea, K., Giles, G.G.: Glycemic Index and Dietary Fiber and the Risk of Type 2 Diabetes. *Diabetes Care*. 27, 2701–2706 (2004). <https://doi.org/10.2337/DIACARE.27.11.2701>.
20. Zafar, M.I., Mills, K.E., Zheng, J., Regmi, A., Hu, S.Q., Gou, L., Chen, L.L.: Low-glycemic index diets as an intervention for diabetes: a systematic review and meta-analysis. *Am J Clin Nutr*. 110, 891–902 (2019). <https://doi.org/10.1093/AJCN/NQZ149>.
21. Vega-López, S., Venn, B.J., Slavin, J.L.: Relevance of the Glycemic Index and Glycemic Load for Body Weight, Diabetes, and Cardiovascular Disease. *Nutrients* 2018, Vol. 10, Page 1361. 10, 1361 (2018). <https://doi.org/10.3390/NU10101361>.
22. Sousa, D.J.M. de, Sousa, L.L.C. de, Rodrigues, J.R., Learte, L.R.S., Santos, G.M. dos: Influência do índice glicêmico dos alimentos sobre a concentração de marcadores pró-inflamatórios. *ARCHIVES OF HEALTH INVESTIGATION*. 9, 2020 (2020). <https://doi.org/10.21270/ARCHI.V9I1.3220>.
23. A good guide to good carbs: The glycemic index - Harvard Health, <https://www.health.harvard.edu/healthbeat/a-good-guide-to-good-carbs-the-glycemic-index>, last accessed 2022/10/14.

24. Zhang, J.Y., Jiang, Y.T., Liu, Y.S., Chang, Q., Zhao, Y.H., Wu, Q.J.: The association between glycemic index, glycemic load, and metabolic syndrome: a systematic review and dose–response meta-analysis of observational studies. *European Journal of Nutrition* 2019 59:2. 59, 451–463 (2019). <https://doi.org/10.1007/S00394-019-02124-Z>.
25. Antelo, C., Martinho, D., Marreiros, G.: A Review on Supervised Learning Methodologies for Detecting Eating Habits of Diabetic Patients. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 13566 LNAI, 374–386 (2022). https://doi.org/10.1007/978-3-031-16474-3_31/COVER.
26. Teixeira, B., Martinho, D., Novais, P., Corchado, J., Marreiros, G.: Diabetic-Friendly Multi-agent Recommendation System for Restaurants Based on Social Media Sentiment Analysis and Multi-criteria Decision Making. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 13566 LNAI, 361–373 (2022). https://doi.org/10.1007/978-3-031-16474-3_30/COVER.
27. Pinto, A., Martinho, D., Vieira, A., Ramalho, A., Freitas, A.: Recommendation Systems in the Context of Diabetes Mellitus Type 2: A Bibliometric Analysis. *Lecture Notes in Networks and Systems*. 468 LNNS, 681–691 (2022). https://doi.org/10.1007/978-3-031-04826-5_67/COVER.
28. Godinho, J., Batista, S., Martinho, D., Conceição, L.: A Recommendation System of Nutrition and Physical Activity for Patients with Type 2 Diabetes Mellitus. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 12490 LNCS, 287–297 (2020). https://doi.org/10.1007/978-3-030-62365-4_28/COVER.
29. ITEA 4 · Project · 18032 Food Friend, <https://itea4.org/project/food-friend.html>, last accessed 2022/01/04.
30. Bitencourt, J.H., Hert Da Corégio, M.: Estimativa da Resposta Glicêmica em Refeições para Diabéticos Utilizando Visão Computacional.
31. Neto, E.C., Reboucas, E.S., de Moraes, J.L., Gomes, S.L., Filho, P.P.R.: Development control parking access using techniques digital image processing and applied computational intelligence. *IEEE Latin America Transactions*. 13, 272–276 (2015). <https://doi.org/10.1109/TLA.2015.7040658>.
32. Glucose Buddy app review - YouTube, https://www.youtube.com/watch?v=I4GaGLLND9E&ab_channel=RyanSitko, last accessed 2022/03/04.

33. MyFitnessPal Meal Scan Tutorial (NEW FEATURE!) - YouTube, https://www.youtube.com/watch?v=DjsbEPy6_AQ&ab_channel=KevTheTrainer, last accessed 2022/03/04.
34. Calorie Mama Food AI - Food Image Recognition and Calorie Counter using Deep Learning, <https://www.caloriemama.ai/app>, last accessed 2022/03/04.
35. Selçuk, A.A.: A Guide for Systematic Reviews: PRISMA. *Turk Arch Otorhinolaryngol.* 57, 57 (2019). <https://doi.org/10.5152/TAO.2019.4058>.
36. Darabi, Z., Zarandi, M.H.F., Solgi, S.S., Turksen, I.B.: An intelligent multi-agent system architecture for enhancing self-management of type 2 diabetic patients. 2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2015. (2015). <https://doi.org/10.1109/CIBCB.2015.7300273>.
37. Zhanle Wang Regina, by: The diabetic patient software agent. (2012).
38. Wang, Z., Paranjape, R.: The self-aware diabetic patient software agent model. *Comput Biol Med.* 43, 1900–1909 (2013). <https://doi.org/10.1016/J.COMPBIOMED.2013.09.007>.
39. Sowah, R.A., Bampoe-Addo, A.A., Armoo, S.K., Saalia, F.K., Gatsi, F., Sarkodie-Mensah, B.: Design and Development of Diabetes Management System Using Machine Learning. *Int J Telemed Appl.* 2020, (2020). <https://doi.org/10.1155/2020/8870141>.
40. Shroff, G., Smailagic, A., Siewiorek, D.P.: Wearable context-aware food recognition for calorie monitoring. *Proceedings - International Symposium on Wearable Computers, ISWC.* 119–120 (2008). <https://doi.org/10.1109/ISWC.2008.4911602>.
41. Anthimopoulos, M.M., Gianola, L., Scarnato, L., Diem, P., Mougiakakou, S.G.: A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE J Biomed Health Inform.* 18, 1261–1271 (2014). <https://doi.org/10.1109/JBHI.2014.2308928>.
42. Yogaswara, R.D., Yuniarno, E.M., Wibawa, A.D.: Instance-Aware Semantic Segmentation for Food Calorie Estimation using Mask R-CNN. *Proceedings - 2019 International Seminar on Intelligent Technology and Its Application, ISITIA 2019.* 416–421 (2019). <https://doi.org/10.1109/ISITIA.2019.8937129>.
43. Konstantakopoulos, F., Georga, E.I., Klampanas, K., Rouvalis, D., Ioannou, N., Fotiadis, D.I.: Automatic estimation of the nutritional composition of foods as part of the glucoseml type 1 diabetes self-management system. *Proceedings - 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering, BIBE 2019.* 470–473 (2019). <https://doi.org/10.1109/BIBE.2019.00091>.
44. Usman, M., Ahmad, K., Sohail, A., Qaraqe, M.: The Diabetic Buddy: A Diet Regulator and Tracking System for Diabetics. *Proceedings - International Workshop on Content-*

- Based Multimedia Indexing. 2021-June, (2021).
<https://doi.org/10.1109/CBMI50038.2021.9461897>.
45. Darapaneni, N., Singh, V., Tarkar, Y.S., Kataria, S., Bansal, N., Kharade, A., Paduri, A.R.: Food image recognition and calorie prediction. 2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings. (2021).
<https://doi.org/10.1109/IEMTRONICS52119.2021.9422510>.
 46. Dehais, J., Shevchik, S., Diem, P., Mougiakakou, S.G.: Food volume computation for self dietary assessment applications. 13th IEEE International Conference on BioInformatics and BioEngineering, IEEE BIBE 2013. (2013).
<https://doi.org/10.1109/BIBE.2013.6701615>.
 47. Yue, Y., Jia, W., Sun, M.: Measurement of food volume based on single 2-D image without conventional camera calibration. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS. 2166–2169 (2012). <https://doi.org/10.1109/EMBC.2012.6346390>.
 48. Sun, Z., Li, L., Liu, Y., Du, X., Li, L.: On the Importance of Building High-quality Training Datasets for Neural Code Search. Proceedings - International Conference on Software Engineering. 2022-May, 1609–1620 (2022). <https://doi.org/10.1145/3510003.3510160>.
 49. Zhang, S., Zhang, C., Yang, Q.: Data preparation for data mining. <http://dx.doi.org/10.1080/713827180>. 17, 375–381 (2010).
<https://doi.org/10.1080/713827180>.
 50. Cortes, C., Vapnik, V., Saitta, L.: Support-vector networks. Machine Learning 1995 20:3. 20, 273–297 (1995). <https://doi.org/10.1007/BF00994018>.
 51. Fraj, M. ben: In Depth: Parameter tuning for SVC,
<https://maviator.github.io/2018/01/05/SVC/>, last accessed 2022/04/16.
 52. Aggarwal, C.C.: Neural Networks and Deep Learning. Neural Networks and Deep Learning. (2018). <https://doi.org/10.1007/978-3-319-94463-0>.
 53. Gurney, K.: An Introduction to Neural Networks. (2018).
<https://doi.org/10.1201/9781315273570>.
 54. Asphalt Institute.: Asphalt in pavement preservation and maintenance. 155 (2009).
 55. Lopez, M.M., Kalita, J.: Deep Learning applied to NLP. (2017).
 56. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for Hyper-Parameter Optimization.
 57. O’Shea, K., Nash, R.: An Introduction to Convolutional Neural Networks. (2015).
<https://doi.org/10.48550/arxiv.1511.08458>.

58. Yamashita, R., Nishio, M., Do, R.K.G., Togashi, K.: Convolutional neural networks: an overview and application in radiology. *Insights Imaging*. 9, 611–629 (2018). <https://doi.org/10.1007/S13244-018-0639-9/FIGURES/15>.
59. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 4510–4520 (2018). <https://doi.org/10.48550/arxiv.1801.04381>.
60. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. (2017). <https://doi.org/10.48550/arxiv.1704.04861>.
61. Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., Hemanth, J.: SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustain Cities Soc.* 66, (2021). <https://doi.org/10.1016/J.SCS.2020.102692>.
62. GitHub - facebookresearch/detectron2: Detectron2 is a platform for object detection, segmentation and other visual recognition tasks., <https://github.com/facebookresearch/detectron2>, last accessed 2022/08/22.
63. Noor, S., Waqas, M., Saleem, M.I., Minhas, H.N.: Automatic Object Tracking and Segmentation Using Unsupervised SiamMask. *IEEE Access*. 9, 106550–106559 (2021). <https://doi.org/10.1109/ACCESS.2021.3101054>.
64. Kirillov, A., Wu, Y., He, K., Girshick, R.: PointRend: Image Segmentation as Rendering.
65. Ahmadyan, A., Zhang, L., Ablavatski, A., Wei, J., Grundmann, M.: Objectron: A Large Scale Dataset of Object-Centric Videos in the Wild with Pose Annotations. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 7818–7827 (2020). <https://doi.org/10.48550/arxiv.2012.09988>.
66. Ren, R., Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. *IEEE Trans Pattern Anal Mach Intell*. XX, 1 (2020).
67. Dehais, J., Anthimopoulos, M., Shevchik, S., Mougialakou, S.: Two-view 3D reconstruction for food volume estimation. *IEEE Trans Multimedia*. 19, 1090–1099 (2017). <https://doi.org/10.1109/TMM.2016.2642792>.
68. Baptistella, M., Teresinha Arns Steiner, M., Chaves Neto, A.: O uso de redes neurais e regressão linear múltipla na engenharia de avaliações: determinação dos valores venais de imóveis urbanos.
69. van Rossum Fred Drake, G.L., PythonLabs, editor: An Introduction to Python Release 2.2.2.

70. scikit-learn: machine learning in Python — scikit-learn 1.1.2 documentation, <https://scikit-learn.org/stable/>, last accessed 2022/04/26.
71. Oliphant, T.E.: Python for scientific computing. *Comput Sci Eng.* 9, 10–20 (2007). <https://doi.org/10.1109/MCSE.2007.58>.
72. Zhou, Q.-Y., Park, J., Koltun, V.: Open3D: A Modern Library for 3D Data Processing.
73. RGBD images — Open3D 0.15.1 documentation, http://www.open3d.org/docs/release/tutorial/geometry/rgbd_image.html, last accessed 2022/09/21.
74. Ringnér, M.: What is principal component analysis? *Nature Biotechnology* 2008 26:3. 26, 303–304 (2008). <https://doi.org/10.1038/nbt0308-303>.
75. Pirjatullah, Kartini, D., Nugrahadi, D.T., Muliadi, Farmadi, A.: Hyperparameter Tuning using GridsearchCV on the Comparison of the Activation Function of the ELM Method to the Classification of Pneumonia in Toddlers. *Proceedings - 2021 4th International Conference on Computer and Informatics Engineering: IT-Based Digital Industrial Innovation for the Welfare of Society, IC2IE 2021.* 390–395 (2021). <https://doi.org/10.1109/IC2IE53219.2021.9649207>.
76. Antelo, C.: Portuguese Meals Dataset | Kaggle, <https://www.kaggle.com/datasets/catarinaantelo/portuguese-meals>, last accessed 2022/09/03.
77. Bossard, L., Guillaumin, M., van Gool, L.: Food-101 - Mining discriminative components with random forests. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 8694 LNCS, 446–461 (2014). https://doi.org/10.1007/978-3-319-10599-4_29/COVER.
78. Mader, K.S.: Food Images (Food-101) | Kaggle, <https://www.kaggle.com/datasets/kmader/food41/metadata>, last accessed 2022/07/24.
79. Shakeel, M.H.: GitHub - haroonshakeel/labelme2coco: How to create custom COCO data set for instance segmentation, <https://github.com/haroonshakeel/labelme2coco>, last accessed 2022/09/04.
80. COCO - Common Objects in Context, <https://cocodataset.org/#detection-eval>, last accessed 2022/10/18.
81. U.S. DEPARTMENT OF AGRICULTURE: FoodData Central, https://fdc.nal.usda.gov/fdc-app.html#/, last accessed 2022/09/10.
82. Nutritionix - Largest Verified Nutrition Database, <https://www.nutritionix.com/>, last accessed 2022/10/13.

83. Glycemic Index Food Chart - Glycemic Index (GI) & Glycemic Load (GL) Guide, <https://glycemic-index.net/>, last accessed 2022/10/14.
84. Portugal, A.P. dos D. de: Controlar a Diabetes: Contagem de Hidratos de Carbono e Ajustes de Insulina - Parte 2, <https://controlardiabetes.pt/educacao-multimedia/contagem-de-hidratos-de-carbono-e-ajustes-de-insulina-parte-2>, last accessed 2022/10/01.