



Reconhecimento e recomendação de hábitos alimentares de pacientes diabéticos do tipo 2

JOÃO CARLOS CAMPOS MOREIRA DA SILVA

Outubro de 2022

Reconhecimento e recomendação de hábitos alimentares de pacientes diabéticos do tipo 2

João Carlos Campos Moreira Da Silva

Engenheiro Biomédico pelo Instituto Superior de Engenharia do Porto

“Dissertação apresentada no Instituto Superior de Engenharia do Porto para a obtenção de grau de Mestre em Engenharia Biomédica”

Orientadores: Jorge Meira e Diogo Martinho

[Outubro 2022]

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota”

Madre Teresa de Calcutá

Agradecimentos

Quero começar por agradecer ao Instituto Superior de Engenharia do Porto por toda a formação recebida, bem como a todos os professores e colegas com quem interagi durante o meu percurso académico e que contribuíram para o meu desenvolvimento quer académico quer pessoal.

Aos professores Jorge Meira e Diogo Martinho, responsáveis pela minha orientação, por toda a atenção, apoio e disponibilidade prestada ao longo do desenvolvimento deste trabalho.

Por fim, gostaria de agradecer a todos os familiares e amigos que, direta ou indiretamente, me apoiaram e ajudaram, ao longo dos últimos 5 anos, pois sem eles nunca teria chegado até aqui.

Resumo

As doenças relacionadas com a alimentação são um dos principais problemas de saúde pública, sendo a DM2 uma das principais doenças. Esta é uma doença crónica caracterizada por apresentar uma resistência à insulina, levando a um aumento dos níveis de glicose sanguínea, e que pode trazer problemas graves para os pacientes. Embora não exista uma cura, um estilo de vida saudável pode ser suficiente para a controlar. Desta forma o controlo da alimentação pode trazer francas melhorias à qualidade de vida destas pessoas, sendo que o desenvolvimento das tecnologias veio oferecer novos meios para tornar este processo mais rápido, simples e cómodo face aos métodos tradicionais.

Assim, os objetivos do presente trabalho passam pelo desenvolvimento de um sistema, simples e intuitivo, que seja capaz de identificar uma refeição a partir de uma foto da mesma, e determinar os valores nutricionais dessa mesma refeição.

Para tal, recorreu-se a métodos de *machine learning* para tentar desenvolver um sistema com estas características. O modelo proposto é baseado em RNC combinadas com métodos de *transfer learning*, sendo que para o desenvolvimento do mesmo recorreu-se a 5 classes da base de dados ETHZ Food-101. O sistema desenvolvido contempla ainda um modelo para identificar e dar *feedback* da informação nutricional relativa às refeições da pessoa.

Nos testes realizados ao modelo obtiveram-se bons resultados a nível de performance, de tal forma que pudesse considerar que o modelo desenvolvido é capaz de realizar previsões com elevada exatidão, mesmo com uma resolução reduzida, para as 5 classes usadas. Embora seja difícil comparar os resultados obtidos com outras abordagens presentes na literatura foi possível verificar que as RNC são o melhor método para este tipo de sistema.

Palavras-chave: Diabetes Tipo 2; *Machine Learning*, Redes Neurais Convolucionais; Reconhecimento Alimentar.

Abstract

Diet-related diseases are a major public health problem, with DM2 being one of the main ones. This is a chronic disease characterised by showing insulin resistance, leading to an increase in blood glucose levels, and which can bring serious problems for patients. Although there is no cure, a healthy lifestyle may be enough to control it. Thus, the development of technology has offered new ways to make this process faster, simpler, and more convenient than traditional methods.

So, the objectives of this work are to develop a simple and intuitive system that can identify a meal from a photo of it and determine the nutritional values of that meal.

For this purpose, machine learning methods were used to try to develop a system with these characteristics. The proposed model is based on the RNC combined with transfer learning methods, and for its development 5 classes from the ETHZ Food-101 database were used. The developed system also includes a model to identify and provide feedback on the nutritional information related to a person's meals.

In the tests performed on the model, good performance results were obtained, such that it could be considered that the model developed can make highly accurate predictions, even with a reduced resolution, for the 5 classes used. Although it is difficult to compare the results obtained with other approaches present in the literature, it was possible to verify that the RNC are the best method for this type of system.

Key words: Type 2 Diabetes; Machine Learning, Convolutional Neural Networks; Food Recognition.

Índice

1.	INTRODUÇÃO.....	3
1.1.	ENQUADRAMENTO E MOTIVAÇÃO	3
1.2.	OBJETIVOS.....	4
1.3.	ORGANIZAÇÃO DO DOCUMENTO	5
2.	ESTADO DA ARTE	9
2.1.	ALIMENTAÇÃO E SAÚDE PÚBLICA	9
2.2.	DIABETES	10
2.3.	MÉTODOS DE RECONHECIMENTO ALIMENTAR	18
2.4.	MACHINE LEARNING	23
2.4.1.	<i>Random Forest</i>	24
2.4.2.	<i>Support Vector Machine</i>	25
2.4.3.	<i>Naive Bayes</i>	26
2.4.4.	<i>Redes Neurais Artificiais</i>	26
3.	DESENVOLVIMENTO.....	31
3.1.	BASE DE DADOS	31
3.2.	API NINJAS	32
3.3.	MÉTODOS DE AVALIAÇÃO	33
3.4.	ABORDAGEM PROPOSTA.....	34
4.	DISCUSSÃO E RESULTADOS	47
4.1.	RESULTADOS	47
4.2.	DISCUSSÃO	50
5.	CONCLUSÕES	59
5.1.	CONCLUSÃO	59
5.2.	TRABALHOS FUTUROS	60
	REFERÊNCIAS	61

Lista de Figuras

Figura 1. Número total, estimado, de adultos (20-79 anos) com diabetes em 2021.	12
Figura 2. Prevalência relatada da DM2 em jovens por região e etnia.	14
Figura 3. Incidência reportada de DM2 em jovens por região e etnia.	15
Figura 4. Etapas de um sistema de classificação alimentar baseado em imagens, com informação nutricional.	19
Figura 5. Esquema representativo do funcionamento de uma rede neuronal artificial inspirado na comunicação que ocorre entre os neurónios dos sistemas biológicos.	27
Figura 6. Diagrama representativo do modelo de <i>machine learning</i> desenvolvido.	35
Figura 7. Divisão do dataset, neste caso apenas da classe <i>chicken_wings</i> , em imagens de treino, validação e teste.	36
Figura 8. Leitura e redimensionamento das imagens.	37
Figura 9. Normalização das imagens de treino, validação e teste.	37
Figura 10. Definição das diferentes camadas presentes na RNC desenvolvida.	38
Figura 11. Definição dos processos que ocorrem na fase de <i>augmentation</i> .	38
Figura 12. Arquitetura da <i>EfficientNetB3</i> .	39
Figura 13. Definição da função <i>compile()</i> .	40
Figura 14. Definição da função <i>fit()</i> .	41
Figura 15. <i>Callbacks</i> usados no modelo proposto.	41
Figura 16. Função usada para avaliar a RNC desenvolvida.	41
Figura 17. Resultados obtidos num dos testes realizados à rede desenvolvida, sendo estes o output da função <i>evaluate()</i> .	42
Figura 18. Função usada para guardar o modelo desenvolvido.	42
Figura 19. Função usada para carregar o modelo anteriormente desenvolvido.	42
Figura 20. Função <i>predict()</i> , usada para se fazer a identificação do prato inserido pelo utilizador.	42
Figura 21. Janela na qual o utilizador indica ao sistema a porção ingerida.	43
Figura 22. Formação da <i>query</i> usada para fazer a chamada da <i>Nutrition API</i> .	43
Figura 23. Chamada da <i>Nutrition API</i> .	43

Figura 24. *Display* das informações relativas ao *input* do utilizador, nomeadamente o nome da refeição, a imagem inserida e ainda a informação nutricional correspondente. 44

Figura 25. Gráfico que demonstra a forma como tanto a training loss como a validation loss variam ao longo das épocas de treino. 49

Figura 26. Gráfico que demonstra a forma como tanto a training accuracy como a validation accuracy variam ao longo das épocas de treino. 50

Lista de Tabelas

Tabela 1. Exemplos de algumas das aplicações de reconhecimento alimentar já desenvolvidas.	23
Tabela 2. Número de imagens em cada uma das classes utilizadas após a filtragem manual.	32
Tabela 3. Valores de <i>accuracy</i> obtidos usando algoritmos de classificação baseados em SVM, RF e NB.	47
Tabela 4. Valores de <i>accuracy</i> , <i>precision</i> e <i>recall</i> obtidos usando a rede neuronal proposta.	48
Tabela 5. Alguns dos principais trabalhos de classificação alimentar baseada em imagens da base de dados ETHZ Food-101 presentes na literatura.	53

Lista de Abreviaturas

- API - Application Programming Interface (Interface de Programação de Aplicações)
- DM2 - Diabetes Mellitus Tipo 2
- ETHZ - Eidgenössische Technische Hochschule Zürich (Instituto Federal Suíço de Tecnologia de Zurique)
- GECAD - Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e o Desenvolvimento
- IDF - International Diabetes Federation (Federação Internacional de Diabetes)
- ISEP - Instituto Superior de Engenharia do Porto
- NB - Naive Bayes
- OMS - Organização Mundial da Saúde
- RF - Random Forest
- RNA - Redes Neurais Artificiais
- RNC - Redes Neurais Convolucionais
- SVM - Support Vector Machines

CAPÍTULO 1 – INTRODUÇÃO

1. Introdução

Este capítulo serve o propósito de introduzir o tema abordado nesta dissertação, na qual foi desenvolvido um sistema de processamento digital de imagem para a detecção de hábitos alimentares de pacientes diabéticos do tipo 2. Deste sistema, resultam ainda algumas informações relativas às refeições ingeridas pelos utilizadores.

Este capítulo, para além de servir para fazer o enquadramento e apresentar as motivações que levaram à realização deste trabalho, também apresenta objetivos que se pretendem alcançar com o mesmo. Para além destes pontos, será ainda descrita a estrutura e organização desta dissertação.

1.1. Enquadramento e Motivação

O presente trabalho insere-se no projeto *Food Friend*, no qual o ISEP-GECAD é um dos parceiros [1]. Este projeto está relacionado com: a prevenção da desnutrição em lares de idosos e em pacientes alimentados via sonda; e com o tratamento transversal nutricional de doenças crónicas. Os objetivos do *Food Friend* passam por desenvolvimento de um conjunto de ferramentas que seja capaz de medir automaticamente a ingestão de alimentos de uma pessoa com o mínimo de entrada necessária e transformar esta informação em feedback personalizado, permitindo uma visão clara da dieta quer para os pacientes quer para profissionais de saúde [2].

As doenças relacionadas com a alimentação são uma grande preocupação de saúde pública e continuam a pôr em risco tanto a saúde da população como a sustentabilidade dos sistemas de saúde. De tal modo, que a ingestão desequilibrada de alimentos pode resultar em distúrbios metabólicos, desnutrição, excesso de peso, baixo desempenho mental, entre outros fatores de risco para a saúde, como hipertensão e osteoporose. Estes efeitos podem levar ao aparecimento de determinadas doenças, como é o caso da diabetes mellitus tipo 2 (DM2) [2].

A DM2 corresponde a cerca de 90% de todos os casos de diabetes. Esta doença é caracterizada por apresentar uma resistência à insulina, o que leva a um aumento dos níveis de glicose sanguínea, e está associada à adoção de um estilo de vida menos saudável [3].

Quando não tratada corretamente, a DM2 pode gerar complicações cerebrais, renais, problemas de visão, entre outras. Assim, embora não exista uma cura, é necessário

haver um controlo da mesma, de modo a evitar estas complicações. Deste modo, recomenda-se que os pacientes que possuam esta doença realizem algumas mudanças no estilo de vida, de modo a reduzirem o seu peso, apresentarem uma alimentação adequada e ainda praticar exercício físico de forma regular, sendo que, em certos casos, estas mudanças podem ser suficientes para manter os valores glicémicos estáveis. Caso estas mudanças só por si não sejam suficientes pode recorrer ao uso de medicamentos antidiabéticos orais, ou até administração de insulina [4].

Por estas razões, torna-se necessário acompanhar e apoiar os pacientes na sua rotina diária para que seja promovido o estilo de vida adaptado e personalizado a cada um [5]. Tradicionalmente, para se realizar o acompanhamento destes pacientes, recorria-se ao uso de papel e caneta para anotar os alimentos ingeridos. Contudo, este era um método inteiramente manual, que não podia ser considerado totalmente confiável ou suficiente para auxiliar o paciente sempre que ele necessita. Dada a baixa comodidade e falta de confiança, vários pacientes acabavam desmotivados o que levava a que estes abandonassem o controlo desta doença [5].

No entanto, o avanço das tecnologias, especialmente ao nível dos smartphones e das suas câmaras, veio oferecer novos métodos para realizar este acompanhamento, de tal forma que o registo dos alimentos ingeridos pode estar à distância de uma fotografia. Estas melhorias levaram a que fossem desenvolvidos sistemas e aplicações móveis, que tornam o acompanhamento destes pacientes mais rápido, cómodo e simples, levando a que estes sejam mais motivadores em relação às técnicas mais tradicionais [6].

Assim, a tecnologia abriu portas à possibilidade do desenvolvimento de aplicações móveis que podem ser usadas para apoiar e acompanhar pacientes com DM2 em tempo real, através de um registo contínuo da sua rotina. Com o registo contínuo da rotina diária do paciente, é possível ter acesso aos valores nutritivos dos alimentos que este ingere e, deste modo, aconselhá-lo de forma a manter os seus níveis glicémicos estáveis. Isto poderá proporcionar um estilo de vida mais saudável aos seus utilizadores, mantendo assim a sua doença controlada de uma forma simples e cómoda.

1.2. Objetivos

O presente trabalho enquadra-se na área da tecnologia aplicada à saúde, e propõe-se ao desenvolvimento de um sistema capaz de detetar as refeições ingeridas pelos pacientes de DM2 através de imagens.

Como tal, o objetivo principal deste trabalho passa pela captura e identificação de informação associada aos hábitos alimentares de pacientes diabéticos do Tipo 2, tais como refeições ingeridas. Para além da identificação das refeições ingeridas, o sistema deverá emitir informações sobre a refeição, nomeadamente os seus dados nutricionais.

Para tal, o sistema deverá ser capaz de realizar as seguintes tarefas:

- Identificar a comida presente na imagem;
- Estimar e apresentar o valor nutricional da refeição.

O sistema deverá ser personalizável de modo que cada utilizador possa estabelecer a quantidade de alimentos ingerida, desta forma o utilizador obterá informações mais próximas da realidade. Para além disso, o sistema deve ser simples de usar e intuitivo dado que a DM2 afeta principalmente pessoas mais envelhecidas e que podem ter mais dificuldade a lidar com tecnologia.

1.3. Organização do documento

A presente dissertação está dividida em cinco capítulos. Este primeiro capítulo serve o propósito de fazer o enquadramento do trabalho, apresentando os objetivos e motivações, bem como a estrutura do documento.

O segundo capítulo intitula-se de estado da arte e tem como finalidade familiarizar os leitores dos problemas associados à alimentação e saúde pública, no geral, e à diabetes em particular. Para além disso, neste consta também uma abordagem aos métodos de reconhecimento alimentar e dos métodos de *machine learning* usados no desenvolvimento do trabalho.

Depois segue-se o terceiro capítulo, o desenvolvimento, onde são apresentados os materiais e os métodos utilizados para desenvolver o sistema proposto, sendo este descrito passo a passo também neste capítulo.

O quarto capítulo, por sua vez, apresenta os resultados obtidos quer da solução proposta quer de outros métodos testados durante o desenvolvimento do presente projeto. Para além disso este capítulo contempla também um paralelismo entre os resultados obtidos com os resultados obtidos por outros autores.

Por fim, o capítulo cinco apresenta as principais conclusões deste trabalho e dá algumas sugestões para trabalhos futuros.

CAPÍTULO 2 – ESTADO DA ARTE

2. Estado da Arte

Nesta secção será abordada a importância que a alimentação tem na saúde, nomeadamente em paciente com DM2. Para além disso também serão identificados e analisados sistemas e/ou aplicações de deteção de alimentos, com especial foco em sistemas orientados para a gestão nutricional.

2.1. Alimentação e Saúde Pública

A alimentação constitui uma das atividades humanas mais importantes, não só por ser uma necessidade biológica, mas também por envolver aspetos económicos, sociais, científicos, psicológicos e culturais [7].

Esta é considerada um fator vital para a vida humana, ainda assim, a alimentação é muito mais do que apenas ingerir nutrientes, uma vez que esta tem um significado muito próprio para cada pessoa e/ou grupo de pessoas, podendo chegar a constituir um traço de identidade de um indivíduo [8].

Além disso, a alimentação é um dos principais determinantes de saúde, sendo que a Organização Mundial da Saúde (OMS) relata que dietas pouco saudáveis e falta de atividade física são alguns dos principais problemas de saúde global. A OMS relata ainda que uma dieta saudável ajuda a prevenir a desnutrição em todas as suas formas, bem como uma série de doenças não transmissíveis e condições [9]. Por outro lado, o consumo desequilibrado de alimentos pode levar ao aparecimento de várias doenças e desnutrição, que tem uma influência negativa nos resultados clínicos, na função corporal, na autonomia e na qualidade de vida, podendo resultar distúrbios metabólicos, desnutrição, excesso de peso, baixo desempenho mental, entre outros fatores de risco para a saúde [2], [10].

Assim, os hábitos alimentares podem estar associados ao aparecimento de problemas de saúde tais como a hipertensão, osteoporose, doenças cardiovasculares, DM2, patologias hepáticas, doença de Alzheimer e cancro.

Atualmente, em Portugal, os hábitos alimentares inadequados são um dos 5 principais fatores de risco para a perda de anos de vida saudável e a mortalidade, contribuindo para 7,3% dos DALYs (*Disability-adjusted life years*) e para 11,4% da mortalidade, no ano de 2019. Estas taxas aumentam ainda mais quando se considera o peso dos hábitos alimentares juntamente com os fatores de risco metabólicos associados

à alimentação inadequada (índice de massa corporal elevado, a glicose plasmática aumentada, a hipertensão arterial e o colesterol LDL elevado), passando a ser 38% e 60%, respetivamente. Assim, pode-se inferir que, em Portugal, a prevalência das doenças crónicas associadas à alimentação inadequada é elevada, sendo provavelmente um dos principais problemas de saúde pública [11].

2.2. Diabetes

A Diabetes trata-se de uma doença metabólica crónica caracterizada por níveis elevados de glicose (ou açúcares) no sangue. Esta é muito observada na nossa sociedade, de tal forma que o número de casos tem vindo a aumentar nas últimas 3 décadas [4], [12], [13].

A Diabetes decorre de uma deficiência na secreção de insulina (responsável pela entrada da glicose nas células), e/ou incapacidade de captar a glicose sanguínea. Assim, a glicose permanece no sangue em vez de fornecer energia às células, levando a concentrações anormalmente altas de glicose no sangue [4].

De acordo com a Direção Geral de Saúde (DGS), os critérios de diagnóstico da diabetes são: apresentar uma glicemia em jejum superior a 126 mg/dL (ou $\geq 7,0$ mmol/L); apresentar, juntamente com sintomas típicos da diabetes, uma glicemia ocasional superior a 200 mg/dL (ou $\geq 11,1$ mmol/L); apresentar valores de Hemoglobina glicada A1c (HbA1c) superiores a 6,5 %; ou então apresentar uma glicemia superior 200 mg/dL (ou $\geq 11,1$ mmol/L) às 2 horas, na prova de tolerância à glicose oral (PTGO) com 75g de glicose [4], [14].

Pessoas com diabetes podem vir a desenvolver uma série de complicações agudas e crónicas à saúde, podendo provocar sérios danos ao coração, vasos sanguíneos, olhos, rins e nervos caso esta não seja diagnosticada e monitorizada de forma correta. Esta é também uma das principais causas de morbilidade e mortalidade na população geral [4], [12].

Existem diferentes tipos de diabetes, sendo que estes podem ser classificados como [15], [16]:

- Diabetes mellitus tipo 1 - que ocorre devido à destruição autoimune de células beta, que resulta numa deficiência absoluta de insulina;

- Diabetes mellitus tipo 2 – ocorre devido a uma diminuição progressiva da secreção de insulina das células beta e conseqüentemente a uma resistência da insulina;
- Diabetes mellitus gestacional - correspondem a diabetes diagnosticado no segundo ou terceiro semestre da gravidez que não tinham sido diabetes evidentes no período anterior à gravidez. Pode decorrer da ação das hormonas produzidas pela placenta que, durante uma gravidez, dificultam a ação da insulina e a estabilização dos valores de glucose no sangue;
- Outros tipos específicos de diabetes por outros motivos, mas que ocorrem com menos frequência, como por exemplo, diabetes Tipo LADA (*Latent Autoimmune Diabetes in Adults*), diabetes tipo MODY (*Maturity-Onset Diabetes of the Young*), diabetes associado a poliendocrinopatias auto-imunes, diabetes associados ao aumento de função das glândulas endócrinas, doenças pancreáticas ou à anormalidade da insulina.

A OMS estima que, atualmente, cerca de 422 milhões de pessoas em todo o mundo têm diabetes. Para além disso, 1,5 milhões de mortes são atribuídas diretamente à diabetes todos os anos. Estes dados ficam ainda mais preocupantes dado que tanto o número de casos como a prevalência da Diabetes têm aumentado constantemente nas últimas décadas [12].

Segundo a décima edição do IDF (*International Diabetes Federation*) Diabetes Atlas, os países que demonstram maior número de casos de Diabetes são os Estados Unidos da América, a China, a Índia e Paquistão, sendo que estes apresentam mais de 20 milhões de casos [17]. Estes dados, bem como os dados relativos aos restantes países do mundo podem ser visualizados na Figura 1.

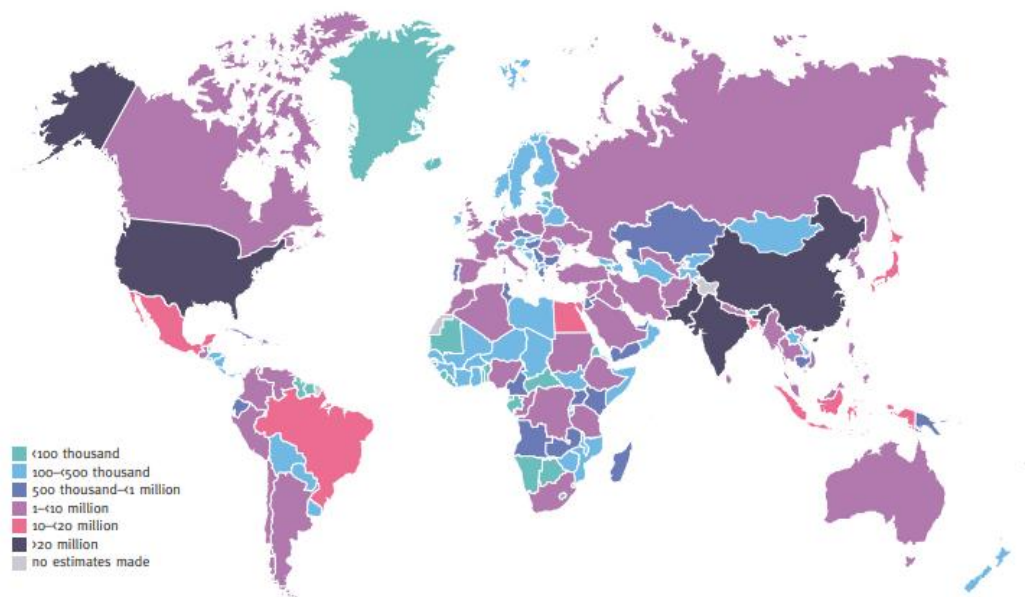


Figura 1. Número total, estimado, de adultos (20-79 anos) com diabetes em 2021 [17].

No ano de 2018, segundo o Relatório Anual do Observatório Nacional da Diabetes, estimou-se que a prevalência da Diabetes na população portuguesa com idades compreendidas entre os 20 e os 79 anos foi de 13,6%, o que corresponde a mais de 1,2 milhões de casos [4].

Dentro dos vários tipos de diabetes, destaca-se a DM2, uma vez que é a mais comum, correspondendo a mais de 90% dos casos. Esta afeta sobretudo pessoas adultas e idosas, com excesso de peso ou obesidade, sedentárias e com estilos de vida pouco saudáveis, e há frequentemente historial familiar [13].

A DM2 é uma doença metabólica caracterizada por alterações na secreção, e por resistência à ação da insulina [4]. No caso das alterações na secreção de insulina, o pâncreas torna-se incapaz de fabricar insulina em quantidade suficiente para controlar os níveis de açúcar, que vão aumentando de forma progressiva levando assim à hiperglicemia [4], [18], [19]. Por outro lado, a resistência à ação da insulina resulta de uma desregulação do metabolismo de carboidratos, lípidos e proteínas e resulta na secreção de insulina prejudicada, resistência à insulina ou uma combinação de ambos [20].

Ao contrário da diabetes mellitus tipo 1, as pessoas não nascem com DM2, desenvolvendo a doença devido a vários anos de hábitos de vida pouco saudáveis, especialmente o consumo excessivo de carboidratos na alimentação e estilo de vida sedentário. No entanto, a tendência genética também pode determinar de uma forma

parcial a suscetibilidade a esta doença. Assim, os vários fatores que capazes de promover o desenvolvimento da DM2 são [4], [13]:

- Obesidade, alimentação inadequada e inatividade física;
- Envelhecimento;
- Resistência à insulina;
- História familiar de diabetes;
- Pressão alta;
- Tolerância à glicose prejudica (glicose sanguínea acima do normal, mas abaixo do limite para se diagnosticar como diabetes);
- Histórico de diabetes gestacional, e má nutrição durante a gravidez;
- Etnia.

O diagnóstico desta doença ocorre geralmente após os 40 anos de idade, mas pode ocorrer mais cedo. A DM2 pode se assintomática, levando a que o diagnóstico muitas vezes seja efetuado devido à manifestação de complicações associadas ou, acidentalmente, através de resultados anormais de glicose em exames ao sangue ou à urina [4], [12].

No passado esta era considerada uma doença de adultos que se manifestava após uma vida sedentária e de anos de obesidade. No entanto, nos últimos anos, a DM2 tem sido cada vez mais observada em grupos etários cada vez mais jovens. O aumento da DM2 nestas faixas etárias presume-se que esteja associado ao aumento da obesidade infantil em muitos países, que resulta de maus hábitos alimentares e estilos de vida sedentários, podendo tornar-se um problema de saúde pública global [17], [21].

No entanto, algumas regiões que têm uma baixa prevalência de obesidade infantil, tais como leste asiático, revelaram taxas de incidência mais elevadas de DM2 juvenil do que as populações com uma maior carga de obesidade infantil. Para tal contribuíram fatores como predisposição genética, estatuto socioeconómico, acesso aos cuidados de saúde e práticas culturais [17].

Segundo a décima edição do IDF Diabetes Atlas, as taxas de incidência mais elevadas de DM2 em jovens foram verificadas no Canadá, nos Estados Unidos da América e na Austrália, tal como é possível verificar pela Figura 2 [17].

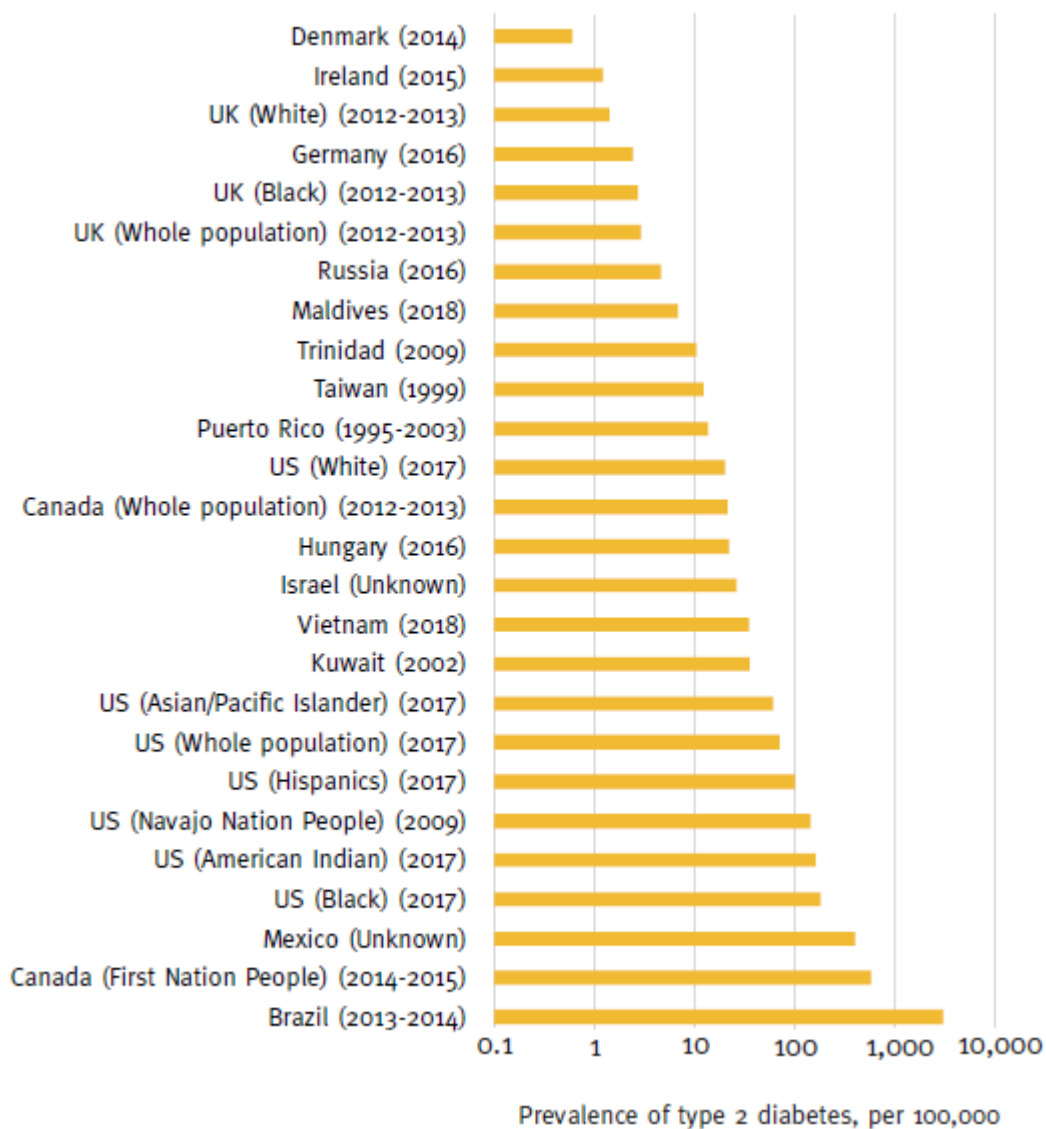


Figura 2. Prevalência relatada da DM2 em jovens por região e etnia [17].

Por outro lado, as estimativas de prevalência, presente na Figura 3, demonstra que a prevalência da DM2 em jovens foi mais acentuada no Brasil e México, bem como nos Estados Unidos da América e na Austrália, novamente [17].

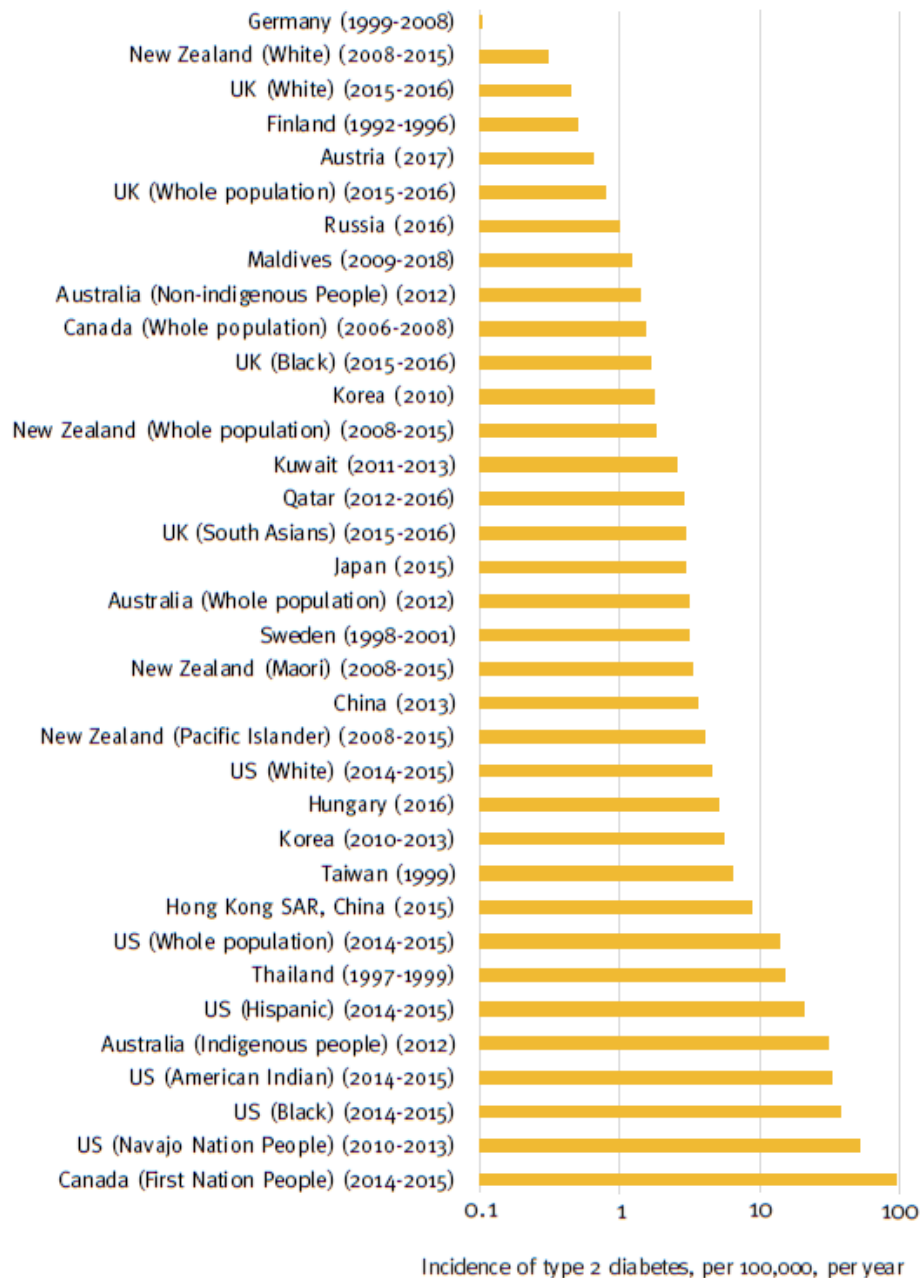


Figura 3. Incidência reportada de DM2 em jovens por região e etnia [17].

Enquanto os pacientes com diabetes tipo 1 são dependentes de insulina exógena, o mesmo não acontece com pacientes com DM2, visto que estes podem não necessitar de administrar insulina exógena para combater esta doença. Para muitos pacientes, uma adaptação na alimentação, tanto ao nível do que se come como quando come, bem como a prática de atividade física regular (permite que o organismo aproveite melhor o açúcar que tem em circulação), podem ser suficientes para manter a Diabetes controlada, pelo menos durante um certo período [4].

Isto torna-se ainda mais importante dado que a obesidade e a diabetes estão fortemente ligados, uma vez que a obesidade está associada a uma grande porção de tecido adiposo que por sua vez influencia a sensibilidade celular à insulina, resultando na insulinoresistência e, conduzindo à DM2 [18], [20].

Assim, a Diabetes é uma doença crónica que exige disciplina e moderação, sendo que apresentar uma alimentação adequada é um dos pilares da terapêutica desta doença. Deste modo, os diabéticos deveram procurar que a sua alimentação respeite os seguintes pontos:

- Comer uma variedade de alimentos saudáveis;
- Limitar o consumo de gordura e de sal na dieta diária;
- Ter em atenção a dimensão das porções e manter a coerência em todas as refeições;
- Instituir uma rotina para os horários das refeições;
- Seguir à regra as indicações nutricionais estabelecidas pelo médico.

No entanto, a alimentação e o exercício físico por si só podem não ser suficientes, sendo necessário fazer o tratamento com recurso a fármacos e, em certos casos à utilização de insulina a fim de evitar o aparecimento de complicações agudas e crónicas [4].

Pacientes com DM2 podem não apresentar sintomas ou ter sintomas leves daí que por vezes os pacientes possam viver vários anos sem que esta doença seja diagnosticada. Os sintomas clássicos da DM2 são semelhantes aos da diabetes tipo 1, a saber [4], [18]:

- Sede anormal e secura de boca;
- Micção frequente;
- Cansaço/falta de energia;
- Fome constante;
- Perda de peso súbita;
- Feridas de cura lenta;
- Infecções recorrentes;
- Visão turva.

Para além destes sintomas, a DM2 pode ainda afetar vários órgãos, podendo levar ao aparecimento de outras complicações, tais como [22], [23]:

- Doenças cardiovasculares e dos vasos sanguíneos;
- Lesões renais;
- Lesões neurológicas e perda de sensibilidade e dor;
- Doenças oculares e cegueira;
- Problemas de audição;
- Doenças digestivas;
- Problemas sexuais e problemas na gestação;
- Síndrome do pé diabético, que pode obrigar à amputação.

Assim, recomenda-se que estes pacientes tenham uma dieta individualizada, que deve ser ajustável ao paciente e adaptada às suas condições de vida. Deste modo, a dieta apresenta-se como um método que deve ser utilizado de forma a prevenir, retardar ou tratar complicações que possam advir da Diabetes [24].

Por todas estas razões, torna-se necessário promover medidas para consciencializar os pacientes bem como medidas que permitam os mesmos controlar e monitorizar os seus comportamentos de modo a combater esta doença, sendo que o avanço das tecnologias veio fazer isso mesmo. A elevada presença da tecnologia do dia a dia de todas as pessoas, especialmente devido ao uso de *smartphones*, abriu portas a novos métodos que possam ajudar pacientes diabéticos a controlar a sua dieta, e seria um desperdício não aproveitar estes recursos.

Assim, têm vindo a ser desenvolvidas algumas plataformas ou aplicações para *smartphones*, de tal forma que, atualmente, já existam soluções eficientes de modo a facilitarem a avaliação dietética [25]. Estes métodos surgiram então como ferramentas promissoras face aos métodos tradicionais, especialmente tendo em conta que estas oferecem algumas vantagens nomeadamente ao nível da portabilidade, aceitabilidade, comodidade, diminuição da carga de trabalho, minimização dos erros e tempo despendido [6].

Para além disso, alguns estudos revelaram que as aplicações dietéticas para *smartphones* mostram maior retenção de utilizadores, quando comparado com os métodos tradicionais [25].

2.3. Métodos de Reconhecimento Alimentar

Normalmente, os programas de avaliação dietética requerem o registo diário dos alimentos consumidos, bem como o seu conteúdo e as porções ingeridas. Tradicionalmente, este era um processo inteiramente manual, o que se torna um processo trabalhoso ou até complicado, especialmente para pessoas idosas [26]. Por estas razões, começaram a ser desenvolvidos sistemas capazes de realizar este processo de uma forma automática. Assim, surgiram os sistemas de reconhecimento alimentar [25].

Estes sistemas são capazes de desempenhar tarefas como a deteção e classificação dos alimentos ingeridos, sendo que alguns sistemas também são capazes de realizar estimativas do volume dos alimentos bem como das calorias, e valores nutricionais dos mesmos. Os sistemas de recomendação alimentar tanto podem ser automáticos como podem necessitar que o utilizador forneça alguns inputs ao sistema [27].

Com o desenvolvimento das tecnologias, a determinação rápida, exata e automática dos alimentos e dos seus atributos torna-se quase uma exigência por parte dos utilizadores deste tipo de sistemas. Deste modo, tem vindo a ser desenvolvidas narizes eletrónicos, sensores acústicos, sistemas de visão por computador, entre outros, capazes de extrair determinadas características dos alimentos a fim de os identificar [10], [28], [29].

Contudo, ainda é muito complicado fazer a determinação de um alimento de forma completamente automática dado o enorme volume de características presentes nos mesmo, para além destes incluírem muito material repetitivo e irrelevante. Desta forma, torna-se muito complexo implementar um sistema deste tipo no mundo real. Ainda assim, várias tem sido as tentativas de desenvolver estes sistemas [25].

Um sistema de classificação alimentar baseado em imagens, por norma, é composto por três etapas, nomeadamente a segmentação, a extração de características, e por fim a classificação. Caso o sistema se destine não só a identificar o alimento, mas também a providenciar informação nutricional do mesmo, poderão existir mais duas etapas: estimar o volume e peso dos alimentos e, por fim, a obtenção da informação nutricional dos mesmos [25]. A Figura 4 ilustra, sequencialmente, as várias etapas realizadas neste tipo de sistema.



Figura 4. Etapas de um sistema de classificação alimentar baseado em imagens, com informação nutricional.

O primeiro passo é a segmentação, sendo esta essencial para identificar as regiões da imagem onde estão presentes alimentos, de modo a extrair essas regiões e excluir as restantes. Esta etapa torna-se muito mais simplificada quando esta é semiautomática, ou seja, quando o utilizador indica quais as zonas da imagem que contem alimentos, melhorando assim a exatidão do sistema, especialmente quando é necessário identificar múltiplos alimentos numa única imagem ou quando é necessário extrair o volume dos alimentos [25], [30].

Por outro lado, a segmentação automática é ainda uma tarefa complicada dado que as imagens podem não apresentar os seus contornos, ou caso existam alimentos que estejam cobertos, escondidos ou parcialmente escondidos por outros alimentos [31]. A segmentação torna-se ainda mais complicada quando os alimentos estão picados ou misturados [32].

Após a segmentação vem o processo de extração de características, onde são extraídas algumas características das regiões identificadas na etapa anterior. Por norma, a classificação é mais eficiente e estável quando relativamente poucas características estão envolvidas [29]. Assim, quando se desenvolve este tipo de sistemas deve-se ter em consideração quais são as características mais relevantes para se identificarem alimentos, sendo que, normalmente, as mais utilizadas são a cor, tamanho, forma e textura [33]–[36].

Estas características são então utilizadas para a classificação, sendo que para tal recorrem-se a modelos de previsão, previamente treinados, baseados em algoritmos de *machine learning* tais como *Support Vector Machines*, *K-Nearest Neighbors*, *Bag of Features*, *Multiple Kernel Learning*, *Random Forests*, *Partial Least Square* e redes neuronais artificiais entre outras, sendo que dentro destas destaca-se as redes neuronais convulsionais [25], [29].

Estes métodos têm se mostrado muito valiosos em várias áreas, sendo que o reconhecimento alimentar não é exceção. Estes têm vindo a ser usados com o intuito de

realizar o reconhecimento de alimentos, uma vez que estes algoritmos se mostraram robustos mesmo trabalhando com grandes volumes de dados, sendo capazes de apresentar resultados exatos num curto espaço de tempo. Sendo estas algumas das características necessárias para aplicar estes sistemas no mundo real, dada existência de uma grande quantidade de material repetitivo e irrelevante [29].

Uma vez identificados os alimentos numa determinada imagem, o passo seguinte passa por determinar o seu volume ou peso, de modo que seja possível estimar a informação nutricional correspondente.

Contudo, determinar o volume dos alimentos pode tornar-se muito complicado, quando se tem apenas uma única imagem bidimensional como fonte de informação. Isto porque estas imagens normalmente não contêm informações sobre a escala e profundidade dos objetos alimentares. Por norma, para se obterem estes parâmetros recorre-se a componentes de hardware especiais, tais como sensores de profundidade, ou usando câmaras de visão estéreo. A profundidade também pode ser estimada utilizando múltiplas imagens de diferentes vistas desde que existam algumas informações adicionais, como medidas e posições de determinados elementos presentes na imagem [25], [37].

Deste modo, parâmetros como a escala e a posição dos objetos são componentes importantes para que se possa contruir um modelo 3D de um objeto. O uso de um objeto de referência também pode ser usado para relacionar as suas dimensões, que são conhecidas, com os restantes elementos da imagem [38], [39].

Atualmente, já existem algumas ofertas deste tipo de sistema no mercado, tais como o *Dishtracker*, *Bite AI*, *LogMeal API*, *Calorie Mama* e o *Foodvisor*.

O *Dishtracker* é um sistema desenvolvido com o intuito de facilitar e acelerar o processo de *checkout* de refeições. Este é um sistema que recorre ao uso de uma câmara para reconhecer objetos, nomeadamente comida, e regista-os automaticamente no sistema para se proceder ao *checkout*, sendo que para tal este usa técnicas de inteligência artificial. O *Dishtracker* é capaz de reconhecer todos os artigos que foram treinados e guardados no sistema, sendo que para cada prato basta adicionar apenas uma imagem [40].

Um processo típico de *checkout* pode demorar até 30 segundos, no entanto, usando o *Dishtracker* este processo pode ser acelerado em até 27 segundos. Isto acontece dado à capacidade de o sistema reconhecer e registar a refeição em 0,3 segundos, levando

a que o tempo total de checkout seja de cerca de 3 segundos. Para além de acelerar o *checkout*, o sistema permite que os clientes realizem este processo de forma independente, é capaz de trabalhar com todos os tipos de caixas registadoras convencionais, e permite que, normalmente, 100% dos pratos comuns sejam reconhecidos sem qualquer interação exigida pelo pessoal de serviço [40].

Esta tecnologia está disponível desde 2017, sendo que atualmente já é usada em mais de 12 000 caixas diariamente, levando a que 14 000 checkouts sejam realizados todos os dias pelo *Dishtracker*. Foi criado também uma aplicação móvel, *ZeroQ*, com este sistema que permite que os utilizadores nem tenham de se dirigir as caixas para proceder ao *checkout* [40].

A *Bite AI* é uma app de reconhecimento e registo alimentar, que usa os mais recentes modelos de *deep learning* para reconhecer milhares de alimentos a partir de texto, imagens ou códigos de barras. No que toca ao reconhecimento via imagem, estes modelos são treinados por milhões de imagens, nomeadamente da *Wikidata* e da *USDA Nutrition Database*. Para além disso, o sistema está sempre a melhorar com cada imagem inserida no sistema, isto porque os modelos são automaticamente treinados para reconhecer refeições previamente [41], [42].

Esta app é capaz de detetar, com uma elevada exatidão, pratos, alimentos integrais e ingredientes para refinar os resultados e criar combinações ilimitadas. Na *Bite AI* a maioria das refeições inclui detalhes nutricionais completos, ingredientes, tamanhos de porções e método de preparação para que possa calcular o conteúdo calórico, apesar de não determinarem o volume/peso da refeição [41], [42].

A *LogMeal API* é uma aplicação usada para reconhecer e rastrear alimentos, sendo que esta utiliza o algoritmo de Inteligência Artificial mais desenvolvido para alimentos no mercado, baseado em visão informática e *deep learning*. Uma das suas principais vantagens é que este sistema está constantemente a aprender com os *inputs* realizados pelos seus utilizadores. Outros fatores positivos passam pelo rápido reconhecimento alimentar, inferior a 1 segundo, elevada exatidão (99%), permite manter um registo do consumo alimentar com estatísticas em tempo real e ainda permite obter informação nutricional dos mesmos [43], [44].

Esta é, atualmente, uma das principais aplicações de reconhecimento de alimentos, pratos e menus a nível mundial, identificando-os, classificando-os,

categorizando-os, e ainda apresentando a sua informação nutricional. Assim, a aplicação é usada por vários tipos de utilizadores, tais como pacientes diabéticos, atletas, pessoas com problemas nutricionais, ou simplesmente pessoas que têm cuidado com a sua saúde e alimentação [43], [44].

Para além de tudo isto, esta aplicação pode ainda ser usada para realizar checkout de forma automática e autónoma em restaurantes que possuam o *LogMeal Kiosk* [43], [44].

Uma outra opção disponível é a *Calorie Mama*. Esta é uma aplicação de reconhecimento de alimentos num período de tempo inferior a um segundo, sendo que para tal basta tirar uma foto aos alimentos para, através de técnica de *deep learning*, detetar os alimentos e apresentar a sua informação nutricional. A *Calorie Mama* é ainda capaz de construir perfis nutricionais detalhados dos utilizadores e recomendar dietas personalizadas [45].

Esta aplicação foi treinada com imagens de comida de todo o mundo, sendo o sistema de identificação alimentar mais diversificado do mercado, de tal forma que esta é capaz de identificar milhares de categorias de alimentos diferentes. Além disso, esta é ainda capaz de diferenciar entre diferentes estilos de apresentação, métodos de preparação, e variações regionais. Uma outra vantagem desta aplicação é que a sua exatidão está constantemente a melhorar, com as imagens usadas pelos utilizadores a serem adicionadas à base de dados [45].

Por outro lado, existe a *Foodvisor*, uma aplicação com a finalidade de ser um guia de saúde e que fornece nutrição personalizada. Esta aplicação fornece o aconselhamento personalizado de dietistas para os utilizadores atingirem os seus objetivos, nomeadamente ao nível de hábitos alimentares saudáveis, fornecendo sugestões de receitas equilibradas. Para tal, o utilizador deve responder a algumas perguntas iniciais de modo a indicar os seus objetivos e depois este irá receber conselhos e cursos diários para ajudá-lo a alcançar seu objetivo pessoal [46].

A *Foodvisor* também é capaz de reconhecer alimentos, estimar a sua quantidade e fornecer informação nutricional acerca dos mesmos em poucos segundos [46].

Na Tabela 1, podemos encontrar um resumo das aplicações/sistemas referidos anteriormente.

Tabela 1. Exemplos de algumas das aplicações de reconhecimento alimentar já desenvolvidas.

Aplicação	Características
<i>Dishtracker / ZeroQ</i>	- Facilita e acelera o processo de <i>checkout</i> de refeições.
<i>Bite AI</i>	- Reconhece alimentos a partir de texto, imagens ou códigos de barras; - Fornece informações nutricionais para grande parte dos alimentos; - Constante aprendizagem; - Permite escolher diferentes porções; - Considera o método de preparação dos alimentos.
<i>LogMeal API</i>	- Permite manter um registo do consumo alimentar; - Fornece informações nutricionais; - Constante aprendizagem; - Determina o volume/peso dos alimentos; - Facilita e acelera o processo de <i>checkout</i> de refeições; - Recursos limitados consoante os diferentes planos;
<i>Calorie Mama</i>	- Capaz de construir perfis nutricionais e recomendar dietas personalizadas; - Base de dados mais diversificada; - Fornece informações nutricionais; - Constante aprendizagem; - Determina o volume/peso dos alimentos; - Tem em consideração o método de preparação e variações regionais.
<i>Foodvisor</i>	- Fornece informações nutricionais; - Constante aprendizagem; - Determina o volume/peso dos alimentos.

2.4. Machine Learning

Machine learning é um dos principais ramos da informática. Esta trata-se de um tipo de inteligência artificial que tem o propósito de realizar a deteção automática de padrões significativos nos dados. Deste modo, o objetivo destas tecnologias passa por

proporcionar aos algoritmos a capacidade de aprenderem e de se adaptarem aos dados fornecidos, de modo a conseguir prever a que conjunto esses dados pertencem.

Os algoritmos de *machine learning* podem ser divididos tendo por base o tipo de aprendizagem, sendo que os mais comuns são os seguintes:

- Aprendizagem supervisionada – onde é apresentada à rede a correspondência entre uma entrada e uma saída;
- Aprendizagem não supervisionada - a rede é treinada de modo a responder a agrupamentos de padrões de entrada, esta é usada quando os dados não estão identificados;
- Aprendizagem semi-supervisionada - combina exemplos rotulados e não rotulados;
- Aprendizagem por reforço - baseada em tentativa erro para encontrar uma solução para o problema, onde cada ação tem algum impacto no ambiente, que fornece feedback para orientar o algoritmo;
- Transdução - semelhante à aprendizagem supervisionada, mas para além das classes apresentadas na fase de treino esta tenta prever novas saídas;
- Aprender a aprender - onde o algoritmo aprende seu próprio viés indutivo com base na experiência anterior.

Para além desta divisão, os problemas de *machine learning* podem ainda ser divididos em problemas de classificação e problemas de regressão. Enquanto os problemas de classificação procuram uma função através da qual os dados são previstos em rótulos de classes discretas, a regressão procura uma função para distinguir os dados em valores reais contínuos em vez de usar classes [47], [48].

Assim, em sistemas de classificação as amostras são divididas em duas ou mais classes para estes aprenderem, de modo a estes serem capazes prever a classe correspondente para cada entrada.

Random Forest, *Support Vector Machine*, *Naive Bayes* e as Redes Neurais Artificiais são alguns dos principais algoritmos de classificação.

2.4.1. Random Forest

Random Forests (RF) trata-se de um método de *machine learning* que tanto pode ser utilizado em problemas de resposta categórica, classificação, ou de resposta contínua,

e regressão. As RF's são uma extensão das árvores de decisão, na medida em que consistem na agregação de várias árvores de decisão formando assim uma “floresta”, onde o *output* é um resultado único.

Estes algoritmos são um dos algoritmos de classificação supervisionados mais conhecidos e eficazes, tendo já sido utilizado com sucesso em várias aplicações. Algumas das características mais favoráveis das RF's são:

- poderem ser usadas, de forma eficaz, tanto para regressão como para classificação, ou seja, tanto podem ser a resposta para sistemas binários como para sistemas com várias classes;
- são relativamente rápidos a treinar e a prever;
- dependem apenas de um ou dois parâmetros de afinação;
- têm uma estimativa construída de erro de generalização;
- podem ser utilizados diretamente para problemas de alta dimensão;
- não é tendenciosa, porque cada árvore é treinada numa porção de dados, desta forma o enviesamento geral do algoritmo é minimizado;
- mesmo que novos dados sejam adicionados ao conjunto de dados, o processo global não é afetado, porque embora novos dados possam alterar uma árvore, é extremamente improvável que afete todas as árvores.
- apresenta bom desempenho mesmo quando os dados têm valores em falta ou não foram escalados corretamente.

Ainda assim, estes algoritmos podem tornar-se altamente complexos em casos onde o número de árvores de decisão combinadas é elevado o que leva a que sejam necessários elevados recursos computacionais para que se possa processar este tipo de algoritmo [49], [50].

2.4.2. Support Vector Machine

Support Vector Machine (SVM), tal como as RF, é um algoritmo de *machine learning* supervisionado que pode ser usado quer para problemas de classificação quer de regressão. Estes algoritmos são baseados na busca de uma linha de separação entre duas classes distintas analisando os dois pontos, um de cada grupo, mais próximos da outra classe. Quando este tipo de algoritmo é usado para problemas multiclasse este é tratado como vários problemas binários.

Desta forma, o processo de formação de uma função de decisão SVM consiste em identificar um hiperplano que maximiza a distância (ou "margem") entre duas classes. Após estas estarem estabelecidas, é possível fazer-se a classificação dos dados tendo em conta a sua posição em relação a estes planos, ou retas quando se trata de um problema onde apenas duas características são consideradas [51], [52].

Assim, um hiperplano ótimo serve para separar duas classes, sendo que estes podem ser definidos por funções lineares ou não lineares (funções de base radial, sigmóides e funções polinomiais).

Em comparação com outros tipos de classificadores, o poder da SVM deriva em grande parte da sua capacidade de alcançar uma precisão elevada e equilibrada, que são generalizáveis mesmo em casos com elevada dimensionalidade [51].

2.4.3. Naive Bayes

Naive Bayes (NB), é um algoritmo de aprendizagem, que consiste na forma mais simples da rede Bayesiana, na qual todos os atributos são independentes dado o valor da variável de classe, ao que se chama independência condicional. É óbvio que o pressuposto da independência condicional raramente é verdadeiro na maioria das aplicações do mundo real.

Ainda assim, este método proporciona frequentemente uma classificação exata e competitiva, sendo um dos métodos mais eficientes e eficazes em *machine learning*. Assim, algumas das principais características do NB são a sua eficiência computacional, baixa variância, aprendizagem incremental, robustez face ao ruído e ainda robustez face a valores em falta [53]–[55].

2.4.4. Redes Neurais Artificiais

Uma rede neuronal artificial (RNA) consiste num conjunto de técnicas computacionais, inspiradas na estrutura neuronal do ser humano, que tentam simular e modelar os processos cognitivos da mente humana. Tal como o cérebro humano, as RNA adquirem conhecimento através da experiência, ou seja, com recurso a exemplos e algoritmos de aprendizagem. Desta forma, depois de treinadas para uma certa tarefa, a RNA consegue relacionar dados nas suas entradas, com a informação desejada à saída.

As RNA's são compostas por várias unidades de processamento, neurónios, interligados, podendo formar várias camadas, de modo a formarem uma "rede". A estas

ligações estão associados valores numéricos aos quais se dá a designação de pesos. Estes são alterados conforme o problema proposto, de modo a obter a saída desejada. Para a alteração destes pesos, são utilizados algoritmos de treino, onde por semelhança ao sistema nervoso humano, um neurónio é ativado quando recebe estímulo suficiente. Um neurónio combina os dados de entrada com uma série de coeficientes, ou pesos, que ampliam ou atenuam essa entrada em função da sua significância face à tarefa para a qual o algoritmo está a ser treinado.

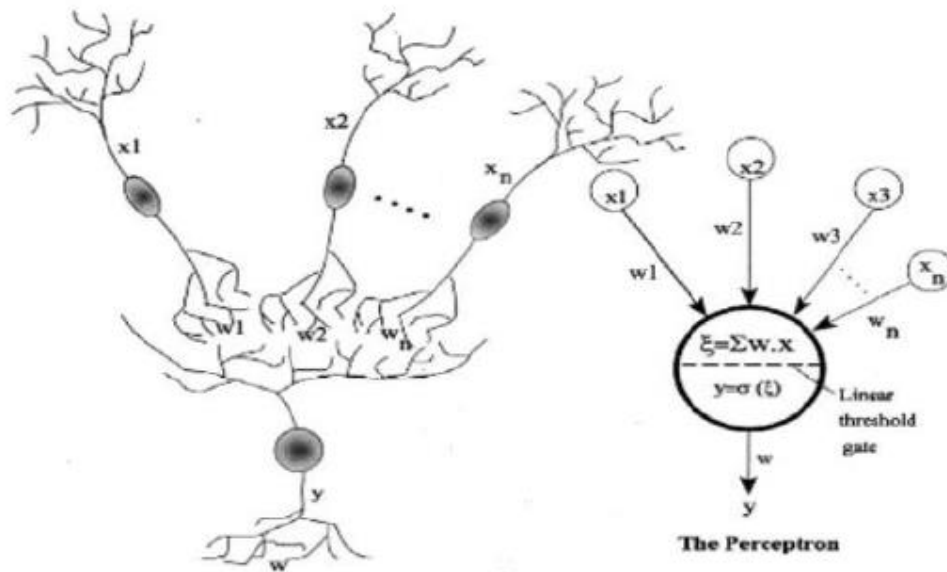


Figura 5. Esquema representativo do funcionamento de uma rede neuronal artificial inspirado na comunicação que ocorre entre os neurónios dos sistemas biológicos [56].

Estas assemelham-se ao cérebro humano em dois pontos principais:

- O conhecimento é adquirido pela rede a partir do ambiente envolvente através de um processo de aprendizagem (algoritmo de treino).
- Os nós da rede (neurónios) estão ligados através de interligações com pesos (pesos sinápticos), usados para armazenar o conhecimento adquirido [57]–[59].

Dentro das RNA existe um caso com uma arquitetura especial, sendo estas as redes neuronais convolucionais (RNC). Estas foram inspiradas pelo mecanismo visual que ocorre ao nível do cérebro, no córtex visual. O córtex visual contém muitas células que são responsáveis pela deteção da luz em pequenas sub-regiões sobrepostas do campo visual, as quais são chamadas campos recetivos. Estas células atuam como filtros locais sobre o espaço de entrada, e as células mais complexas têm campos recetivos maiores.

Uma RNC consiste numa rede com uma ou mais camadas convolucionais, sendo estas as responsáveis pela função desempenhada pelas células no córtex visual. Desta forma, estas são principalmente utilizadas em reconhecimento de objetos, identificação e agrupamento de imagens [56], [60].

A arquitetura deste tipo de redes é adaptada para reduzir número de parâmetros, permitindo lidar com maiores bases de dados, e manter a dependência espacial e temporal da entrada. Assim, uma RNC tem como objetivo estabelecer automaticamente uma hierarquia de características relevantes, extrair a sua representação e integrá-la no problema de classificação [61].

CAPÍTULO 3 – DESENVOLVIMENTO

3. Desenvolvimento

Este capítulo tem o propósito de apresentar as etapas, os materiais e os métodos levados a cabo no desenvolvimento do presente projeto. Este é um projeto que foi inteiramente desenvolvido com recurso à linguagem de programação *Python* na versão 5.1.5. Deste modo, foram usadas algumas das bibliotecas disponíveis nesta linguagem, tais como a *tensorflow*, muito usada para a criação de redes neuronais, a *numpy*, usada para trabalhar com matrizes e funções matemáticas, a *cv2*, que é uma das principais bibliotecas para trabalhar com imagens, e ainda *sklearn* também ela uma linguagem muito usada em projetos envolvendo *machine learning*.

Assim, tendo isto em conta este capítulo aborda a base de dados, os métodos usados, e a arquitetura para se chegar ao sistema proposto.

3.1. Base de Dados

Para se desenvolver um sistema capaz de realizar deteção automática de comida através de algoritmos de *machine learning* é necessário recorrer-se a bases de dados com imagens de comida, sendo que o presente sistema não é exceção.

Deste modo, a base de dados disponibilizada pelo *Eidgenössische Technische Hochschule Zürich* (ETHZ) foi a escolhida para se implementar o sistema. Intitulada de ETHZ Food-101, esta trata-se de uma base de dados multimodal, e encontra-se dividida em 101 categorias, onde cada categoria representa um prato diferente. Cada uma destas categorias é constituída por 1.000 imagens do mundo real, ou seja, no total esta base de dados é constituída por 101.000 imagens.

As imagens que constituem a base de dados foram adquiridas no site *foodspotting.com*, site este que permite aos utilizadores tirar fotos do que eles estão a comer, anotar local e tipo de alimentos e carregar estas informações. Para além das imagens, também as 101 categorias foram escolhidas de acordo com as categorias mais populares do site.

Aquando da recolha das imagens, os responsáveis pela criação desta base de dados selecionaram aleatoriamente 750 imagens para treino para cada classe, sendo que estas imagens contêm ruído, associado a cores intensas e até imagens que não corresponde à categoria na qual estão inseridas. Além disso, foram recolhidas e limpas manualmente 250 imagens de teste para cada classe, ou seja, não contêm ruído. Além disso, todas as

imagens foram redimensionadas para terem uma largura máxima de 512 pixels e as imagens demasiado pequenas foram excluídas [56]. O conjunto de dados está disponível para download¹.

No entanto, de modo a testar o sistema diversas vezes com diferentes dados de treino e teste, foi realizada uma filtragem manual, onde as imagens que não correspondiam à classe na qual estavam inseridas acabaram por ser removidas pois não fazia sentido que estas fossem usadas como dados de teste, algo que poderia acontecer.

Para além disso, devido a limitações de *hardware* tornou-se inviável usar muitas classes para o desenvolvimento do sistema. Assim, foram apenas selecionadas 5 das 101 classes para a sua execução. A Tabela 2 ilustra as classes utilizadas bem como o seu identificador associado e o número de imagens que contem cada classe.

Tabela 2. Número de imagens em cada uma das classes utilizadas após a filtragem manual.

Identificador	Classe	Número de imagens
0	'chicken_wings'	954
1	'edamame'	975
2	'pizza'	933
3	'macarons'	924
4	'poutine'	948

3.2. API Ninjas

De modo que fosse possível estimar os valores nutricionais das refeições identificadas recorreu-se a aplicações (APIs) da *API Ninjas*. A *API Ninjas* é uma empresa privada, que atua no setor de desenvolvimento de *software*, que oferece dezenas de APIs de dados reais para os programadores possam construir projetos reais [56]. Para se ter acesso a estas APIs é necessário registrar-se no site da *API Ninjas*².

No presente trabalho recorreu-se a uma das APIs da *API Ninjas*, sendo estas a a *Nutrition API*.

¹ <http://www.vision.ee.ethz.ch/datasets/food-101/>

² <https://api-ninjas.com/>

A *Nutrition API* é uma aplicação capaz de retomar a informação nutricional de um dado alimento, a partir de texto utilizando processamento em linguagem natural.

Uma característica desta API é que permite a obtenção dos dados nutricionais referentes a uma porção personalizada. Assim, esta retorna a informação nutricional do alimento indicado para a porção definida pelo utilizador. Caso não seja inserida a porção desejada a API retorna os dados nutricionais do alimento para uma porção pré-definida de 100g [62].

3.3. Métodos de Avaliação

Para se avaliarem as previsões feitas pelo sistema recorreu-se a três métricas, *accuracy*, *precision* e *recall*, e ainda à função *loss*.

A *accuracy*, em português exatidão, é uma métrica usada para avaliar modelos de classificação, como é o caso neste exemplo. Esta métrica é expressa sob a forma de uma percentagem e permite avaliar a capacidade do sistema em realizar previsões corretas, ou seja, permite avaliar o quão perto ou distante de um determinado conjunto de previsões estão ao seu valor verdadeiro. A *accuracy* pode ser calculada fazendo-se a razão entre o número de previsões corretas e o número total de previsões, tal como expresso na equação 1.

$$accuracy = \frac{\text{número de previsões corretas}}{\text{número total de previsões}} \quad (1)$$

A *precision*, em português precisão, é uma métrica que indica a qualidade das previsões positivas feitas pelo modelo, em modelos de classificação binários. No entanto, esta não está limitada ao uso de modelos binários, podendo também ser usada em problemas multimodais. Assim, esta pode ser calculada, em modelos multiclasse, através da razão entre a soma do número de verdadeiros positivos (VP) de todas as classes, pela soma do número total de previsões positivas (número de verdadeiros positivos (VP) mais o número de falsos positivos (FP)) de todas as classes, a fórmula de cálculo desta métrica encontra-se na equação 2.

$$precision = \frac{\sum VP}{\sum (VP + FP)} \quad (2)$$

O *recall*, também designado por sensibilidade, por sua vez representa o número de previsões positivas corretas feitas a partir de todas as previsões positivas que poderiam

ter sido feitas, ou por outras palavras, mede a capacidade do modelo para encontrar todas as unidades positivas no conjunto de dados. Esta pode ser calculada, tal como mostra a equação 3, a partir da razão entre a soma dos verdadeiros positivos (VP) em todas as classes pela soma dos verdadeiros positivos (VP) e dos falsos negativos (FN) em todas as classes [63], [64].

$$recall = \frac{\sum VP}{\sum (VP + FN)} \quad (3)$$

Por fim, a *loss*, ou função de perda, é uma função que compara os valores alvo e os valores de saída, sendo usada para medir o quão bem a rede neuronal modela os dados. Na fase de treino, o objetivo passa por tentar minimizar esta perda entre os valores reais e os valores de saída previstos [65].

Existem diversos tipos de *loss*, sendo que a função usada neste caso foi a *Categorical Cross-entropy*. Esta função é usada quando se está na presença de problemas com duas ou mais classes e é uma medida muito boa de quão distinguíveis duas distribuições de probabilidade discretas são uma da outra. Para tal esta usa uma função do tipo *softmax* para resolver os problemas de classificação multiclasse, tal como se pode ver na equação 4 [66], [67].

$$Categorical\ Cross - Entropy\ Loss = -\frac{1}{n} \times \sum_{i=1}^N \sum_{j=1}^M y_{ij} \times \log(p_{ij}) \quad (4)$$

3.4. Abordagem Proposta

A abordagem proposta no presente trabalho está assente em diversas fases, tal como se pode observar na Figura 6, sendo que estas estão divididas em dois blocos, o primeiro consiste no desenvolvimento do modelo (Modelo de classificação) e o segundo na aplicação do mesmo (Aplicação do Modelo). O bloco de desenvolvimento do modelo é constituído pela fase de carregamento dos dados, a fase de pré-processamento e a fase de configuração da rede neuronal. Por outro lado, o bloco da aplicação do modelo consiste na previsão da classe correspondente à imagem de input, e apresentar a informação nutricional da mesma, sendo que para tal recorreu-se à *Nutricional API*.

Em oposição às etapas descritas na Figura 4, a abordagem proposta não contempla as etapas de segmentação nem de estimativa do volume ou do peso dos alimentos.

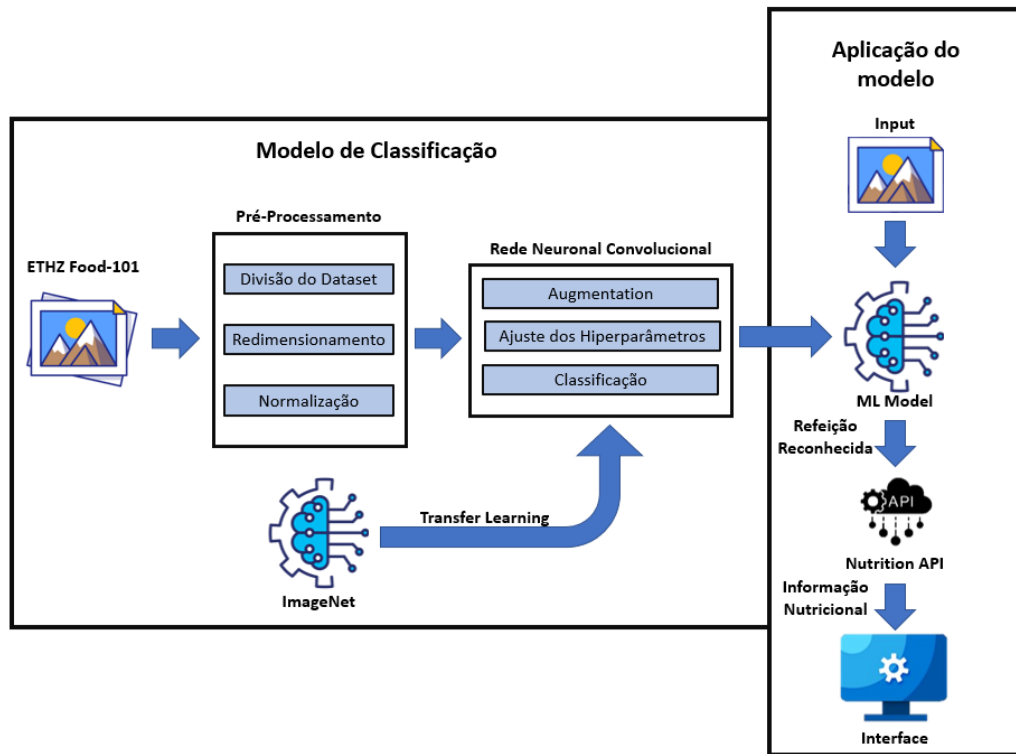


Figura 6. Diagrama representativo do modelo de *machine learning* desenvolvido.

Assim, o desenvolvimento do modelo inicia-se com a importação dos dados para o sistema. Esta fase está parcialmente misturada com a fase de pré-processamento, isto porque a divisão do *dataset* ocorre em simultâneo com a importação dos dados. Estes dois processos são realizados com o objetivo de se carregar e dividir as imagens, de cada uma das classes usadas, em imagens de treino, validação e teste.

Assim, em primeiro lugar fornece-se ao sistema o *path* da pasta onde estão armazenadas as imagens de cada uma das classes usadas à função *glob()*, da biblioteca *glob*, que retorna um *array* com o *path* para cada uma das imagens que é armazenado no *array image_list*.

De seguida, recorreu-se à função *train_test_split()*, que faz parte da biblioteca *sklearn*. Esta função permite dividir *arrays* ou matrizes em subconjuntos aleatórios de treino e teste. Deste modo, usou-se como parâmetros de entrada o *array* obtido anteriormente, *image_list*, a percentagem de dados que queremos usar como treino (*train_size*), e um inteiro (*random_state*) que controla o embaralhamento dos dados,

permitindo que os dados sejam distribuídos da mesma forma sempre que este valor é igual.

A função anteriormente referida foi usada duas vezes para cada classe, tal como se pode ver na Figura 7, sendo que na primeira utilização usou-se um *train_size* de 0,75 o que resulta num output de 2 *arrays*, um que contém o *path* de 75% das imagens e que serão usadas para treino, e outro que voltará a ser dividido. Na segunda utilização, usou-se o segundo *array* anteriormente com um *train_size* usado foi de 0,5 o que origina, novamente, dois outputs que contém os *path* das imagens que serão usadas como validação e como teste. Assim, 75% dos dados são usados com treino, 12,5% são usados para validação e os restantes 12,5% são usados como teste.

```
image_list = glob.glob(r".\imagem\chicken_wings\*")
train_aux, test_aux1 = train_test_split(image_list, train_size=0.75, random_state = rs)
train += train_aux
test_aux, val_aux = train_test_split(test_aux1, train_size=0.5, random_state = rs)
test += test_aux
val += val_aux
```

Figura 7. Divisão do dataset, neste caso apenas da classe *chicken_wings*, em imagens de treino, validação e teste.

O segmento de código presente na Figura 7 foi usado para cada uma das classes usadas no desenvolvimento deste modelo. Para além disso, para se testar o modelo, o valor usado do *random_state* foi se diferente em cada um dos testes, tendo-se feito variar o parâmetro *rs* entre 1 e 5.

Continuando na fase de pré-processamento, já depois dos dados estarem divididos nos três conjuntos anteriormente referidos, as imagens de cada um dos conjuntos são lidas, redimensionadas e convertidas num *array*. Para se ler as imagens e redimensioná-las recorreu-se à biblioteca *cv2*, tendo-se usado as funções *imread()* e *resize()*, respetivamente.

A função *resize()* usa como parâmetros de entrada, uma imagem, que corresponde ao output da função *imread()*, e as dimensões que se querem que a imagem apresente, sendo o seu output uma imagem igual à imagem de entrada mas com as dimensões que foram especificadas. O uso da função *resize()* foi uma necessidade e não uma opção, dado que as imagens da base de dados já apresentavam todas as mesmas dimensões mas por questões de *hardware*, neste caso de memória, foi necessário diminuir o tamanho das imagens, ficando estas como 124x124 *pixels*.

Já após as imagens terem sido redimensionadas estas são convertidas num *array*, com auxílio da função *array()* da biblioteca *numpy*. O uso destas 3 funções encontra-se ilustrado na Figura 8.

```
im = cv.resize(cv.imread(f), (min_side, min_side))
arr = np.array(im)
images.append(arr)
```

Figura 8. Leitura e redimensionamento das imagens.

Os *arrays* resultantes destas imagens são posteriormente divididos por 255 (Figura 9). Isto acontece porque os píxeis de uma imagem RGB (*red*, *green* and *blue*, do inglês vermelho, verde e azul) podem variar entre 0 e 255 para cada uma destas cores, o que representa um código de cor.

```
Train_img = Train_img / 255.0
Test_img = Test_img / 255.0
val_img = val_img / 255.0
```

Figura 9. Normalização das imagens de treino, validação e teste.

Esta abordagem foi usada dado que utilizar uma imagem numa rede neuronal pode tornar os cálculos muito complexos, assim, para reduzir esta complexidade, optou-se por normalizar os valores para estes variarem entre 0 e 1. Desta forma, os números serão pequenos o que torna o cálculo mais simples e rápido.

Estando a fase de pré-processamento dos dados terminada, passamos à terceira fase, que é a fase de configuração da RNC. Nesta fase fez-se variar a arquitetura da rede e os seus hiperparâmetros de modo que se pudesse obter os melhores resultados possíveis.

O modelo proposto que, tal como se pode observar pela Figura 10, inicia-se com uma fase de *augmentation*. Este é um processo muito usado em *machine learning* e serve o propósito de aumentar artificialmente a quantidade de dados a partir dos dados já existentes. Para tal, fazem pequenas alterações aos dados, que no caso de imagens estas podem pôr rotações, modificações na cor, *zoom* ou uma combinação destas alterações.

```

model = tf.keras.Sequential([
    data_augmentation,
    tf.keras.applications.EfficientNetB3(
        input_shape=(size, size, 3),
        weights='imagenet',
        include_top=False,
        drop_connect_rate=0.5
    ),
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.6),
    tf.keras.layers.Dense(5, activation='softmax')
])

```

Figura 10. Definição das diferentes camadas presentes na RNC desenvolvida.

No caso do modelo desenvolvido, a fase de *augmentation* produz novas imagens, sendo que para tal as imagens sofrem uma rotação horizontal que pode ser quer em sentido horário quer em sentido anti-horário, seguidamente sofrem uma rotação aleatória dentro do intervalo $[-20\% * 2 * \pi, 20\% * 2 * \pi]$, depois sofrem ainda um aumento entre -10% a 10%, e por fim sofrem uma variação de contraste que é dada pela fórmula presente na equação 5.

$$\text{Variação de contraste} = (x - \bar{x}) \times 0,2 + \bar{x} \quad (5)$$

Para se realizarem estas alterações, recorreu-se à biblioteca *keras*, que é capaz de correr na *biblioteca tensorflow*, e a algumas das suas funções, a saber: *RandomFlip()*, *RandomRotation()*, *RandomZoom()* e *RandomContrast()*, tal como se pode constatar na Figura 11.

```

data_augmentation = tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal", input_shape=(size, size, 3)),
    layers.experimental.preprocessing.RandomRotation(0.2),
    layers.experimental.preprocessing.RandomZoom(0.1),
    layers.experimental.preprocessing.RandomContrast(0.2)
])

```

Figura 11. Definição dos processos que ocorrem na fase de *augmentation*.

Estas imagens criadas permitem, não só que hajam mais dados para treinar o modelo, mas também que o modelo seja treinado com imagens de posições e contrastes diferentes o que pode traduzir numa maior robustez do sistema.

Após o bloco de *augmentation*, usou-se mais uma função da biblioteca *keras*, neste caso a *EfficientNetB3()*. *EfficientNet* é uma arquitetura de rede neuronal convolucional e um método que utiliza um coeficiente composto para escalar uniformemente a largura, a profundidade e a resolução da rede. Para tal esta recorre a conjunto de coeficientes de escala fixos, ao contrário da prática convencional que escala as dimensões baseando-se em fatores arbitrários. A *EfficientNet*, cuja arquitetura esta

descrita na Figura 12, tem sido usada pois apresenta-se como mais eficiente e com melhores valores de *accuracy* quando comparada com outras *ConvNets* [68].

Assim a *EfficientNetB3()* foi usada para que fosse possível carregar um modelo de classificação de imagem com pesos pré-treinados no *ImageNet*, num processo ao qual se dá o nome de *transfer learning*.

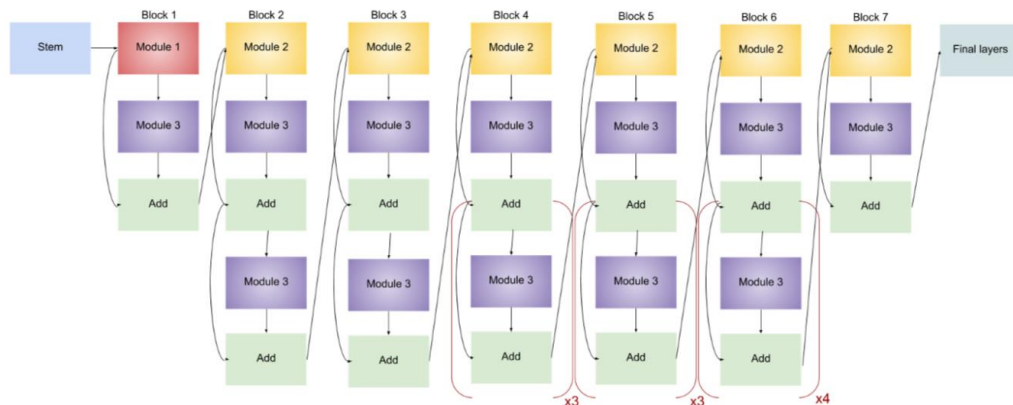


Figura 12. Arquitetura da *EfficientNetB3* [69].

Além do modelo carregado, a proposta contempla ainda outras camadas, todas elas formadas a partir de funções do *keras*. Na primeira dessas camadas usou-se a função *GlobalAveragePooling2D()*, sendo esta responsável por calcular um valor médio de todos os valores da matriz, calculado o produto da largura de entrada pela altura de entrada, para cada um dos canais de entrada.

Seguidamente, consta uma camada densa, que se trata de uma camada simples de neurónios em que cada neurónio recebe a entrada de todos os neurónios da camada anterior. Neste caso em particular, para se formar esta camada usou a função *dense()*, onde os parâmetros de entrada são o número de nós, que é de 512, e a função de ativação '*relu*', ou seja uma função ativação linear retificada.

Esta camada é seguida por uma camada de *droupout*, onde, neste caso específico, 60% das unidades de entrada, durante a fase de treino, são colocadas, de forma aleatória, a 0, ou seja, são ignoradas. O peso das entradas que não são colocadas a 0 é aumentado em $1 / (1 - 0,6)$ de modo que a soma sobre todas as entradas permaneça inalterada. O uso deste tipo de camadas é útil pois ajude a prevenir a ocorrência de *overfitting*.

Por fim, a última camada do modelo proposto é novamente uma camada densa, mas desta vez esta é constituída por 5 nós, dado que é a última camada de uma rede

neuronal com várias classes, e como tal o número de nós deve ser igual ao número de classes. A função de ativação nesta camada também é diferente da camada densa anterior, dado que a função *'relu'* não é a mais apropriada para problemas multiclasse. Para este tipo de problema a função *'sigmoid'*, por norma, é a mais apropriada, tendo sido a função de ativação usada aqui.

Estando a estrutura da rede definida, passou-se à definição dos métodos de avaliação da mesma, onde para tal mais uma vez usou-se uma função da biblioteca *keras*, neste caso a função *compile()*. Esta função recebe a lista de métricas que são usadas para avaliar a rede, tal como se pode observar na Figura 13.

```
model.compile(optimizer = Adam( learning_rate=0.001),  
              loss = 'categorical_crossentropy',  
              metrics=[tf.keras.metrics.Accuracy(),  
                      tf.keras.metrics.Precision(),  
                      tf.keras.metrics.Recall()])
```

Figura 13. Definição da função *compile()*.

Esta proposta assenta na implementação do algoritmo de otimização *Adam*. Este trata-se de um método estocástico de gradiente descendente que se baseia na estimativa adaptativa de momentos de primeira e segunda ordem. Algumas das suas características são que este tem pouca necessidade de memória e que é adequado para problemas que são grandes em termos de dados, duas características que vão de encontro às necessidades do presente modelo, quer pela quantidade e dados, quer pelos recursos computacionais usados.

Tal como referido no anteriormente nesta dissertação, no ponto 3.3, as métricas selecionadas para se fazer a avaliação da presente rede neuronal são a *loss*, a *accuracy*, a *precision* e o *recall*. Todas estas métricas foram calculadas usando as funções que possuem nomes iguais a elas mesmo e que pertencem à biblioteca *keras*. Para além disso definiu-se ainda uma *learning rate* de 0,001, sendo este um hiperparâmetro que controla o quanto o modelo é alterado face ao erro registado.

Por fim, para treinar o modelo usou-se a função *fit()*, Figura 14. Definição da função *fit()*. Nesta função tem como parâmetros de entrada as imagens de treino, *Train_img*, as classes de cada uma das imagens de treino, *Train_labels_cat*, os dados para validação, que inclui tanto as imagens, *Val_img*, quer as suas classes, *Val_labels_cat*, e ainda o número de épocas, 50.

```

history = model.fit(
    Train_img,
    Train_labels_cat,
    validation_data = (Val_img, Val_labels_cat),
    epochs= 50,
    callbacks=[early_stop, reduce_lr])

```

Figura 14. Definição da função *fit()*.

Para além destes dados, à função *fit()* foram ainda adicionados dois *callbacks*, que são objetos capazes de executar ações em diferentes estágios da fase de treino. Neste caso os *callbacks* usados servem o propósito de fazer uma paragem antecipada (*early_stop*) e de reduzir a *learning rate* (*reduce_lr*). As funções usadas para definir estes *callbacks* estão presentes na Figura 15.

```

early_stop = EarlyStopping(monitor = 'val_accuracy',
                           patience = 10,
                           verbose = 1,
                           restore_best_weights = True)
reduce_lr = ReduceLRonPlateau(monitor = 'val_accuracy',
                              patience = 3,
                              verbose = 1)

```

Figura 15. *Callbacks* usados no modelo proposto.

Para a definição do *early_stop* e do *reduce_lr* recorreu-se as funções *EarlyStopping()* e *ReduceLRonPlateau()*, respetivamente. Em ambas funções o parâmetro a analisar foi a *accuracy* de validação. Por outro lado, a *patience*, ou seja, o número de épocas para que uma ação ocorra é diferente, sendo este 3 para o *reduce_lr*, e 10 para *early_stop*. Isto significa que, a *learning rate* reduz quando a *accuracy* da validação não melhora durante 3 épocas, e que passado 10 épocas sem melhorias na mesma *accuracy* a fase de treino termina de forma antecipada.

Para terminar o bloco do desenvolvimento do modelo, falta apenas a fase de classificação. Esta fase é usada para testar o bom funcionamento do modelo desenvolvido. Assim, recorreu-se os dados de teste, juntamente que a função *evaluate()*, da biblioteca *keras*, para testar o modelo criado, Figura 16.

```

score = model.evaluate(Test_img, Test_labels_cat)

```

Figura 16. Função usada para avaliar a RNC desenvolvida.

A função *evaluate()*, usa como parâmetros de entrada as imagens de teste, *Test_img*, bem como as classes correspondentes, *Test_labels_cat*, e retorna a *loss*, e as valores das métricas para o modelo no modo de teste, como se pode ver na Figura 17.

```
19/19 [=====] - 20s 1s/step - loss: 0.0746 -  
accuracy: 0.9797 - precision: 0.9813 - recall: 0.9763
```

Figura 17. Resultados obtidos num dos testes realizados à rede desenvolvida, sendo estes o output da função *evaluate()*.

De modo a guardar-se o modelo obtido, para que pudesse ser usado numa aplicação real de reconhecimento de alimentar, usou-se a função *save()*, como demonstrado na Figura 18.

```
model.save('model.h5')
```

Figura 18. Função usada para guardar o modelo desenvolvido.

Esta função permite guardar o modelo para este mais tarde poder ser chamado. Para tal, esta função guarda a arquitetura, os valores dos pesos, as informações de compilação e o otimizador do modelo.

Terminado o bloco desenvolvimento do modelo, passamos então ao bloco de aplicação do modelo. Este bloco foi implementado de modo a se desenvolver um sistema que recebe uma imagem, e classifica-a consoante as classes para o qual o modelo foi treinado. O sistema recebe ainda a porção do alimento ingerida de modo a ser capaz de apresentar a informação nutricional ao alimento ingerido.

Para que esta implementação fosse possível, foi necessário recorrer, novamente, à biblioteca *keras*, onde esta foi usada para importar o modelo criado anteriormente (Figura 19).

```
model = load_model('model.h5')
```

Figura 19. Função usada para carregar o modelo anteriormente desenvolvido.

De seguida, a imagem inserida pelo utilizador passa pelo mesmo processo de redimensionamento e de normalização que ocorre no bloco de pré-processamento. Depois, é realizada a identificação do prato inserido, sendo que para tal usou-se a função *predict()*, Figura 20, também ela biblioteca *keras*.

```
img_pred = model.predict(image)
```

Figura 20. Função *predict()*, usada para se fazer a identificação do prato inserido pelo utilizador.

Esta função permite gerar previsões de saída para as amostras de entrada. Assim, como parâmetros de entrada usou-se a variável *image*, que corresponde à imagem inserida pelo utilizador após passar pela fase de pré-processamento. Já a saída corresponde a um

array com as probabilidades de cada uma das classes ser a correspondente à imagem inserida pelo utilizador.

Desta forma, seleciona-se a classe com maior probabilidade de ser a correta como a correspondente ao *input*. De seguida, é pedido ao utilizador para inserir a porção da refeição em gramas (*portion*), tal como representado na Figura 21.

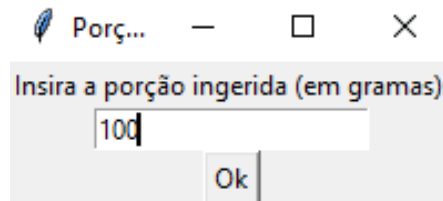


Figura 21. Janela na qual o utilizador indica ao sistema a porção ingerida.

Assim que se obtém uma resposta válida, esta é combinada com a *string* “g” e com a classe obtida anteriormente (*food*) para formar a variável *query*, Figura 22.

```
query = str(portion) + 'g' + food
```

Figura 22. Formação da *query* usada para fazer a chamada da *Nutrition API*.

A *query* obtida é depois usada para se fazer a pesquisa pela informação nutricional da refeição inserida pelo utilizador na *Nutrition API*, Figura 23. Para fazer a chamada desta API é ainda necessário uma chave de acesso (*api_key*).

```
api_url = 'https://api.api-ninjas.com/v1/nutrition?query={}'.format(query)
response = requests.get(api_url, headers={'X-API-Key': api_key})
```

Figura 23. Chamada da *Nutrition API*.

A resposta desta chamada da API irá fornecer, entre outras coisas, o nome do prato selecionado, a porção indicada e os valores nutricionais correspondentes ao prato e porção indicados. Dentro dos valores nutricionais a partir desta é possível obter-se informação relativamente ao nível de calorias, gorduras, gorduras saturadas, proteínas, sódio, potássio, colesterol, carboidratos, fibras e açúcares.

Assim estes valores são apresentados ao utilizador da forma demonstrada na Figura 24.

Informação Nutricional

CHICKEN WINGS



Calorias: 261.2
Porção: 100.0g
Gorduras: 16.7g
Gorduras Saturadas: 4.9g
Proteínas: 24.0g
Sódio: 96mg
Potássio: 144mg
Colesterol: 140mg
Carboidratos: 0.0g
Fibras: 0.0g
Açúcares: 0.0g

Figura 24. *Display* das informações relativas ao *input* do utilizador, nomeadamente o nome da refeição, a imagem inserida e ainda a informação nutricional correspondente.

CAPÍTULO 4 – DISCUSSÃO E RESULTADOS

4. Discussão e Resultados

Neste capítulo são apresentados e analisados os resultados obtidos durante o desenvolvimento da presente dissertação. Para tal, recorreu-se a diferentes métodos para gerar os melhores resultados possíveis, sendo que depois estes foram comparados com alguns dos melhores resultados presentes na literatura.

4.1. Resultados

De modo que fosse possível desenvolver o presente sistema, numa fase embrionária do projeto, recorreu-se a diferentes abordagens para se descobrir qual a que poderia ser mais útil para se realizar a deteção automática de alimentos. O objetivo consistiu em implementar-se diferentes sistemas e avaliar qual aquele que é capaz de fazer uma deteção mais precisa dos alimentos apresentados.

Assim, recorreu-se a 4 métodos de *machine learning* diferentes, a saber: *Naive Bayes* (NB), *Random Forest* (RF), *Support Vetor Machine* (SVM) e, por fim, Redes Neurais. No entanto, desde cedo ficou perceptível que as Redes Neurais eram o método mais eficaz para a realização desta tarefa. Isto porque, de acordo com os resultados apresentados na Tabela 3, os restantes métodos não se mostraram suficientes para realizar esta tarefa com a confiança que se espera de um sistema deste tipo.

Tabela 3. Valores de *accuracy* obtidos usando algoritmos de classificação baseados em SVM, RF e NB.

Método	<i>Accuracy</i>
SVM (linear)	0,4793
SVM (rbf)	0,5283
SVM (poly)	0,5646
SVM (sig)	0,4371
<i>Random Forest</i>	0,5762
<i>Naive Bayes</i>	0,4785

Analisando os dados da Tabela 3 podemos ver que estes métodos são um pouco limitados para se realizar uma deteção automática de alimentos, pelo menos para a base de dados escolhida. Isto pode-se comprovar pelos valores de *accuracy* alcançados, dado que estes não passam dos 56%. Dentro destes métodos de classificação, o RF foi aquele

que apresentou melhores resultados, tendo apresentado uma *accuracy* de 57,62. Para avaliar o método SVM foram usados 4 *kernels* diferentes, onde os *kernels* poly e rbf foram os que se mais aproximaram dos resultados obtidos pelo RF, com 56,46 e 52,83%, respetivamente. Por outro lado, os kernels linear e sig, mostraram-se ser menos capazes de realizar esta tarefa, tendo obtido valores de *accuracy* de 47,93 e 43,71%, respetivamente. Por fim, o NB tal como os dois anteriores também mostrou mais dificuldades em prever a classe correta, tendo obtido o segundo pior resultado de *accuracy* dos métodos aqui experimentados, 47,85%, superando apenas o SVM com *kernel* sig.

Tendo em conta que os resultados obtidos usando estes métodos não foram suficientemente satisfatórios e que a literatura retrata um maior uso de redes neuronais para a realização deste tipo de sistemas, a tentativa de desenvolvimento do sistema usando estes métodos de classificação foi abandonada. Deste modo, o presente estudo focou-se em tentar desenvolver a melhor rede possível. Depois de várias experiências, onde se fez variar os vários parâmetros chegou-se à solução proposta, que se encontra descrita na secção 3.4.

Para se avaliar esta solução recorreu-se a três parâmetros, *accuracy*, *precision* e *recall*, sendo que cada um destes parâmetros foi avaliado 5 vezes, com dados de treino, validação e teste diferentes em cada um destes testes. Os resultados obtidos aquando destes testes encontram-se representados na Tabela 4.

Tabela 4. Valores de *accuracy*, *precision* e *recall* obtidos usando a rede neuronal proposta.

Teste	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
1	0,9695	0,9728	0,9695
2	0,9729	0,9746	0,9729
3	0,9797	0,9813	0,9763
4	0,9763	0,9779	0,9746
5	0,9646	0,9645	0,9629
Média	0,9734	0,9742	0,9712

Olhando para a Tabela 4 podemos verificar que os resultados obtidos, em cada um dos testes, são significativamente melhores quando comparados com os da Tabela 3,

apresentando um valor de *accuracy*, aproximadamente, 40% superior à *accuracy* obtida usando o método do RF, que tinha sido a melhor de entre os outros métodos.

Passando para os valores médios obtidos, podemos verificar que os valores obtidos de *accuracy*, *precision* e *recall* são muito satisfatórios para sistemas deste tipo, sendo estes de 97,34, 97,42 e 97,12%, respectivamente. Além disso, os resultados obtidos em cada um teste são muito semelhantes aos resultados obtidos nos restantes testes, o que significa que o sistema não funciona apenas para um determinado conjunto de dados de treino e teste, não caindo assim em situações de *overfitting* ou *underfitting*.

Outro fator avaliado que também ajuda a verificar se existe um bom ajuste é a *training loss* e a *validation loss*, estando estas representadas na Figura 25.

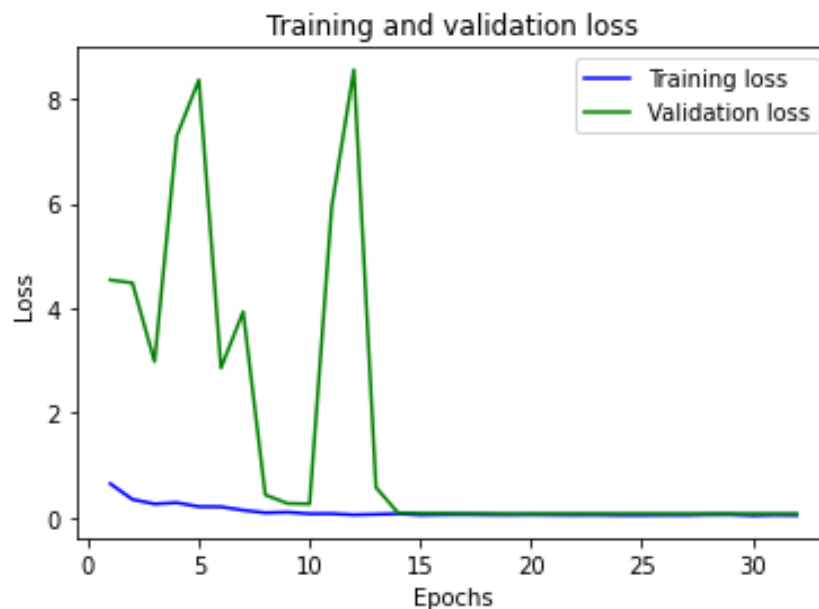


Figura 25. Gráfico que demonstra a forma como tanto a *training loss* como a *validation loss* variam ao longo das épocas de treino.

Observando a Figura 25, podemos verificar que, durante a fase de treino, tanto a *training* como a *validation loss* tendem a diminuir com o avanço das épocas de treino até estabilizarem, sendo este mais um indicador de um bom ajuste. No entanto, a *validation loss* não serve apenas para verificar em que estado se encontra o ajuste do sistema, isto porque esta está diretamente relacionada com a *validation accuracy* (que se encontra na Figura 26).

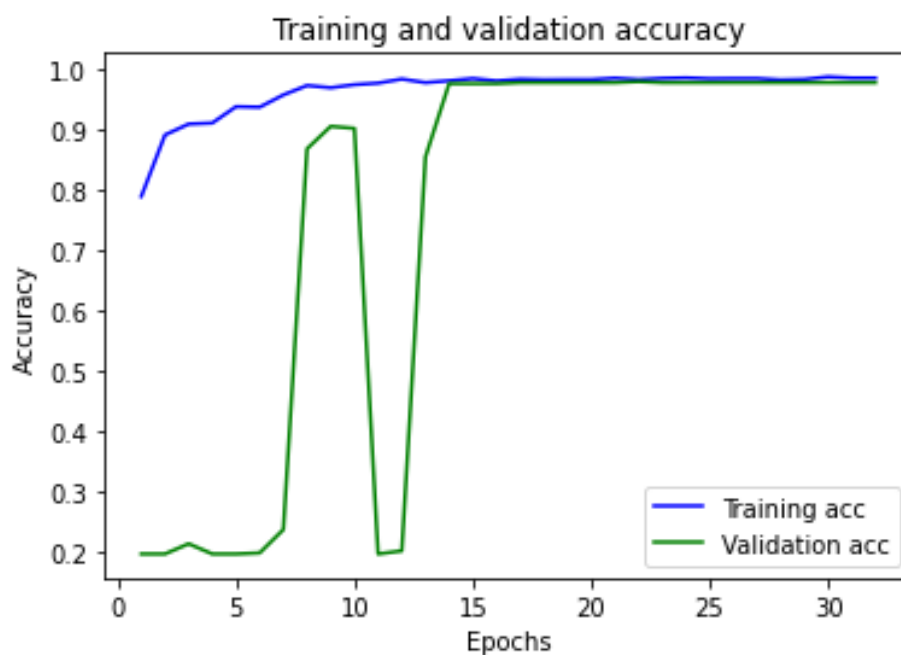


Figura 26. Gráfico que demonstra a forma como tanto a training accuracy como a validation accuracy variam ao longo das épocas de treino.

Analisando em simultâneo as Figura 25 e Figura 26, é de fácil compreensão que quando a *validation loss* começa a aumentar, a *validation accuracy* começa a diminuir, o que significa que não está a aprender. Por outro lado, quando a *validation loss* começa a diminuir, a *validation accuracy* começa a aumentar, o que por sua vez significa que o modelo está a aprender e a funcionar de forma correta. De realçar ainda que não se observa um aumento simultâneo da *validation loss* e da *validation accuracy*, o que seria um indício de que o sistema estaria a sofrer de *overfitting*.

Outro fator a analisar é que o uso do *callback early_stop* foi bem empregue neste sistema dado que em nenhum dos testes se atingiu as 50 épocas. O uso desta ferramenta permite assim que haja uma paragem precoce da fase de treino, quando a *validation accuracy* deixa de aumentar, o que evita que o modelo caia em situações de *overfit*.

4.2. Discussão

Tendo em conta que a ETHZ Food-101 é uma das bases de dados de comida mais conhecidas, existem muitos outros trabalhos de classificação alimentar a partir de imagens baseados nesta base de dados.

Pandey et al (2017), no seu trabalho mostraram que as RNCs têm um desempenho superior comparativamente com outras técnicas de *machine learning* para realizar a deteção alimentar baseada no dataset da ETHZ Food-101. Após os autores terem

verificado este facto estes focaram-se em tentar desenvolver a RNC que melhor se adaptasse a esta função. Assim, estes desenvolveram uma rede que consiste num agrupamento de 3 RNC diferentes, *AlexNet*, *GoogLeNet* e *ResNet*. Estas são treinadas com as imagens da Food-101, após estas sofrerem pré-processamento, nomeadamente equalização do histograma.

Existe depois uma camada que concatena as características obtidas das redes anteriores, tendo esta uma função de ativação de *ReLU*. A saída desta camada passa para uma camada que combina as saídas para o comprimento desejado do número de classes presentes.

Por fim, este modelo tem ainda uma camada *softmax* para fazer a previsão das classes de cada imagem de entrada. Assim, os resultados obtidos para esta arquitetura uma *accuracy* de 72.12%, 91.61% e 95.95% para top-1, top-5 e top-10, respetivamente [70].

Aguilar et al (2017), utilizaram a base de dados ETHZ Food-101 para realizar reconhecimento alimentar a partir da fusão de classificadores baseados em redes neuronais. Nesta abordagem as imagens são redimensionadas, sendo que as menores dimensões foram de 224x224 *pixels*. Para avaliar a capacidade de previsão do modelo desenvolvido neste estudo usou-se a *accuracy* quer para 1-crop, quer para 10-crops. Neste trabalho, os autores experimentaram diferentes abordagens tanto baseadas na fusão de redes neuronais como também abordagens baseadas em classificadores usados em separado.

Os resultados obtidos mostraram que a *accuracy* global é superior para as 10-crops quando comparado com 1-crop, sendo estes cerca de 1% melhores. Da mesma forma, os resultados obtidos também mostraram que a *accuracy* é superior quando há a fusão de redes neuronais, sendo esta cerca de 1,5% mais elevada do que o melhor resultado dos classificadores avaliados separadamente. Assim, os melhores resultados, obtidos para as situações de fusão de redes, foram de 86.07% e de 86.71%, para 1-crop e para 10-crops, respetivamente.

Ainda assim, foi relatado que abordagens baseadas em classificados isolados InceptionV3, ResNet200, WRN e WISeR conseguiram obter melhores resultados do que os atingidos pelos autores. Contudo, os últimos três modelos necessitam de um servidor multi-GPU para atingir esses resultados. Além disso, esses modelos têm 2,5 vezes mais

parâmetros do que os modelos escolhidos, o que envolve um elevado custo computacional [71].

Isto vai de encontro aos resultados obtidos por *Hassannejad et al* (2016) e de *Martinel et al* (2018).

Por um lado, *Hassannejad et al* (2016) recorreram a uma abordagem baseada no classificador *InceptionV3*, para realizarem a classificação de imagens alimentares com base na de ETHZ Food-101. A arquitetura usada é então uma RNC profunda com 54 camadas, contando apenas as camadas com parâmetros a serem aprendidos. Esta abordagem, quando treinada com a Food-101, onde a resolução das imagens usada foi de 299×299 pixels, obteve uma *accuracy* de top-1 de 88,28% e uma *accuracy* de top-5 de 96,88%. Outra questão que ficou visível pelos resultados mostrados pelos autores é que as RNC se mostraram mais capazes de realizar esta tarefa quando comparado com outros métodos.

Os autores relataram ainda que os resultados obtidos tinham sido os melhores até à data da publicação, apresentando uma menor complexidade computacional e por isso consideraram que esta era a abordagem mais próxima para se usar em sistemas móveis [72].

Por outro lado, *Martinel et al* (2018), introduziram uma nova forma de lidar com a estrutura alimentar, tendo-se se focado nas características verticais dos alimentos. Para tal, o modelo desenvolvido pelos autores consiste numa rede com dois ramos principais para a extração de características: um ramo de uma rede residual que codifica representações visuais genéricas das imagens dos alimentos, sendo capaz de proporcionar uma hierarquia profunda e de captar as características da maioria dos alimentos; um ramo de uma *slice network* com uma camada convolucional em fatias que captura os traços verticais dos alimentos.

As características extraídas são então combinadas numa rede com uma arquitetura *WISeR*. O modelo proposto pelos autores foi capaz de gerar bons resultados, tendo alcançado uma top-1 *accuracy* de 90,27% e 98,71% para uma *accuracy* top-5 para o dataset da Food-101 onde as imagens tinham uma dimensão de 256×256 pixels [73]. Mais uma vez, foi visível que as RNC são melhores para este tipo de tarefa.

Mais tarde, *Tripathi* (2021), propôs uma arquitetura *DenseNet161* pré-treinada com blocos de *augmentation* antes de alimentar o modelo. No bloco de *augmentation* as

imagens sofreram redimensionamento de rotação aleatória, rotação horizontal aleatória e normalização. Para além disso, as imagens usadas para treino tinham uma resolução de 384x384 *pixels*.

Com esta abordagem o autor conseguiu atingir uma top-1 *accuracy* de 93,26% e uma top-5 *accuracy* de 99,01%, sendo estes os melhores resultados encontrados até então. Além disso os autores também relataram que as RNC são os melhores métodos para fazer a classificação alimentar baseando-se neste *dataset* [74].

Uma outra abordagem a considerar é a de *Yunus et al* (2018), que propuseram um sistema capaz estimar automaticamente atributos alimentares, tais como ingredientes e valor nutricional, baseado numa imagem de entrada. Para realizar a classificação dos alimentos os autores usaram um RNC baseada numa arquitetura *InceptionV4* pré-treinada com a *Imagenet*. Para além desta fase de *transfer learning* o modelo contempla ainda camadas de *dropout*, ativações *ReLU* e *softmax*. Os resultados obtidos com este modelo, para o dataset da Food-101, com várias *crops* foram de 79,22% e 90,63% para *accuracy* de top-1 e top-5, respetivamente.

O método proposto permite ainda obter os ingredientes e estimar os seus valores nutricionais, sendo estes estimados tendo por base informações recolhidas da internet, tal como blogs de nutrição e bases de dados [74].

A Tabela 5 apresenta um resumo dos trabalhos anteriormente referidos, sendo que para cada um consta os autores, a resolução das imagens usada, os métodos, a *accuracy* obtida bem como alguns comentários adicionais.

Tabela 5. Alguns dos principais trabalhos de classificação alimentar baseada em imagens da base de dados ETHZ Food-101 presentes na literatura.

Autores	Resolução	Métodos	<i>Accuracy</i>	Comentários
<i>Pandey et al</i> (2017) [70]	512x512 <i>pixels</i>	Equalização do histograma Agrupamento de 3 RNC: <i>AlexNet</i> , <i>GoogLeNet</i> , <i>ResNet</i>	Top-1=72.12% Top-5=91.61% Top-10=95.95%	N/A
<i>Aguilar et al</i> (2017) [71]	224x224 <i>pixels</i>	<i>Transfer learning</i> Fusão de RNC: <i>InceptionV3</i> e <i>ResNet50</i>	Top-1=86.71%	N/A

<i>Hassannejad et al</i> (2016) [72]	299×299 pixels	<i>Transfer learning</i> <i>Google InceptionV3</i>	Top-1=88,28% Top-5=96,88%	Verificaram que RNC obtiveram melhores resultados comparativamente com outros métodos;
<i>Martinel et al</i> (2018) [73]	256x256 pixels	<i>Transfer learning</i> <i>Augmentation</i> <i>WISeR</i>	Top-1=90,27% Top-5=98,71%	Rede com dois ramos para a extração de características; Verificaram que RNC obtiveram melhores resultados comparativamente com outros métodos;
<i>Tripathi</i> (2021) [74]	384x384 pixels	<i>Transfer learning</i> <i>Augmentation</i> <i>DenseNet161</i>	Top-1=93,26% Top-5=99,01%	Verificaram que RNC obtiveram melhores resultados comparativamente com outros métodos;
<i>Yunus et al</i> (2018) [75]	Não especificado	<i>Transfer learning</i> <i>Augmentation</i> <i>InceptionV4</i>	Top-1=79,22% Top-5=90,63%	Fornecer <i>feedback</i> dos valores nutricionais

Embora não seja possível fazer um paralelismo entre o trabalho desenvolvido e os outros trabalhos presentes na literatura, dado que o presente trabalho foi desenvolvido apenas para 5 classes e os restantes foram desenvolvidos para as 101 classes da ETHZ Food-101, é possível tirar-se algumas ilações ou criar algumas hipóteses.

Desta forma, comparando os resultados obtidos neste trabalho com os trabalhos analisados é facilmente perceptível que as RNC são o método que melhor se adequa para realizar a identificação de refeições usando a ETHZ Food-101 como base para treinar. Isto é comprovado porque neste trabalho os classificadores NB, RF e SVM obtiveram piores resultados comparativamente com as RNC, algo que vai de encontro ao que os autores em [70], [72], [73] e [74] afirmam.

De forma a facilitar a fase de treino quer a abordagem proposta quer as restantes abordagens analisadas recorreram ao uso de *transfer learning*. Para além disso, outra técnica que, mais uma vez também foi usada nesta abordagem, é a *augmentation* para criar mais dados e variados para se poder treinar os modelos com mais dados.

Muitas das abordagens encontradas na literatura recorrem à *accuracy* top-5 ou top-10 para avaliarem os modelos desenvolvidos algo que não acontece na presente proposta. Ainda assim estas abordagens também recorrem à *accuracy* top-1, que é o mesmo que a *accuracy* usada para avaliar o sistema proposto. Assim, analisando esta métrica é possível verificar-se que a *accuracy* alcançada com este modelo é superior a todas as outras presentes na literatura. No entanto, há que ter em atenção que as propostas disponíveis na literatura contemplam o uso das 101 classes disponíveis na base de dados enquanto no presente trabalho apenas foram usadas 5 classes. Deste modo, é esperado que ao aumentar o número de classes de 5 para 101 a *accuracy* registada desça consideravelmente.

Por outro lado, a abordagem proposta usa uma resolução de imagem de 124x124 *pixels*, sendo esta pelo menos metade da resolução usada nas outras abordagens. Ora este pode ser um fator a ter em conta dado que o sistema proposto mostrou-se capaz de prever com elevada exatidão a classe correspondente às imagens inserida mesmo após esta ser redimensionada numa imagem muito pequena. Isto é relevante pois trata-se de uma questão de *downscaling*, onde imagens maiores serão reduzidas, o que torna mais difícil para a RNC aprender os recursos necessários para classificação ou deteção, pois o número de pixels onde o recurso vital estará presente é significativamente reduzido. Esta questão ganha ainda mais força pelo facto da arquitetura presente na literatura com melhor *accuracy* ser aquele que apresenta uma resolução maior sendo esta de 384x384 *pixels*, o que é cerca de 3 vezes superior, quer para largura quer para altura, às imagens usadas no modelo proposto.

Além disso, ao usar-se uma resolução de imagem reduzida o tempo de processamento será menor. Isto deve-se ao facto de uma imagem reduzida corresponder a uma matriz mais pequena, ou seja, apresentam menos atributos. Ora como o número de atributos é menor isto significa que o tempo de processamento necessário e os recursos computacionais usados serão menores. Ainda assim, isto não significa que a abordagem proposta seja necessariamente mais rápida que as restantes abordagens presentes na literatura, mas implica que o tempo de processamento é menor comparativamente com

abordagens que usem uma arquitetura similar à arquitetura usada, mas com imagens de maiores dimensões.

Yunus et al (2018), no seu modelo também possibilitam obter informações nutricionais, tal como no modelo proposto. No entanto, o modelo proposto por Yunus e os restantes colegas usa informações que são retiradas de diferentes páginas web, sendo que as porções podem não ser estabelecidas da mesma forma entre todos os sites. Desta forma, não existindo um padrão, nem a estimativa da porção que é ingerida, as informações nutricionais fornecidas podem estar algo distantes da realidade. Por outro lado, o modelo proposto neste trabalho pede ao utilizador para inserir a porção ingerida o que levará a apresentar um valor mais próximo daquele que é real, embora o sistema ainda não esteja preparado para distinguir diferentes receitas.

CAPÍTULO 5 – CONCLUSÕES

5. Conclusões

Esta secção apresenta um resumo geral de todo o trabalho realizado, explorando os principais resultados obtidos e os objetivos alcançados, e referindo as principais limitações do trabalho.

5.1. Conclusão

A presente tese enquadra-se na área da tecnologia aplicada à saúde, e propõe-se ao desenvolvimento de um sistema capaz de detetar informações relativamente aos hábitos alimentares de pacientes de DM2, através de imagens.

O modelo proposto nesta tese está assente na utilização de métodos de *transfer learning* e de RNC, tendo ficado visível que estes métodos podem ser implementados para realizar o reconhecimento alimentar com elevada eficácia. Para além disso, as RNC mostraram-se como o melhor método para a realização desta tarefa, não só pelos resultados obtidos neste trabalho, mas também por resultados obtido por outros autores.

A abordagem adotada nesta dissertação ficou limitada pelas capacidades de computação dos recursos disponíveis. Estas limitações impediram não só o uso de imagens com maiores dimensões, como também impediu que fossem usadas as 101 classes presentes na base de dados da ETHZ Food-101.

Ainda assim, foi possível atingir os dois principais objetivos desta tese, sendo estes o desenvolvimento de um sistema capaz de identificar uma refeição consoante uma imagem inserida pelo utilizador, bem como estimar os valores nutricionais dessa refeição.

Embora os resultados obtidos não possam ser considerados os melhores resultados presentes na literatura, dado que não contemplam o uso das 101 classes, a proposta apresentada pode vir a ser uma hipótese a testar num ambiente com melhores recursos e que poderá apresentar-se a um nível robusto para desempenhar a função de identificar refeições a partir de imagens. Além disso, a abordagem proposta é uma abordagem que não requer muitos recursos computacionais, tornando assim a sua implementação mais fácil em ambientes com menos recursos.

5.2. Trabalhos Futuros

Com vista a dar seguimento a este trabalho, existem várias melhorias que podem ser aplicadas de forma a complementar o mesmo e assim torná-lo numa aplicação que possa ser utilizada no dia a dia.

As várias melhorias podem passar por aumentar o número de classes, pelo menos para as 101 da ETHZ Food-101, implementar uma fase de segmentação, de modo a diminuir-se o ruído presente nas imagens e assim facilitar a identificação da classe correta.

Para além destas questões que melhorariam a rede desenvolvida, existem melhorias que podem melhorar o sistema em si, nomeadamente transformá-lo uma aplicação móvel, guardar os dados das refeições inseridas de modo a criar-se um perfil nutricional do utilizador e considerar diferentes formas de cozinhar os alimentos.

Com estas melhorias o sistema proposto pode ser uma ferramenta bastante útil no controlo da DM2, e assim facilitar e melhorar a vida dos pacientes que sofrem desta doença.

Referências

- [1] Food Friend, “Dissemination plan,” pp. 1–11, 2019, [Online]. Available: <https://itea4.org/project/food-friend.html>.
- [2] Food Friend, “Actionable insights on nutritional intake,” pp. 1–2, 2019, [Online]. Available: <https://itea4.org/project/food-friend.html>.
- [3] R. Goyal and I. Jialal, “Diabetes Mellitus Type 2 - StatPearls - NCBI Bookshelf.” <https://www.ncbi.nlm.nih.gov/books/NBK513253/> (accessed Apr. 09, 2022).
- [4] Sociedade Portuguesa de Diabetologia, *Relatório Anual do Observatório Nacional da Diabetes*. 2019.
- [5] J. S. Gonzalez, E. Shreck, C. Psaros, S. A. Safren, C. Author, and H. Psychol Author manuscript, “Distress and Diabetes Treatment Adherence: A Mediating Role for Perceived Control HHS Public Access Author manuscript,” *Heal. Psychol*, vol. 34, no. 5, pp. 505–513, 2015, doi: 10.1037/hea0000131.
- [6] X. Cai, S. Qiu, D. Luo, L. Wang, Y. Lu, and M. Li, “Mobile Application Interventions and Weight Loss in Type 2 Diabetes: A Meta-Analysis,” *Obesity*, vol. 28, no. 3, pp. 502–509, 2020, doi: 10.1002/oby.22715.
- [7] R. P. da C. Proença, “Alimentação e globalização: algumas reflexões,” *Ciênc. cult. (São Paulo)*, vol. 62, no. 4, pp. 43–47, 2010.
- [8] I. Loureiro, “A importância da educação alimentar: o papel das escolas promotoras de saúde,” *Rev. Port. Saúde Pública*, vol. 22, pp. 43–55, 2004, [Online]. Available: <http://www.cdi.ensp.unl.pt/docbweb/MULTIMEDIA/RPSP2004-2/2-04-2004.PDF>.
- [9] “Healthy diet.” <https://www.who.int/news-room/fact-sheets/detail/healthy-diet> (accessed Feb. 15, 2022).
- [10] Food Friend, “Nutrition assessment and coaching overview,” pp. 1–31, 2019, [Online]. Available: Nutrition assessment and coaching overview.
- [11] M. J. Gregório, D. Teixeira, R. Monteiro, S. M. de Sousa, S. Irving, and P. Graça, “Para a Alimentação Saudável Nos Cuidados De Saúde Primários: Modelo de Intervenção e Ferramentas,” p. 51, 2020.

- [12] “Diabetes.” https://www.who.int/health-topics/diabetes#tab=tab_1 (accessed Feb. 22, 2022).
- [13] “What is diabetes,” Mar. 26, 2020. <https://www.idf.org/aboutdiabetes/what-is-diabetes.html> (accessed Feb. 22, 2022).
- [14] “Diagnóstico e Classificação da Diabetes Mellitus - Portal das Normas Clínicas.” <https://normas.dgs.min-saude.pt/2011/01/14/diagnostico-e-classificacao-da-diabetes-mellitus/> (accessed Feb. 22, 2022).
- [15] “ABC da Diabetes | Portal da Diabetes.” <https://apdp.pt/diabetes/abc-da-diabetes/> (accessed Feb. 22, 2022).
- [16] D. Care and S. S. Suppl, “Classification and diagnosis of diabetes: Standards of Medical Care in Diabetes-2020,” *Diabetes Care*, vol. 43, no. January, pp. S14–S31, 2020, doi: 10.2337/dc20-S002.
- [17] I. D. Federation, *International Diabetes Federation Diabetes Atlas*, 10th ed., vol. 102, no. 2. 2021.
- [18] S. Vijan, “Type 2 diabetes,” 2010. <https://www.idf.org/aboutdiabetes/type-2-diabetes.html> (accessed Mar. 03, 2022).
- [19] S. Chatterjee, K. Khunti, and M. J. Davies, “Type 2 diabetes,” *Lancet*, vol. 389, no. 10085, pp. 2239–2251, 2017, doi: 10.1016/S0140-6736(17)30058-2.
- [20] R. A. DeFronzo *et al.*, “Type 2 diabetes mellitus,” *Nat. Rev. Dis. Prim.*, vol. 1, no. July, pp. 1–23, 2015, doi: 10.1038/nrdp.2015.19.
- [21] Y. Zheng, S. H. Ley, and F. B. Hu, “Global aetiology and epidemiology of type 2 diabetes mellitus and its complications,” *Nat. Rev. Endocrinol.*, vol. 14, no. 2, pp. 88–98, 2018, doi: 10.1038/nrendo.2017.151.
- [22] “Complications of diabetes | Guide to diabetes | Diabetes UK.” <https://www.diabetes.org.uk/guide-to-diabetes/complications> (accessed Mar. 05, 2022).
- [23] “Type 2 diabetes - Health problems - NHS.” <https://www.nhs.uk/conditions/type-2-diabetes/health-problems/> (accessed Mar. 05, 2022).
- [24] W. Sami, T. Ansari, N. S. Butt, M. Rashid, and A. Hamid, “Effect Of Diet Counseling On Type 2 Diabetes Mellitus,” *Int. J. Sci. Technol. Res.*, vol. 4, no. 8,

- pp. 112–118, 2015.
- [25] M. A. Subhi, S. H. Ali, and M. A. Mohammed, “Vision-Based Approaches for Automatic Food Recognition and Dietary Assessment: A Survey,” doi: 10.1109/ACCESS.2019.2904519.
- [26] A. A. Fatehah, B. K. Poh, S. N. Shanita, and J. E. Wong, “Feasibility of Reviewing Digital Food Images for Dietary Assessment among Nutrition Professionals,” *Nutrients*, vol. 10, no. 8, Aug. 2018, doi: 10.3390/NU10080984.
- [27] J. Ngo, A. Engelen, M. Molag, J. Roesle, P. García-Segovia, and L. Serra-Majem, “A review of the use of information and communication technologies for dietary assessment,” *Br. J. Nutr.*, vol. 101 Suppl 2, no. SUPPL. 2, Jul. 2009, doi: 10.1017/S0007114509990638.
- [28] Y. Bi, M. Lv, C. Song, W. Xu, N. Guan, and W. Yi, “AutoDietary: A Wearable Acoustic Sensor System for Food Intake Recognition in Daily Life,” *IEEE Sens. J.*, vol. 16, no. 3, pp. 806–816, Feb. 2016, doi: 10.1109/JSEN.2015.2469095.
- [29] X. Hu *et al.*, “Study for Food Recognition System Using Deep Learning You may also like Research on Intrusion Detection and Target Recognition System Based on Deep Learning Study for Food Recognition System Using Deep Learning,” *ICPAS 2021) J. Phys. Conf. Ser.*, p. 12014, 2021, doi: 10.1088/1742-6596/1963/1/012014.
- [30] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, “A new approach to image-based estimation of food volume,” *Algorithms*, vol. 10, no. 2, 2017, doi: 10.3390/a10020066.
- [31] M. Bosch, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, “Combining Global and Local Features for Food Identification in Dietary Assessment,” *Proceedings. Int. Conf. Image Process.*, vol. 2011, pp. 1789–1792, 2011, doi: 10.1109/ICIP.2011.6115809.
- [32] J. Siswantoro, A. S. Prabuwono, A. Abdullah, and B. Idrus, “Automatic Image Segmentation using Sobel Operator and k-Means Clustering: A Case Study in Volume Measurement System for Food Products,” 2015.
- [33] W. Zhang, Q. Yu, B. Siddiquie, A. Divakaran, and H. Sawhney, “‘Snap-n-Eat’: Food Recognition and Nutrition Estimation on a Smartphone Weiyu,” *J. Diabetes*

- Sci. Technol.*, vol. 9, no. 3, pp. 525–533, 2015, doi: 10.1177/1932296815582222.
- [34] B. T. Nguyen, D.-T. Dang-Nguyen, T. X. Dang, T. Phat, and C. Gurrin, “A Deep Learning based Food Recognition System for Lifelog Images,” 2018, doi: 10.5220/0006749006570664.
- [35] P. Pouladzadeh, G. Villalobos, R. Almaghrabi, and S. Shirmohammadi, “A novel SVM based food recognition method for calorie measurement applications,” *Proc. 2012 IEEE Int. Conf. Multimed. Expo Work. ICMEW 2012*, pp. 495–498, 2012, doi: 10.1109/ICMEW.2012.92.
- [36] M. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, “A Food Recognition System for Diabetic Patients Based on an Optimized Bag-of-Features Model,” *IEEE J. Biomed. Heal. INFORMATICS*, vol. 18, no. 4, p. 1261, 2014, doi: 10.1109/JBHI.2014.2308928.
- [37] D. Allegra, S. Battiato, A. Ortis, S. Urso, and R. Polosa, “A review on food recognition technology for health applications Dario,” *Heal. Psychol. Res.*, vol. 8, no. 9297, 2020, doi: 10.4081/hpr.2020.9297.
- [38] S. Fang, C. Liu, F. Zhu, E. J. Delp, and C. J. Boushey, “Single-View Food Portion Estimation Based on Geometric Models HHS Public Access,” pp. 385–390, 2015, doi: 10.1109/ISM.2015.67.
- [39] Y. Yue, W. Jia, M. Sun, and : Drsun@pitt Edu, “Measurement of Food Volume Based on Single 2-D Image without Conventional Camera Calibration*,” 2012, doi: 10.1109/EMBC.2012.6346390.
- [40] “Camera recognition - Fully automatic booking.” <https://dishtracker.at/?lang=en> (accessed Apr. 05, 2022).
- [41] “Bite AI - Food Recognition API.” <https://bite.ai/food-recognition/> (accessed Apr. 05, 2022).
- [42] “Bite AI - Food Intelligence Platform.” <https://bite.ai/> (accessed Apr. 05, 2022).
- [43] “LogMeal Food AI - Image API for Food Detection Food Tracking based on the most Advanced Deep Learning.” <https://www.logmeal.es/api> (accessed Apr. 05, 2022).
- [44] “Home - AIGecko - Artificial Intelligence Machine Learning Computer Vision

- Deep Learning.” <https://www.aigecko.com/> (accessed Apr. 05, 2022).
- [45] “Calorie Mama Food AI - Food Image Recognition and Calorie Counter using Deep Learning.” <https://www.caloriemama.ai/#CalorieMama> (accessed Apr. 05, 2022).
- [46] “Foodvisor App - Nutrition Coaching - Eat Healthy & Lose Weight.” <https://www.foodvisor.io/en/> (accessed Apr. 05, 2022).
- [47] G. Rebala, A. Ravi, and S. Churiwala, “An Introduction to Machine Learning,” 2019, doi: 10.1007/978-3-030-15729-6.
- [48] T. O. Ayodele, “Types of Machine Learning Algorithms,” in *New Advances in Machine Learning*, 2010.
- [49] D. Denisko and M. M. Hoffman, “Classification and interaction in random forests,” doi: 10.1073/pnas.1800256115.
- [50] A. Cutler, D. R. Cutler, and J. R. Stevens, “Random Forests,” *Ensemble Mach. Learn.*, pp. 157–175, 2012, doi: 10.1007/978-1-4419-9326-7_5.
- [51] D. A. Pisner and D. M. Schnyer, “Support vector machine,” *Mach. Learn. Methods Appl. to Brain Disord.*, pp. 101–121, Jan. 2019, doi: 10.1016/B978-0-12-815739-8.00006-7.
- [52] S. Huang, C. A. I. Nianguang, P. Penzuti Pacheco, S. Narandes, Y. Wang, and X. U. Wayne, “Applications of Support Vector Machine (SVM) Learning in Cancer Genomics,” *Cancer Genomics Proteomics*, vol. 15, no. 1, pp. 41–51, Jan. 2018, doi: 10.21873/CGP.20063.
- [53] I. Rish, “An empirical study of the naive Bayes classifier.”
- [54] H. Zhang, “The Optimality of Naive Bayes,” Accessed: Oct. 09, 2022. [Online]. Available: www.aaai.org.
- [55] G. I. Webb, “Naïve Bayes,” 2016, doi: 10.1007/978-1-4899-7502-7_581-1.
- [56] K. O’shea and R. Nash, “An Introduction to Convolutional Neural Networks.”
- [57] S. Shalev-Shwartz and S. Ben-David, “Understanding Machine Learning: From Theory to Algorithms,” *Cambridge Univ. Press*, 2014, Accessed: Oct. 09, 2022. [Online]. Available: <http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning>.

- [58] G. Rebala, A. Ravi, and S. Churiwala, “(Artificial) Neural Networks,” *An Introd. to Mach. Learn.*, pp. 103–116, 2019, doi: 10.1007/978-3-030-15729-6_9.
- [59] A. K. Jain, J. Mao, and K. M. Mohiuddin, “Artificial neural networks: A tutorial,” *Computer (Long Beach, Calif.)*, vol. 29, no. 3, pp. 31–44, Mar. 1996, doi: 10.1109/2.485891.
- [60] Samer Hijazi, Rishi Kumar, and Chris Rowen, “Using Convolutional Neural Networks for Image Recognition,” San Jose, CA, USA, 2015. Accessed: Oct. 09, 2022. [Online]. Available: www.cadence.com.
- [61] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, “A Guide to Convolutional Neural Networks for Computer Vision,” *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, pp. 1–207, Feb. 2018, doi: 10.2200/S00822ED1V01Y201712COV015.
- [62] “Nutrition API - API Ninjas.” <https://api-ninjas.com/api/nutrition> (accessed Oct. 09, 2022).
- [63] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: An Overview,” AN OVERVIEW, 2020.
- [64] M. Junker, R. Hoch, and A. Dengel, “On the evaluation of document analysis components by recall, precision, and accuracy,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 717–720, 1999, doi: 10.1109/ICDAR.1999.791887.
- [65] “Loss Functions and Their Use In Neural Networks | by Vishal Yathish | Towards Data Science.” <https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9> (accessed Oct. 09, 2022).
- [66] “tf.keras.losses.CategoricalCrossentropy | TensorFlow v2.10.0.” https://www.tensorflow.org/api_docs/python/tf/keras/losses/CategoricalCrossentropy (accessed Oct. 09, 2022).
- [67] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, “A Comprehensive Survey of Loss Functions in Machine Learning,” *Ann. Data Sci.*, vol. 9, no. 2, pp. 187–212, Apr. 2022, doi: 10.1007/S40745-020-00253-5.
- [68] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, May 2019, doi: 10.48550/arxiv.1905.11946.

- [69] “Complete Architectural Details of all EfficientNet Models | by Vardan Agarwal | Towards Data Science.” <https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142> (accessed Oct. 14, 2022).
- [70] P. Pandey, A. Deepthi, B. Mandal, and N. B. Puhan, “FoodNet: Recognizing Foods Using Ensemble of Deep Networks,” *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1758–1762, Dec. 2017, doi: 10.1109/LSP.2017.2758862.
- [71] E. Aguilar, M. Bolaños, and P. Radeva, “Food Recognition Using Fusion of Classifiers Based on CNNs,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10485 LNCS, pp. 213–224, 2017, doi: 10.1007/978-3-319-68548-9_20/FIGURES/5.
- [72] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, “Food image recognition using very deep convolutional networks,” *MADiMa 2016 - Proc. 2nd Int. Work. Multimed. Assist. Diet. Manag. co-located with ACM Multimed. 2016*, pp. 41–49, Oct. 2016, doi: 10.1145/2986035.2986042.
- [73] N. Martinel, G. L. Foresti, and C. Micheloni, “Wide-slice residual networks for food recognition,” *Proc. - 2018 IEEE Winter Conf. Appl. Comput. Vision, WACV 2018*, vol. 2018-January, pp. 567–576, May 2018, doi: 10.1109/WACV.2018.00068.
- [74] P. Tripathi, “Transfer Learning on Deep Neural Network: A Case Study on Food-101: Food Classifier,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 5, pp. 229–232, 2021, Accessed: Oct. 09, 2022. [Online]. Available: <http://www.ijeast.com>.
- [75] R. Yunus *et al.*, “Special Section on Mobile Multimedia for Healthcare a Framework to Estimate the Nutritional Value of Food in Real Time Using Deep Learning Techniques,” doi: 10.1109/ACCESS.2018.2879117.