Redes neurais convolucionais para classificação e avaliação da maturação de frutos de café

Anagê Calixto Mundim Filho

Universidade Federal de Uberlândia
Faculdade de Computação
Bacharelado em Sistemas de Informação

Monte Carmelo - MG
2021

Anagê Calixto Mundim Filho

Redes neurais convolucionais para classificação e avaliação da maturação de frutos de café

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Murillo G. Carneiro
Coorientador: Cleyton B. Alvarenga

Monte Carmelo - MG
2021

Agradecimentos

Agradeço primeiramente ao meu pai e à minha mãe que me criaram e me ajudaram a chegar até aqui, me ensinando o essencial da vida, como humildade, bondade e gratidão. Toda minha família que me acompanhou e desejou sorte durante minha graduação. Amigos que foram essenciais nessa fase da minha vida e que me ajudavam sempre a passar os momentos difíceis e conturbados durante o curso.

Agradeço também a todos professores do curso que sempre deram suporte e ensinavam tudo o que sabia para ajudar os alunos, agradeço principalmente o meu orientador Murillo G. Carneiro que sempre me apoiou e me incentivou a participar de tudo, me orientando tanto no TCC quanto na minha Iniciação Científica, tutor do PETSIMC ao qual fiz parte no fim da graduação e agradeço novamente pelo carinho e esforço com os seus estudantes.

Obrigado a todos que me ajudaram chegar até o fim e me ensinaram a não desistir e continuar tentando e ter perseverança e não desanimar, pois cheguei até aqui. Aprendi bastante e foi uma parte importantíssima da minha vida.

Resumo

Neste trabalho foi desenvolvido um sistema baseado em aprendizado profundo voltado para classificação de frutos de café em seus cinco estádios de maturação bem como para auxiliar os cafeicultores na tomada de decisão sobre o momento mais apropriado para iniciar a colheita. Estes são problemas com elevado grau de dificuldade devido ao amadurecimento desuniforme dos frutos dessa planta. Nesse sentido, foi criada uma base de dados com mais de 10000 imagens de frutos de café capturadas em seus diferentes estádios de maturação, com uma parcela representativa delas analisadas e rotuladas por especialistas da área. Além disso, foram triplicadas tais imagens rotuladas a partir de aumento de dados e essa base de dados resultante para o treinamento da rede neural convolucional. Foram consideradas três arquiteturas de redes neurais convolucionais apropriadas para dispositivos móveis, MobileNet, MobileNet V2 e NasNetMobile, as quais foram treinadas e tiveram seu desempenho preditivo comparado entre si. Os resultados mostraram que a abordagem foi bem sucedida, uma vez que o melhor modelo obteve uma acurácia de 93% e F1 de 92% em um conjunto separado de imagens de teste. Um protótipo considerando esse modelo foi inclusive desenvolvido e disponibilizado para testes com cafeicultores.

Palavras-chave: Redes neurais convolucionais, Classificação do café, Aprendizado profundo, Android, Aprendizado supervisionado.

Abstract

In this work, we investigated deep learning methods to classify coffee trees maturation and assist farmers in the decision-making of the coffee harvest. These are little explored problems in the literature and they are difficult because the fruits ripen unevenly in the plant. In this sense, a database was created with more than 10,000 images of coffee fruits in their five stages of maturation, which was labeled by experts in the field. In addition, we tripled the classified images using data augmentation methods and a trained convolutional neural network over this data set. As these models are intended to be available in smartphones, we focused on the investigation of convolutional neural networks appropriate to this scenario, such as MobileNet, MobileNet V2, and NasNetMobile, which were trained and had their predictive performance compared. The results show that our best model obtained an accuracy of 93 % and F1 of 92 % in a separate set of test images. A prototype that implemented this model was created to allow coffee farmers to test our system.

Keywords: Convolutional neural networks, Coffee classification, Deep learning, Android, Supervised learning.

Lista de ilustrações

Lista de tabelas

Lista de símbolos

AM Aprendizado de Máquina

$ReLU$ Rectified Linear Unit

RNA Rede neural artificial

RNC Rede neural convolucional

RNP Rede neural profunda

Sumário

Capítulo 1

Introdução

Este trabalho está contextualizado em estudos de Aprendizado Profundo, especificamente em Redes Neurais Convolucionais (RNCs) e também na agricultura, especificamente o café. Relacionando ambas áreas que já são bem conhecidas por conta da Agricultura 4.0 na qual a tecnologia está cada vez mais envolvida, esse trabalho visa caracterizar a maturação do café através de redes convolucionais podendo inclusive recomendar a colheita ou não do café. Existe um problema na cafeicultura que é quando determinar a colheita, apesar de existirem métodos para isso na literatura, eles são limitados em vários aspectos. O trabalho levanta o seguinte questionamento: como o processamento de imagens pode ajudar o cafeicultor a fazer a colheita no momento ideal de forma a uma colheita de maior qualidade e sem desperdício?

1.1 Justificativa

Os métodos atuais não conseguem tirar o melhor proveito da colheita do cafeeiro, e diferentemente de outras culturas, é colhido com uma grande variação no teor de água dos frutos, o que torna o processo bastante trabalhoso. Além disso, as diversas floradas tornam a determinação do ponto ideal de colheita ainda mais complicado, propiciando desuniformidade de maturação com frutos em diferentes estádios. Diante disso, é possível notar como esse uso da tecnologia para a colheita do café pode ser importante e revolucionária, assim evitando perdas e melhorando o lucro das colheitas.

1.2 Objetivo geral e específico

O objetivo geral é a classificação e avaliação da maturação dos frutos o café por meio de um dispositivo móvel. Com isso, os seguintes objetivos específicos são vislumbrados:

- Desenvolver um algoritmo de processamento de imagens para trabalhar com diversos tipos de fotos de frutos de café para auxiliar na decisão do cafeicultor entre em colher

ou não colher, e também desenvolver um modelo multi-classe para classificação dos estádios de maturação dos cafeeiros.

- ❏ Investigar o desempenho de redes neurais convolucionais utilizadas para classificação Mobile, especificamente, as redes Mobile Net, Mobile Net V2 e NasnetMobile.
- ❏ Criar um padrão para retirar as fotos e organização do banco de dados de imagens, e colaboração com especialistas em café para rotular os diferentes estádios de maturação dos frutos de café.
- ❏ Criar um protótipo de aplicativo mobile com a implantação do melhor modelo para auxiliar na tomada de decisão do cafeicultor.

1.3 Hipótese

A hipótese investigada neste projeto é que se o uso de redes neurais convolucionais é uma alternativa eficiente do ponto de vista preditivo em medidas de acurácia e precisão e praticidade para automatizar o processo de caracterização da maturação do café.

1.4 Organização da dissertação

- ❏ Fundamentação teórica: Foi descrito o que é necessário para o entendimento do trabalho, onde é feito uma parte focada em café, redes neurais artificiais, convolucionais e profundas e focando em arquiteturas de redes neurais convolucionais. Há sessões descrevendo os trabalhos relacionados e as métricas para entendimento da avaliação da metodologia proposta.
- ❏ Materiais e métodos: são apresentadas as principais etapas do trabalho e como foram realizadas.
- ❏ Resultados experimentais: É apresentado e discutido o resultado dos modelos e apresentamos o protótipo desenvolvido a partir do melhor modelo.
- ❏ Conclusão: É apresentado as contribuições do trabalho, conclusões e os novos caminhos para seguir com a pesquisa e foi mencionado sobre publicação de artigo e premiação com este trabalho.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão apresentados a cultura do café bem como os principais conceitos, técnicas de colheita do café, conceitos, técnicas e algoritmos de aprendizado de máquina utilizados no desenvolvimento deste trabalho. Além disso, e será abordados os principais trabalhos encontrados na literatura relacionados à pesquisa.

2.1 Cultura do café

O agronegócio é um dos setores mais importantes da economia brasileira, que representa uma das principais fontes de economia, com o café tendo grande papel nisso (FRAGA, 1963). O Brasil teve sua safra total estimada em 49 milhões de sacas, número que representa 28% da produção mundial, incluindo as espécies arábica e canéfora. Tal produção contempla 34,5 milhões de sacas de café arábica, que equivalem a 33% da produção mundial, e 14,5 milhões de sacas de canéfora, as quais correspondem a 21% do volume físico produzido dessa espécie no mundo (EMBRAPA; T; S., 2019). Essa performance posiciona o Brasil em primeiro lugar na produção de café arábica e em segundo lugar na produção de café canéfora. O cafeeiro, que é o nome dado ao arbusto do café é da família *Rubiaceae*, sendo *Coffea* seu gênero, podendo se dividir em centenas de espécies, sendo as principais *Coffea arabica*; *Coffea canephora*; *Coffea liberica*; *Coffea racemosa*, destacando o *Coffea arabica* o principal a ser cultivado em Minas Gerais, São Paulo, Paraná e Bahia (CAMARGO, 1985). A espécie arábica se divide em inúmeras variedades. Ela é responsável por cerca de três quartos da produção mundial da bebida! O formato do grão é oval e na lavoura as plantas são sensíveis e mais fáceis de terem pragas e intempéries por conta do seu formato e precisam de mais cuidado para garantir sua qualidade, sua colheita é seletiva e, geralmente, feita à mão, ou ainda é utilizado a colheita mecanizada (CAMARGO, 1985).

2.1.1 Maturação do café

Em relação ao amadurecimento do café e as suas maturações, começando na sua formação que ocorre com o vingamento da flor até a completa maturação, o fruto de café passa por diversas fases, cada uma delas de importância decisiva na obtenção de grãos cereja sadios e graúdos, os frutos chumbinhos, permanecem no estádio de dormência durante aproximadamente seis semanas (MESQUITA et al., 2016).

Inicia-se, então, a formação da semente. Neste estádio, o crescimento é interrompido por certo período, no qual ocorre o endurecimento dela, etapa conhecida como granação. Nesta fase, também, a formação do fruto pode ser prejudicada por estiagens prolongadas, temperatura elevada, deficiência nutricional, com aparecimento de frutos chochos e mal granados, é um ponto que o cafezal deve ser muito bem cuidado para manter a qualidade do café (MESQUITA et al., 2016).

Posteriormente, os frutos do café ganham cores, começando do verde para o verde-cana e evoluindo para o cereja ou o amarelo conforme a cultivar. Os constituintes químicos atingem teores que conferem características peculiares de maturação completa, destacando-se a presença da mucilagem, que é um hidrogel solúvel e coloidal, parte integrante do fruto, composta de 85% de água e 15% de sólidos (MESQUITA et al., 2016). Depois da maturação, inicia-se a senescência do fruto e a seca gradativa da mucilagem. Neste período, podem ocorrer infecções microbianas influenciadas principalmente pela umidade relativa do ar, tanto em frutos na planta, tanto naqueles já caídos, que constituem a parcela denominada varrição (CAMARGO, 1985). Na Figura 1 é possível ver com fotos os seus estágios de maturação, essas fotos foram usadas para treino do algoritmo e são de nossa autoria.

Figura 1 – Fotos de estádios de maturação de frutos do café. (Fonte: autor)

2.1.2 Estratégias para a colheita

Em relação à determinação do ponto de colheita, isto é, a avaliação do grau de maturação do café em cada talhão, devendo ser feita próximo à colheita. Ela possui como finalidade orientar na tomada de decisão de iniciar ou não a colheita, como também definir com melhor precisão e clareza por qual talhão deverá iniciar-se a colheita. Esse aspecto é muito importante pois certas regiões do cafezal são mais apropriadas para iniciar a colheitas, especialmente por conta a colheita um elevado número de café cereja (BARTHOLO; GUIMARÃES, 1997).

Basicamente, existem dois tipos de colheita do café: manual e mecânica. A colheita mecânica possui a derriça total, onde, nela o trabalhador vai por meio da vibração fazer os frutos do talhão do café cair e após isso coletá-los. Importante relatar que são colhidos todos os frutos do talhão como acontece na própria colheita mecânica, impedindo assim que os frutos verdes e verde-cana se desenvolvam. Ademais, o trabalhador também deve evitar danos excessivos aos ramos e às folhas, para preservar a produção seguinte. Outra forma de realizar a colheita manual é a colheita seletiva que realiza a derriça no pano somente dos frutos cereja. Os frutos verdes são colhidos mais adiante quando estiverem maduros. Nesse caso, poderão ser necessárias duas a três colheitas por planta ou talhão devido à desuniformidade existente na maturação do café. Esta é influenciada por diversos fatores como clima, altitude, número de floradas, adensamento da lavoura, entre outros. Por ser uma operação que necessita de maior mão-de-obra, é mais empregada por alguns cafeicultores com o objetivo de se obter um café superior, uma vez que cereja é a matéria prima adequada (MESQUITA et al., 2016).

A colheita mecânica veio juntamente com a expansão da cafeicultura e com a implementação de lavouras em regiões de topografia mais adequadas à mecanização. Então foram desenvolvida várias máquinas, equipamentos e estratégias, visando a derriça e o recolhimento mecânico do café, com maior rendimento, menor custo e em menor tempo, com isso gerando muita vantagem para o agricultor, porém precisa preservar a qualidade do produto na realização da colheita no momento mais adequado da maturação(SILVA et al., 2013). Com a terceirização da colheita mecanizada está possibilitando a adoção desta prática para pequenas propriedades, pois antes somente eram usadas em propriedades de larga escala, devido principalmente ao preço elevado, tanto de compra quanto de manutenção. Existem dois tipos de colhedoras: automotrizes ou tracionadas. Essas máquinas, através de sistemas hidráulicos, com varetas vibratórias, fazem o trabalho de derriça, recolhimento, abanação e descarga do café na forma ensacada ou a granel. As automotrizes, como o nome sugere, têm propulsão própria. Já as tracionadas necessitam ser acopladas a um trator através da barra de tração e da tomada de força, assim conseguem colher os talhões com sua própria vibração (ALVES et al., 1999).

2.2 Aprendizado profundo

Será explicado nessa seção sobre aprendizado de máquina, sendo ele supervisionado e não-supervisionado, redes neurais artificiais (RNAs), redes neurais profundas (RNPs), redes neurais convolucionais (RNCs), arquiteturas de redes neurais convolucionais fundamentais e também arquiteturas voltadas para dispositivos móveis.

2.2.1 Aprendizado de máquina (AM)

No início da era computacional foi criado algoritmos para realização de tarefas simples e complexas. Algumas tarefas, no entanto, não possuem um algoritmo definido para serem executadas. Como exemplo, é possível citar o reconhecimento facial, separação de e-mail legítimos de spam e classificação de imagens que é o caso de uso (ALPAYDIN, 2020)

Aprendizado de Máquina é uma área da Inteligência Artificial que tem como objetivo o desenvolvimento de técnicas computacionais capazes de aprender e adquirir conhecimento a partir de dados de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores (MONARD; BARANAUSKAS, 2003). Existem três paradigmas principais no AM, esses podem ser definidos da seguinte maneira:

1. Resumida na 4, o aprendizado supervisionado é fornecido ao algoritmo de aprendizado, um conjunto de exemplos de treinamento para os quais o rótulo da classe associada é conhecido. No geral, cada exemplo é descrito por um vetor de valores de características, ou atributos, e o rótulo da classe associada. O objetivo do algoritmo de aprendizado é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados, ou seja, exemplos que não tenham o rótulo da classe. Para rótulos de classe discretos, esse problema é conhecido como **classificação**(MONARD; BARANAUSKAS, 2003). Este é o tipo de aprendizado que será utilizado.

2. Já no aprendizado não-supervisionado os dados de entrada do modelo são descritos exclusivamente por suas características, isto é, sem rótulos. O próprio modelo analisa as informações da base para separá-la em grupos de acordo com algum critério de similaridade. A Figura 3 mostra o esquema de como funciona esse aprendizado, os dados são interpretados e processados, e o algoritmo os agrupa de maneira automática.

3. No aprendizado por reforço é usado o treinamento de modelos de AM para tomar uma sequência de decisões. O agente aprende a atingir uma meta em um ambiente incerto e potencialmente complexo, porém ainda pode se treinar em ambientes extremamente simples e determinísticos. No aprendizado por reforço, o sistema de

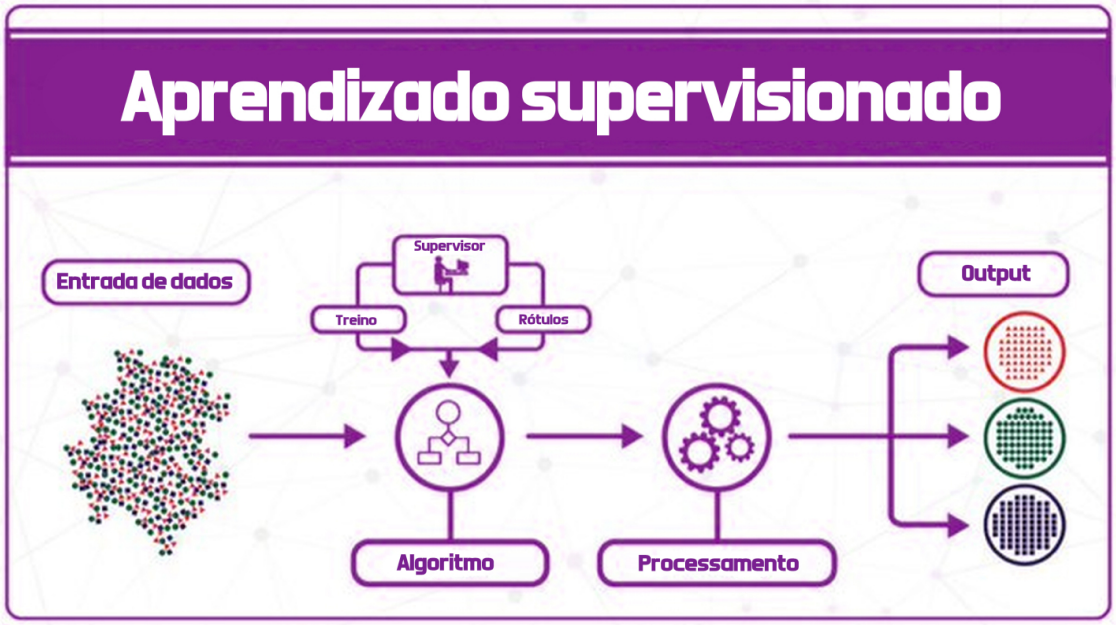

Figura 2 – Ilustração do funcionamento do aprendizado supervisionado - (adaptado de (PARIKH, 2018).

Figura 3 – Ilustração do funcionamento do aprendizado não-supervisionado - (adaptado de (PARIKH, 2018).

inteligência artificial enfrenta uma situação. O computador utiliza diferentes abordagens, inclusive baseadas em tentativa e erro, para encontrar uma solução para o problema. Para que o algoritmo aprenda o que o programador deseja, o agente recebe recompensas ou penalidades pelas ações que executa. É bastante usada em games, robótica e várias outras aplicações envolvendo aprendizado de agentes autônomos.

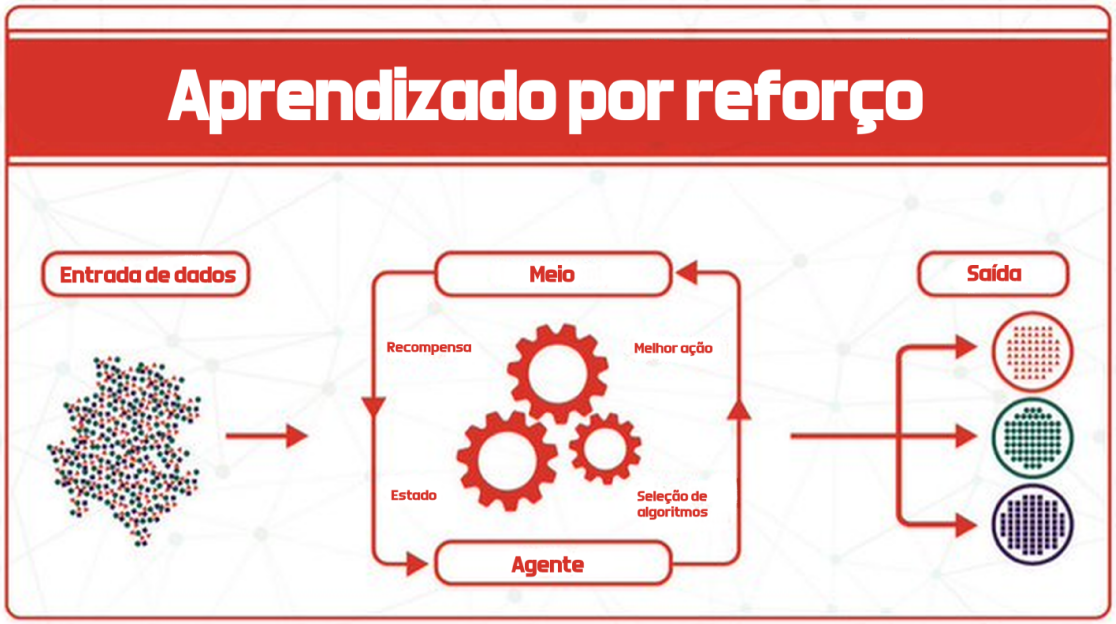

Figura 4 – Ilustração do funcionamento do aprendizado por reforço - (adaptado de (PARIKH, 2018).

2.2.2 Redes Neurais Artificiais (RNA)

Uma RNA é um sistema conexionista composto de neurônios artificiais ou nós (HOPFIELD, 1982). Estes neurônios tentam funcionar e ter o mesmo uso dos neurônios não-artificiais que são células do tecido nervoso dos seres vivos e funcionam, basicamente, da seguinte maneira: são ativados por impulsos elétricos, e trocam informações com outros neurônios sempre que são ativados (YEGNANARAYANA, 2009). Esses sistemas conseguem aprender a realizar tarefas considerando exemplos, geralmente sem serem programados com regras específicas da tarefa. O objetivo original da RNA era resolver problemas da mesma maneira que um cérebro humano resolveria. Mas, com o tempo, a atenção passou a ser executar tarefas específicas, levando a desvios da biologia (HOPFIELD, 1982). As RNAs têm sido usadas em diversas tarefas, incluindo visão computacional (KHAN et al., 2018), tradução automática (AKMELIAWATI; OOI; KUANG, 2007), engenharia (SHAHIN; JAKSA; MAIER, 2001), gestão de riscos de cartão de crédito (PACELLI; AZZOLLINI et al., 2011), filtragem de spam em redes sociais (JAIN; SHARMA; AGARWAL, 2019), jogos (CHAROENKWAN; FANG; WONG, 2010), diagnósticos médicos (AMATO et al., 2013), o uso dessas redes estão aumentando a cada descoberta de novos algoritmos e novas aplicações.

2.2.3 Redes Neurais Profundas (RNP)

Uma RNP é uma RNA com múltiplas camadas entre a entrada e a saída dos dados (SCHMIDHUBER, 2014). A RNP encontra a manipulação matemática correta para

transformar a entrada na saída, seja uma relação linear ou não linear. A informação de entrada se move pelas camadas, calculando a probabilidade de cada saída. Por exemplo, uma RNP treinada para reconhecer raças de cachorros examinará a imagem fornecida e calculará a probabilidade de o cão na imagem ser uma determinada raça. O usuário pode revisar os resultados e selecionar quais probabilidades a rede deve exibir (acima de um determinado limite, etc.) e retornar o rótulo proposto. Cada manipulação matemática é considerada uma camada e a RNP possui muitas camadas, daí o nome redes "profundas". A Figura 5 mostra bastante da complexidade de uma RNP.

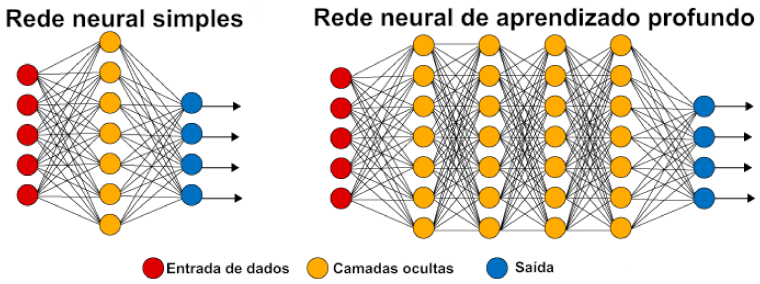

Figura 5 – Ilustração da complexidade de uma rede neural profunda comparada a uma rede neural simples. Adaptado de Favio Vázquez 2017.

As RNP podem modelar relacionamentos não lineares e complexos. Suas arquiteturas geram modelos composicionais em que o objeto é expresso como uma composição em camadas. As camadas extras permitem a composição de recursos das camadas anteriores, modelando potencialmente dados complexos com menos unidades do que uma rede rasa com desempenho semelhante (BENGIO, 2009). As RNPs geralmente são redes de *feedforward*, onde, os dados fluem da camada de entrada para a camada de saída sem retorno. A princípio, a RNP cria um mapa de neurônios virtuais e atribui valores numéricos aleatórios, ou pesos, às conexões entre eles. Os pesos e entradas são multiplicados e normalmente retornam uma saída entre 0 e 1. Se a rede não reconhece com precisão um padrão específico, um algoritmo ajustaria os pesos. Dessa forma, o algoritmo pode tornar determinados parâmetros mais influentes, até determinar a manipulação matemática correta para processar completamente os dados. Como exemplo de uso das RNP temos as RNCs, que é a RNP normalmente aplicada em imagens, reconhecimento de fala e até mesmo para o domínio de textos (RICHARDSON; REYNOLDS; DEHAK, 2015).

2.3 Redes neurais convolucionais (RNC)

Uma RNC é um dos algoritmos de aprendizado profundo, que tem como dados de entrada uma imagem. Elas conseguem identificar e diferenciar aspectos dos dados de entrada, fazendo o mapeamento e transformação espacial dos mesmos. Estas são muito utilizadas no processamento de imagens e muito importante na área de visão computacional. Diferentemente de redes neurais simples, que tem vetores de atributos como entrada,

a RNC trata imagens como atributos multidimensionais, sendo consideradas a altura, largura e a profundidade e isso representa os canais de cores (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).A RNC recebe uma imagem de entrada, a processa e classifica a mesma em determinadas categorias (por exemplo, Cão, Gato, Peixe, etc). Os computadores recebem uma imagem de entrada como uma matriz de pixels e isso depende da resolução da imagem. Com base na resolução da imagem, temos h x w x d (h = Altura, w = Largura, d = Dimensão). Por exemplo, uma imagem de matriz 6 x 6 x 3 da matriz de RGB (3 canais de cores RGB) e uma imagem de matriz 4 x 4 x 1 da matriz da imagem em escala de cinza, a Figura 6 se refere a uma matriz RGB de entrada 6x6x3.

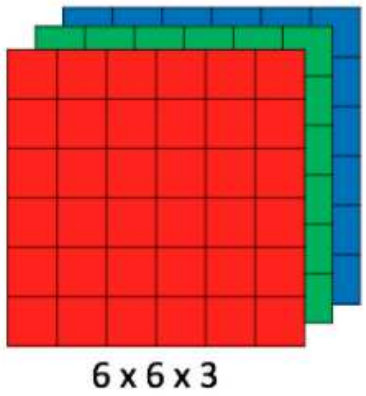

Figura 6 – Ilustração de uma matriz RGB de 3 dimensões e seus vetores

2.4 Funcionamento da Rede Neural Convolucional

Modelos de RCN de aprendizado profundo são usados para treinar algoritmos com milhares de imagens e testá-los imagens desconhecidas. Cada imagem de entrada passa por uma série de processos, como: convolução com filtros (*Kernals*), *Pooling*, camadas totalmente conectadas, $ReLU$, todas serão explicadas em tópicos abaixo.

- **Camadas de convolução:** É por onde os atributos da imagem de entrada serão extraídos a partir das operações de convolução de matrizes. Uma matriz menor serve como filtro,onde será lido todos os pixels e com isso produzir uma matriz de dimensões menores que a matriz de entrada. Na Figura 7 é mostrado como isso é feito.

  Com isso, é feito o cálculo do valor da matriz que será convolucionada da seguinte forma: $(1 \cdot 1) + (1 \cdot 0) + (1 \cdot 1) + (0 \cdot 0) + (1 \cdot 1) + (1 \cdot 0) + (0 \cdot 1) + (0 \cdot 0) + (1 \cdot 1) = 4$. O valor do próximo elemento da primeira linha da matriz é calculado da mesma forma, e assim até terminar de percorrer toda a matriz, a Figura 8 mostra como é feito isso passo a passo.

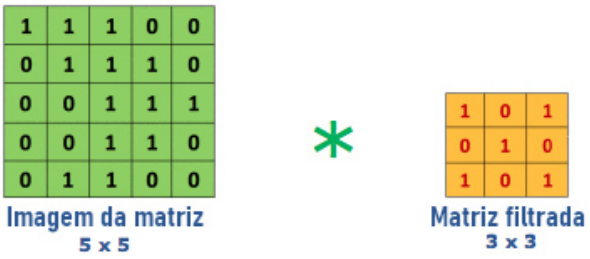

Figura 7 – Ilustração de uma matriz da imagem multiplicando pelo núcleo (kernel)

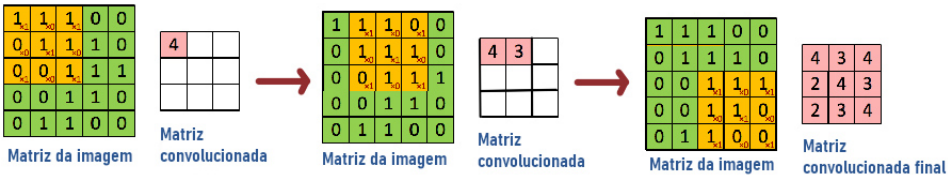

Figura 8 – Ilustração da construção de uma Matriz convolucional se formando por um a matriz de imagem

- **Stride:** É o numero de cada passo que será dado pelo filtro da convolução. Quando o Stride é 1, movemos os filtros 1 pixel de cada vez. Quando o Stride é 2, movemos os filtros 2 pixels por vez e assim por diante. Da mesma forma, para qualquer número inteiro S > 0, um Stride de S faz com que o filtro seja convertido S unidades por vez por saída. Os campos receptivos se sobrepõem menos e o volume de saída resultante tem dimensões espaciais menores quando o comprimento do Stride é aumentado. Na Figura 9 é apresentado um exemplo sobre como esse filtro se comporta na matriz imagem.

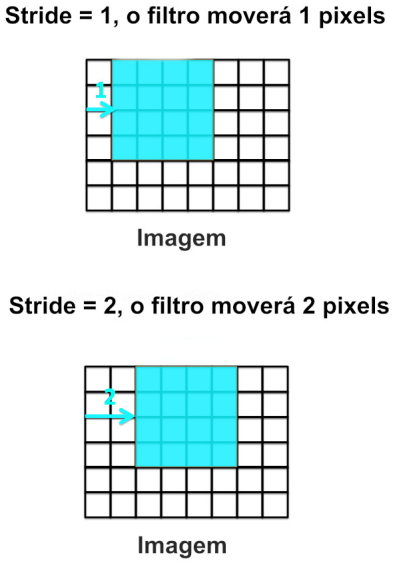

Figura 9 – Ilustração do funcionamento *Stride*, ou passos.

- ***Padding***: é usado quando o filtro não se encaixa perfeitamente na imagem e precisamos de algo mais ajustar isso, podemos fazer uso do zero-padding que é preencher a matriz imagem com 0 nos valores negativos, ou então, deletar a parte que não se encaixou no filtro e usar o que é necessário da imagem. Ele também é usado para deixar a imagem que foi convoluída com o mesmo tamanho da original.

- **Não Linearidade (ReLU)**: função de ativação Unidade Linear Retificada (Rectified Linear Unit - *ReLU*) é aplicada sempre após a convolução da imagem,com o propósito de deixar a convolução sem números negativos (NAIR; HINTON, 2010). A saída dessa função é: f(x) = max(0,x) com isso os números positivos se manterão e e os negativos ficarão zerados 0.

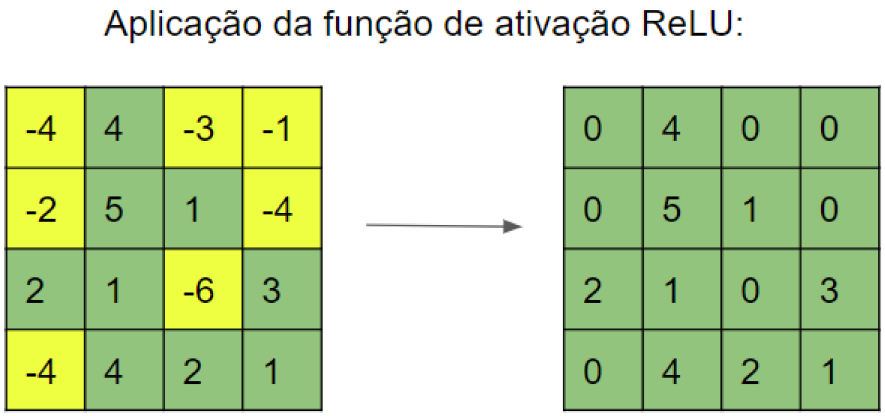

Figura 10 – Exemplo para ilustrar o funcionamento da função de ativação Rectified Linear Unit (ReLU).

- **Pooling** - Camada de agrupamento: O *pooling* é quando a imagem é redimensionada para outro tamanho, sendo obtida das camadas anteriores. O mais popular tipo utilizado é o Max Pooling, com a Figura 11 podemos ver como ele funciona. Suponha que se pretenda redimensionar a camada em uma proporção de 2. Isso significa que a altura e a largura da sua imagem serão metade da original. Portanto, se precisa compactar a cada 4 pixels e mapeá-lo para um novo pixel único sem perda de dados "importantes" dos pixels ausentes. O Pool máximo é feito usando o maior valor desses 4 pixels. Portanto, um novo pixel representa 4 pixels antigos, usando o maior valor desses 4 pixels. Isso é feito para cada grupo de 4 pixels em toda a imagem. Isso ajuda muito a reduzir o tamanho da matriz original, que faz ficar mais "leve" e sem perder informações importantes. É também muito usado o AVG Pooling, que utiliza a média da soma dos números para ser feito o procedimento.

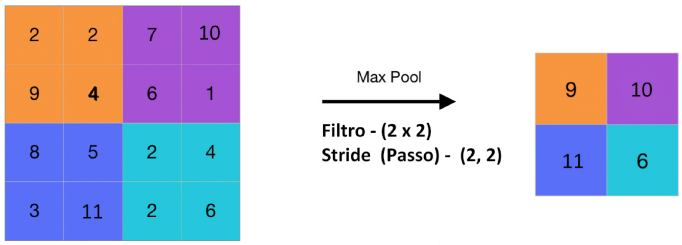

Figura 11 – Exemplo para ilustrar o funcionamento do Pool máximo (Max Pooling)

2.4.1 Camada totalmente conectada

É a camada que chamamos de **Fully connected layer** ou FC, na qual achatamos a matriz em um vetor. Como podemos ver na Figura 12, após o achatamento os vetores são passados nessa camada totalmente conectada para criarmos um modelo, parecendo uma RNA. Por fim, temos uma função de ativação como *softmax* ou *sigmoid* para classificar as saídas como gato, cachorro, carro, caminhão etc.

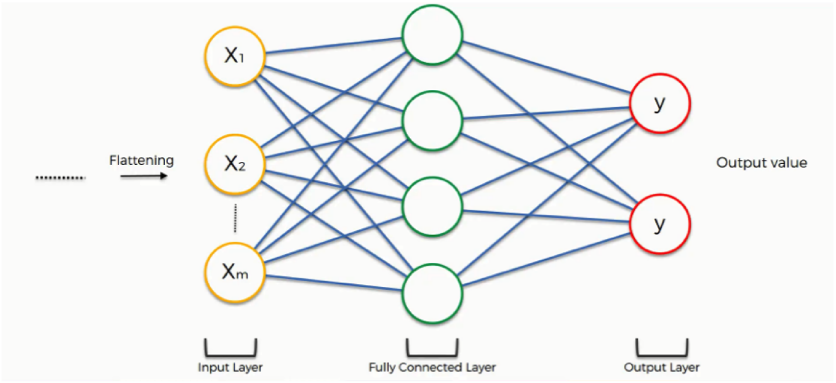

Figura 12 – Exemplo de uma camada totalmente conectada (Fully connected layer. FC)

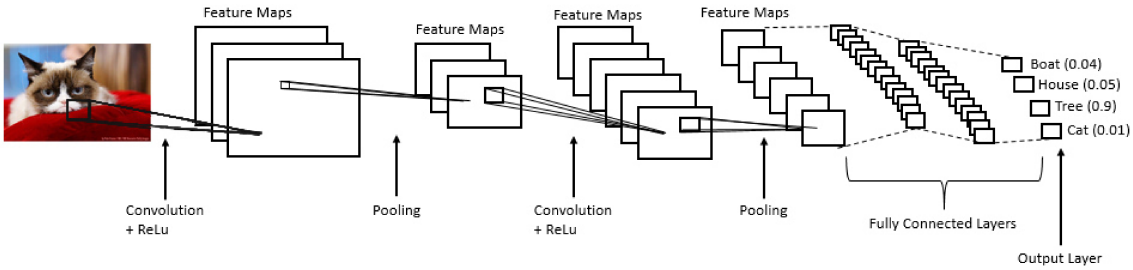

Figura 13 – Processamento da imagem até sua classificação dentro de uma RNC

Na Figura 13 podemos ver como o processamento da imagem é feita por todas essas camadas que foram explicadas anteriormente, diminuindo cada vez mais e sendo processada diversas vezes pelo algoritmo até ele tomar uma decisão do que está naquela imagem.

2.4.2 Arquiteturas de redes neurais convolucionais

Essa sessão detalha alguns tipos de arquiteturas das RNC, bem como pontos importantes para compreender mais sobre essas redes e seus parâmetros. Apresentaremos as redes LeNet, VGG-16 e AlexNet.

2.4.3 LeNet

Yann LeCun, Leon Bottou, Yosuha Bengio e Patrick Haffner propuseram uma arquitetura de RNC para reconhecimento de caracteres manuscritos e impressos à máquina nos anos 90, que eles chamaram de LeNet-5 (LECUN et al., 2015). Como a Figura 14 nos mostra como que funciona esta arquitetura, vemos que ela é direta e simples de entender, por isso é usada principalmente como primeiro passo para o ensino da RNC.

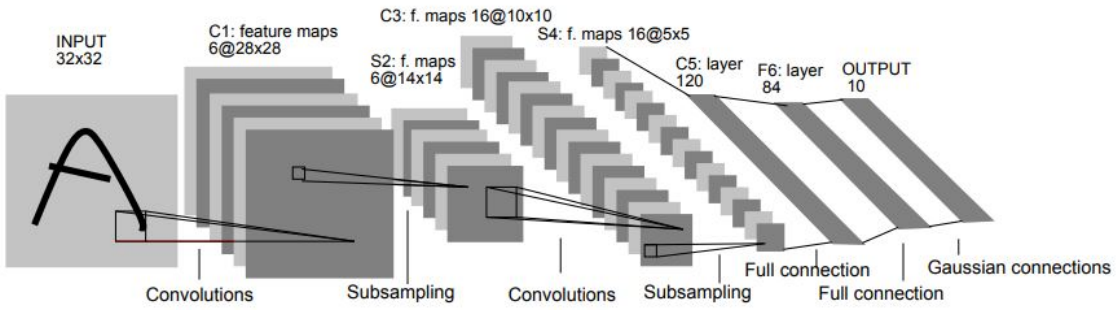

Figura 14 – Ilustração da arquitetura da RNC LeNet-5 (Autor: (LECUN et al., 2015))

A arquitetura LeNet-5 consiste em 2 partes de camadas convolucionais e camadas de **AVG pooling** que significa o pooling pegando a média ao invés do maior valor que foi o que vimos no Max Pooling, duas camadas totalmente conectadas (FC) e no fim um softmax para classificar. A seguir será explicado com mais detalhes como funciona essa arquitetura.

- ❏ Primeira camada: A entrada para LeNet-5 é uma imagem em escala de cinza de tamanho 32x32 que passa pela primeira camada convolucional com 6 filtros com tamanho 5x5 e *Stride* de 1. As dimensões da imagem mudam de 32x32x1 para 28x28x6.Esse processo está explicado na Figura 15.

- ❏ Segunda camada: Em seguida, o LeNet-5 aplica uma camada de AVG Pooling ou subamostragem com um tamanho de filtro 2x2 e um *Stride* de 2. As dimensões da imagem resultantes serão reduzidas para 14x14x6. Esse processo está explicado na Figura 15.

- ❏ Terceira camada: Teremos uma segunda camada convolucional com 16 filtros de tamanho 5x5 e Stride 1, apenas 10 dos 16 filtros são conectados a 6 filtros da

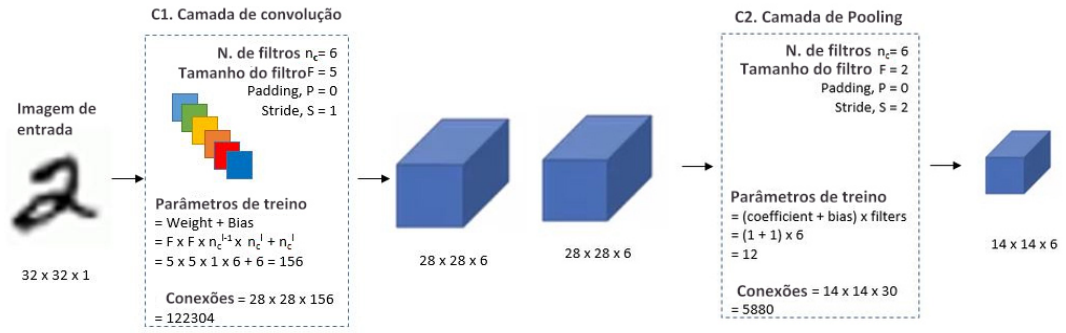

Figura 15 – Funcionamento da LeNet: Processo da Camada de convolução e de Pooling (Adaptado de Muhammad Rizwan)

camada anterior. Esse processo está explicado na Figura 16. Na qual cada coluna indica qual filtro em C2 é combinado pelas unidades em um filtro específico de C2.

- ❏ Quarta camada: Ocorrerá novamente uma AVG Pooling layer, com filtro 2x2 e Stride 2. Essa camada é exatamente igual a segunda camada que foi usado o AVG Pooling, a saída será de 5x5x16. Esse processo está explicado na Figura 16.

- ❏ Quinta camada: É uma camada convolucional totalmente conectada, com 120 filtros, cada um do tamanho 1x1. Cada uma das 120 unidades em C5 está conectada a todos os 400 nós (5x5x16) na quarta camada S4. Esse processo está explicado na Figura 17.

- ❏ Sexta camada: sexta camada é uma camada totalmente conectada (F6) com 84 unidades. Esse processo está explicado na Figura 17.

- ❏ Por fim, existe uma camada de saída *softmax* X totalmente conectada com 10 valores possíveis correspondentes aos dígitos de 0 a 9.

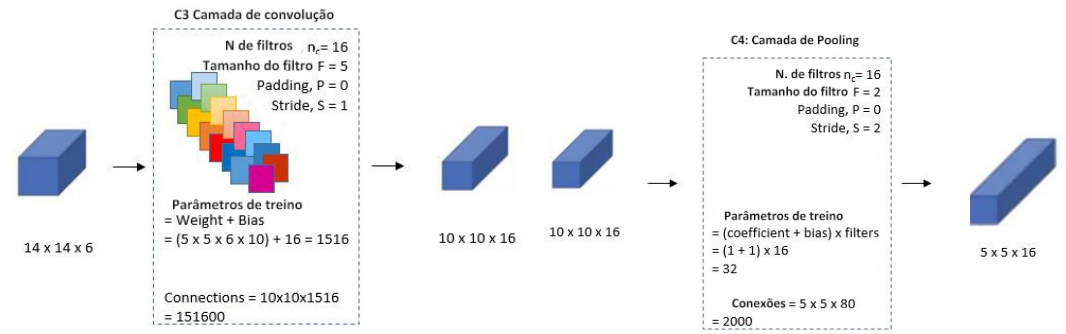

Figura 16 – Funcionamento da LeNet: Camada de convolução renovada e pooling novamente (Adaptado de Muhammad Rizwan)

Resumindo o funcionamento da LeNet-5 com essa tabela realizando cada uma das etapas do que aconteceu com a entrada da imagem até a saída e decisão da arquitetura:

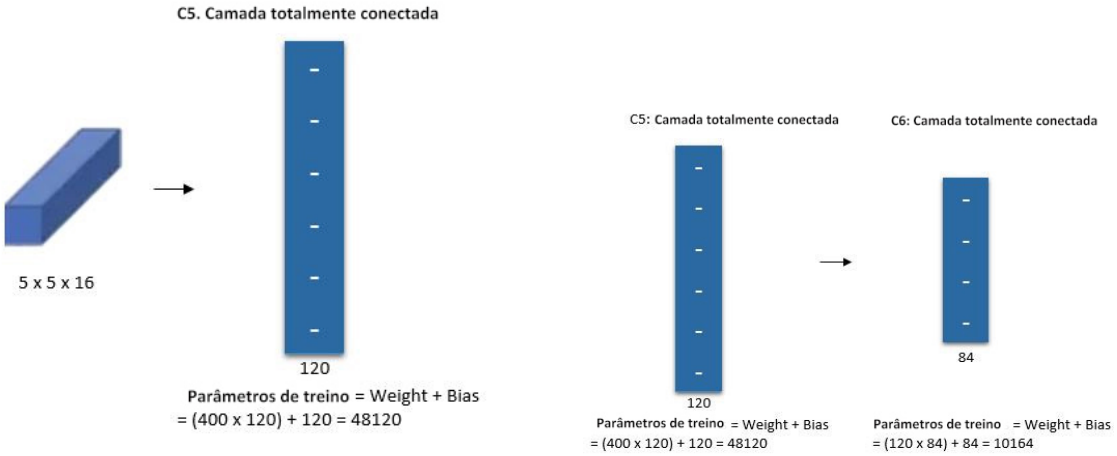

Figura 17 – Funcionamento da LeNet: Camada de convolução renovada e pooling novamente (Adaptado de Muhammad Rizwan)

Camadas		Filtros	Tamanho	Tamanho do Kernel	Stride	Ativação
Input	Imagem	1	32 x 32	-	-	-
1	Convolução	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolução	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolução	120	1x1	5x5	1	tanh
6	Tot. Conect.	-	84	-	-	tanh
Output	Tot. Conect.	-	10	-	-	softmax

Tabela 1 – Camadas da RNC LeNet-5

2.4.4 AlexNet

AlexNet é o nome de uma RNC, projetada por Alex Krizhevsky, e publicada com Ilya Sutskever e o orientador de doutorado de Krizhevsky, Geoffrey Hinton. Essa arquitetura é bastante semelhante a LeNet de Yann Lecun, mas é mais profunda, com mais filtros por camada e com camadas convolucionais empilhadas. Eles treinaram sua rede na IMAGE NET, que possui 1,2 milhão de imagens de alta resolução em 1000 classes diferentes, com 60 milhões de parâmetros e 650.000 neurônios (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

A arquitetura AlexNet consiste em 5 camadas convolucionais, algumas das quais são seguidas por camadas de Max Pooling e, em seguida, três camadas totalmente conectadas e, finalmente, um classificador. Para melhor compreensão das camadas e das etapas que serão explicadas abaixo, acompanhe as etapas pela Figura 18.

- ❏ Primeira etapa:

  A entrada para AlexNet é uma imagem RGB de 227x227x3 que passa pela primeira camada convolucional com 96 filtros de recursos com tamanho 11x11 e um Stride 4.

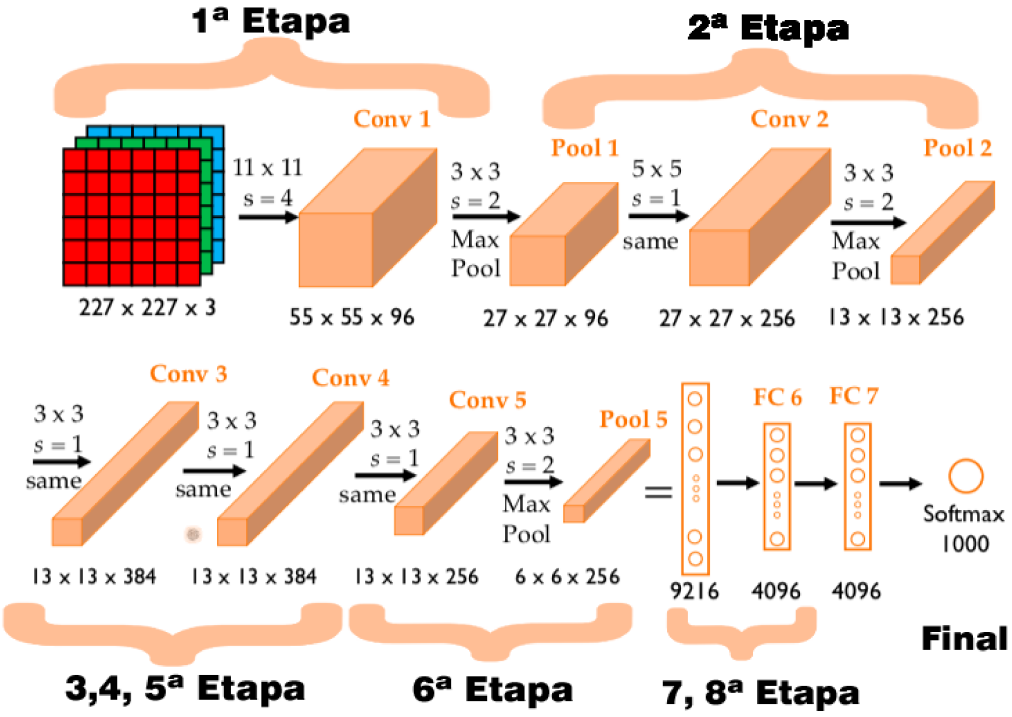

Figura 18 – Ilustração do funcionamento da RNC AlexNet (Adaptado de Anh H. Reynolds)

As dimensões da imagem mudam para 55x55x96. Em seguida, a AlexNet aplica a camada de Max Pooling com um tamanho de filtro 3x3 e um Stride 2. As dimensões da imagem resultantes serão reduzidas para 27x27x96.

- Segunda etapa: Em seguida, há uma segunda camada convolucional com 256 filtros de recurso com tamanho 5x5 e com padding = 2. Depois, há novamente uma camada de MaxPool com tamanho de filtro 3x3 e um Stride 2. Essa camada é igual à segunda camada, exceto que possui 256 filtro de recursos, portanto a saída será reduzida para 13x13x256.

- Terceira, quarta e quinta etapas:

  A terceira, quarta e quinta camadas são camadas convolucionais com tamanho de filtro 3x3 e com Stride 1. Os dois primeiros usaram 384 filtros de recursos, enquanto o terceiro usou 256 filtros. As três camadas convolucionais são seguidas por uma camada de Max pooling com tamanho de filtro 3x3, um Stride de 2 e Padding de 1 e possui 384 mapas de características passando para 256 na próxima interação.

- Sexta etapa: A saída da camada convolucional é achatada através de uma camada totalmente conectada com mapas de recursos 9216, sendo uma camada convolucional de 6x6x256 que foi gerada com o Max Pooling 3x3 e Stride de 2.

- Sétima e oitava etapa: Duas camadas totalmente conectadas de 4096 vetores cada.

- ❏ Camada de saída: Finalmente, há uma camada de saída softmax X com 1000 valores possíveis.

Com mais detalhes sobre como funciona resumidamente na Tabela 2.

Camadas		Filtros	Tamanho	Tamanho do Kernel	Stride	Ativação
Input	Imagem	1	227x227x3	-	-	-
1	Convolução	96	55x55x96	11x11	4	relu
	Max Pooling	96	27x27x96	3x3	2	relu
2	Convolução	256	27x27x256	5x5	1	relu
	Max Pooling	256	13x13x256	3x3	2	relu
3	Convolução	384	13x13x384	3x3	1	relu
4	Convolução	384	13x13x384	3x3	1	relu
5	Convolução	256	13x13x256	3x3	1	relu
	Max Pooling	256	6x6x256	3x3	2	relu
6	Tot. Conect.	-	9216	-	-	relu
7	Tot. Conect.	-	4096	-	-	relu
8	Tot. Conect.	-	4096	-	-	relu
Output	Tot. Conect.	1000	-	-	-	softmax

Tabela 2 – Camadas da RNC AlexNet

2.4.5 VGG

A AlexNet foi lançada em 2012 e foi um avanço que revolucionou o aprendizado profundo e melhorou nas tradicionais redes neurais convolucionais (RNCs), após isso foi criado o VGG, desenvolvido por Simonyan e Zisserman para competição da ILSVRC (Large Scale Visual Recognition Challenge) Desafio de reconhecimento visual em grande escala, realizado em 2014. Enquanto os derivados anteriores da AlexNet focavam em tamanhos e avanços menores na primeira camada convolucional, o VGG aborda outro aspecto muito importante das RNCs, que é a profundidade da imagem. A seguir explicarei mais sobre a arquitetura, como foi feito no artigo original (SIMONYAN; ZISSERMAN, 2014).

- ❏ A entrada para a camada 1 é de imagem fixa de tamanho 224 x 224 RGB.
- ❏ A imagem é passada através de uma pilha de camadas convolucionais, onde os filtros foram usados com um campo receptivo muito pequeno 3x3.
- ❏ A convolução é fixada em 1 pixel, dessa forma, o preenchimento espacial da camada de entrada de convolução, é de tal modo, que a resolução espacial é preservada após a convolução.
- ❏ A cada final da execução de uma pilha de convolução é realizado a camada de Max-pooling, sendo utilizada 5 vezes no total, sendo executada numa janela de 2x2.

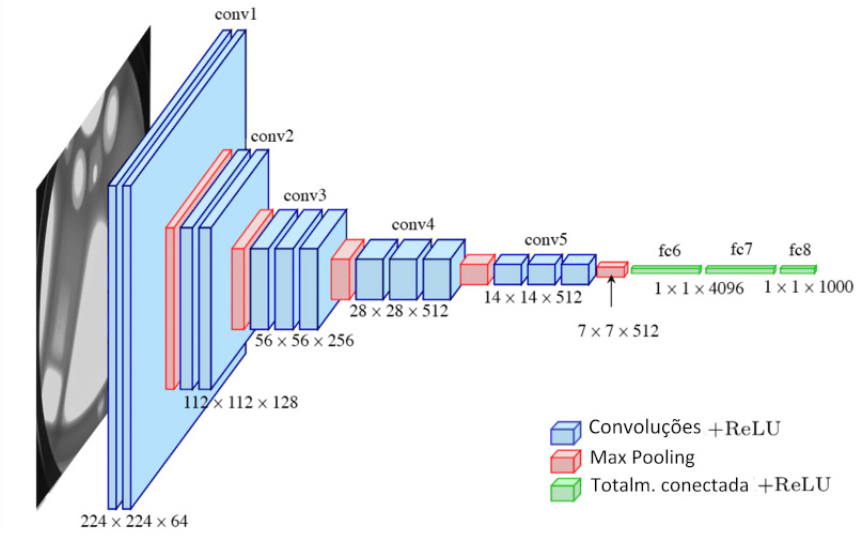

Figura 19 – Ilustração do funcionamento da RNC VGG (Adaptado de Max Ferguson)

- Após essas execuções, são acionadas 3 camadas Totalmente Conectadas (FC), os dois primeiros possui 4096 canais de cada, a terceira possui 1000 canais, que é 1 para cada classe a ser classificada. Todas as camadas FC possuem camadas ocultas ReLU, por fim é utilizado uma função de ativação Soft-max para saída do algoritmo.

As configurações da ConvNet estão descritas na Figura 20. As redes são referidas aos seus nomes (A-E). Todas as configurações seguem o design genérico presente na arquitetura e diferem apenas em profundidade: de 11 camadas de peso na rede A (8 camadas conv.) a 19 camadas de peso na rede E (16 camadas conv.). A largura de convoluções de camadas é bem pequeno, começando em 64 na primeira camada e aumentando em um fator de 2 após cada camada de Max pooling, até atingir 512. Aí podemos ver a diferença entre a VGG 16 e a 19.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 20 – Resumo de camadas da RNC VGG (Adaptado de Max Ferguson)

2.4.6 MobileNet, MobileNetV2 e NasNetMobile

As MobileNets podem trabalhar com muitas tarefas, incluindo detecção de objetos, atributos de face e localização geográfica em grande escala. MobileNets são baseados em uma arquitetura simplificada que usa convoluções separáveis em profundidade para construir RNPs que ocupam menos espaço (HOWARD et al., 2017). Os autores apresentam dois hiper parâmetros globais simples que trocam de forma eficiente a latência e a precisão. Esses hiper-parâmetros permitem que o construtor de modelo escolha o modelo de tamanho certo para seu aplicativo com base nas restrições do problema. Já a MobileNetV2 proposta pelo Google baseia-se nas ideias da MobileNet, usando convolução separável com inteligência de profundidade como blocos de construção eficientes (SANDLER et al., 2018). No entanto, a MobileNetV2 apresenta dois novos recursos à arquitetura: 1) linear bottleneck, onde a última convolução de um bloco residual tem uma saída linear antes de ser adicionada às ativações iniciais, e 2) conexões curtas entre os bottlenecks.

Podemos ver a arquitetura de ambas e também como são feitas suas convoluções na

Figura 21, com isso vemos que a arquitetura da MobileNet V1 é maior porém cada entrada de convolução não executa tantos passos quanto às convoluções da MobileNet V2 que é mais robusta como podemos ver na parte de convoluções da imagem citada anteriormente.

Já a NasNet Mobile ou *Mnasnet* funciona de uma forma diferente, os desenvolvedores dela no Google Brain buscaram criar uma RNC otimizada utilizando aprendizado por reforço na sua arquitetura. Ela utiliza esse aprendizado por reforço para pesquisar em um espaço de configurações de rede neural, encontrando um equilíbrio entre precisão e latência levando em conta o aparelho e os modelos escolhidos para o treinamento. A NAS foi usado com conjuntos de dados padrão como CIFAR10 e ImageNet para otimizar RNCs para tamanhos diferentes. A versão reduzida e preparada para o mobile alcança performances melhores e bem mais rápidas que a original NasNet, se tornando uma boa alternativa para arquiteturas mobile (ZOPH et al., 2018).

Arquitetura MobileNetV1

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Arquitetura MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Convoluções MobileNet e MobileNetV2

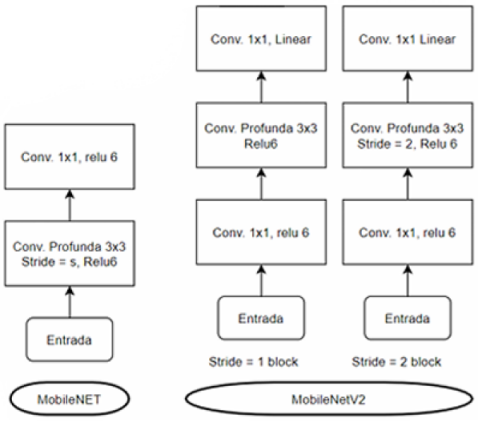

Figura 21 – Comparação entre arquiteturas e convoluções da MobileNet e MobileNetV2

O motivo de serem escolhidas foram por conta de seu desempenho, tamanho e eficiência em dispositivos móveis. Isso, é por conta da *depthwise separable convolution* (KAISER; GOMEZ; CHOLLET, 2017) que foi desenvolvida primeiramente no modelo da Mobile Net, com ela é possível fazer convoluções de forma mais rápida e eficiente por conta da sua separação de dois processos para realizar a convolução, esses processos são o . Essa convolução necessita notavelmente bem menos memória e processamento computacional

		Classe verdadeira	
		Positivo	**Negativo**
Predição	**Positivo**	Verdadeiro positivo (VP)	Falso positivo (FP)
	Negativo	Falso negativo (FN)	Verdadeiro negativo (VN)

Figura 22 – Ilustração do funcionamento da matriz de confusão

para ser realizada, assim conseguindo ser 9x mais eficiente do que a Mobile Net padrão.

2.4.7 Métricas de desempenho preditivo

As métricas escolhidas para avaliação do desempenho das RNC foram os seguintes: Acurácia, F1-Score, tamanho do modelo em KBs e Matriz de confusão. Para explicar melhor as métricas e ficarem de fácil entendimento, precisamos compreender essa primeira matriz de confusão que contém os verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos, tanto da predição quanto do *ground truth* (classe verdadeira). Podemos ver como funciona na figura (**??**).

- Acurácia: É a métrica de classificação mais conhecida. É muito fácil de entender, e é facilmente adequada para problemas de classificação binária e multi-classe, basicamente:

$$\textbf{Acurácia} = \frac{VP + VN}{VP + FP + FN + VN}$$

- Precisão e *Recall*: A precisão é basicamente a proporção das predições positivas que são verdadeiramente positivas.

$$\textbf{Precisão} = \frac{VP}{VP + FP}$$

  Já o Recall, nos dá quantos verdadeiros positivos da classe verdadeira (ground truth) foram corretamente classificados.

$$\textbf{Recall} = \frac{VP}{VP + FN}$$

- F1-Score: O F1 é um número entre 0 e 1 que significa a média harmônica da precisão e do recall juntos. Com ela temos um equilíbrio e melhor visão de como é o resultado da classificação de forma mais clara e de forma mais robusta.

$$\textbf{F1} \ = 2 * \frac{precisão * recall}{precisão + recall}$$

- Tamanho: Outra métrica utilizada para nosso caso foi o tamanho de arquivo dos modelos, pois esses serão utilizados voltados para o mobile, logo não podem ter um tamanho muito grande. O tamanho foi avaliado em KB (kilobytes).

2.5 Trabalhos Relacionados

Relacionados à cultura do café, destacam-se estudos voltados para estimativa de produção (CARRIJO et al., 2017), controle de pragas (OLIVEIRA et al., 2019) e caracterização dos frutos (KAZAMA, 2019). Investigado neste trabalho, a caracterização dos frutos do café é ainda pouco pesquisada na literatura, com alguns trabalhos investigando soluções para a classificação de frutos isolados retirados da planta (e normalmente com imagens registradas com controle de fundo) e outros investigando essa classificação através de imagens tomadas diretamente da planta. Apesar da facilidade em treinar modelos para frutos isolados e com controle de fundo, essa abordagem possui aplicação limitada, especialmente no contexto do nosso trabalho, que almeja desenvolver soluções que possam ser usadas de maneira prática e fácil pelo homem do campo. Por outro lado, a caracterização dos frutos considerando imagens tomadas da própria planta é muito menos explorada na literatura, pois está sujeita a todas as interferências decorrentes do campo tais como iluminação, fundo, sobreposição, etc. Contudo, apesar de ser mais difícil, este cenário foi o escolhido para o nosso estudo uma vez que faz um retrato fiel da realidade na qual o nosso usuário, o agricultor, está inserido. Trabalho mais relacionado com esta investigação, (KAZAMA, 2019) desenvolveu uma metodologia para classificar os grãos de café a partir da caracterização individual de seus estádios de maturação considerando a tomada de imagens de um ramo em lona (com controle de fundo), um ramo na planta (sem controle de fundo) ou do próprio terço da planta (inferior, médio e superior). Para classificar individualmente os frutos da imagem, a metodologia proposta pelos autores utiliza uma RNC baseada em região (R-CNN), a qual realiza a identificação dos frutos de café no ramo ou terço da planta bem como a classificação desses frutos em quatro estádios de maturação: verde, verde-cana, cereja e passa. Como resultado, o modelo alcançou um desempenho razoável para a identificação dos frutos, porém com dificuldade para classificar estádios de maturação transitivos, tal como o verde-cana. Diferente do que é proposto neste artigo, o trabalho de (KAZAMA, 2019) caracteriza a maturação dos grãos de café de maneira individual, ou seja, não retorna se aquela planta de café deveria ou não ser colhida, mas o estádio de cada um dos frutos representados na imagem. Outro trabalho que utiliza automação para classificar a maturação do café é o artigo (OLIVEIRA et al., 2016) da Universidade de Lavras. A cor dos grãos de café verdes pode variar amplamente, o que dificulta sua classificação por inspeção visual. Uma Rede Neural Artificial foi utilizada como modelo de transformação e o classificador de Bayes foi utilizado para classificar os grãos de café em quatro grupos: esbranquiçado, verde cana, verde e verde-azulado. O autor também explica sobre os métodos para coleta de imagens dos frutos do café com um esquema e especificações de cada passo dos materiais usados e as amostras de café. Além disso é explicado também como é realizado o processo para o modelo de treino da rede neural que irá transformar a matriz RGB em padrões de valores CIE L*a*b*, explicando em detalhes como tipo do arquivo da imagem, fazendo a imagem do tipo TIFF ser convertida em

matrizes tridimensionais no MATLAB para serem usadas depois no código. Esse artigo chegou a um resultado melhor que o olho humano para classificação desses frutos, porém a máquina utilizada para captura dessas imagens é complexa, precisando de uma caixa preta de metal com luz para iluminar o café, câmera posicionada em cima para captura das imagens e os frutos de café dentro dela.

Capítulo 3

Materiais e Métodos

Os métodos que desenvolvemos e etapas conduzidas desse trabalho, estão resumidos no seguinte fluxograma da Figura 23. Em síntese, foi realizado a coleta das imagens e anotação dos dados, preparação e aumento dos dados, seleção e treinamento de algoritmos voltados para aparelhos móveis e no fim o modelo com melhor performance foi adaptado em um protótipo de aplicativo que classifica os cafeeiros de acordo com os frutos de café. A seguir é explicado em detalhe cada uma dessas etapas.

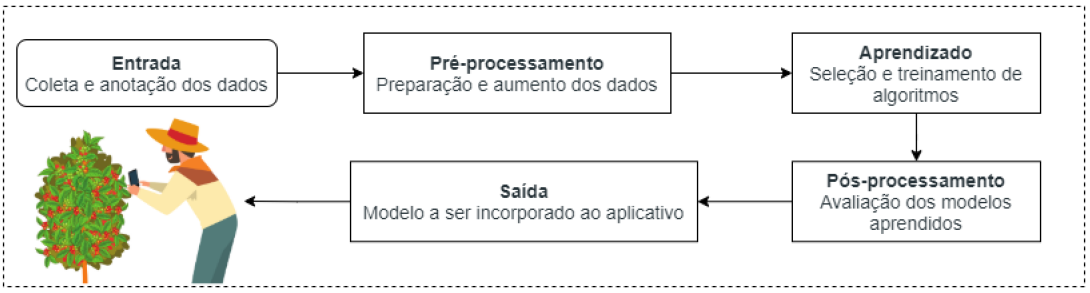

Figura 23 – Fluxograma do processo de desenvolvimento do sistema.

3.1 Coleta e anotação dos dados

Como primeira parte do trabalho, foram obtidas imagens de frutos de café de diferentes regiões do Brasil de forma colaborativa com produtores, pesquisadores e estudantes, pois foi necessário imagens para alimentação da RNC em todos os cinco estádios do grão do café. Mantemos um padrão para captura de imagens onde o usuário define a posição do dossel (local onde é localizado os frutos do café), abre a copa da planta na posição e aproxima o celular na posição vertical ou horizontal, o foco seria mostrar o cafeeiro aberto com os frutos aparecendo para a classificação do mesmo, como explicado no folder de divulgação da Figura 24 que enviamos para quem estaria apto a captura de imagens. Foram concebidas exatamente 10.251 imagens no total contando com todas as classes e a partir delas projetamos as etapas do nosso algoritmo de aprendizagem supervisionado.

Contamos com ajuda de 5 especialistas em café e agronomia da UFU para classificar as imagens capturadas para o projeto. Separamos de forma aleatória uma parcela dessas 10.251, totalizando 2500 imagens que foram enviadas para os especialistas, pois estes supervisionaram o aprendizado , para cada um rotular 200 imagens de cada estádio a partir desse subconjunto da base fornecida a eles. Este processo de classificação foi realizado através de tabelas e no fim criamos um algoritmo para pegar as anotações das tabelas exportadas para CSV e conseguir localizar as imagens e organizá-las de forma automática para poupar tempo. Uma segunda rodada de anotação foi realizada com os cinco especialistas para resolver os casos conflitantes, especialmente relacionados com os estádios transitivos, que são verde cana, passa e seco. Por conta disso algumas classes ficaram com um pouco mais de 200 imagens enquanto outras tiveram um pouco menos. Aliás, este problema da classificação se refletiu até o fim do projeto nos testes da nossa rede neural por conta da variabilidade dos frutos deixarem bem difícil a rotulação. Por fim obtivemos aproximadamente 1000 imagens para serem usadas no treinamento, validação e teste da RNC .

Figura 24 – Fôlder divulgado para captura das imagens de café

3.2 Preparação e aumento dos dados

Após a etapa de coleta e anotação dos dados, conseguimos 1000 imagens classificadas no total para serem utilizadas no treinamento da RNC. Foram atribuídas 90% das imagens

para treino, e 10% para teste. A fim de melhorar o desempenho do algoritmo com mais imagens, foi utilizada estratégias de aumento de dados (*data augmentation*). Utilizamos técnicas como *Flip* horizontal, vertical e a combinação dos dois nas imagens originais para aumentar a base de dados e desempenho do modelo, isso fez aumentar de 885 imagens para **3540** imagens no total. Exemplos de imagens transformadas na Figura 25.

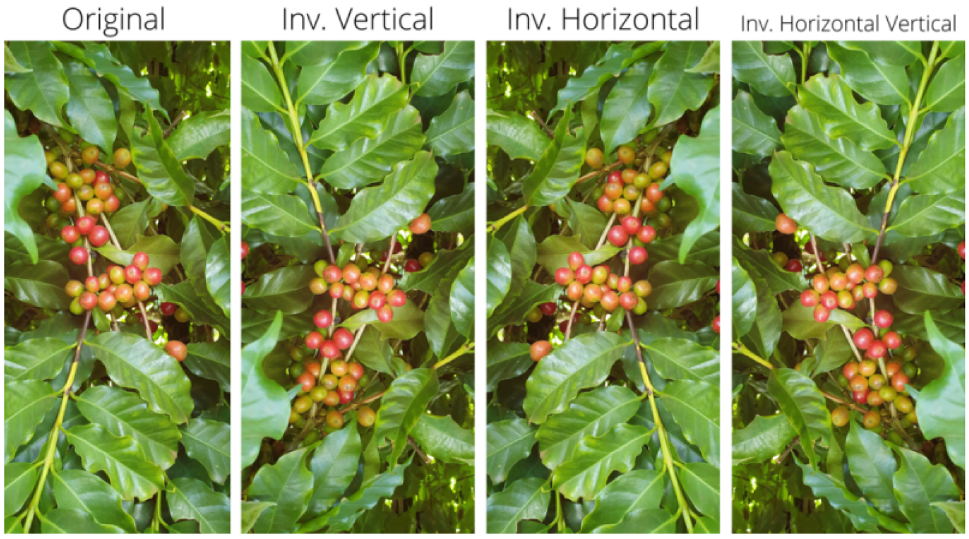

Figura 25 – Imagens geradas a partir de uma imagem original para o aumento do banco de dados

3.3 Seleção e treinamento de algoritmos

Como discutido no trabalho, foi estudado redes neurais convolucionais como VGG-16 e LeNet, porém para utilização do modelo final foi estudado outros modelos voltados para o uso Mobile, onde o modelo final treinado deveria ser utilizado para a classificação das fotos do café. Como objetivo final é a utilização do mesmo para os cafeicultores, com as arquiteturas complexas como VGG-16 o modelo poderia ter sua utilização comprometida em celulares. Como o modelo final será utilizado em dispositivos móveis, não pode consumir recursos demasiados de memória e processamento. Dado isso, esta etapa de pesquisa permitiu com que fossem identificadas as arquiteturas mais apropriadas para o nosso problema (voltadas para o uso de smartphones): MobileNet, MobileNet V2 e NasNetMobile. Cada uma substitui a convolução clássica por uma nova mais eficiente e de baixo custo computacional, que é a *depthwise separable convolution*. Foram utilizadas com a transferência de aprendizado padrão, onde é carregado dados pré-treinados da ImageNET, onde são alocados os pesos treinados nas arquiteturas selecionadas são alocados na nossa rede com novos dados de entrada, congelando-se os pesos das camadas de extração de características e treina-se apenas as camadas de classificação. (DENG et al., 2009), um banco de dados de imagens em grande escala que se tornou até um desafio de competição entre

redes neurais convolucionais para ver qual tem a melhor performance. Com esses modelos pré treinados ganhamos em termos de desempenho preditivo e também economizamos tempo por conta do Transfer Learning (RIBANI; MARENGONI, 2019).

Capítulo 4

Resultados Experimentais

Com os modelos e métodos escolhidos, iniciamos o treinamento após a preparação da base de dados. O treino foi feito utilizando arquiteturas MobileNet, MobileNetV2 e NasNetMobile. Foi feito experimentos e avaliações da técnica de aumento de dados além de experimentos comparando os três modelos selecionados. Ao final desse processo, foi gerado um protótipo de classificação de frutos de café em tempo real a partir do melhor modelo aqui encontrado.

Após a decisão dos modelos e métodos, foi feito o treino pelos três modelos, esses foram treinados com diversos parâmetros, mudando épocas de treino, alterando o *learning rate* do otimizador Adam empiricamente e aumentando o tamanho da entrada das imagens para o algoritmo absorver mais detalhes e conseguir separar as fotos de acordo com cada classe.

4.1 Base de dados e configuração dos parâmetros

Nossa base de dados foi composta por mais de 10.251 imagens de cafés em seus cinco estádios (verde, verde cana, cereja, passa e seco). Com isso, foram separadas 2500 imagens em forma de sorteio para a classificação dos especialistas, sendo classificadas mais de 1000 imagens. Após a classificação, foram separadas 90% para treino e 10% para testes. Utilizamos apenas treino e teste, pois, a validação não é usada quando utilizamos *Transfer Learning*. A Tabela 4 nos mostra a quantidade de imagens na base de dados antes de passar pelo método de aumento de dados, após ela ser aplicada quadruplicamos as imagens de treino e de teste, isto é, 1 imagem gerou 3 novas imagens. No total, 885 imagens rotuladas foram usadas para treino e mantivemos 100 para os testes, como podem ser vistas na 4, após o aumento de dados temos 3540 imagens para treino e ainda as 100 para os testes.

Após a organização da base de dados, as imagens foram separadas para dois experimentos, o de classificação binária (colher ou não colher) e o multi-classe. Os parâmetros

Tabela 3 – Quantidade das imagens da base de dados original

Classes	Treino (Rotuladas)	Teste (Rotuladas)
Verde	233	20
Verde Cana	177	20
Cereja	137	20
Passa	145	20
Seco	193	20
		Total: 985

Tabela 4 – Banco de dados após o aumento de dados nas imagens de treino

Classes	Treino (Rotuladas)	Teste (Rotuladas)
Verde	932	20
Verde Cana	708	20
Cereja	548	20
Passa	580	20
Seco	772	20
		Total: 3540

de treino foram ajustados diversas vezes, e de forma empírica o melhor resultado foi o seguinte:

- ❏ Entrada de arquivos em 224x224.
- ❏ 15 Épocas de treino, 55 Passos por época e com tamanho do batch de 64.
- ❏ Otimizador: Adam learning rate = 0.0001 (Decidimos de forma empírica, isto é, testamos vários valores para ver qual o algoritmo aprendia melhor e esta foi a melhor combinação até agora).
- ❏ 3540 Imagens para treinamento.

  (Classe: Colher (Cafés nos estádios cereja, passa e seco, onde deve haver menos de 20% de cafés verdes) 1900 exemplos.

  (Classe: Não colher (Cafés nos estádios verde e verde-cana).
- ❏ Foram usadas 100 imagens para o teste (20 de cada classe) 1640 exemplos.

4.2 Experimentos

Nesta sessão é mostrado os resultados de ambos experimentos e discussões dos resultados. Os testes foram feito em uma máquina utilizando apenas o processador para treinamento, um I5-9400F, 16GB de Ram, não foi possível utilizar a placa de vídeo por conta dela ser da marca AMD Radeon, e o TensorFlow não possui suporte para esta

marca, apenas para a NVidia. Os três modelos treinados, foram utilizados com o apoio de Transfer Learning, a maioria deles atingiram 99% de acurácia durante o treinamento e *loss* chegando a média de 0.00032, o que é uma ótima medida e nos dá confiança do desempenho deste modelo. Por outro lado vale ressaltar a nossa cautela com os resultados de treino, pois nem sempre se refletem nos testes devido a problemas como *overfitting*, situação em que o modelo é muito bem sucedido nos dados do treinamento porém nos testes não consegue um bom resultado (KOEHRSEN, 2018).

4.2.1 Classificação do problema de colher ou não

Na Tabela a seguir é mostrado o resultado dos modelos que foram treinados utilizando as arquiteturas selecionadas (MobileNet, MobileNetV2 e NasNetMobile).

Tabela 5 – Comparação dos modelos e desempenhos em duas bases de dados.

	Original			Aumento de dados		
Modelos	Ac.(%)	F1(%)	Tam.(KB)	Ac.(%)	F1(%)	Tam.(KB)
MobileNet	0.90	0.89	23.079	0.93	0.92	20.489
MobileNetV2	0.86	0.86	20.483	0.92	0.91	20.280
NASNetMobile	0.74	0.78	28.709	0.87	0.88	28.709

A Tabela 5 apresenta o resultado das arquiteturas do estado-da-arte após o seu treinamento sobre a base original e sobre a base de dados resultante das estratégias de aumento de dados. Analisando os resultados da tabela, vemos que o MobileNet se adaptou melhor ao nosso problema, conseguindo o melhor resultado tanto em desempenho quanto em tamanho, mesmo sendo maior que os outros um pouco porém não é grande essa diferença. Ainda temos muito a testar para melhorar essa performance, como aumentar épocas do treinamento, testar outros otimizadores e tamanhos da entrada de imagem, porém já obtivemos um resultado de 93% de acurácia que é muito relevante para um problema complexo como o nosso. É importante observar ainda que as estratégias de aumento de dados melhoraram o desempenho de todas as arquiteturas investigadas.

4.2.2 Classificação categórica do problema multi-classe

Após os testes feitos no problema de identificar o momento adequado para colheita, testamos como o modelo se comportaria no problema de caracterização dos estádios de maturação utilizando a MobileNet, que foi o modelo com melhor desempenho na Tabela 5. A acurácia foi 63% de acordo com os resultados apresentados na Tabela 6, com uma boa performance na classificação do cafeeiro verde, cereja e seco, porém ruim ($< 60\%$ de F1) para as classes verde-cana e passa. Isso, se dá pelo fato daqueles serem estádios não transitivos, já os testes de classificação do verde-cana e o seco possuem diversos frutos

em estádios diferentes na mesma planta. As configurações de entrada de dados foram as mesmas do problema binário. Foram inseridas 3540 Imagens de treino essas sendo: 932 Verde; 708 Verde-Cana; 548 Cereja; 580 Passa; 772 Seco.

Tabela 6 – Teste da Mobile Net para o problema multi-classe.

Classe	Precisão	Recall	F1-Score
Verde	0.79	0.95	0.86
verde-cana	0.67	0.50	0.57
Cereja	0.56	0.70	0.62
Passa	0.45	0.45	0.45
Seco	0.69	0.55	0.61

Tabela 7 – Matriz de confusão da Mobile Net no problema multi-classe

Classe	Verde	Verde-Cana	Cereja	Passa	Seco
Verde	**19**	0	0	1	0
Verde-Cana	4	**10**	6	0	0
Cereja	0	5	**14**	1	0
Passa	1	0	5	**9**	5
Seco	0	0	0	9	**11**

Consideramos que o modelo teve um desempenho mediano. Através da Tabela 7 fica claro que as classes com mais dificuldades são a verde-cana e a passa por serem estados transitivos, na Figura 26 é mostrado imagens que foram classificadas pelo modelo. Como podemos observar algumas que foram classificadas erradas podem ter sido por conta da iluminação como no caso da cereja que foi classificada como passa e também apresentação de mais cores como na verde que foi classificada como verde cana, e também classificou um verde como passa, isso provavelmente foi consequência da foto meio escura em alguns pontos da imagem onde o modelo pode ter achado que teria grãos de café do estádio passa. Assim podemos ter uma ideia do que se pode melhorar, seja na captura de fotos ou na seleção delas e também no modelo, onde poderemos testar novos, pois esse funcionou muito bem para a classificação de colher ou não, obtendo ótimo resultado porém houve complicações para o problema das várias classes.

4.2.3 Aplicativo Android

Além disso foram feitos protótipos funcionais executando nosso modelo em um aplicativo Android. O protótipo consegue executar o algoritmo treinado e retornar em tempo real a classificação do cafeeiro capturado pela câmera, porém precisa estar no local para ser feita essa classificação. Um segundo protótipo foi feito que recebe imagens e nos dá a classificação de imagens enviadas para ele, com isso pode ser testado diretamente em fotos

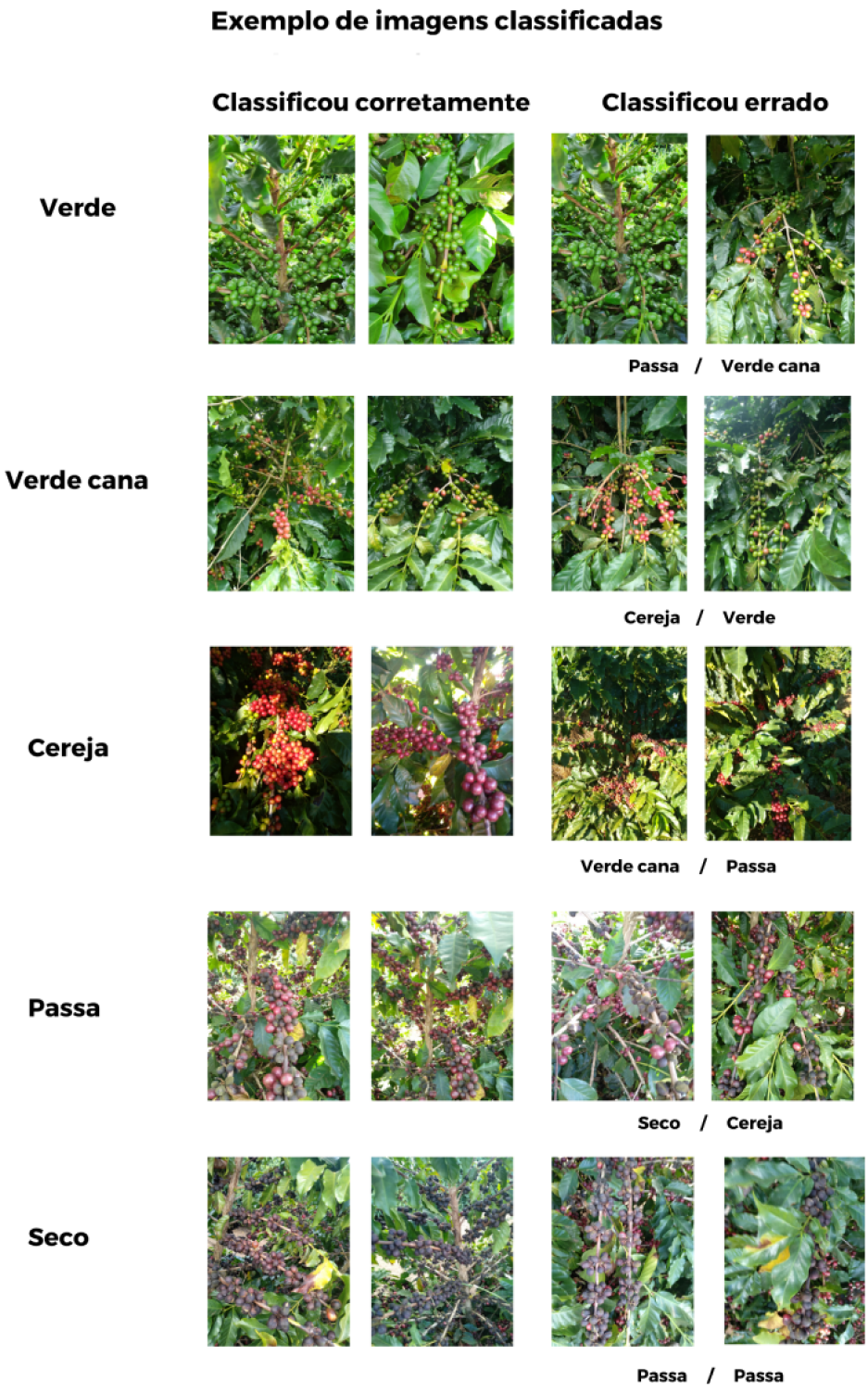

Figura 26 – Resultados de imagens classificadas

de cafeeiros sem precisar de estar no local, como mostrado na Figura 28. O protótipo foi feito utilizando a ferramenta Android Studio e a biblioteca TensorFlow Lite que é voltada para uso Mobile, aplicando o aprendizado de máquina para dispositivos com menor poder de processamento.

Com isso, conseguimos avançar no projeto para a realização de testes no campo em

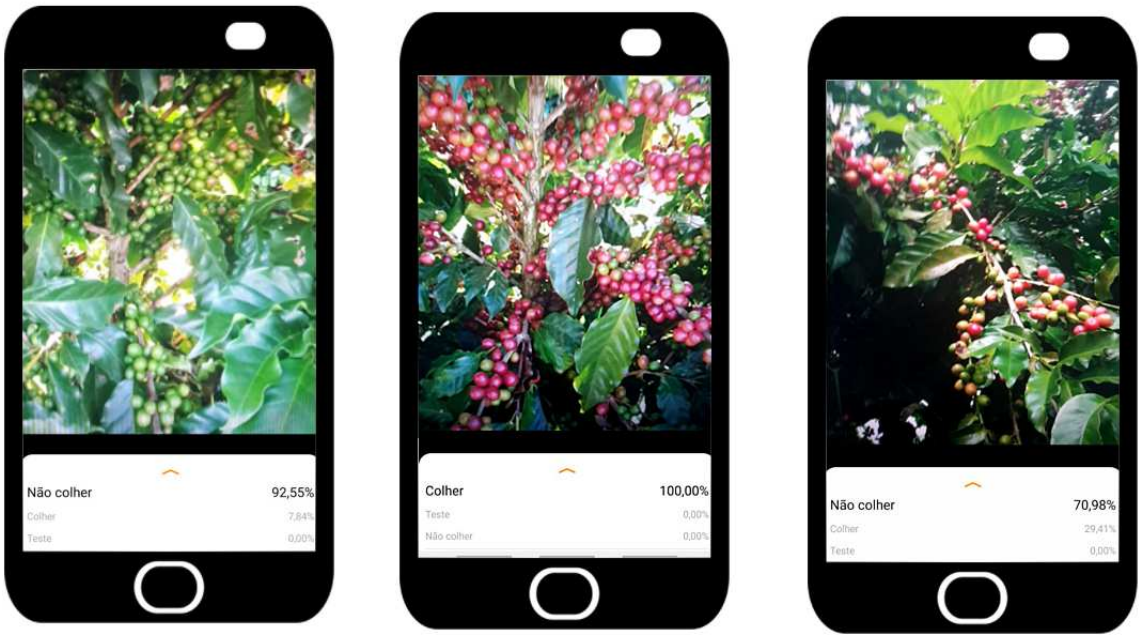

Figura 27 – Protótipo 1 em funcionamento.

tempo real. O aplicativo foi configurado para identificar a adequabilidade dos frutos para a colheita. Como é mostrado no terceiro celular da Figura 27 até para casos transitivos está funcionando devidamente. As próximas etapas envolvem a realização de testes no campo e com imagens diretamente para o protótipo 2 para ver como será o desempenho do aplicativo desenvolvido na prática.

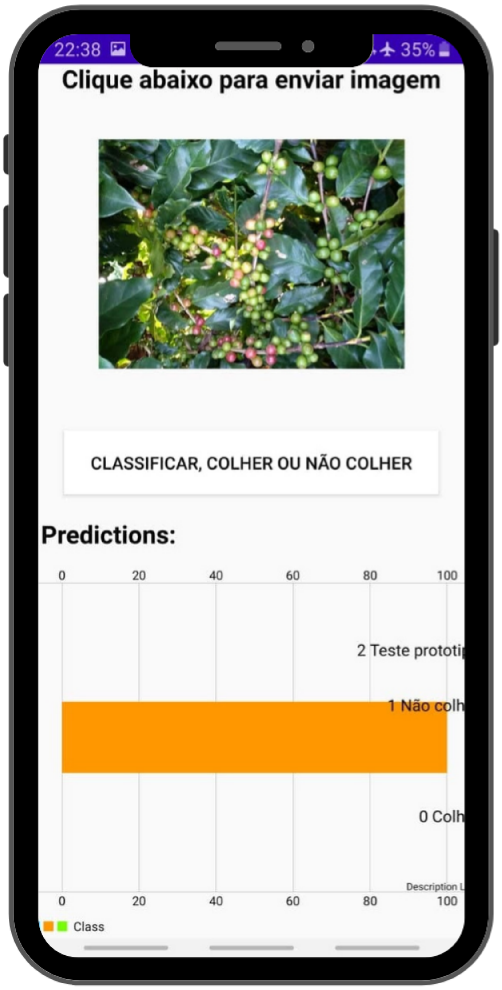

Figura 28 – Protótipo 2 em funcionamento.

Capítulo 5

Conclusão

Foi estudado e implementado os modelos de arquiteturas de RNCs voltadas para uso em *smartphones* e conseguimos um bom resultado no modelo de colher ou não colher, foram criados protótipos para ser executados testes sistemáticos e criada a base de dados com milhares de imagens classificadas por especialistas que ainda pode ser expansível e melhorada. A hipótese foi respondida, sendo ela:"Se o uso de redes neurais convolucionais é uma alternativa eficiente em medidas de acurácia e precisão e praticidade para automatizar o processo de caracterização da maturação do café", e tomamos a conclusão que sim, por conta da eficiência de classificação em 93% pode ser levado em conta na hora de tomar uma decisão da colheita dos frutos, porém ainda é necessário testes sistemáticos para realmente comprar isto.

5.1 Principais Contribuições

As principais contribuições desse trabalho foram a criação da base de dados com o auxílio dos especialistas supervisionando o aprendizado, ela conta com mais de 10000 imagens e não foi totalmente classificada ainda, porém contém muitas imagens classificadas de cafeeiros que podem ser utilizadas em projetos futuros do mundo todo, e também podemos melhorá-la expandindo e também . O melhor modelo obtido pelos experimentos aqui realizados alcançou um desempenho de 93% de acurácia e F1-Score de 92%, além disso comprovamos que o uso de aumento de dados pode melhorar o desempenho no nosso projeto. Um protótipo utilizando o modelo vencedor para classificação de cafeeiros diretamente pelo celular é um ponto que chama bastante atenção e foi uma contribuição boa para o projeto em um todo.

5.2 Trabalhos Futuros

Para trabalhos futuros deixamos o aplicativo, a fim de que possa ser avaliado e testado por especialistas e cafeicultores no campo. Outra direção é a investigação de abordagens

de aprendizado semi-supervisionado, a fim de lidar com as milhares de imagens recebidas durante o processo de aprendizado. Também pretendemos melhorar nosso desempenho considerando um conjunto maior de parâmetros de treinamento para as arquiteturas. Ademais, deixamos espaço para trabalhos futuros estarem buscando melhorias nos nossos métodos, como já foi dito, existem outros tipos de RNCs que ainda precisam ser testadas e comparadas, como por exemplo as várias RCNNs (Region-based Convolutional Neural Network) para detecções de regiões e segmentações, onde provavelmente teremos grandes melhorias para o problema multi-classe, pois essas redes neurais convolucionais por região conseguem separar os frutos do restante da imagem, assim se conseguiria classificar os frutos da imagem e depois dar um veredito da classificação do cafeeiro como um todo.

5.3 Contribuições em Produção Bibliográfica

- ❑ WICSI - SBSI 2021 (MUNDIM-FILHO; ALVARENGA; CARNEIRO, 2021): Artigo aceito para publicação no Workshop de Iniciação Científica do Simpósio Brasileiro de Sistemas de Informação, apresentando parte do nosso estudo dos modelos e resultados sobre a base de dados apresentada.
- ❑ Vencedor (3^o lugar) do III CODESII (Competição de Desenvolvimento de Sistemas de Informação Inovadores), onde a ideia foi expandida para trazer mais ferramentas no aplicativo como mapeamento espacial da maturação dos cafeeiros de uma certa área onde o cafeicultor teria melhor gestão da sua fazenda.

Referências

AKMELIAWATI, R.; OOI, M. P.-L.; KUANG, Y. C. Real-time malaysian sign language translation using colour segmentation and neural network. In: IEEE. **2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007**. [S.l.], 2007. p. 1–6. Citado na página 18.

ALPAYDIN, E. **Introduction to machine learning**. [S.l.]: MIT press, 2020. Citado na página 16.

ALVES, O. A. A. R. et al. **Colheita e Preparo do Café**. [S.l.]: SENAR, 1999. Citado na página 15.

AMATO, F. et al. **Artificial neural networks in medical diagnosis**. [S.l.]: Elsevier, 2013. Citado na página 18.

BARTHOLO, G.; GUIMARÃES, P. Cuidados na colheita e preparo do café. **Informe Agropecuário**, v. 18, n. 187, p. 33–42, 1997. Citado na página 15.

BENGIO, Y. **Learning deep architectures for AI**. [S.l.]: Now Publishers Inc, 2009. Citado na página 19.

CAMARGO, Â. P. de. Florescimento e frutificação de café arábica nas diferentes regiões (cafeeiras) do brasil. **Pesquisa Agropecuária Brasileira**, v. 20, n. 7, p. 831–839, 1985. Citado 2 vezes nas páginas 13 e 14.

CARRIJO, G. L. et al. Automatic detection of fruits in coffee crops from aerial images. In: IEEE. **2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)**. [S.l.], 2017. p. 1–6. Citado na página 33.

CHAROENKWAN, P.; FANG, S.-W.; WONG, S.-K. A study on genetic algorithm and neural network for implementing mini-games. In: IEEE. **2010 International Conference on Technologies and Applications of Artificial Intelligence**. [S.l.], 2010. p. 158–165. Citado na página 18.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255. Citado na página 37.

EMBRAPA; T, L.; S., S. d. P. A. D. d. F. e. I. C.-G. d. A. d. P. e. I. J. **Sumário café outubro 2019**. [S.l.]: EMBRAPA, 2019. Citado na página 13.

FRAGA, C. C. Resenha histórica do café no brasil. **Agricultura em São Paulo**, v. 10, n. 1, p. 1–21, 1963. Citado na página 13.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the National Academy of Sciences**, National Academy of Sciences, v. 79, n. 8, p. 2554–2558, 1982. ISSN 0027-8424. Disponível em: <https://www.pnas.org/content/79/8/2554>. Citado na página 18.

HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, arXiv, p. 1–7, 2017. Citado na página 30.

JAIN, G.; SHARMA, M.; AGARWAL, B. Spam detection in social media using convolutional and long short term memory neural network. **Annals of Mathematics and Artificial Intelligence**, Springer, v. 85, n. 1, p. 21–44, 2019. Citado na página 18.

KAISER, L.; GOMEZ, A. N.; CHOLLET, F. Depthwise separable convolutions for neural machine translation. **arXiv preprint arXiv:1706.03059**, 2017. Citado na página 31.

KAZAMA, E. H. Colheita de prescrição para o café, é possível? Universidade Estadual Paulista (UNESP), p. 1–15, 2019. Citado na página 33.

KHAN, S. et al. A guide to convolutional neural networks for computer vision. **Synthesis Lectures on Computer Vision**, Morgan & Claypool Publishers, v. 8, n. 1, p. 1–207, 2018. Citado na página 18.

KOEHRSEN, W. Overfitting vs. underfitting: A complete example. **Towards Data Science**, Towards Data Science, p. 6–7, 2018. Citado na página 41.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). **Advances in Neural Information Processing Systems 25**. Curran Associates, Inc., 2012. p. 1097–1105. Disponível em: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>. Citado 2 vezes nas páginas 20 e 26.

LECUN, Y. et al. Lenet-5, convolutional neural networks. **URL: http://yann. lecun. com/exdb/lenet**, v. 20, n. 5, p. 14, 2015. Citado 2 vezes nas páginas 5 e 24.

MESQUITA, C. M. et al. **Manual do café: colheita e preparo (Coffea arábica L.)**. [S.l.]: EMATER-MG, 2016. Citado 2 vezes nas páginas 14 e 15.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, Manole Ltda, v. 1, n. 1, p. 32, 2003. Citado na página 16.

MUNDIM-FILHO, A. C.; ALVARENGA, C. B.; CARNEIRO, M. G. Colher ou não colher? Uma rede neural convolucional para apoiar cafeicultores na decisão da colheita. In: **Anais Estendidos do XVII Simpósio Brasileiro de Sistemas de Informação**. [S.l.]: SBC, 2021. p. 57–60. Citado na página 47.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Icml**. [S.l.: s.n.], 2010. Citado na página 22.

OLIVEIRA, A. J. et al. Analysis of nematodes in coffee crops at different altitudes using aerial images. In: IEEE. **2019 27th European Signal Processing Conference (EUSIPCO)**. [S.l.], 2019. p. 1–5. Citado na página 33.

OLIVEIRA, D. S. L. Emanuelle Morais de et al. A computer vision system forcoffee beans classification based on computational intelligence techniques, classificador de bayes. **URL: www.elsevier.com/locate/jfoodeng**, p. 22–27, 2016. Citado na página 33.

PACELLI, V.; AZZOLLINI, M. et al. An artificial neural network approach for credit risk management. **Journal of Intelligent Learning Systems and Applications**, Scientific Research Publishing, v. 3, n. 02, p. 103, 2011. Citado na página 18.

PARIKH, D. Learning paradigms in machine learning. Medium, 2018. Disponível em: <https://medium.datadriveninvestor.com/learning-paradigms-in-machine-learning-146ebf8b5943.> Citado 3 vezes nas páginas 5, 17 e 18.

RIBANI, R.; MARENGONI, M. A survey of transfer learning for convolutional neural networks. In: IEEE. **2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)**. [S.l.], 2019. p. 47–57. Citado na página 38.

RICHARDSON, F.; REYNOLDS, D.; DEHAK, N. A unified deep neural network for speaker and language recognition. **arXiv preprint arXiv:1504.00923**, 2015. Citado na página 19.

SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.]: arXiv, 2018. p. 4510–4520. Citado na página 30.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **CoRR**, abs/1404.7828, 2014. Disponível em: <http://arxiv.org/abs/1404.7828>. Citado na página 18.

SHAHIN, M. A.; JAKSA, M. B.; MAIER, H. R. Artificial neural network applications in geotechnical engineering. **Australian geomechanics**, Citeseer, v. 36, n. 1, p. 49–62, 2001. Citado na página 18.

SILVA, F. C. d. et al. Desempenho operacional da colheita mecanizada e seletiva do café em função da força de desprendimento dos frutos. Editora UFLA, 2013. Citado na página 15.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014. Citado na página 28.

YEGNANARAYANA, B. **Artificial neural networks**. [S.l.]: PHI Learning Pvt. Ltd., 2009. Citado na página 18.

ZOPH, B. et al. Learning transferable architectures for scalable image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 8697–8710. Citado na página 31.