

**UNIVERSIDADE ESTADUAL DE PONTA GROSSA
SETOR DE CIÊNCIAS AGRÁRIAS E TECNOLOGIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA**

MATHEUS MORESCO

**ANÁLISE DE ARQUITETURAS DE REDES NEURAIS SIAMESAS PARA A
CLASSIFICAÇÃO DE ESPÉCIES DE PLANTAS**

PONTA GROSSA

2022

MATHEUS MORESCO

**ANÁLISE DE ARQUITETURAS DE REDES NEURAIS SIAMESAS PARA A
CLASSIFICAÇÃO DE ESPÉCIES DE PLANTAS**

Dissertação apresentada como um dos requisitos para obtenção do título de Mestre em Computação Aplicada na Universidade Estadual de Ponta Grossa, Área de concentração: Computação para Tecnologias Agrícolas.

Orientador: Prof. Dr. Alceu de Souza Britto Jr.
Coorientador: Prof. Dr. Yandre Maldonado e Gomes da Costa

PONTA GROSSA

2022

M843 Moresco, Matheus
Análise de arquiteturas de redes neurais siamesas para a classificação de espécies de plantas / Matheus Moresco. Ponta Grossa, 2022.
54 f.

Dissertação (Mestrado em Computação Aplicada - Área de Concentração: Computação para Tecnologias em Agricultura), Universidade Estadual de Ponta Grossa.

Orientador: Prof. Dr. Alceu de Souza Brito Júnior.
Coorientador: Prof. Dr. Yandre Maldonado e Gomes Costa.

1. Redes neurais siamesas. 2. Classificação de espécies de plantas. 3. Características multicamadas. I. Brito Júnior, Alceu de Souza. II. Costa, Yandre Maldonado e Gomes. III. Universidade Estadual de Ponta Grossa. Computação para Tecnologias em Agricultura. IV.T.

CDD: 004



UNIVERSIDADE ESTADUAL DE PONTA GROSSA
Av. General Carlos Cavalcanti, 4748 - Bairro Uvaranas - CEP 84030-900 - Ponta Grossa - PR - <https://uepg.br>

TERMO

TERMO DE APROVAÇÃO

Matheus Moresco

ANÁLISE DE ARQUITETURAS DE REDES NEURAIS SIAMESAS PARA A CLASSIFICAÇÃO DE ESPÉCIES DE PLANTAS

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Computação Aplicada da Universidade Estadual de Ponta Grossa, pela seguinte banca examinadora:

Prof. Dr. Alceu de Souza Britto Júnior (UEPG - Presidente)

Prof. Dr. Luciano José Senger (UEPG)

Prof(a). Dra. Simone Bello Kaminski Aires (UTFPR-PG)

Ponta Grossa, 10 de março de 2022.



Documento assinado eletronicamente por **Alceu de Souza Britto Junior, Professor(a)**, em 10/03/2022, às 17:12, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Simone Bello Kaminski Aires, Usuário Externo**, em 25/05/2022, às 09:47, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Luciano Jose Senger, Professor(a)**, em 25/05/2022, às 18:39, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida no site <https://sei.uepg.br/autenticidade> informando o código verificador **0904550** e o código CRC **58E00B64**.

AGRADECIMENTOS

Ao meu orientador, Alceu de Souza Britto Jr, agradeço pelo apoio durante toda a pesquisa, e também toda a ajuda prestada durante minha formação acadêmica. Agradeço, também, ao meu coorientador Yandre Maldonado e Gomes da Costa, que acompanhou o projeto e contribuiu muito com ideias, sugestões e melhorias ao trabalho. Aos Professores da banca, agradeço pelas sugestões, críticas e elogios que fizeram, pois foram de grande ajuda para evolução e melhoria deste trabalho.

Agradeço a Universidade Estadual de Ponta Grossa, pela oportunidade e estrutura cedida para os estudos. Em especial, ao Professor José Carlos Ferreira da Rocha, coordenador do Programa de Mestrado da UEPG, agradeço por ter me incentivado a ingressar no programa e pelo auxílio prestado durante minha permanência no curso.

A minha Família, namorada e amigos, agradeço imensamente pelo apoio durante todo esse tempo, sempre me incentivaram a continuar e acreditaram no meu potencial para concluir esta etapa de minha vida. Em especial, agradeço as amigas que fiz durante o programa, que mesmo remotamente mantivemos contato, trocando informações e nos ajudando durante o curso.

Finalmente, agradeço a CAPES pelo apoio financeiro durante este período, que permitiu me dedicar exclusivamente a este projeto.

RESUMO

A classificação de espécies de plantas, em especial a que utiliza imagens de folhas, é tida como um grande desafio, que requer a análise de especialistas em botânica. Isso pode estar relacionado ao grande número de espécies de plantas já catalogadas e também ao fato de algumas folhas de plantas serem semelhantes, apesar de pertencerem a espécies diferentes. Com o intuito de facilitar o processo de classificação de plantas, sistemas automáticos que usam técnicas de aprendizagem de máquina e visão computacional para diferenciar as espécies têm sido propostos, em especial modelos de Redes Neurais Convolucionais (CNN, do inglês, “*Convolutional Neural Networks*”), vem sendo amplamente utilizados para extração de características e classificação de espécies de plantas. Porém, o uso de CNN necessita de um grande número de imagens para realizar seu treinamento. Além disso, tal ferramenta não é escalável, de maneira que, caso uma nova classe seja adicionada, a rede precisa ser treinada novamente. Assim, uma alternativa tem sido o uso de Redes Neurais Siamesas (SNN, do inglês, “*Siamese Neural Network*”). A partir disso, este estudo tem por objetivo avaliar diferentes arquiteturas de Redes Neurais Siamesas na classificação de espécies de plantas a partir da imagem da folha. Adicionalmente, pretende-se investigar o uso de características extraídas de camadas intermediárias das SNNs, observando o impacto na classificação e buscando priorizar abordagens que sejam eficientes mesmo com poucas imagens de treinamento. Experimentos realizados com as bases de imagens de folhas *Flavia* e *MalayaKew*, demonstram que o uso de características intermediárias, dependendo da combinação de camadas, pode melhorar o desempenho de uma SNN. Os modelos VGG-16, MobileNet e DenseNet, apresentaram, respectivamente, uma melhora na acurácia de 0,35, 1,05 e 1 pontos percentuais para a base Flavia e 3,37, 3,33 e 11 pontos percentuais para a base MalayaKew.

Palavras-chave: Redes Neurais Siamesas, Classificação de espécies de plantas, Características multicamadas

ABSTRACT

The classification of plant species, specifically those that use leaf images, has been a great challenge, which requires the analysis of experts in botany. It may be related to a large number of plant species already cataloged and the fact that some plant leaves are similar, despite belonging to different species. To facilitate the plant classification process, automatic systems that use machine learning and computer vision techniques to differentiate species have been proposed, especially Convolutional Neural Network (CNN) models, which have been widely used for feature extraction and classification of plant species. However, the use of CNN requires a large number of images to carry out its training. Also, this tool is not scalable, so if a new class is added, the network needs to retrain. Thus, an alternative has been the use of Siamese Neural Networks (SNN). This study aims to evaluate different architectures of Siamese Neural Networks to plant species classification from leaf images. Additionally, the use of features extracted from the intermediate layers of SNNs is investigated, observing the impact on the results and seeking to prioritize approaches that are efficient even with few training images. Experiments on Flavia and MalayaKew leaf image databases have shown that the fusion of intermediate features improves SNN performance. For SNNs composed of VGG16, MobileNet, and DenseNet models, we observed an accuracy improvement of 0.35, 1.05, and 1 percentage points respectively for the Flavia database, while 17.42, 3.33, and 11 percentage points respectively for the MalayaKew database.

Keywords: Siamese Neural Networks, Plant Species Classification, Multi-layer Features

LISTA DE ILUSTRAÇÕES

Figura 1 -	Exemplos de semelhança interespecie retirados da base <i>Flavia</i> . . .	12
Figura 2 -	Exemplos de variabilidade intraespecie retirados da base <i>LifeCLEF 2015</i>	13
Figura 3 -	Arquitetura do modelo AlexNet.	21
Figura 4 -	Arquitetura do modelo VGG-16	21
Figura 5 -	Arquitetura do modelo ResNet	23
Figura 6 -	Arquitetura do modelo DenseNet	24
Figura 7 -	Arquitetura do modelo GoogleNet	25
Figura 8 -	Exemplo de Módulo Inception	26
Figura 9 -	Arquitetura do modelo Inception V3	26
Figura 10 -	Arquitetura do modelo MobileNet	27
Figura 11 -	Exemplos de Imagens da Base Flavia	35
Figura 12 -	Exemplos de Imagens da Base MalayaKew	35
Figura 13 -	Modelo Proposto	37
Figura 14 -	Redução de exemplos na base de treino	45
Figura 15 -	Desempenho do Modelo SNN- <i>DenseNet</i> em uma base de dados cruzada	47
Figura 16 -	Desempenho do Modelo SNN- <i>DenseNet</i> com características multicamadas em uma base de dados cruzada	48

LISTA DE TABELAS

Tabela 1 -	Matriz de confusão	28
Tabela 2 -	Exemplos de trabalhos relacionados a classificação de espécies de plantas usando imagem de folhas.	32
Tabela 3 -	Distribuição de imagens por classe na base <i>Flavia</i>	36
Tabela 4 -	Acurácia da CNN e SNN (%) nas bases <i>Flavia</i> e <i>MalayaKew</i> . . .	41
Tabela 5 -	Camadas internas selecionadas para extração de características . .	42
Tabela 6 -	Acurácia (%) dos modelos de SNN com características multicamadas na Base <i>Flavia</i>	43
Tabela 7 -	Acurácia (%) dos modelos de SNN com características multicamadas na Base <i>MalayaKew</i>	44
Tabela 8 -	Acurácia (%) dos modelos considerando 10 exemplos de treinamento por classe, na base <i>Flavia</i>	45
Tabela 9 -	Acurácia (%) dos modelos considerando 10 exemplos de treinamento por classe, na base <i>MalayaKew</i>	46

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Motivação	11
1.2	Problema	12
1.3	Objetivo	14
1.3.1	Objetivos específicos	14
1.4	Hipóteses	14
1.5	Contribuições	14
1.6	Estrutura do Trabalho	15
2	REVISÃO DA LITERATURA	16
2.1	Aprendizagem Profunda	16
2.2	Aprendizado Incremental	16
2.3	Redes Neurais Convolucionais	17
2.3.1	Aprendizado por transferência	20
2.3.1.1	AlexNet	20
2.3.1.2	VGGNet	21
2.3.1.3	ResNet	22
2.3.1.4	DenseNet	22
2.3.1.5	GoogLeNet	22
2.3.1.6	Inception	22
2.3.1.7	MobileNet	24
2.4	Redes Neurais Siamesas	27
2.5	Medidas de Avaliação	28
2.6	Trabalhos Correlatos	30
2.7	Considerações Finais	33
3	MATERIAL E MÉTODOS	34

3.1	Bases de Dados	34
3.1.1	<i>Flavia</i>	34
3.1.2	<i>MakayaKew</i>	34
3.2	Método proposto	36
3.2.1	Arquitetura da CNN	38
3.2.2	Arquitetura da SNN	38
3.2.3	SNN com características multicamadas	39
3.2.4	Escolha das imagens de referência	40
4	RESULTADOS EXPERIMENTAIS	41
4.1	Definição das Arquiteturas de CNN e SNN	41
4.2	Uso de Multicamadas nas Redes Siamesas	42
4.3	Avaliação dos Modelos em Bases Com Poucos Exemplos	45
4.4	Avaliação de Escalabilidade	46
5	CONCLUSÃO	49
	REFERÊNCIAS	51

1 INTRODUÇÃO

As plantas desempenham papel fundamental na natureza, servindo como fonte de oxigênio e nutrientes, tendo grande importância no ecossistema do planeta terra. Além disso, também servem como matéria prima de vários setores da indústria (ANTONELLI; SMITH; SIMMONDS, 2019).

Embora as plantas estejam presentes no nosso cotidiano, a classificação de espécies de plantas é tida como um grande desafio, que requer a análise de especialistas em botânica, e mesmo assim, devido ao grande número de espécies de plantas já catalogadas, tais especialistas têm dificuldades nesta tarefa.

Com o intuito de facilitar este processo de classificação de plantas, sistemas automáticos que usam técnicas de aprendizagem de máquina e visão computacional para diferenciar as espécies têm sido propostos (ZHANG *et al.*, 2020; AZLAH *et al.*, 2019).

Os modelos de classificação automática de espécie de plantas usam imagens do caule, folha, flor ou até mesmo da planta inteira. A partir dessas imagens é possível extrair um conjunto de características que será usado na classificação. Tais características podem ser obtidas de maneira manual, que permite obter informações como cor, forma, textura e tamanho das imagens (WÄLDCHEN; MÄDER, 2018). Porém, a extração manual dessas características é trabalhosa e precisa ser alterada de acordo com o tipo de imagem que se está trabalhando. Dessa forma, abordagens que usam extração automática de características têm recebido atenção especial. Com Redes Neurais Convolucionais (CNN, do inglês, “*Convolutional Neural Network*”) é possível ajustar automaticamente a extração de características de acordo com o tipo de imagem que está sendo usada (LEE *et al.*, 2017).

Apesar disso, o uso de Redes Neurais Convolucionais demanda um grande número de imagens para realizar seu treinamento e não é uma ferramenta de aprendizado incremental, de maneira que, caso uma nova classe seja adicionada, a rede precisa ser treinada novamente. Uma alternativa tem sido o uso de Redes Neurais Siamesas (SNN, do inglês, “*Siamese Neural Network*”), em particular para a classificação de espécies de plantas. Neste contexto, estas redes têm se destacado na classificação usando poucas imagens para treino e garantindo escalabilidade do classificador (WANG; WANG, 2019). Além disso, o artigo de Rahman *et al.* (2018) mostra que é possível melhorar o desempenho de Redes Neurais Siamesas usando características extraídas de camadas intermediárias das CNNs constituintes, dando ainda mais vantagens a esta abordagem.

1.1 Motivação

Em um Ecossistema com grande diversidade biológica de seres vivos a classificação de espécies é uma tarefa muito abordada pelos pesquisadores, a fim de obter informações sobre o *habitat* em estudo. A classificação de espécies de plantas tem um papel fundamental no desenvolvimento de novas tecnologias. Em especial, na agricultura, vemos a classificação de espécies de plantas presente na identificação de plantas daninhas (SØGAARD, 2005), tarefa que auxilia o agricultor no controle de pragas e, conseqüentemente, na melhoria da produção.

Aliada à evolução tecnológica dos últimos anos, principalmente com o surgimento de câmeras fotográficas e smartphones, a classificação de espécies de plantas, tanto na agricultura como na botânica em geral, encontrou um novo meio de coleta de dados. Porém, a aquisição em larga escala de imagens de plantas, nos deixa o desafio de analisá-las e classificá-las, um trabalho quase impossível de ser feito manualmente.

A grande variedade de bases de dados contendo imagens de plantas, chamou a atenção de pesquisadores na área de aprendizagem de máquina e visão computacional, que têm proposto métodos automáticos para a análise dessas imagens (WÄLDCHEN; MÄDER, 2018).

Abordagens automáticas de reconhecimento de plantas costumam destacar a folha como um componente muito importante na tarefa de identificação, devido às propriedades visuais presentes na folha, tais como: contornos, bordas, cor, textura e padrões de veias (ZHAO *et al.*, 2015).

Apesar da utilização de técnicas como CNNs serem comuns na classificação de plantas, estas são treinadas em uma base contendo um grande número de imagens distribuídas de maneira uniforme entre as classes, porém, ainda é comum encontrarmos distribuições desbalanceadas em bases de imagens de folhas, devido à dificuldade de acesso e a escassez de plantas de uma determinada espécie. O problema de desbalanceamento de dados causa um aumento na complexidade da classificação quando usamos CNNs (BUDA; MAKI; MAZUROWSKI, 2018).

Além do mais, o constante surgimento de novas espécies de plantas (ANTONELLI *et al.*, 2020), obriga a CNN tradicional a ser reformulada e treinada novamente sempre que uma nova classe é adicionada à base, processo que, dependendo do número de espécies e quantidade de imagens presentes na base de dados, pode gastar muito tempo.

De tal maneira que, o uso de SNNs tem sido estudado como alternativa de clas-

sificação, devido a este método apresentar bons resultados na classificação da bases de dados com poucos exemplos (WANG; WANG, 2019), além de permitir a construção de classificadores escaláveis.

1.2 Problema

A classificação de espécies de plantas é uma tarefa difícil até mesmo para especialistas em botânica. A criação de uma solução automática para reconhecimento de espécies a partir da imagem da folha apresenta a seguintes dificuldades (ARAUJO *et al.*, 2021):

a) Classificação em subcategorias: diferente da classificação tradicional, que visa classificar objetos muito diferentes, como por exemplo, uma pessoa e um carro, na classificação em subcategorias, todas as imagens pertencem a uma mesma categoria geral básica, porém o objetivo é dividir essas imagens em subcategorias, como no caso da classificação de imagens de folhas, todas as imagens pertencem a categoria de folhas de plantas, porém queremos dividir as imagens em espécies (WANG; SUN; WANG, 2017; ŠULC; MATAS, 2017).

Existem dois aspectos que tornam o reconhecimento de subcategorias um problema desafiador à visão computacional, são eles:

Semelhança interespécie: quando ocorrem casos onde imagens de classes diferentes, tem forma, cor e textura semelhantes, tornando difícil a tarefa de diferenciá-las. Esse fenômeno pode ser observado na Figura 1.

Figura 1: Exemplos de semelhança interespécie retirados da base *Flavia*.



Fonte: (WU *et al.*, 2007).

Nota: Imagens pertencentes as classes (a) *Big-fruited holly*, (b) *Crepe myrtle*, (c) *Japanese flowering cherry* e (d) *Wintersweet*.

Alta variabilidade intraespécies: as imagens de folhas geralmente são capturadas em diferentes cenários e suas características morfológicas podem variar dependendo da maturação da planta, gerando imagens de diferentes cores e formas dentro da mesma classe, fenômeno que pode ser observado na Figura 2.

b) Dados desbalanceados: devido ao difícil acesso e a raridade de algumas espécies

Figura 2: Exemplos de variabilidade intraespécie retirados da base *LifeCLEF 2015*.



Fonte: (JOLY *et al.*, 2015).

Nota: Todas as imagens pertencem a classe *Broussonetia papyrifera* (L.) Vent.

de plantas, podemos encontrar em algumas bases de dados, categorias que possuam uma quantidade menor de imagens comparadas às demais, causando um desbalanceamento. Na maioria dos casos, isso resulta em muitas informações a respeito das categorias mais incidentes, e menos das minoritárias, o que pode, em muitos casos, interferir no resultado da classificação, causando um desempenho ruim nas classes mal representadas (BUDA; MAKI; MAZUROWSKI, 2018).

c) Quantidade de espécies: A grande diversidade de plantas existentes se apresenta como uma dificuldade na visão computacional. De acordo com Thorn e Gardens (2016) estima-se que existem cerca de 391 mil espécies de plantas já catalogadas. Com o grande número de espécies, as dificuldades intraespécies e interespécies se tornam ainda mais complexas. Além disso, bases de dados com um grande número de classes aumentam a complexidade da classificação.

Além de aumentar a complexidade do modelo a ser construído para a classificação, o grande número de categorias de objetos, requer um grande número de pesos e parâmetros de modelos de reconhecimento. Tais circunstâncias tornam difícil treinar, de forma eficiente, modelos de aprendizagem.

Além disso, novas espécies de plantas são encontradas a todo momento, em 2019, foram descobertas 1942 novas espécies de plantas (ANTONELLI *et al.*, 2020), sendo assim necessário o uso de técnicas escaláveis, que estejam preparadas para a inserção de novas categorias a qualquer momento.

1.3 Objetivo

Este trabalho tem por objetivo avaliar diferentes arquiteturas de Redes Neurais Siamesas para a classificação de espécies de plantas a partir da imagem da folha. Para tal, pretende-se investigar a combinação de características extraídas de camadas intermediárias das CNNs constituintes, observando o impacto na classificação e buscando priorizar abordagens de aprendizado incremental.

1.3.1 Objetivos específicos

Destacam-se como objetivos específicos deste trabalho:

- Investigar diferentes formas de combinar camadas de uma SNN.
- Investigar o impacto no desempenho das SNNs variando sua arquitetura e parâmetros.
- Avaliar as SNNs propostas considerando os desafios inerentes ao processo de classificação de espécies de plantas.

1.4 Hipóteses

As hipóteses principais que norteiam este trabalho de pesquisa são as seguintes:

- Hipótese 1 - a escolha da CNN constituinte tem um papel importante na construção da SNN.
- Hipótese 2 - a combinação de camadas intermediárias de uma SNN tem um impacto positivo no contexto de classificação de espécie de plantas.
- Hipótese 3 - uma SNN com combinação de camadas mantém as mesmas características de plasticidade e estabilidade quando comparadas a uma SNN convencional.

1.5 Contribuições

As contribuições científicas deste trabalho podem ser descritas como:

- Um comparativo entre o desempenho de Redes Neurais Convolucionais e Redes Neurais Siamesas, no âmbito da classificação de espécies de plantas usando imagens de folha.
- Uma análise que mostre o desempenho de SNNs quando combinadas diferentes características extraídas de camadas intermediárias da rede.
- Uma abordagem incremental para classificação de espécies de plantas usando imagens de folha.

Além disso, espera-se que este trabalho possa auxiliar pesquisadores nas áreas de:

- Aprendizado de máquina, em especial aqueles que desenvolvem pesquisas com Redes Neurais Siamesas, espera-se que este trabalho seja usado para avaliação de modelos de SNNs e que possa fornecer informações úteis sobre a combinação de características multicamadas em modelos de Redes Neurais.
- Botânica, espera-se que o presente trabalho possa trazer avanços na área de classificação de espécies de plantas, principalmente com relação à criação de métodos automáticos de classificação, que sejam escaláveis.

1.6 Estrutura do Trabalho

O presente trabalho foi organizado em capítulos, o primeiro capítulo apresenta a identificação e contextualização do problema. O segundo capítulo faz uma breve revisão dos conceitos usados durante o trabalho e traz estudos relacionados ao tema. Durante o terceiro capítulo foi descrito o método proposto e os materiais utilizados. No quarto capítulo, foram expostos os resultados obtidos. E, por fim, no quinto capítulo estão presentes as conclusões do trabalho.

2 REVISÃO DA LITERATURA

Neste capítulo são descritos os principais conceitos relacionados ao método proposto, como a aprendizagem profunda, Redes Neurais Convolucionais e Redes Neurais Siamesas. Além disso, é feita uma revisão dos trabalhos relacionados ao tema.

2.1 Aprendizagem Profunda

O aprendizado profundo (DL, do inglês, “*Deep Learning*”) consiste em redes neurais que possuem várias camadas de processamento trabalhando juntas, similar ao sistema nervoso humano. Todas as camadas são interconectadas por nós, ou neurônios, e cada camada interna usa a saída da camada anterior como sua entrada. A primeira camada em uma rede é chamada de camada de entrada, enquanto a última é chamada de camada de saída. Todas as camadas entre as duas são referidas como camadas ocultas (LECUN; BENGIO; HINTON, 2015).

O aprendizado profundo é superior na extração de informações e é muito utilizado na etapa de extração de características, pois apresenta vantagens, comparado às técnicas convencionais usadas para extração manual de características na visão computacional (O’MAHONY *et al.*, 2019). Algumas das arquiteturas de DL mais usadas são as Redes Neurais Convolucionais, Redes Multilayer Perceptrons, Redes Neurais Recorrentes, Redes de Crença Profunda, entre outras. O DL é responsável por avanços recentes em visão computacional, classificação de imagens e detecção de objetos (RUSSAKOVSKY *et al.*, 2015).

2.2 Aprendizado Incremental

Modelos de aprendizagem tradicionais assumem que os dados presentes durante o treinamento representam a totalidade do problema. No entanto, há situações no mundo real, em que adquirir dados que representem o problema como um todo é um trabalho difícil e demorado (CIARELLI; OLIVEIRA; SALLES, 2012). Além disso, existem casos em que a aquisição de dados continua, até mesmo sem previsão de término, como é o caso do tema tratado neste trabalho (classificação de espécies de plantas).

Em tais situações, é necessário um sistema que seja capaz de se adaptar a mudança de ambiente, sem que seu desempenho seja afetado. A arquitetura deve ser desenvolvida de forma que ela aprenda com os novos dados sem esquecer o conhecimento já adquirido. As estruturas que atendem estes requisitos são chamadas de modelos de Aprendizado

Incremental (CIARELLI, 2012). Tais sistemas tem duas propriedades principais:

- **Estabilidade:** adquirir novos conhecimentos sem esquecer drasticamente o conhecimento já adquirido. Garante a preservação do conhecimento já adquirido, mas não garante que os novos dados serão entendidos por completo.
- **Plasticidade:** capacidade de aprender continuamente, porém, não tem a garantia de preservar o conhecimento adquirido anteriormente.

A plasticidade e estabilidade estão intimamente relacionadas. Um modelo ideal deve ser capaz de ser estável e plástico ao mesmo tempo. Deve ter plasticidade para se adaptar a futuros cenários do ambiente do problema, mas deve também permanecer estável com relação a eventos referentes aos cenários antigos (CARPENTER; GROSSBERG, 1987).

Em sua tese Ciarelli (2012) cita como exemplos de Redes Neurais com Aprendizado Incremental os modelos de Rede Neural Probabilística Incremental (RNPI), Perceptron Multicamadas Evolutivo (eMLP), Rede Neural Fuzzy Evolutiva (EFuNN), Rede Neural Probabilística Incremental Baseada em Expectation Maximization (RNPI-EM), entre outros.

2.3 Redes Neurais Convolucionais

Uma Rede Neural Convolutiva é um algoritmo de aprendizado profundo que, a partir de uma imagem de entrada, é capaz de extrair características de vários aspectos e até mesmo diferenciar imagens de classes diferentes. O pré-processamento exigido em uma CNN é muito menor em comparação com outros algoritmos de classificação. Enquanto os métodos convencionais de extração de características de imagens são fixos e devem ser configurados manualmente, os filtros das CNNs têm a capacidade de se modificar e se atualizam, durante o treinamento, para se adequar ao tipo de características presentes nas imagens (VOULODIMOS *et al.*, 2018).

O processamento em cada neurônio de uma CNN se dá pelo que chamamos de função de ativação. A escolha das funções de ativação de uma rede neural são uma consideração importante uma vez que definem como devem ser seus dados de entrada. As funções de ativação mais usadas são (ALOYSIUS; GEETHA, 2017):

- **Linear:** é a função de ativação mais básica porque não altera a entrada de

um neurônio. Geralmente é utilizada nas camadas de saída em redes neurais de regressão. É definida pela Equação 1.

$$f(x) = x \quad (1)$$

- **Sigmóide:** é comumente utilizada por redes neurais com propagação positiva (*Feedforward*) que precisam ter como saída apenas números positivos, em redes neurais multicamadas e em outras redes com sinais contínuos. A função sigmóide transforma todos os valores de entrada em valores que variam entre 0 e 1. É definida pela Equação 2.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

- **Tangente Hiperbólica(tanh):** muito semelhante à função sigmóide. Na verdade, é apenas uma versão escalonada da função sigmoide, sendo simétrico em relação à origem, variando os valores de saída entre -1 e 1. É definida pela Equação 3.

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (3)$$

- **ReLU:** é a função de ativação mais amplamente utilizada ao projetar redes neurais atualmente. A função ReLU é não linear, e sua principal característica é que ela converte as entradas que possuem valores negativos para 0. Definida pela Equação 4.

$$f(x) = \begin{cases} 0 & \Rightarrow x < 0 \\ x & \Rightarrow x \geq 0 \end{cases} \quad (4)$$

- **Leaky ReLU:** É uma melhoria da função ReLU, em vez de definir o gradiente 0 para $x < 0$, que fez com que as informações desses neurônios sejam desconsideradas, a Leaky ReLU define um pequeno componente linear de x , como mostra a Equação 5. A principal vantagem de substituir a linha horizontal é remover o gradiente zero.

$$f(x) = \begin{cases} \alpha x & \Rightarrow x < 0 \\ x & \Rightarrow x \geq 0 \end{cases} \quad (5)$$

- **Softmax:** força a saída de uma rede neural a representar a probabilidade dos dados serem de uma das classes definidas. As saídas desta função variam entre 0 e 1 e a soma de toda as saídas desta camada devem ser igual a 1. Geralmente é usada na camada de saída do classificador, onde são geradas as probabilidades para definir a classe de cada entrada. Definida pela Equação 6.

$$\sigma(Z)_j = \frac{e^{Z_j}}{\sum_{k=1}^K e^{Z_k}} \quad (6)$$

A CNN consiste em uma ou mais camadas convolucionais que são seguidas por uma ou mais camadas totalmente conectadas. Dentre os principais tipos de camadas presentes em Redes Neurais Convolucionais (ALOYSIUS; GEETHA, 2017), estão:

- **Camada de Convolução:** as imagens de entrada são filtradas usando um conjunto de filtros convolucionais e cada filtro extrai um conjunto de características das imagens.
- **Camada de Pooling:** seu principal objetivo é reduzir o número de parâmetros e cálculos do modelo. Isto não só acelera os cálculos, como evita o problema de *overfitting*. Um procedimento comum para o pooling é conhecido como pool máximo (ou *Max-Pooling*). No *Max-Pooling*, uma unidade de pooling simplesmente gera a ativação máxima na região de entrada $n \times n$.
- **Camadas Não Lineares:** o objetivo principal desta camada é transformar o sinal de entrada em sinal de saída, e esse sinal será utilizado como entrada na próxima camada. As funções mais comuns usadas nas camadas não lineares são: sigmóide, Tanh, ReLU, etc.
- **Camadas Totalmente Conectadas:** Os neurônios desta camada estão conectados a todos os neurônios da camada anterior. Conecta e achata os blocos contendo os dados extraídos pela rede, transformando em uma única linha que contém todas a informações extraídas. Nesta camada se inicia o processo de classificação das informações extraídas pelas camadas anteriores.
- **Camadas de Classificação:** É a camada que determina a probabilidade da imagem sendo processada pertencer a cada uma das classes definidas no modelo. Uma das funções mais usadas neste tipo de camada é a função Softmax.

2.3.1 Aprendizado por transferência

O aprendizado por transferência visa melhorar o desempenho de modelos de um determinado domínio, transferindo conhecimento adquiridos em domínios diferentes do original. Dessa forma, pode ser reduzida a dependência de um grande número de dados durante o treinamento no domínio original. Esta técnica tem sido promissora e popular no contexto do aprendizado de máquina (ZHUANG *et al.*, 2020).

As redes Neurais Convolucionais vem se destacando em vários domínios da computação, como a classificação e reconhecimento de imagens, dessa forma, alguns padrões definidos em trabalhos de destaque se tornam amplamente conhecidos (DHILLON; VERMA, 2020).

Os modelos de CNNs com melhores desempenhos são treinados e salvos pelos pesquisadores que os desenvolveram, depois são disponibilizados ao público. É comum encontrar na literatura pesquisadores que carregam esses modelos pré-treinados e adaptam aos seus trabalhos usando técnicas de aprendizado por transfência, esta prática é conhecida como ajuste fino (do inglês, “*fine-tuning*”).

As próximas subseções tem como objetivo descrever as arquiteturas de CNNs comumente utilizadas (ALOYSIUS; GEETHA, 2017).

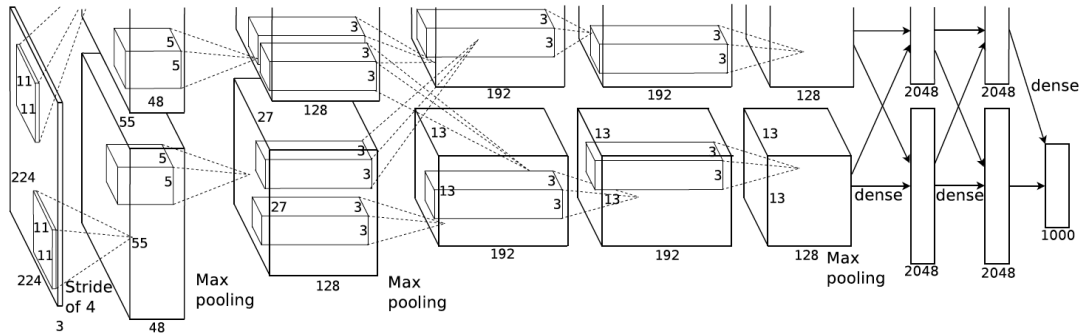
2.3.1.1 AlexNet

AlexNet é um modelo projetado por Alex Krizhevsky em conjunto com Ilya Sutskever e Geoffrey Hinton (KRIZHEVSKY; SUTSKEVER; HINTON, 2017). Esta rede foi vitoriosa no “*ImageNet Large Scale Visual Recognition Challenge*”, que ocorreu em setembro de 2012, onde, no teste de top-5, atingiu 84,6% de acurácia para a base *ImageNet*.

O trabalho em questão concluiu que a profundidade do modelo era essencial para o seu alto desempenho, procedimento que era muito custoso até então, porém, passou a ser viável devido à possibilidade de usar unidades de processamento gráfico (GPUs, do inglês “*Graphics Processing Unit*”) durante o treinamento. A rede proposta foi treinada em duas GPUs GTX 580 por 5 a 6 dias.

A arquitetura consiste em oito camadas, cinco camadas convolucionais e três camadas totalmente conectadas, como mostra a Figura 3. AlexNet usa a função ReLU em vez da tanh, que era padrão na época.

Figura 3: Arquitetura do modelo AlexNet.

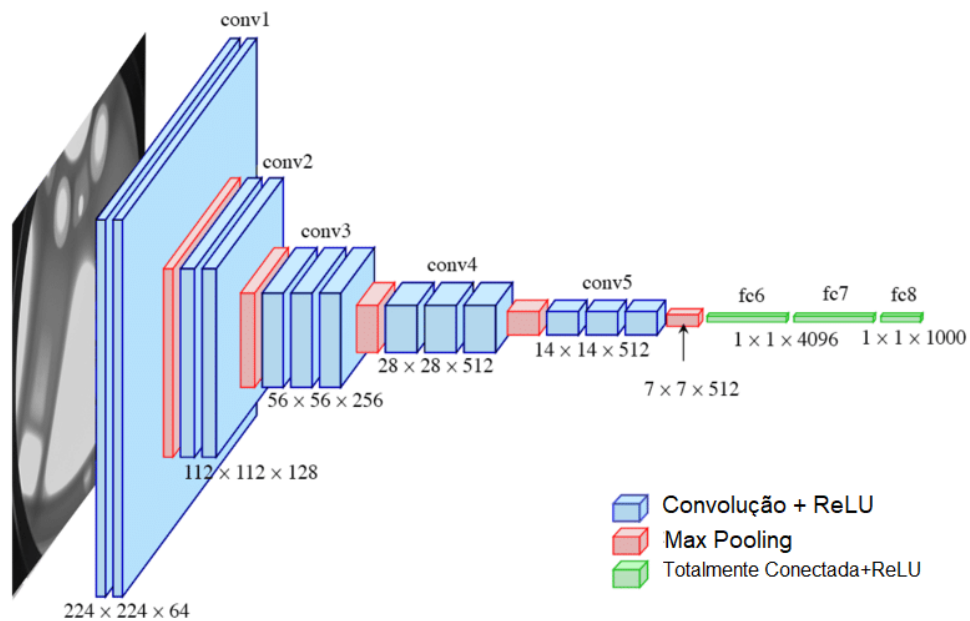


Fonte: (KRIZHEVSKY; SUTSKEVER; HINTON, 2017)

2.3.1.2 VGGNet

Desenvolvida por Simonyan e Zisserman em 2014 (SIMONYAN; ZISSERMAN, 2014), o VGGNet é muito usado com um extrator de características e sua configuração de pesos está disponível publicamente. Possui duas variações, o VGG-16, mostrado na Figura 4, que contém 13 camadas convolucionais com 3 camadas totalmente conectadas, e o VGG-19, que contém 3 camadas convolucionais extras. Foi treinado na base ImageNet em 4 Nvidia Titan Black GPUs por 2 a 3 semanas. A VGGNet possui 138 milhões de parâmetros, o que o torna mais complexo com relação a quantidade de parâmetros.

Figura 4: Arquitetura do modelo VGG-16



Fonte: (FERGUSON *et al.*, 2017)

2.3.1.3 ResNet

ResNet foi desenvolvido por He et.al. em 2015 (HE *et al.*, 2016). A arquitetura ResNet foi destaque no desafio de classificação “*ILSVRC 2015*”, demonstrando que redes extremamente profundas podem ser treinadas usando SGD como otimizador. A ideia por trás dessa abordagem é que a próxima camada deve aprender algo novo e diferente do que a entrada já foi fornecida. O modelo Resnet com camadas residuais é ilustrado na Figura 5 e apresenta uma complexidade menor do que VGGNet, com relação a quantidade de parâmetros.

2.3.1.4 DenseNet

Desenvolvida por Huang *et al.* (2017), possui L camadas, tem $L(L+1/2)$ conexões diretas e cada camada é acoplada de maneira *Feedforward*. Ele concatena os mapas de características de saída junto com os mapas de características de entrada; portanto, cada camada adquire conhecimento coletivo de todas as camadas anteriores. Este modelo apresenta bons resultados em várias bases de dados, incluindo as bases CIFAR-100 e ImageNet. Sua maior desvantagem é utilizar muita memória extra, devido à concatenação de tensores. A arquitetura do modelo DenseNet pode ser observada na Figura 6.

2.3.1.5 GoogLeNet

O GoogLeNet foi desenvolvido por C. Szegedy et al. (SZEGEDY *et al.*, 2015) e conseguiu o melhor desempenho no desafio de detecção e classificação da *ImageNet* em 2014, com precisão de 93,3%, na classificação de top-5. O diferencial desse modelo é o uso aprimorado de recursos de computacionais no modelo. É usado um bloco de construção conhecido como “*inception module*”, que leva em consideração a profundidade e largura ampliadas do modelo. Este módulo é baseado em várias convoluções muito pequenas para reduzir drasticamente o número de parâmetros.

Esta arquitetura contém 22 camadas, o número de parâmetros é reduzido de 60 milhões para 4 milhões quando comparado com AlexNet. A arquitetura da GoogLeNet pode ser observada na Figura 7.

2.3.1.6 Inception

O módulo *Inception* de extração de características que surge junto com a *GoogLeNet*, com o objetivo de extrair características em vários níveis, computando convoluções

Figura 5: Arquitetura do modelo ResNet

34-layer residual

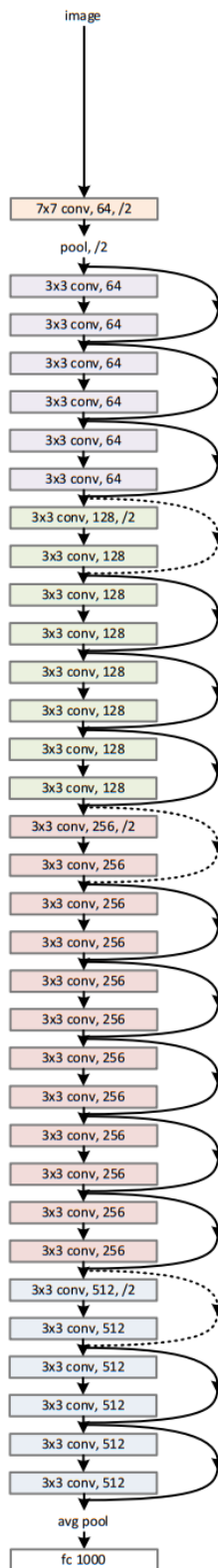
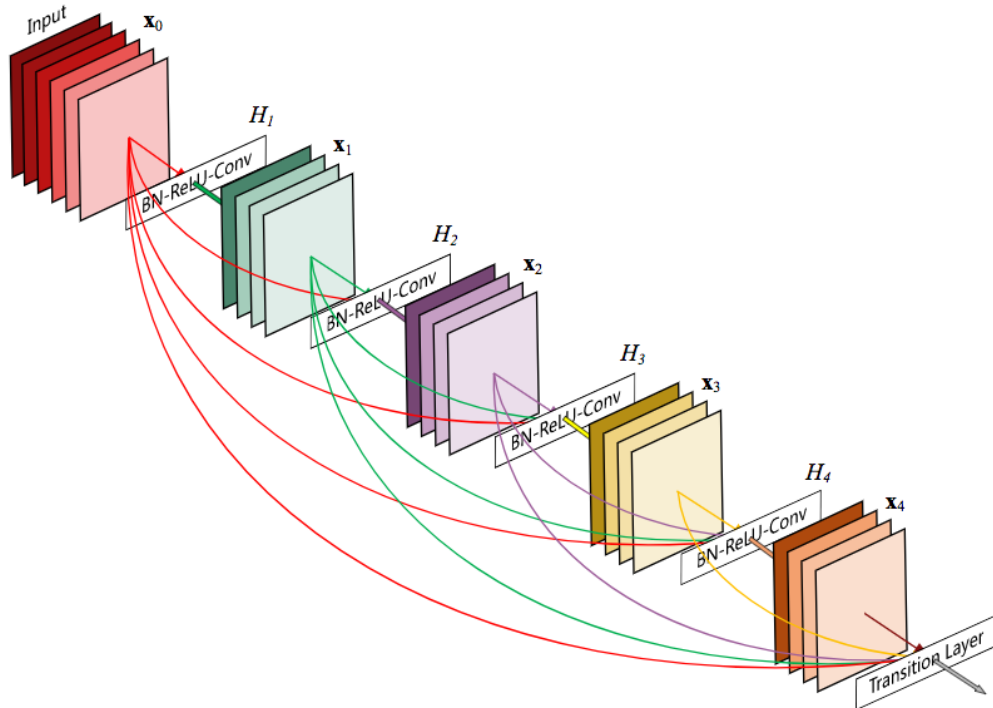
Fonte: (HE *et al.*, 2016)

Figura 6: Arquitetura do modelo DenseNet



Fonte: (HUANG *et al.*, 2017)

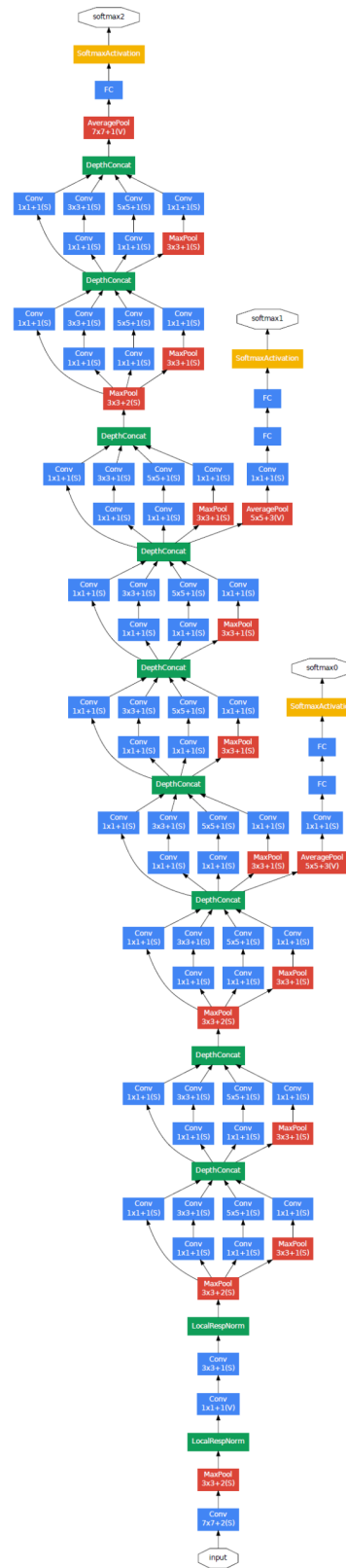
1×1 , 3×3 e 5×5 dentro do mesmo módulo da rede, como mostra a Figura 8, é a base para a criação da rede Inception (SZEGEDY *et al.*, 2015). O modelo em si é composto de componentes simétricos e assimétricos, incluindo convoluções, agrupamentos médios, agrupamentos máximos, concatenações, desistências e camadas totalmente conectadas. A normalização em lote é usada extensivamente em todo o modelo e aplicada às entradas de ativação.

O modelo Inception V3 foi desenvolvido em 2015 para a classificação de imagens. Treinado no conjunto de dados ImageNet, apresentou um bom desempenho e um custo computacional relativamente baixo, pois reduziu o número de parâmetros usados pela rede, quando comparado a arquiteturas anteriores (SZEGEDY *et al.*, 2017). A Figura 9 mostra como é a arquitetura do Inception V3.

2.3.1.7 MobileNet

MobileNet é uma arquitetura mais adequada para aplicativos de visão baseados em dispositivos móveis e embarcados, onde há restrições na capacidade de computação. Esta arquitetura foi proposta pelo Google e é baseada em uma arquitetura simplificada

Figura 7: Arquitetura do modelo GoogleNet



Fonte: (SZEGEDY *et al.*, 2015)

Figura 8: Exemplo de Módulo Inception

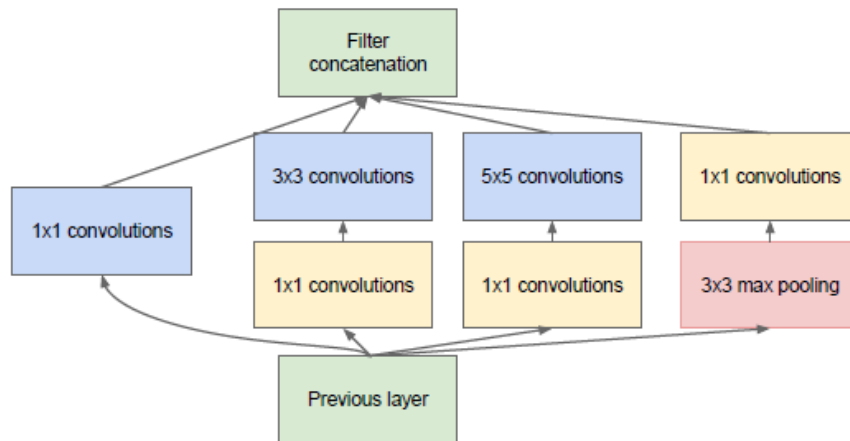
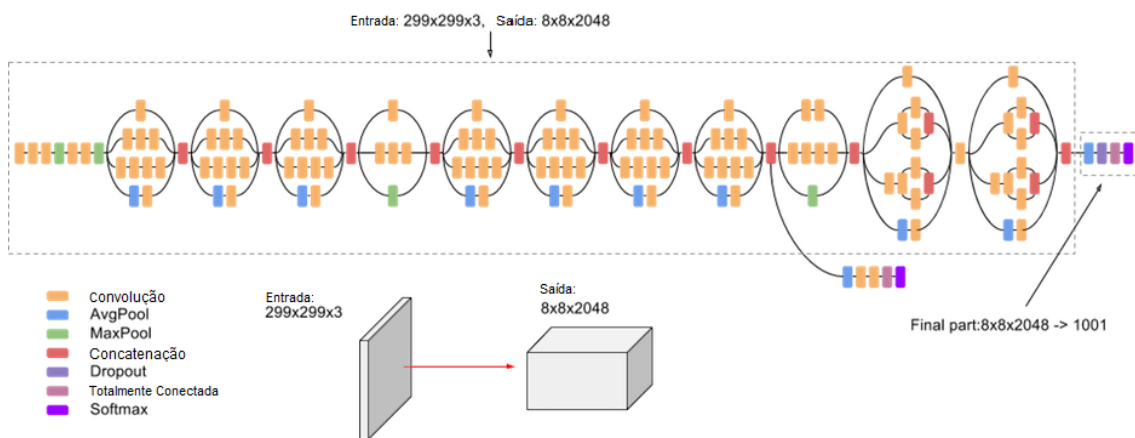
Fonte: (SZEGEDY *et al.*, 2015)

Figura 9: Arquitetura do modelo Inception V3

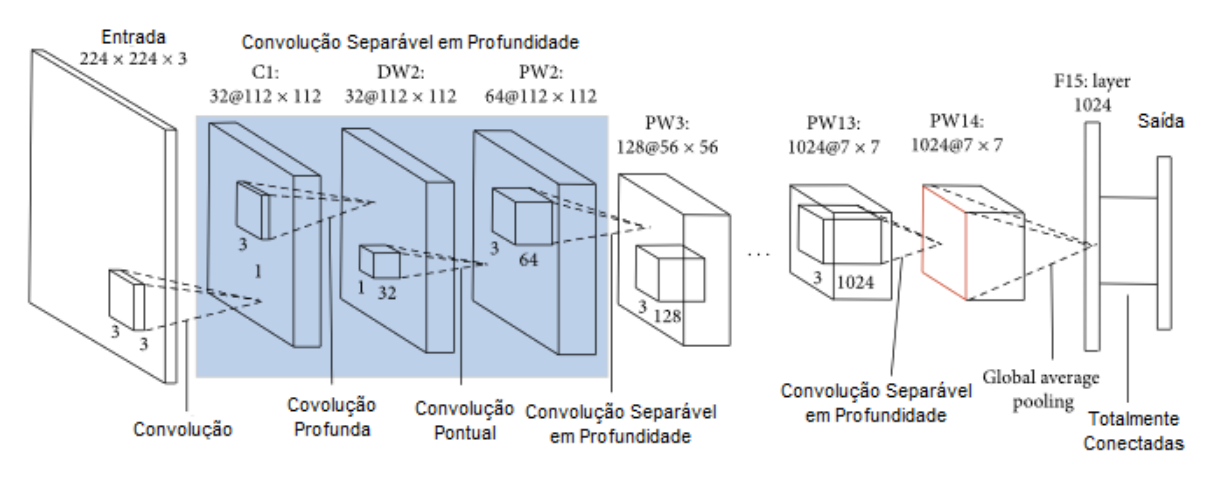


Fonte: (GOOGLE DEVELOPERS, 2021)

que usa convoluções separáveis em profundidade para construir redes neurais profundas de peso leve. A arquitetura geral da MobileNet possui 30 camadas.

Dividida em MobileNet V1 (HOWARD *et al.*, 2017) e V2 (SANDLER *et al.*, 2018), sendo que a MobileNet V2 possui gargalos lineares entre as camadas e conexões de atalho entre os gargalos, o que, conforme os autores, a deixa mais rápida, especificamente, este modelo atinge maior precisão utilizando duas vezes menos operações e 30% menos parâmetros. A arquitetura da MobileNet esta ilustrada na Figura 10.

Figura 10: Arquitetura do modelo MobileNet



Fonte: (WANG *et al.*, 2020)

2.4 Redes Neurais Siamesas

O primeiro trabalho usando Redes Neurais Siamesas foi o de Bromley *et al.* (1994), e tinha como objetivo detectar assinaturas falsas. Desde o seu surgimento, essa arquitetura tem sido empregado em diversos campos diferentes, e se generalizou em aplicações de informática e tecnologia da informação (CHICCO, 2021).

A Rede Neural Siamesa é composta por duas CNNs, que compartilham os mesmos pesos. As CNNs processam duas imagens de entrada e têm como saída um vetor de características para cada uma das entradas. O algoritmo compara a saída da rede neural superior e a saída da rede neural inferior por meio de uma medida de distância. Por meio dessa medida de similaridade, a rede neural afirma se os dois perfis são iguais ou diferentes (CHICCO, 2021).

Dentre as métricas de distância mais usadas em Redes Neurais Convolucionais, estão (BELLET; HABRARD; SEBBAN, 2015):

- **Distância Euclidiana:** medida amplamente usada em SNNs, é a distância entre dois pontos, que pode ser provada pela aplicação repetida do teorema de Pitágoras, A Distância Euclidiana e calculada conforme a Equação 7.

$$D_{Euclidiana}(x_i, x_j) = \sqrt{(x_i - x_j)^2} = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (7)$$

- **Distância de Manhattan:** uma forma de geometria em que a usual métrica

da geometria euclidiana é substituída por uma nova métrica em que a distância entre dois pontos é a soma das diferenças absolutas de suas coordenadas, definida pela Equação 8.

$$D_{Manhattan}(x_i, x_j) = \sqrt{\sum_{k=1}^d |x_{ik} - x_{jk}|} \quad (8)$$

- **Similaridade de Cosseno:** é uma medida da similaridade entre dois vetores num espaço vetorial que calcula o valor do cosseno do ângulo compreendido entre eles. É definido pela Equação 9.

$$D_{Cosseno}(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \cdot \|x_j\|} \quad (9)$$

2.5 Medidas de Avaliação

Ao desenvolvermos classificadores usando técnicas de aprendizado de máquina, é crucial que usemos as métricas de avaliação apropriadas para cada problema. O valor delas reflete a qualidade de um modelo, portanto se forem mal escolhidas, será impossível avaliar se o modelo de fato está atendendo aos requisitos necessários. Existem casos que determinados erros apresentam pesos diferentes que os demais. Dentre as principais métricas de avaliação, temos (SAMMUT; WEBB, 2011):

- **Matriz de confusão:** uma forma bastante simples de visualizar a performance de um modelo de classificação é através de uma matriz de confusão. Esta matriz indica quantos exemplos existem em cada grupo: falso positivo (FP), falso negativo (FN), verdadeiro positivo (TP) e verdadeiro negativo (TN). É interessante visualizar a contagem destes grupos tanto em números absolutos quanto em porcentagens da classe real, já que o número de exemplos em cada classe pode variar.

Tabela 1: Matriz de confusão

	Positivo	Negativo
Verdadeiro	TP	TN
Falso	FP	FN

Fonte: O autor

- **Acurácia:** Indica quantos exemplos foram classificados corretamente, independente da classe à qual pertencem. A fórmula é definida pela razão entre os casos classificados corretamente e todos os exemplos, como mostrado na Equação 10. A desvantagem desta métrica, é ela não representar de maneira ideal o desempenho em bases desbalanceadas. Se temos uma base com 100 exemplos, sendo 90 verdadeiros e 10 falsos, se o classificador atribuir que todos os exemplos têm valor verdadeiro, obterá uma acurácia de 90%, que parece ser um bom resultado, porém este resultado não representa o modelo adequadamente.

$$\mathbf{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

- **Precisão:** Esta medida é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e o total de exemplos classificados como positivos, expresso na Equação 11. Esta métrica dá ênfase ao erro por falso positivo.

$$\mathbf{Precisão} = \frac{TP}{TP + FP} \quad (11)$$

- **Revocação:** Do inglês “Recall”, ao contrário da precisão, dá ênfase aos erros por Falso Negativo, é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e a quantidade de exemplos que são de fato positivos, Equação 12.

$$\mathbf{Recall} = \frac{TP}{TP + FN} \quad (12)$$

- **F1-Score:** Leva em consideração tanto a precisão quanto a revocação. Ela é definida pela média harmônica entre as duas, como mostra a Equação 13. Uma característica do F1-Score, é que se a precisão ou o recall forem baixos, esta métrica também terá um valor baixo. Desta forma, F1-Score só será alto, quando a precisão e a revocação também forem altas.

$$\mathbf{F1-Score} = \frac{Preciso \times Recall}{Preciso + Recall} \quad (13)$$

2.6 Trabalhos Correlatos

A classificação de espécies de plantas tem sido uma aplicação muito abordada na visão computacional (WÄLDCHEN; MÄDER, 2018). Em especial, a abordagem usando imagens de folhas vem sendo amplamente explorada, para tal, técnicas de aprendizado profundo podem ser usadas para a extração de diversas características das imagens de folhas (LEE *et al.*, 2017). Dessa maneira, tais técnicas vem sendo propostas para a classificação de espécie de plantas a partir de imagens de folhas (AZLAH *et al.*, 2019).

Devido ao aumento do número de pesquisas relacionadas a classificação de espécies de plantas, foram criadas bases de dados contendo imagens de folhas de diversas espécies, a fim de criar um meio de comparar os métodos de classificação desenvolvidos. Neste sentido, se destacam as bases de dados *Flavia* (WU *et al.*, 2007), *Swedish* (SÖDERKVIST, 2001), *LifeCLEF* (JOLY *et al.*, 2015), *LeafSnap* (KUMAR *et al.*, 2012), *Foliage* (KADIR *et al.*, 2012) e *MalayaKew* (LEE *et al.*, 2015).

São encontradas na literatura pesquisas usando diferentes características da folha para a classificação, como forma (ZHAO *et al.*, 2015), textura (KUMAR; REDDY; BORA, 2017) e cor (CAGLAYAN; GUCLU; CAN, 2013). Também existem trabalhos que fazem uma extração automática de características, usando principalmente CNNs (YALCIN; RAZAVI, 2016).

Com o objetivo de desenvolver um sistema de aprendizado profundo, capaz de extrair características automaticamente para a classificação de folhas de plantas por espécie, foi criado um modelo de CNN chamado *LeafNet* (BARRÉ *et al.*, 2017). A avaliação da acurácia do modelo proposto nas bases de dados *LeafSnap*, *Flavia* e *Foliage* revela um desempenho de 86,3%, 95,8% e 97,9%, respectivamente. Os resultados desse estudo, mostram que uma CNN pode fornecer uma melhor representação de características para imagens de folhas em comparação com as características extraídas manualmente.

Além disso, vem sendo usadas Redes Neurais pré-treinadas para a extração de características de imagens de folhas e classificação de espécie de plantas, conforme pode ser visto nos parágrafos a seguir.

O estudo de Dyrmann, Karstoft e Midtiby (2016) faz o ajuste fino da rede neural pré-treinada *VGGNet* para a classificação de espécies usando 10,413 imagens de folhas divididas de maneira desbalanceada em 22 classes, formada pela junção de 6 bases de dados. O objetivo da junção das bases seria aumentar a quantidade de imagens para o treinamento. O resultado final apresenta uma acurácia de 86,2% para a classificação

das bases juntas, o artigo enfatiza que o modelo teve problema para classificar classes específicas e relaciona isso a baixa taxa obtida em razão do número reduzido de imagens nessas classes.

O artigo de Tan *et al.* (2018) compara o uso da rede pré-treinada *AlexNet* para a extração de características de imagens da base MalayaKew, após extrair as características ele testa o uso dos classificadores *Support Vector Machine*(SVM), Rede Neural Artificial(ANN, do inglês, “*Artificial Neural Network*”), *K-Nearest Neighbor*(KNN) e *Naive Bayes* para as características obtidas. O melhor resultado obtido foi a combinação da rede *AlexNet* como extrator de características e a ANN como classificador, atingindo uma acurácia de 95,54%.

Raj e Vajravelu (2019) propõem uma arquitetura de aprendizagem profunda dupla (DDLA, do inglês “*Dual Deep Learning Architecture*”), que combina das arquiteturas *MobileNet* e *DenseNet* para extração de características das bases de dados *Flavia*, *Folio*, *Swedish* e *Leaf-12*. A pesquisa concatena as características extraídas pelas duas redes neurais e usa regressão logística para fazer a classificação. O método proposto obteve uma acurácia superior aos modelos que usam apenas as características de uma arquitetura de rede neural. A DDLA obteve acurácias de 98,71% (*Flavia*), 96,38% (*Folio*), 99,41% (*Swedish*) e 99,39% (*Leaf-12*).

No artigo de Beikmohammadi e Faez (2018) são realizados vários experimentos, no contexto da classificação de espécies de plantas, combinando a Rede Neural pré-treinada *MobileNet* com um classificador de Regressão Logística. A eficiência do método proposto é avaliada em dois conjuntos de dados, *Flavia* e *Leafsnap*, onde obteve acurácias de 99,6% e 90,54%, respectivamente.

Uma arquitetura que vem sendo usada para detectar a similaridade entre imagens é a Rede Neural Siamesa (DEY *et al.*, 2017; Melekhov; Kannala; Rahtu, 2016; Caye Daudt; Le Saux; Boulch, 2018). Esta técnica já vem sendo utilizada para classificar imagens de folhas. Wang e Wang (2019) usa Redes Neurais Siamesas para classificar espécies de plantas que possuem poucos exemplos no conjunto de teste, usando imagens das bases *Flavia*, *Swedish* e *Leafsnap*. Os autores apresentam o desempenho das Redes Siamesas com 5, 10, 15 e 20 exemplos por classe, e variam sua taxa de acerto entre 31% até 95%.

Outra abordagem usando Redes Neurais Siamesa é a de Araujo *et al.* (2022) que define o índice de similaridade das imagens com cada classe, através de uma análise da imagem da folha inteira e de imagens de partes aproximadas da folha. Foram usadas as bases *LeafSnap* e *LifeClef 2015* para testar a abordagem, onde foram observadas acurácias

de 96% e 86%, respectivamente.

Rahman *et al.* (2018), com o objetivo de aumentar o número de características extraídas, e conseqüentemente, melhorar o desempenho de uma Rede Neural Siamesa, usa em seu trabalho multi níveis de características. A abordagem consiste em concatenar na camada de saída original, a saída de cada um dos 5 blocos convolucionais presentes em um modelo de arquitetura *VGG-16*. O trabalho consiste em detectar mudanças em imagens de satélite e foi constatado que a maior acurácia é obtida quando combinados os blocos 4 e 5.

Um comparativo das abordagens apresentadas nesta seção seguidas de seus métodos de avaliação e resultados, pode ser observado na Tabela 2.

Tabela 2: Exemplos de trabalhos relacionados a classificação de espécies de plantas usando imagem de folhas.

Referência	Características	Modelo	Base	Resultado (acc)
(DYRMANN; KARSTOFT; MIDTIBY, 2016)	VGG-16	CNN	União de 6 datasets, totalizando 10413 imagens, divididas em 22 classes	86,2%
(KUMAR; REDDY; BORA, 2017)	forma e textura (KMs, Gabor e HOG)	SVM	Flavia	97,64% (Gabor + KMs)
(CAGLAYAN; GUCLU; CAN, 2013)	forma e cor	SVM, kNN, Naive Bayes e Random Forest	Flavia	96,32% (RF)
(BEIKMOHAMMADI; FAEZ, 2018)	MobileNet	CNN	Flavia LeafSnap	99,6% 90,54%
(TAN <i>et al.</i> , 2018)	AlexNet e D-leaf (criação dos autores)	SVM, ANN, kNN, Naive Bayes e CNN	Flavia Malaya Kew Swedish	94,63% 90,38% 98,09%
(RAJ; VAJRA-VELU, 2019)	MobileNet e DenseNet-121	DDNN	Flavia Folio Swedish Leaf-12	98,71% 96,36% 99,41% 99,39%
(BARRÉ <i>et al.</i> , 2017)	LeafNet (criação dos autores)	CNN	Leafsnap Foliage Flavia	86,3% 95,8% 97,9%
(WANG; WANG, 2019)	VGG, ResNet, ResNeXt, SENet e DenseNet	SNN	Flavia Swedish Leafsnap . .	95,32% 91,67% 95,75% S-Inception (n=20)
(ARAUJO <i>et al.</i> , 2022)	VGG-16	SNN com 2 pontos de vista	LifeCLEF Leafsnap	(S) 0,87 96%

Fonte: O Autor

2.7 Considerações Finais

A partir do que foi descoberto em trabalhos correlatos, podemos perceber que o uso de redes neurais siamesas pode gerar bons resultados na classificação de espécies de plantas. Em especial, o trabalho de Rahman *et al.* (2018) motiva o aprofundamento dos estudos sobre a combinação de camadas intermediárias de uma SNN.

3 MATERIAL E MÉTODOS

Este capítulo traz uma breve apresentação das bases de dados que serão utilizadas no desenvolvimento do trabalho, além de descrever as arquiteturas das Redes Neurais escolhidas e o método proposto para compará-las.

3.1 Bases de Dados

Para testar as abordagens propostas foram usadas as bases de dados *Flavia* e *MalayaKew*. Estas bases de imagens são descritas nas subseções a seguir.

3.1.1 *Flavia*

A base de dados *Flavia* (WU *et al.*, 2007) contém 1875 imagens de folhas de plantas, que são divididas em 32 classes, cada uma contendo entre 50 e 77 imagens, como mostra a Tabela 3. Essas folhas foram coletadas no campus da Universidade de Nanjing e no arboreto Sun Yat-Sen, Nanking, China. As imagens foram adquiridas por câmeras digitais ou scanners, em um fundo plano e branco, tendo apenas informações da folha na imagem. Exemplos de imagens da base *Flavia* podem ser observadas na Figura 11.

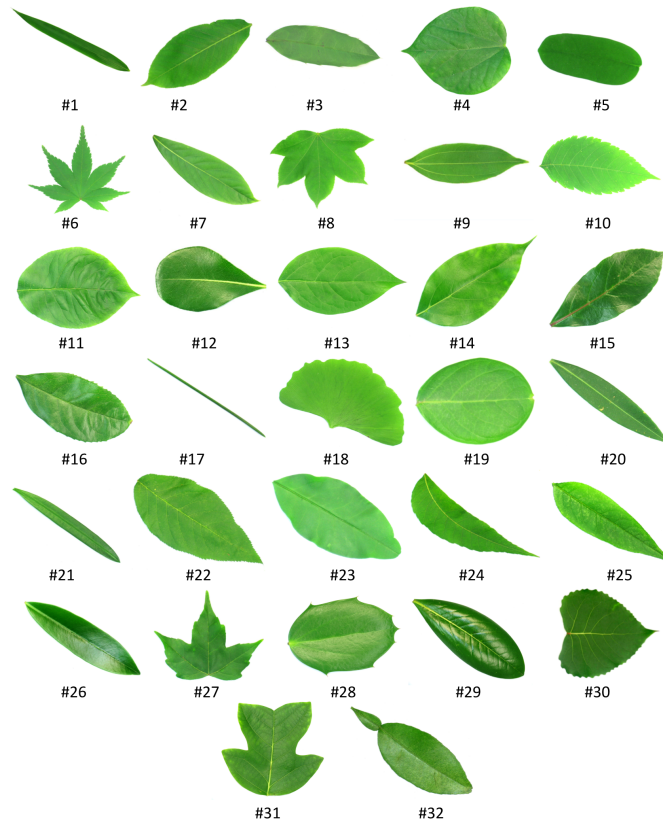
O principal problema encontrado nesta base é o de semelhança interespecie, tendo folhas de espécies diferentes muito similares, também são encontrados casos de variabilidade intraespécies, tendo imagens de folhas de mesma classe com formas e colorações diferentes.

3.1.2 *MalayaKew*

A base de dados *MalayaKew* (LEE *et al.*, 2015) foi coletada no *Royal Botanic Gardens*, Kew, Inglaterra. Consiste em imagens digitalizadas de folhas de 44 classes de espécies. É dividido em duas bases, a primeira contendo imagens da folha inteira, de tamanho 256×256 pixels, que contém 52 exemplos por classe na base de treinamento, totalizando 2288 imagens e 12 exemplos por classe na base de teste, totalizando 528 imagens, a segunda contém recortes do centro da folha, com resolução de 256×256 , sendo 34672 imagens para treinamento e 8800 para teste.

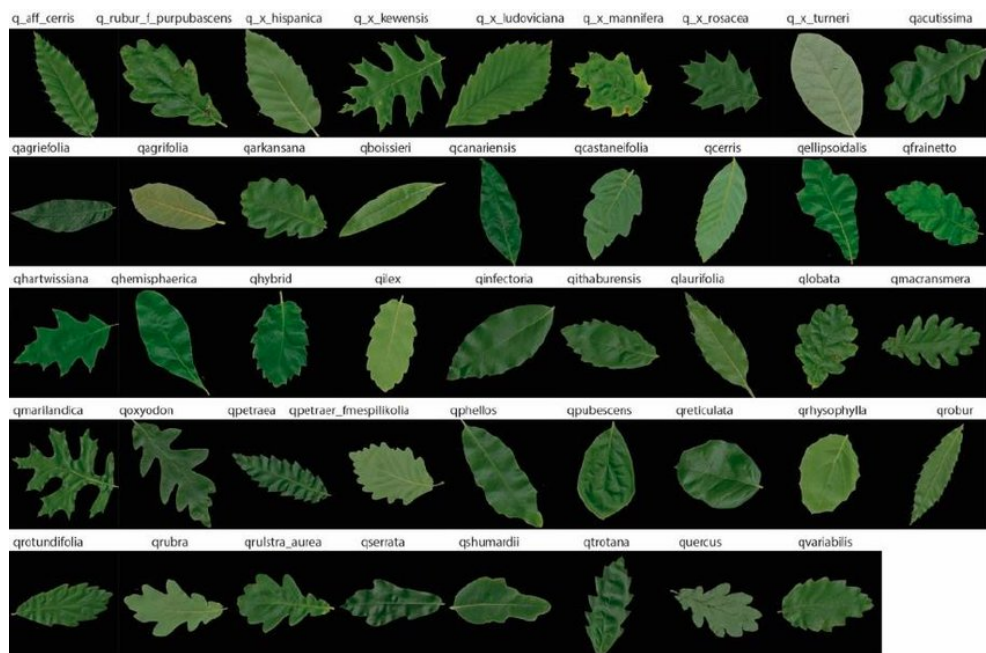
O principal problema encontrado nessa base é o de semelhança interespecie, tendo casos de folhas de espécies diferentes bem parecidas, como pode ser observado na Figura 12.

Figura 11: Exemplos de Imagens da Base Flavia



Fonte: Adaptado de (WU *et al.*, 2007)

Figura 12: Exemplos de Imagens da Base MalayaKew



Fonte: (LEE *et al.*, 2015)

Tabela 3: Distribuição de imagens por classe na base *Flavia*

Classe	Nome da Espécie	N de exemplos
1	pubescent bamboo	58
2	Chinese horse chestnut	62
3	Anhui Barberry	64
4	Chinese redbud	71
5	true indigo	72
6	Japanese maple	55
7	Nanmu	61
8	castor aralia	51
9	Chinese cinnamon	54
10	goldenrain tree	58
11	Big-fruited Holly	49
12	Japanese cheesewood	62
13	wintersweet	51
14	camphortree	64
15	Japan Arrowwood	59
16	sweet osmanthus	55
17	deodar	76
18	ginkgo, maidenhair tree	61
19	Crape myrtle, Crepe myrtle	60
20	oleander	65
21	yew plum pine	59
22	Japanese Flowering Cherry	54
23	Glossy Privet	54
24	Chinese Toon	64
25	peach	53
26	Ford Woodlotus	51
27	trident maple	52
28	Beale's barberry	54
29	southern magnolia	56
30	Canadian poplar	63
31	Chinese tulip tree	52
32	tangerine	55
-	Total	1875

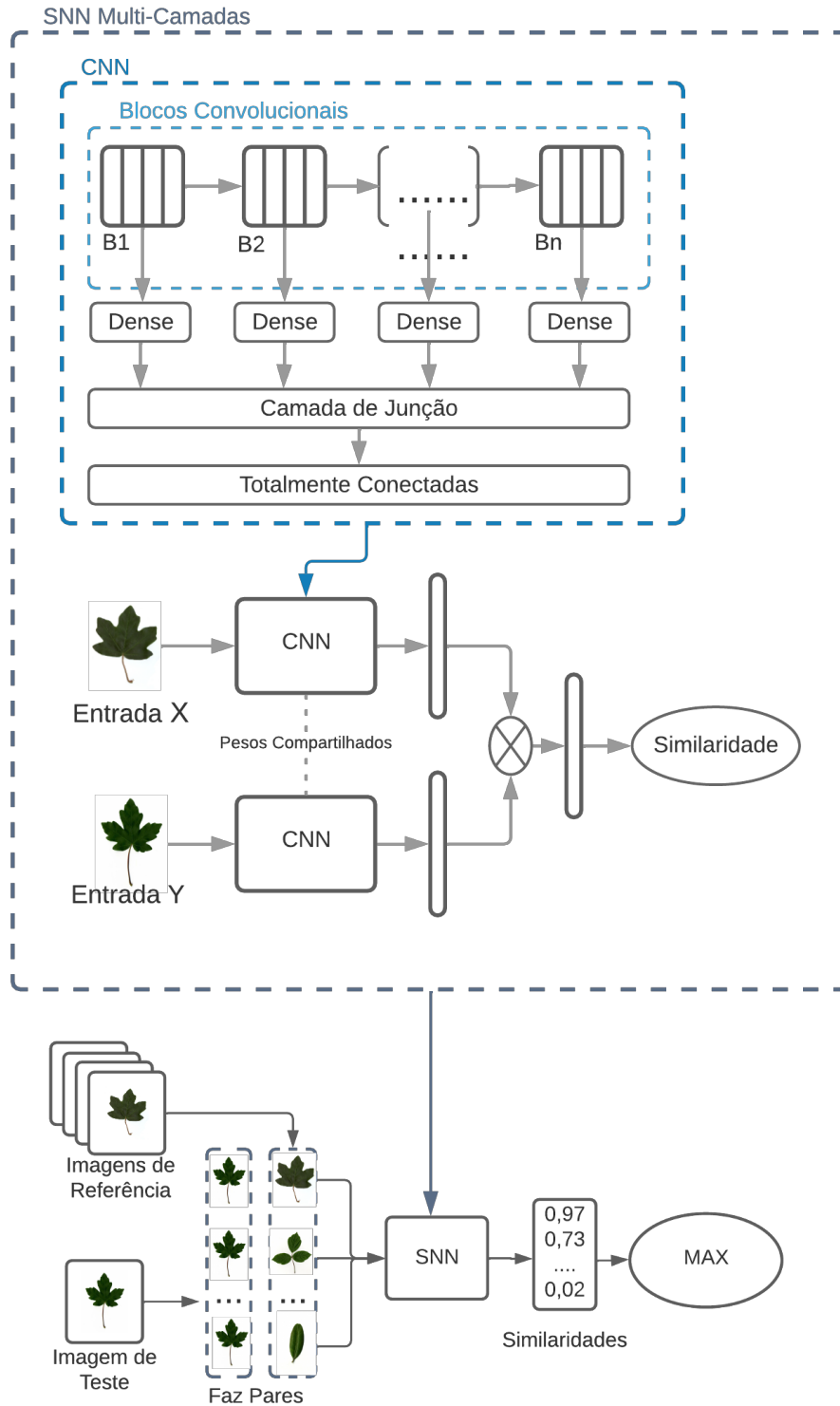
Fonte: Adaptado de (WU *et al.*, 2007)

3.2 Método proposto

O método para avaliar as arquiteturas SNN propostas é apresentado na Figura 13. A partir de um par de entrada (X, Y), uma SNN, constituída por CNNs e explorando a combinação de camadas intermediárias, calcula a similaridade das imagens de entrada. Durante o treino, em cada época, são formados pacotes contendo 64 exemplos de pares de imagens, onde 32 são exemplos positivos (par de imagens da mesma classe) e 32 são

exemplos negativos (par de imagens de classes diferentes). Na etapa de teste, uma imagem é comparada com imagens de referência que representam todas as classes do problema, e a maior similaridade determina a espécie da planta.

Figura 13: Modelo Proposto



Fonte: O Autor

Nota: CNN com características multicamadas, Arquitetura da Rede Neural Siamesa, e processo de classificação usando SNN.

No restante desta seção, são detalhadas as arquiteturas de CNN (Seção 3.2.1), arquiteturas de SNN (Seção 3.2.2), extração de características multicamadas (Seção 3.2.3) e o processo de seleção de imagens de referências (Seção 3.2.4).

3.2.1 Arquitetura da CNN

A fim de comparar o desempenho das redes neurais pré-treinadas encontradas na literatura, foi realizado o *fine-tuning* das redes e usando-as para classificar uma determinada base de dados.

O *fine-tuning* das redes foi feito a partir dos modelos encontrados na biblioteca *Keras*. Os modelos foram carregados já com os pesos treinados na *ImageNet* (RUSSAKOVSKY *et al.*, 2015). Para adaptação do modelo aos parâmetros exigidos na base de dados selecionada, última camada do modelo original, que se trata de uma camada densa com 1000 neurônios com ativação “*Softmax*”, foi substituída por uma camada densa com ativação “*Softmax*”, contendo n neurônios, sendo este o número de espécies presentes na base de dados. Além disso, foram congelados os pesos das camadas convolucionais das redes usadas, ou seja, tais pesos não serão alterados durante o treino da rede, apenas os pesos das camadas totalmente conectadas serão atualizados. Tais ajustes são realizados com a finalidade de padronizar as características extraídas e diminuir a complexidade do modelo, uma vez que, com os pesos congelados, diminui a quantidade de pesos a serem atualizados durante o treinamento.

Para o treino da rede foi usada a função de perda “*Categorical Crossentropy*” e como *optimizer* foi usada a função “adam”, com taxa de aprendizado de 10^{-3} . A rede foi treinada por 500 épocas e com um batch de tamanho igual a 64.

3.2.2 Arquitetura da SNN

A estrutura das Redes Neurais Siamesas, tem como entrada duas imagens, internamente, possui duas CNNs gêmeas com pesos compartilhados, sendo estas as mesmas descritas na Seção 3.2.1. Cada CNN extrai um vetor de características de uma imagem de entrada. Posteriormente, é calculada a distância entre os vetores extraídos usando a Eq. 7. O vetor de distâncias resultante passa por uma camada Densa com função de ativação do tipo “*sigmoid*”, que tem como saída apenas um valor variando entre 0 e 1, que representa a similaridade das duas imagens de entrada.

Durante o treino da Rede Neural Siamesa, as imagens foram organizadas em

batches contendo 64 exemplos cada. Os exemplos consistem de pares de imagens que podem ser da mesma classe (exemplos positivos) ou de classes diferentes (exemplos negativos). Metade de um *batch* é composto por exemplos positivos, rotulados com valor igual a 1, e a outra metade contém exemplos negativos, rotulados com o valor 0. Esse processo pode ser observado no Algoritmo 1. Os modelos foram treinados por 5000 épocas, usando a função de perda “*Binary Crossentropy*” e como *optimizer* a função “adam”, com taxa de aprendizado de 10^{-3} .

Algoritmo 1 Gerar batch

Entrada: $X, Y, tamanho_batch$

Saída: $Batch_x, Batch_y$

```

1▶  $Batch_y[1 : tamanho\_batch/2] \leftarrow 1$ 
2▶  $Batch_y[tamanho\_batch/2 : tamanho\_batch] \leftarrow 0$ 
3▶ para  $i=1$  até  $tamanho\_batch$  faça
4▶    $Batch_x[i][1] = selecionaImagemAleatoria(X)$ 
5▶   se  $Batch_y[i] == 1$  então
6▶      $Batch_x[i][2] = selecionaImagemDaMesmaClasse(X, Batch_x[i][1])$ 
7▶   senão
8▶      $Batch_x[i][2] = selecionaImagemDeOutraClasse(X, Batch_x[i][1])$ 
9▶   fim se
10▶ fim para

```

3.2.3 SNN com características multicamadas

Uma maneira de melhorar a representação de dados é extrair características de várias camadas de uma arquitetura convolucional é chamada de Característica Multinível ou Concatenação de Características Multicamada. A lógica é que as primeiras camadas fornecem representações mais próximas dos dados e as últimas estão próximas do problema. Portanto, a mesclagem dessas características deve fornecer representações mais detalhadas.

A estrutura da camada varia de acordo com as arquiteturas utilizadas. No entanto, passamos a saída dessas camadas por uma camada *Global Average Pooling* e depois por uma camada Densa com ativação *sigmoid*, com o objetivo de reduzir o número de características para 256. Depois, os vetores de 256 características extraídos de diferentes camadas são combinadas. Avaliamos dois esquemas de combinação: Concatenação (KERAS, b) e Adição (KERAS, a). Na concatenação, todas as informações presentes nas camadas são concatenadas em apenas uma camada. Já na Adição, as os neurônios das camadas de entrada são somados, gerando um único vetor, de tamanho igual ao das camadas de

entrada.

Tal processo é repetido para cada arquitetura, adaptando a estratégia aos blocos convolucionais de cada CNN.

3.2.4 Escolha das imagens de referência

Determinar as imagens de referência é um passo essencial para calcular a similaridade de classe. Para evitar casos de variabilidade intraclasse, um algoritmo (Algoritmo 2) determina um conjunto de imagens de referência para cada classe nos dados de treinamento. Inicialmente, o algoritmo seleciona aleatoriamente uma imagem de uma classe específica. Depois disso, o SNN calcula a similaridade desta imagem com as demais amostras de classe. Aquelas amostras que alcançaram uma similaridade abaixo de um limite pertencem ao conjunto de referência. Nesse caso, o limiar foi definido empiricamente em 0,9.

Algoritmo 2 Definição de Imagens de referência

Entrada: *Classes, Dados*

Saída: *referencias*

```

1▶ referencias ← []
2▶ para cada cls em Classes faça
3▶   img ← SelecionaImagemAleatoria(dados[cls])
4▶   referencias[cls].add(img)
5▶   similaridade_vec ← SNN(img, dados[cls])
6▶   para x até similaridade_vec faça
7▶     se x < 0.9 então
8▶       referencias[cls].add(dados[cls][x])
9▶     fim se
10▶   fim para
11▶ fim para

```

O limiar alto (0,9) faz-se necessário uma vez que deseja-se definir um conjunto de referência, que possua imagens distintas da mesma classe. Cada imagem de teste é comparada com todas as imagens de referência para realizar a classificação. A classe da imagem de referência com a maior similaridade determinará a classe da imagem de teste.

4 RESULTADOS EXPERIMENTAIS

Esta seção detalha os experimentos realizados para avaliar diferentes arquiteturas de SNN. O protocolo experimental está organizado da seguinte forma, três conjuntos de experimentos são descritos e os resultados correspondentes são discutidos. No primeiro conjunto, avaliamos o desempenho de diferentes CNNs e correspondentes SNNs compostos por elas, considerando dois conjuntos de dados para classificação de espécies vegetais. O segundo conjunto de experimentos trata da avaliação de estratégias distintas para combinar as camadas SNN. Adicionalmente, experimentos foram realizados considerando casos onde a base de treinamento tem o número de imagens reduzido. Por fim, a SNN foi testada levando em conta suas propriedades relacionadas ao aprendizado incremental.

As abordagens propostas foram desenvolvidas na linguagem Python, usando a plataforma *Google Colab* e os modelos de Rede Neural foram desenvolvidos usando a biblioteca *Keras*.

4.1 Definição das Arquiteturas de CNN e SNN

Para avaliar o desempenho das arquiteturas CNN e SNN, descritas na Seção 3.2.1 e 3.2.2, foram inicializadas com seus pesos da *Imagenet* e ajustadas para o reconhecimento de espécies de plantas de acordo com os hiperparâmetros definidos na Seção 3.2.1.

Os experimentos buscam investigar qual apresenta maior capacidade de generalização. O conjunto de treinamento é composto por várias amostras rotuladas dos conjuntos de dados *Flavia* e *MalayaKew* detalhados na Seção 3.1. As acurácias alcançadas estão descritas na Tabela 4.

Tabela 4: Acurácia da CNN e SNN (%) nas bases *Flavia* e *MalayaKew*

Arquiteturas	Bases de dados			
	<i>Flavia</i>		<i>MalayaKew</i>	
	CNN	SNN	CNN	SNN
<i>VGG-16</i>	97,9	97,7	90,9	70,8
<i>MobileNet</i>	99,5	98,8	96,5	84,5
<i>DenseNet201</i>	98,6	99,4	95,6	82,9
<i>ResNet50</i>	92,1	74,2	50,2	28,4
<i>Inception v3</i>	98,2	95,7	87,9	66,8

Fonte: O Autor

Conforme observado na Tabela 4, a CNN supera a SNN na maioria dos casos. A

lógica é que as características computadas pela última camada da arquitetura da CNN discrimina melhor que o espaço de similaridade calculada pela SNN. Embora o CNN-*MobileNet* tenha alcançado excelentes resultados, essas arquiteturas ainda necessitam de vastos conjuntos de dados e um número fixo de classes para aprender corretamente o problema.

Uma das hipóteses apresentadas é que a arquitetura CNN tem um impacto considerável no desempenho do modelo SNN. Um exemplo é um desempenho contrastante entre uma SNN usando a *DenseNet* e a *Resnet50* para o conjunto de dados *Flavia*. Percebe-se que ao mudar a arquitetura do *Densenet* pra uma SNN, houve um ganho de 0,8 pontos percentuais no desempenho, enquanto, para a mesma ação realizada com a arquitetura do *Resnet50*, houve uma queda de 17,9 pontos percentuais. Assim, confirmamos de forma assertiva à Hipotese 1 apresentada na Seção 1.4.

4.2 Uso de Multicamadas nas Redes Siamesas

Com o intuito de manter um método escalável, avaliou-se a utilização de multicamadas para melhorar a acurácia das arquiteturas de SNN, o que demonstrou ser uma abordagem viável para superar a falta de dados e a dificuldade de adicionar novas classes.

Para esta arquitetura foram selecionados o 3 modelos que tiveram melhor desempenho com SNN (Tabela 4), neste caso, *VGG-16*, *DenseNet* e *MobileNet*.

Para avaliar o desempenho dessa estratégia, combinamos até 5 blocos convolucionais de cada arquitetura. A organização dos blocos é diferente em cada arquitetura. A seleção das camadas de extração de características foi distribuída de maneira a abranger todo o modelo. As camadas selecionadas para cada modelo são as citadas na Tabela 5.

Tabela 5: Camadas internas selecionadas para extração de características

	<i>VGG-16</i>	<i>DenseNet</i>	<i>MobileNet</i>
B1	'block1_pool'	'pool1'	'conv_pw_2_relu'
B2	'block2_pool'	'pool2_pool'	'conv_pw_4_relu'
B3	'block3_pool'	'pool3_pool'	'conv_pw_6_relu'
B4	'block4_pool'	'pool4_pool'	'conv_pw_8_relu'
B5	'block5_pool'	'conv5_block32_concat'	'conv_pw_12_relu'

Fonte: O Autor

Lembrando que, como explicado na Seção 3.2.3, foram implementadas diferentes maneiras de combinar os vetores resultantes de cada bloco. O primeiro é uma concatenação direta de vetores inteiros. A segunda forma implementa um pool médio e uma camada

densa na parte inferior de cada bloco para padronizar sua saída em 256 características. O vetor de características final é gerado usando a camada de concatenação ou adição.

Embora a SNN tenha apresentado um alto desempenho no conjunto de dados *Flavia*, sem uma estratégia de combinação de camadas, uma questão razoável é se incluir a representação multicamada reduziria significativamente ou não o desempenho. Os resultados alcançados são mostrados na Tabela 6. Definimos as iniciais ‘ B_n ’, onde $n = \{1,2,3,4,5\}$, para informar qual conjunto de blocos convolucionais é utilizado para fornecer representações.

Tabela 6: Acurácia (%) dos modelos de SNN com características multicamadas na **Base *Flavia***

Arquitetura <i>VGG-16</i>			
Camadas Intermediárias	Vetor completo	Padronizado (256)	
	<i>Concatenate</i>	<i>Concatenate</i>	<i>Add</i>
Sem combinação	97,3	-	-
B5	92,8	95,9	93,3
B5+B4	96,3	98,4	97,0
B5+B4+B3	97,0	98,6	98,2
B5+B4+B3+B2	98,4	98,4	98,2
B5+B4+B3+B2+B1	97,0	97,2	97,9
Arquitetura <i>MobileNet</i>			
Camadas Intermediárias	Vetor completo	Padronizado (256)	
	<i>Concatenate</i>	<i>Concatenate</i>	<i>Add</i>
Sem combinação	98,7	-	-
B5	99,3	97,2	99,3
B5+B4	98,6	98,9	98,4
B5+B4+B3	98,2	99,4	99,4
B2+B3+B4+B5	98,0	99,3	99,8
B1+B2+B3+B4+B5	98,7	98,7	98,2
Arquitetura <i>DenseNet</i>			
Camadas Intermediárias	Vetor completo	Padronizado (256)	
	<i>Concatenate</i>	<i>Concatenate</i>	<i>Add</i>
Sem combinação	99,4	-	-
B5	98,0	99,1	99,5
B5+B4	98,7	99,4	99,8
B5+B4+B3	98,9	99,3	100,0
B2+B3+B4+B5	99,1	99,3	99,4
B1+B2+B3+B4+B5	99,1	99,8	99,8

Fonte: O Autor

Como se pode notar, a estratégia de multicamadas oferece pequenas alterações sem inconvenientes em relação ao desempenho apresentado na Tabela 4. De acordo com o processo usado para mesclar as informações fornecidas pelas diferentes camadas, essas representações de profundidade podem elevar as taxas de acurácia para perto de 99% para

todas as arquiteturas, corroborando o emprego de características multicamadas.

As vantagens da estratégia multicamadas ficam ainda mais claras no experimento com o conjunto de dados *MalayaKew* (Tabela 7). Sem características multicamadas, o *MobileNet* teve a melhor precisão, atingindo uma taxa de 84,4%. No entanto, aplicando uma representação multicamadas, as taxas aumentam até 3,4%. Outro resultado relevante foi obtido com o *DenseNet*, permitindo 94,3% de classificação correta. Da mesma forma, o *VGG-16* observa 18,4% de melhoria em comparação apenas com a SNN convencional.

Tabela 7: Acurácia (%) dos modelos de SNN com características multicamadas na **Base *MalayaKew***

Arquitetura <i>VGG-16</i>			
Camadas Intermediárias	Vetor completo	Padronizado (256)	
	<i>Concatenate</i>	<i>Concatenate</i>	<i>Add</i>
Sem combinação	70,8	-	-
B5	69,4	62,8	63,6
B5+B4	69,5	71,5	73,3
B5+B4+B3	84,4	75,5	76,1
B5+B4+B3+B2	81,2	88,2	85,0
B5+B4+B3+B2+B1	78,0	84,0	73,1
Arquitetura <i>MobileNet</i>			
Camadas Intermediárias	Vetor completo	Padronizado (256)	
	<i>Concatenate</i>	<i>Concatenate</i>	<i>Add</i>
Sem combinação	84,4	-	-
B5	80,6	81,2	83,3
B5+B4	80,4	82,9	85,8
B5+B4+B3	82,1	87,8	84,3
B2+B3+B4+B5	74,6	84,1	84,1
B1+B2+B3+B4+B5	83,9	83,5	81,1
Arquitetura <i>DenseNet</i>			
Camadas Intermediárias	Vetor completo	Padronizado (256)	
	<i>Concatenate</i>	<i>Concatenate</i>	<i>Add</i>
Sem combinação	82,9	-	-
B5	82,6	91,8	93,7
B5+B4	88,2	90,7	94,3
B5+B4+B3	88,1	87,8	92,4
B2+B3+B4+B5	89,0	89,3	87,3
B1+B2+B3+B4+B5	89,7	90,7	88,2

Fonte: O Autor

Esses experimentos em ambos os conjuntos de dados permitem concluir que um SNN treinado com arquitetura de combinação multicamada fornece uma solução viável para esse fim. Nesse caso, o *DenseNet* combinando os blocos B_5 e B_4 , fornece o melhor desempenho geral.

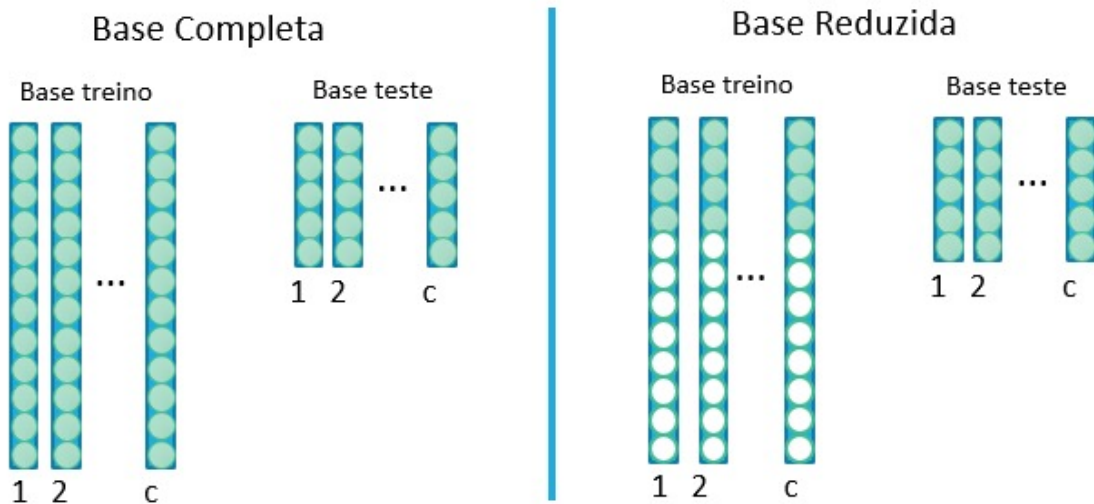
Além disso podemos confirmar a Hipótese 2 feita na Seção 1.4, uma vez que, o uso

de características multicamadas causou um impacto positivo no desempenho dos modelos em ambas as bases.

4.3 Avaliação dos Modelos em Bases Com Poucos Exemplos

A fim de avaliar o desempenho da SNN quando a base de dados está restrita a poucos exemplos para treinamento, os algoritmos apresentados na Seção 3.2 serão novamente utilizados, porém a base de treino foi reduzida e a base de teste mantida, assim como mostra a Figura 14. Neste caso, foi usada uma base de treino contendo apenas 10 imagens por classe.

Figura 14: Redução de exemplos na base de treino



Fonte: O Autor

Nota: (c) representa o número de classes presentes na base

Foram selecionadas as arquiteturas *VGG-16*, *MobileNet* e *DenseNet*, usados como CNN e SNN, para a SNN-multicamadas, selecionamos apenas as combinações de camadas que obtiveram melhores desempenhos nas Tabelas 4, 6 e 7 para realizar os testes com a base reduzida. Os resultados obtido foram organizados na Tabela 8, para a base *Flavia*, e na Tabela 9, para a base *MalayaKew*.

Tabela 8: Acurácia (%) dos modelos considerando 10 exemplos de treinamento por classe, na base *Flavia*

Modelo	CNN	SNN	SNN-multicamadas
<i>VGG-16</i>	92,1	91,8	95,3
<i>MobileNet</i>	98,1	98,4	98,7
<i>DenseNet</i>	98,4	97,5	98,6

Fonte: O Autor

Tabela 9: Acurácia (%) dos modelos considerando 10 exemplos de treinamento por classe, na base *MalayaKew*

Modelo	CNN	SNN	SNN-multicamadas
<i>VGG-16</i>	48,1	46,0	48,3
<i>MobileNet</i>	80,4	62,5	73,5
<i>DenseNet</i>	81,6	79,9	74,2

Fonte: O Autor

Nos resultados obtidos para a base *Flavia*, ocorreu uma queda de desempenho em todas as arquiteturas, porém, percebe-se que os modelos que usam de características multicamadas apresentam as melhores acurácias em todos os cenários, conforme a Tabela 8.

Já na base *MalayaKew* a queda de desempenho foi mais severa, com destaque para as arquiteturas *MobileNet* e *DenseNet*, onde o uso de CNNs proporcionou uma acurácia superior ao uso de SNNs-multicamadas, conforme a Tabela 9.

4.4 Avaliação de Escalabilidade

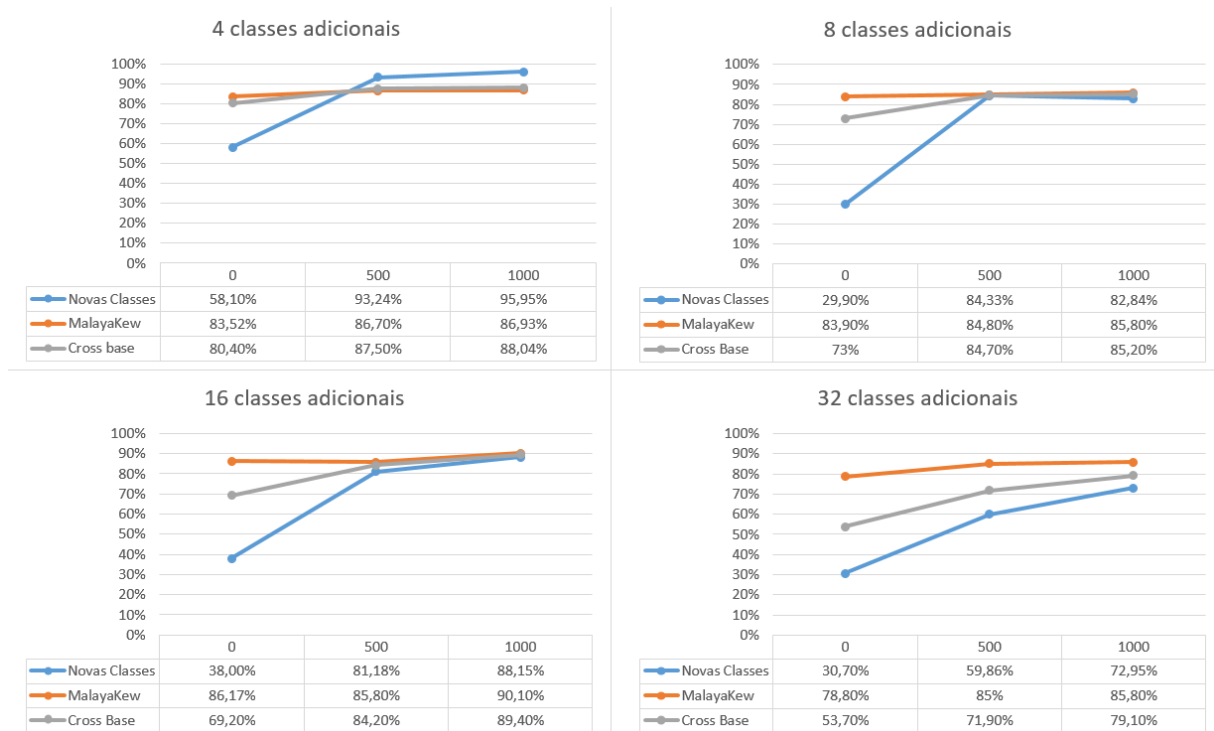
Para avaliar os modelos baseados em SNN quando novas classes são adicionadas ao problema, foram realizados experimentos com o *DenseNet*, adicionalmente, foram combinados os blocos convolucionais B_5 e B_4 , para representar os modelos de SNN-multicamadas. Este modelo foi selecionado por apresentar a melhor precisão nos experimentos descritos em Seção 4.2.

Um experimento foi realizado considerando o cruzamento das bases *MalayaKew* e *Flavia*. Para tal, considera-se o modelo SNN treinado com o conjunto de dados *MalayaKew* e é avaliado incrementalmente seu desempenho considerando $n = \{4, 8, 16, 32\}$ classes adicionais do conjunto de dados *Flavia* (não visto durante o treinamento). Além disso, tal experimento foi realizado considerando nenhum treinamento e um ajuste fino baseado em 500 e 1000 épocas. A fim de avaliar a SNN no contexto do aprendizado incremental, foi calculado para cada teste a acurácia do modelo usado para as imagens pertencentes a base *malayaKew*, a base *Flavia* e as bases cruzadas.

Os gráficos da Figura 15 mostram que os valores da acurácia para as imagens pertencentes originalmente a base *malayaKew*, representados pela cor Laranja, sofrem alterações leves durante todo o processo de aprendizado incremental, mostrando ser um modelo estável. Porém, apesar de manter o conhecimento adquirido anteriormente o modelo não se mostra eficiente em tratar os novos dados incorporados ao problema, neste

caso, as classes adicionais adquiridas da base *Flavia*, representadas nos gráficos em azul. As baixas taxas de acurácia obtidas para as novas classes, para os experimentos que não realizam treino extra, demonstram pouca plasticidade do modelo.

Figura 15: Desempenho do Modelo SNN-*DenseNet* em uma base de dados cruzada



Fonte: O Autor

Nota: Acurácia (%) do Modelo SNN-*DenseNet* na classificação da base de dados cruzada entre *MalayaKew* e *Flavia*, com 4, 8, 16 e 32 classes adicionais, quando variado o número de épocas de treino

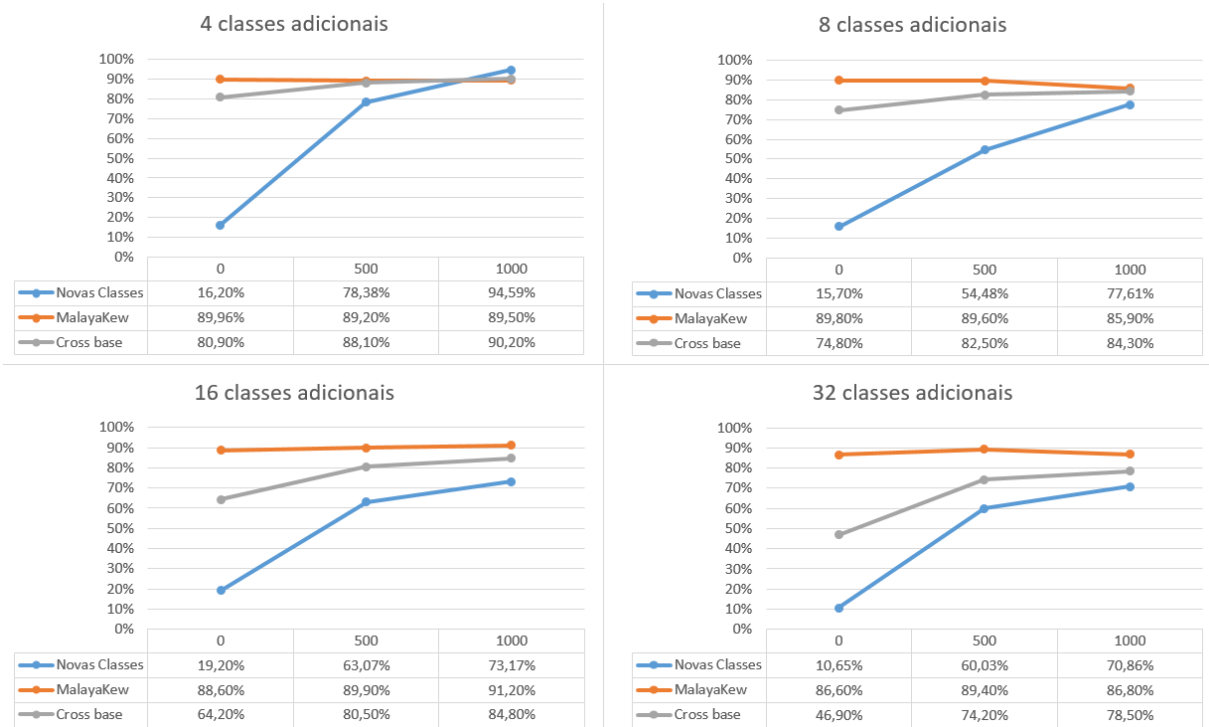
Apesar disso, os experimentos que fazem o ajuste fino do modelo, com treinamento extra por 500 e 1000 épocas, mostram uma melhora no desempenho nas novas classes e, ao mesmo tempo que se mantém o desempenho para as imagens das classes originais (base *MalayaKew*), mostrando ser capaz de adquirir novos conhecimentos, sem perder os conhecimentos adquiridos anteriormente.

Dessa forma, pode-se dizer que o modelo de aprendizado incremental usando uma SNN é estável e plástico, uma vez que ele mantém o desempenho na classificação das classes já conhecidas, e, com o retreino, aprende a classificar as novas classes inseridas.

Um comportamento semelhante é observado para os experimentos usando a SNN-multicamadas, como pode ser visto na Figura 16. Nestes é possível observar, que a acurácia do modelo nas imagens pertencentes a base *MalayaKew* (Linha Laranja), não sofrem alterações drásticas nos experimentos, mostrando a estabilidade do modelo. Ao mesmo tempo, a acurácia nas imagens pertencentes as classes adicionais (Linha Azul) mostram um

baixo desempenho logo após a inserção, e conforme é aplicado o retreino, o conhecimento atrelado aos novos dados é adquirido pelo modelo, aumentando a acurácia nas novas classes, mostrando a plasticidade do modelo.

Figura 16: Desempenho do Modelo SNN-*DenseNet* com características multicamadas em uma base de dados cruzada



Fonte: O Autor

Nota: Acurácia (%) do Modelo SNN-*DenseNet* com características multicamadas na classificação da base de dados cruzada entre *MalayaKew* e *Flavia*, com 4, 8, 16 e 32 classes adicionais, quando variado o número de épocas de treino

Sendo assim, pode-se dizer que o método proposto neste trabalho, referente à combinação de camadas intermediárias de uma SNN, mantém as propriedades de plasticidade e estabilidade, do aprendizado incremental, presentes na SNN convencional.

Conclui-se então, que o ajuste fino após inserção de novas classes, traz benefícios a plasticidade do modelo, de forma que após o retreino o modelo torna-se estável e plástico, tanto para a SNN convencional, quanto para a que combina camadas intermediárias, confirmando a Hipótese 3 definida na Seção 1.4, onde é dito que ao usar características multicamadas em uma SNN suas propriedades de aprendizado incremental são mantidas.

5 CONCLUSÃO

Podemos concluir que, apesar de ser uma abordagem recente neste tipo de aplicação, as Redes Neurais Siamesas apresentam um bom desempenho na classificação de espécies de plantas e mostram um grande potencial para melhorias, visto que, apresentam vantagens, quando comparadas com as Redes Neurais Convolucionais regulares.

Foi verificado que as Redes Neurais Siamesas podem apresentar perda de desempenho quando comparadas a modelos de CNN, apesar disso, é justificável o estudo de suas funcionalidades, como escalabilidade e desempenho em bases com poucas imagens para treino. Na verdade, um menor desempenho pode ser compensado pelo fato das CNN necessitarem de grandes quantidades de dados para treinar e de não serem escaláveis, enquanto as SNN se mostram melhores nestes cenários. Além disso, este trabalho mostra que o uso de características extraídas de camadas intermediárias pode aumentar o desempenho da Rede Siamesa, podendo superar até mesmo modelos de CNN similares, compensando a perda de desempenho inicial.

Com relação as hipóteses feitas no início do estudo, sobre a Hipótese 1, foi visto que as SNNs variam seu desempenho, dependendo da CNN pré-treinada usada, e a escolha do modelo pode sim influenciar no desempenho da Rede, destacam-se os modelos *MobileNet* e *DenseNet*, que obtiveram os melhores resultados na maioria dos testes. Já em relação a Hipótese 2, pode-se dizer que o uso de características multicamadas influencia no desempenho da SNN, os testes realizados com as arquiteturas *VGG-16*, *MobileNet* e *DenseNet* mostram um ganho no desempenho com a combinação de características extraídas de camadas intermediárias. Por fim, os experimentos relacionados à Hipótese 3 mostram que o método proposto mantém suas propriedades de Aprendizado Incremental, mostrando ter plasticidade e estabilidade, durante o aprendizado de novas classes.

Por fim, pode ser dito, que os objetivos descritos no começo deste trabalho foram atingidos, a proposta inicial de analisar variações de SNNs foi satisfeita. Além disso, por decorrência deste trabalho, obteve-se um método de classificação de espécies de plantas usando imagens de folhas, baseado em Redes Neurais Siamesas, que combinam características extraídas de camadas intermediárias da rede. O método proposto se mostrou promissor durante os testes realizados nas bases *Flavia* e *MalayaKew*, obtendo acurácias que chegam a 100% e 94,3%, respectivamente. Adicionalmente, o método possui um aprendizado incremental, que possibilita a inserção de novas classes ao ambiente do problema.

É possível também, utilizar o modelo proposto em conjunto com um software, que

facilite a submissão de imagens de teste para a classificação, possibilitando que qualquer pessoa, através de um smartphone ou computador, utilize o sistema de classificação.

Futuramente, planeja-se investigar o impacto da combinação de camadas intermediárias em CNNs, além de verificar a capacidade da SNN-multicamadas em outros contextos, que não a classificação de espécies de plantas. Além disso, planeja-se investigar estratégias para reduzir o tempo e o consumo de armazenamento aprendendo uma representação baseada em vetores binários. A ideia é considerar na arquitetura SNN uma camada particular para transformar características de valores flutuantes para binários.

REFERÊNCIAS

- ALOYSIUS, N.; GEETHA, M. A review on deep convolutional neural networks. In: IEEE. **2017 International Conference on Communication and Signal Processing (ICCSP)**. [S.l.], 2017. p. 0588–0592.
- ANTONELLI, A. *et al.* **State of the World's Plants and Fungi 2020**. Royal Botanic Gardens, Kew. [S.l.: s.n.], 2020.
- ANTONELLI, A.; SMITH, R. J.; SIMMONDS, M. S. Unlocking the properties of plants and fungi for sustainable development. **Nature Plants**, v. 5, n. 11, p. 1100–1102, 2019.
- ARAUJO, V. M. *et al.* **Reconhecimento de espécies de plantas a partir da imagem da folha e do uso da Aprendizagem Profunda**. 125 p. Tese (Doutorado em Informática) — Pontifícia Universidade Católica do Paraná – PUCPR, Curitiba - PR, 2021.
- ARAUJO, V. M. *et al.* Two-view fine-grained classification of plant species. **Neurocomputing**, v. 467, p. 427–441, 2022.
- AZLAH, M. A. F. *et al.* Review on techniques for plant leaf classification and recognition. **Computers**, v. 8, n. 4, p. 77, 2019.
- BARRÉ, P. *et al.* Leafnet: A computer vision system for automatic plant species identification. **Ecological Informatics**, v. 40, p. 50–56, 2017.
- BEIKMOHAMMADI, A.; FAEZ, K. Leaf classification for plant recognition with deep transfer learning. In: IEEE. **2018 4th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)**. [S.l.], 2018. p. 21–26.
- BELLET, A.; HABRARD, A.; SEBBAN, M. Metric learning. **Synthesis lectures on artificial intelligence and machine learning**, v. 9, n. 1, p. 1–151, 2015.
- BROMLEY, J. *et al.* Signature verification using a "siamese" time delay neural network. **Advances in neural information processing systems**, p. 737–737, 1994.
- BUDA, M.; MAKI, A.; MAZUROWSKI, M. A. A systematic study of the class imbalance problem in convolutional neural networks. **Neural Networks**, v. 106, p. 249–259, 2018.
- CAGLAYAN, A.; GUCLU, O.; CAN, A. B. A plant recognition approach using shape and color features in leaf images. In: SPRINGER. **International Conference on Image Analysis and Processing**. [S.l.], 2013. p. 161–170.
- CARPENTER, G. A.; GROSSBERG, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. **Computer vision, graphics, and image processing**, v. 37, n. 1, p. 54–115, 1987.
- Caye Daudt, R.; Le Saux, B.; Boulch, A. Fully convolutional siamese networks for change detection. In: **2018 25th IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2018. p. 4063–4067.
- CHICCO, D. Siamese neural networks: An overview. **Artificial Neural Networks**, p. 73–94, 2021.

CIARELLI, P. M. **Modelo de Aprendizado Incremental Baseado em Uma Rede Neural com Arquitetura Adaptativa**. Tese (Doutorado) — Universidade Federal do Espírito Santo, Brazil, 2012.

CIARELLI, P. M.; OLIVEIRA, E.; SALLES, E. O. An incremental neural network with a reduced architecture. **Neural networks**, v. 35, p. 70–81, 2012.

DEY, S. *et al.* Signet: Convolutional siamese network for writer independent offline signature verification. **arXiv preprint arXiv:1707.02131**, 2017.

DHILLON, A.; VERMA, G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. **Progress in Artificial Intelligence**, v. 9, n. 2, p. 85–112, 2020.

DYRMANN, M.; KARSTOFT, H.; MIDTIBY, H. S. Plant species classification using deep convolutional neural network. **Biosystems Engineering**, v. 151, p. 72–80, 2016.

FERGUSON, M. *et al.* Automatic localization of casting defects with convolutional neural networks. In: . [S.l.: s.n.], 2017. p. 1726–1735.

GOOGLE DEVELOPERS. **Guia avançado do Inception v3 no Cloud TPU | Google Cloud**. 07 2021. Disponível em: <https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=pt-br>. Acesso em: 18 ago. 2021.

HE, K. *et al.* Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

HOWARD, A. G. *et al.* Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.

HUANG, G. *et al.* Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 4700–4708.

JOLY, A. *et al.* Lifeclef 2015: multimedia life species identification challenges. In: SPRINGER. **International Conference of the Cross-Language Evaluation Forum for European Languages**. [S.l.], 2015. p. 462–483.

KADIR, A. *et al.* Performance improvement of leaf identification system using principal component analysis. **International Journal of Advanced Science and Technology**, v. 44, p. 113–124, 2012.

KERAS, D. do. **Add layer**. Disponível em :https://keras.io/api/layers/merging_layers/add/. Acesso em: 17 mar. 2022.

KERAS, D. do. **Concatenate layer**. Disponível em: https://keras.io/api/layers/merging_layers/concatenate/. Acesso em: 17 mar. 2022.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Communications of the ACM (A1)**, v. 60, n. 6, p. 84–90, 2017.

KUMAR, N. *et al.* Leafsnap: A computer vision system for automatic plant species identification. In: SPRINGER. **European conference on computer vision**. [S.l.], 2012. p. 502–516.

- KUMAR, T. P.; REDDY, M. V. P.; BORA, P. K. Leaf identification using shape and texture features. In: SPRINGER. **Proceedings of international conference on computer vision and image processing**. [S.l.], 2017. p. 531–541.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, v. 521, n. 7553, p. 436–444, 2015.
- LEE, S. H. *et al.* How deep learning extracts and learns leaf features for plant classification. **Pattern Recognition**, v. 71, p. 1–13, 2017.
- LEE, S. H. *et al.* Deep-plant: Plant identification with convolutional neural networks. In: IEEE. **2015 IEEE international conference on image processing (ICIP)**. [S.l.], 2015. p. 452–456.
- Melekhov, I.; Kannala, J.; Rahtu, E. Siamese network features for image matching. In: **2016 23rd International Conference on Pattern Recognition (ICPR)**. [S.l.: s.n.], 2016. p. 378–383.
- O'MAHONY, N. *et al.* Deep learning vs. traditional computer vision. In: SPRINGER. **Science and information conference**. [S.l.], 2019. p. 128–144.
- RAHMAN, F. *et al.* Siamese network with multi-level features for patch-based change detection in satellite imagery. In: IEEE. **2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)**. [S.l.], 2018. p. 958–962.
- RAJ, A. P. S. S.; VAJRAVELU, S. K. Ddla: dual deep learning architecture for classification of plant species. **IET Image Processing**, v. 13, n. 12, p. 2176–2182, 2019.
- RUSSAKOVSKY, O. *et al.* Imagenet large scale visual recognition challenge. **International journal of computer vision**, v. 115, n. 3, p. 211–252, 2015.
- SAMMUT, C.; WEBB, G. I. **Encyclopedia of machine learning**. [S.l.]: Springer Science & Business Media, 2011.
- SANDLER, M. *et al.* Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 4510–4520.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- SÖDERKVIST, O. **Computer vision classification of leaves from swedish trees**. 2001.
- SØGAARD, H. T. Weed classification by active shape models. **Biosystems engineering**, v. 91, n. 3, p. 271–281, 2005.
- ŠULC, M.; MATAS, J. Fine-grained recognition of plants from images. **Plant Methods**, v. 13, n. 1, p. 1–14, 2017.
- SZEGEDY, C. *et al.* Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2017. v. 31, n. 1.

- SZEGEDY, C. *et al.* Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9.
- TAN, J. W. *et al.* Deep learning for plant species classification using leaf vein morphometric. **IEEE/ACM transactions on computational biology and bioinformatics**, 2018.
- THORN, J.; GARDENS, K. **State of the World’s Plants 2016**. [S.l.: s.n.], 2016.
- VOULODIMOS, A. *et al.* Deep learning for computer vision: A brief review. **Computational intelligence and neuroscience**, v. 2018, 2018.
- WÄLDCHEN, J.; MÄDER, P. Plant species identification using computer vision techniques: A systematic literature review. **Archives of Computational Methods in Engineering**, v. 25, n. 2, p. 507–543, 2018.
- WANG, B.; WANG, D. Plant leaves classification: A few-shot learning method based on siamese network. **IEEE Access**, v. 7, p. 151754–151763, 2019.
- WANG, G.; SUN, Y.; WANG, J. Automatic image-based plant disease severity estimation using deep learning. **Computational intelligence and neuroscience**, v. 2017, 2017.
- WANG, W. *et al.* A novel image classification approach via dense-mobilenet models. **Mobile Information Systems**, v. 2020, 2020.
- WU, S. G. *et al.* A leaf recognition algorithm for plant classification using probabilistic neural network. In: IEEE. **2007 IEEE international symposium on signal processing and information technology**. [S.l.], 2007. p. 11–16.
- YALCIN, H.; RAZAVI, S. Plant classification using convolutional neural networks. In: IEEE. **2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)**. [S.l.], 2016. p. 1–5.
- ZHANG, S. *et al.* Plant species recognition methods using leaf image: Overview. **Neuro-computing**, 2020.
- ZHAO, C. *et al.* Plant identification using leaf shapes—a pattern counting approach. **Pattern Recognition**, v. 48, n. 10, p. 3203–3215, 2015.
- ZHUANG, F. *et al.* A comprehensive survey on transfer learning. **Proceedings of the IEEE**, v. 109, n. 1, p. 43–76, 2020.