



Erico André da Silva

Comparação de Métodos baseados em Redes Convolucionais para Classificação de Fava

Recife

2022

Erico André da Silva

Comparação de Métodos baseados em Redes Convolucionais para Classificação de Fava

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Rodrigo Gabriel Ferreira Soares

Recife

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

E68c Silva, Erico André da Silva
Comparação de Métodos baseados em Redes Convolucionais para Classificação de Fava / Erico André da Silva Silva.
- 2022.
50 f. : il.

Orientador: Rodrigo Gabriel Ferreira Soares.
Inclui referências.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, , Recife, 2022.

1. Aprendizado de máquina. 2. Classificação de imagens. 3. Reconhecimento de imagens. 4. Classificação de fava. I. Soares, Rodrigo Gabriel Ferreira, orient. II. Título

CDD

Erico André da Silva

Comparação de Métodos baseados em Redes Convolucionais para Classificação de Fava

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Recife, 26 de maio de 2012:

Prof. Dr. Rodrigo Gabriel Ferreira Soares
Orientador
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Prof. Dr. Roberta Macedo Marques
Gouveia
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Recife
2022

*Este trabalho é dedicado à Minha família e
especialmente à Vó Maria do Carmo (em memória).*

Agradecimentos

A minha esposa Graciele, por toda paciência e compreensão. Por entender meus momentos de ausência e por estar sempre presente nos momentos mais difíceis da minha vida acadêmica. À meu orientador professor Rodrigo Gabriel Ferreira Soares, por toda atenção, paciência, apoio e incentivo, pelo exemplo de professor. Pela dedicação na orientação deste trabalho. Os agradecimentos principais são direcionados à toda equipe da PROGET, Prof. Severino Mendes Jr, Prof. Marliete, Betânia, Severino, João, Simone e toda a equipe do O Restaurante Universitário (RU), Pela dedicação que me fizeram chegar até aqui e ir além assim como faz com todos os discentes.

*“A imaginação muitas vezes nos leva a mundos que nunca
existiram, mas sem isso nós não vamos a lugar algum.”
(Carl Sagan)*

Resumo

A cultura da fava tem recebido pouca atenção por parte dos órgãos de pesquisa e extensão, o que tem resultado em limitações do conhecimento das características agronômicas da cultura. Isso tem afetado a precisão em classificá-las. Tal classificação é de grande importância porque a identificação correta de plantas permite boa resposta da cultura em termos de produtividade e comportamento em diferentes condições ambientais. Neste contexto de informações limitadas sobre as características apresentamos uma solução que aplica o poder da visão computacional à agronomia, que visa melhorar a produtividade, reduzir desperdícios e auxiliar na tomada de decisões e na seleção de cultura que mais se adequa a uma região em particular. As técnicas de visão computacional são um conjunto de métodos utilizados para interpretar imagens, extraindo padrões e características. Visando contribuir com esse cenário de desenvolvimento tecnológico do setor do agrônomo, este trabalho compara algumas das abordagens de classificação supervisionada para identificação de forma automática de espécies de favas. O escopo deste trabalho consiste em classificar imagens de mudas geradas por produtores rurais em duas categorias de favas: orelha de vó e cearense. A partir das comparações realizadas entre métodos de classificadores que utilizam redes convolucionais como extratores de características, selecionamos o melhor método que foi a combinação de rede convolucional com máquina de vetores de suporte (SVM), para ao final apresentarmos esse método para automatizar a classificação das imagens de favas.

Palavras-chave: Aprendizado de máquina, Classificação de imagens, reconhecimento de imagens, classificação de fava.

Abstract

The bean crop has received little attention from research and extension agencies, which has resulted in limitations in the knowledge of the agronomic characteristics of the crop. This has affected the accuracy in classifying them. Such classification is of great importance because the correct identification of plants allows a good response of the culture in terms of productivity and behavior in different environmental conditions. In this context of limited information about the characteristics, we present a solution that applies the power of computer vision to agronomy, which aims to improve productivity, reduce waste and assist in decision-making and in the selection of culture that best suits a particular region. Computer vision techniques are a set of methods used to interpret images, extracting patterns and features. Aiming to contribute to this scenario of technological development in the agronomic sector, this work compares some of the supervised classification approaches for the automatic identification of broad bean species. The scope of this work is to classify images of seedlings generated by rural producers into two categories of beans: grandma's ear and cearense. From the comparisons made between methods of classifiers that use convolutional networks as feature extractors, we selected the best method that was the combination of a convolutional network with a support vector machine (SVM), to finally present this method to automate the classification of bean images.

Keywords: Machine Learning, Image Classification, Image Recognition, Fava Classification.

Lista de ilustrações

Figura 1 – Imagem Digital	16
Figura 2 – Representação de imagem preta e branca	16
Figura 3 – Representação de imagem no Modelo RGB.	17
Figura 4 – O espectro eletromagnético.	17
Figura 5 – Representação do Padrão de Cores RGB.	18
Figura 6 – Data Augmentation	19
Figura 7 – Diagrama de árvore de decisão	20
Figura 8 – Entropia	21
Figura 9 – O hiperplano ótimo separando os dados com a máxima margem ρ	22
Figura 10 – mostra a distância entre hiperplanos e os vetores suporte.	24
Figura 11 – Neurônios	25
Figura 12 – O modelo do neurônio artificial perceptron	26
Figura 13 – funções de ativação	26
Figura 14 – Exemplo de rede neural artificial do tipo <i>feedforward</i> com k camadas escondidas	27
Figura 15 – Esquema de uma Rede Neural Convolutacional.	28
Figura 16 – convolução em uma imagem	29
Figura 17 – Aplicação de max pooling/average pooling em uma imagem 4x4 utili- zando um filtro 2x2	29
Figura 18 – Operação de Flatten da camada Totalmente conectada	30
Figura 19 – A arquitetura da Rede	31
Figura 20 – Ilustração de cada tipo de partição, a versão clássica e o cross validation.	32
Figura 21 – Metodos de otimização dos hiper parâmetros	33
Figura 22 – Matriz de Confusão	33
Figura 23 – Curva ROC	35
Figura 24 – Arquitetura proposta para a detecção automatizada de mudas <i>feijão-fava</i>	37
Figura 25 – Favas	38
Figura 27 – Arquitetura da CNN	41
Figura 28 – Representação do método proposto nos experimentos	42
Figura 29 – CNN MLP Resultados	43
Figura 30 – CNN SVM Resultados	44
Figura 31 – CNN TREE Resultados	45
Figura 32 – Arquitetura VGG16 com MLP	46
Figura 33 – CNN VGG16 Resultados	47

Lista de tabelas

Tabela 1 – Kernel	24
Tabela 2 – Distribuição dos conjuntos de imagens	40
Tabela 3 – Hiperparâmetros	41
Tabela 4 – Hiper parâmetros dos Melhor Modelo CNN	44
Tabela 5 – Hiperparâmetros do Melhor Modelo SVM	45
Tabela 6 – Hiperparâmetros do Melhor Modelo Arvore	46
Tabela 7 – Hiperparâmetros do Melhor Modelo Arvore	47
Tabela 8 – Contem as métricas precisão e Sensibilidade	48

Lista de abreviaturas e siglas

RGB	Red Green Blue
HSL	Hue, Saturation, Lightness
HSV	Hue, Saturation, Value
IA	Inteligência Artificial
AM	Aprendizado de Máquina
SVM	Máquina de Vetores de Suporte
RNA	Rede Neural Artificial
CNN	Convolutacional Neural Network
MLP	Perceptrons de Múltiplas Camadas
DL	Deep Learning
ANN	Artificial Neural Networks
DNN	Deep Neural Network
PNN	redes neurais probabilísticas
ReLU	Unidade linear retificada
VP	Verdadeiro Positivo
FP	Falso Positivo
VN	Verdadeiro Negativo
FN	Falso Negativo
PIB	Produto Interno Bruto

Sumário

	Lista de ilustrações	8
1	INTRODUÇÃO	13
1.1	Justificativa	14
1.2	Objetivo	14
1.3	Objetivos Específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Processamento de Imagens	15
2.1.1	Imagem Digital	15
2.1.2	Cores e Padrão RGB	17
2.2	Extração de Características e Data Augmentation	18
2.3	Aprendizado de Máquina	19
2.3.0.1	Árvores de Decisão	20
2.3.0.2	Support Vector Machine (SVM)	22
2.3.0.3	Redes Neurais Artificiais	25
2.3.0.4	Rede Neural Convolucional	28
2.3.0.4.1	Convolução	28
2.3.0.4.2	Pooling	29
2.3.0.4.3	Camada totalmente Conectada	30
2.3.0.5	VGG 16	30
2.4	Métodos de Amostragem e Avaliação de Desempenho	31
2.4.1	Cross-Validation	31
2.4.2	Hiperparâmetros	32
2.4.3	Pesquisa em grade e aleatória	32
2.4.4	Medidas de Avaliação de Desempenho	33
3	TRABALHOS RELACIONADOS	36
4	METODOLOGIA	37
4.1	Aquisição das imagens e pré-processamento	38
4.2	Seleção de Modelos	40
5	RESULTADOS	43
5.0.1	CNN	43
5.0.2	SVM	44
5.0.3	Árvore de decisão	45

5.0.4	VGGNet	46
6	CONCLUSÕES	48
	REFERÊNCIAS	49

1 Introdução

A agricultura faz parte do setor primário da economia sendo fornecedor de alimento e de matéria-prima. Com a modernização e profissionalização, o setor deu um salto de importância na economia Brasileira. Atualmente é responsável por quase um quarto do Produto Interno Bruto (PIB) ([FIELDVIEW, 2021](#)), e praticamente metade das exportações do país.

A fava (*Phaseolus lunatus* L.) também conhecida como feijão-lima ou feijão-fava é uma leguminosa cultivada em quase todas as regiões do mundo, sendo que no Brasil possui ampla distribuição em todo o território, principalmente no Nordeste, é uma das quatro espécies do gênero *Phaseolus* exploradas comercialmente. A espécie foi domesticada na América do Sul ou Central, ou em ambas, e é subtropical ([ZIMMERMANN; TEIXEIRA, 1996](#)). É uma das principais leguminosas cultivadas na região tropical e apresenta potencial para fornecer proteína vegetal à população, diminuindo a dependência quase exclusiva dos feijões do grupo carioca ([VIEIRA, 1992](#)).

A cultura da fava apresenta-se como fonte de renda e alimento, sendo um recurso natural disponível e resistente aos períodos de seca ([REVISTACULTIVAR, 2015](#)). Permitindo que comunidades rurais, desestimuladas pela exploração agropecuária limitada principalmente pela falta de água, possam recuperar o desejo de permanência no campo, possibilitando melhoria na qualidade de vida, segurança alimentar e nutricional para a agricultura familiar apesar das adversidades climáticas.

Destacada a importância do feijão-fava à agricultura familiar, buscamos classificar de forma precoces as mudas, evitando assim o desperdício de tempo e recurso com variedades que tenham uma baixa produtividade. Pois foi verificado que as mudas de feijão-fava são praticamente idênticas nas etapas iniciais tornando difícil mesmo para especialistas classificá-las. A importância de se identificar as espécies se dá porque cada variedade tem características que se diferem uma das outras como o tamanhos de vagens e quantidade de grãos que podem variar para mais ou menos por safra, já a variedade orelha de vó tem o maior comprimento de suas vagens e maior peso de sementes que variedade fava-cearense. A escolha dessas duas espécies foi para resolver um problema de um aluno do curso de agronomia da Universidade Federal Rural de Pernambuco (UFRPE). Que era classificar de forma rápida e precisa as variedades de Fava Orelha de Vó e Fava Cearense.

Espera-se por meio desta pesquisa contribuir com o processo de classificação de determinadas variedades de fava como orelha de vó e fava Cearense, através das imagens podem ser obtidas a partir de dispositivos portáteis, trazendo economia com essa automação, neste contexto, a análise de imagens tem função valiosa ao transformar os dados coletados

no campo em informações úteis na tomada de decisão dos agricultores com a construção de um classificador automático.

1.1 Justificativa

A cultura da fava apresenta-se como fonte de renda e alimento, sendo um recurso natural disponível e resistente aos períodos de seca, permitindo que comunidades rurais, desestimuladas pela exploração agropecuária limitada principalmente pela falta de água, possam recuperar o desejo de permanência no campo, possibilitando melhoria na qualidade de vida, segurança alimentar e nutricional para a agricultura familiar apesar das adversidades climáticas. Além de ter sido um problema levando na época por um aluno do curso de agronomia da Universidade Federal Rural de Pernambuco (UFRPE).

Buscamos classificar nos estágios iniciais as mudas de feijão-fava, para que o agricultor saiba o manejo adequado evitando desperdícios de recursos, pois sabendo qual variedades tem um consumo mínimo de recursos com uma baixa produtividade ou maior consumo de recursos com produtividade maior. Assim o agricultor terá uma melhor tomada de decisão no manejo adequado a cultura de feijão-fava classificado, evitando o desperdício e uma baixa produtividade.

1.2 Objetivo

Este trabalho foca-se na construção de um agente classificador utilizando redes convolucionais como extrator de características em diferentes abordagem de classificadores para distinguir espécies de feijão-lima ou feijão-fava, dentre elas a fava cearense e a orelha de vó auxiliando o agricultor na escolha do manejo adequado a espécie classificada.

1.3 Objetivos Específicos

1. Estudar o estado da arte em classificação de imagens com CNN.
2. Testar métodos classificação automatizada de feijão-lima usando imagens digitais.
3. Demonstrar a viabilidade do tema.

2 Fundamentação Teórica

2.1 Processamento de Imagens

De acordo com Gonzalez e Woods (2010), “A visão é o mais avançado dos nossos sentidos, por isso não é surpreendente que as imagens desempenhem o papel mais importante na percepção humana”. Gonzalez e Woods (2010) relatam que não estão claros os limites entre o processamento de imagens digitais e a visão computacional, podendo-se considerar a divisão segundo o nível de operações realizadas em:

- Processo de Baixo Nível, que envolve as operações mais básicas e primitivas, sendo um pré-processamento de imagens, reduzindo imperfeições, realçando contrastes, aumentando a nitidez. E é caracterizado pelo fato de que a entrada e a saída serem imagens.
- Processo de Nível Médio, que envolve as operações de segmentação, de descrição e classificação. E é caracterizado pela entrada ser em geral uma imagem e a saída ser atributos extraídos dessa imagem.
- Processo de Alto Nível, que está mais relacionado a visão computacional propriamente dita. E é caracterizado pela análise dos atributos para dar sentido, para dar interpretação mais próxima possível da compreensão humana.

O pré-processamento tem a função de aprimorar a qualidade da imagem. Nesta etapa são utilizados filtros, ou seja, funções e matrizes matemáticas aplicadas na imagem original para que uma nova imagem seja formada. As transformações ou operações podem ocorrer no domínio espacial e/ou no domínio da frequência. As técnicas aplicadas no domínio espacial baseiam-se em filtros que manipulam o plano da imagem, enquanto que as do domínio da frequência se baseiam em filtros que agem sobre o espectro da imagem (GONZALEZ ; WOODS, 2010).

2.1.1 Imagem Digital

Uma imagem digital é considerada como uma matriz no qual os índices de linhas e colunas identificam um ponto na imagem, sendo que o respectivo valor do elemento dessa matriz identifica a intensidade (brilho) da imagem naquele ponto. A esses elementos de uma matriz digital dar-se o nome de “pixel” figura 1.

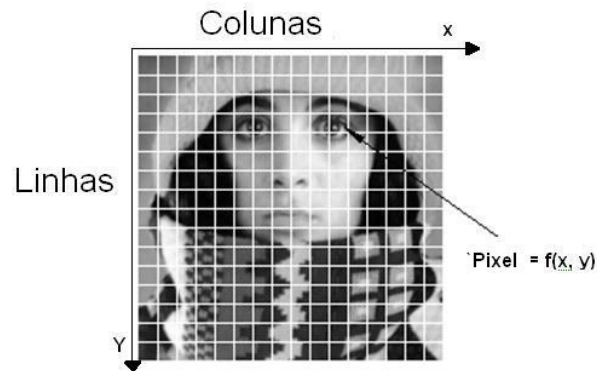


Figura 1 – Imagem Digital

Fonte: (GONZALEZ ; WOODS, 2010)

Em cada coordenada (x, y) da imagem preta e branca representa um valor que pode variar entre 0 a 255 denominado nível de cinza.

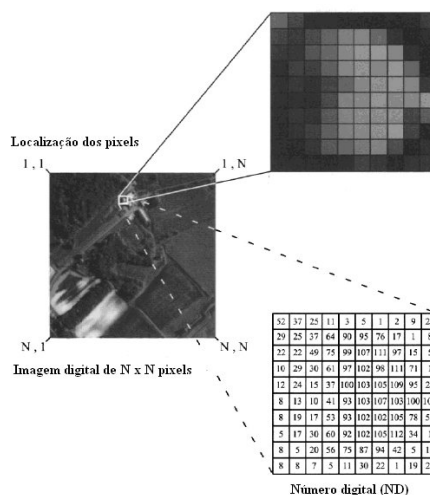


Figura 2 – Representação de imagem preta e branca

Fonte: (Ipma, 2002)

Já no caso de imagens coloridas as informações são armazenadas em intervalos ou bandas de frequência. No padrão RGB, por exemplo, as cores primárias vermelho (R -red), verde (G -green) e azul (B -blue), onde um pixel guarda a informação dos três canais de cores, essa informação está em um intervalo de 8 bits.

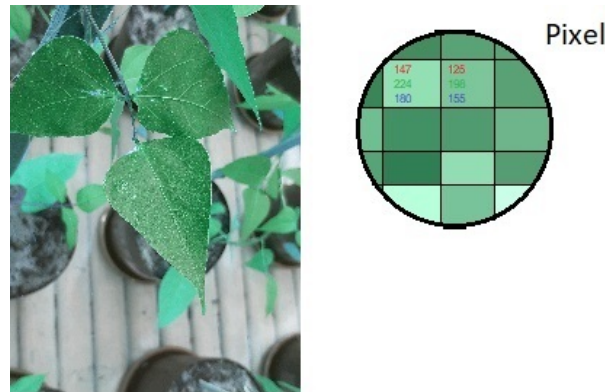


Figura 3 – Representação de imagem no Modelo RGB.

Fonte: Elaborada pelo autor (2022)

2.1.2 Cores e Padrão RGB

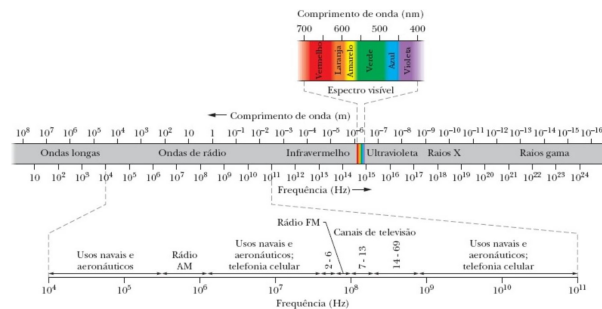


Figura 4 – O espectro eletromagnético.

Fonte: (HALLIDAY, 2016)

O que vemos como cores em uma luz, é na verdade, uma diferença de comprimento de onda eletromagnética apresentado na figura 4, ou seja, uma onda composta por campos elétricos e campos magnéticos alternados. Em um determinado intervalo de frequências do espectro eletromagnético, correspondente à luz visível, cada frequência vai resultar em uma sensação de cor (HALLIDAY, 2016).

Padrão de Cores RGB é um modelo aditivo no qual o vermelho, o verde e o azul (usados em modelos aditivos de luzes) são combinados de várias maneiras para reproduzir outras cores. O nome do modelo e a abreviação RGB vêm das três cores primárias: vermelho, verde e azul (Red, Green e Blue, em inglês).



Figura 5 – Representação do Padrão de Cores RGB.

Fonte: (GONZALEZ ; WOODS, 2010)

De acordo com Gonzalez e Woods (2010), todas as cores desse modelo são representadas por um cubo, onde as oito extremidades são a representação das cores vermelha, amarela, verde, ciano, azul, magenta, branca e preta. A diagonal que parte da origem (branco) até a extremidade mais distante do cubo (preto) forma uma escala de cinza, como mostra a Figura 5. Uma cor no modelo de cores RGB pode ser descrita pela indicação da quantidade de vermelho, verde e azul que contém. Cada uma pode variar entre o mínimo (completamente escuro) e máximo (completamente intenso). Quando todas as cores estão no mínimo, o resultado é preto. Se todas estão no máximo, o resultado é branco. Uma das representações mais usuais para as cores é a utilização da escala de 0 à 255, bastante encontrada na computação pela conveniência de se guardar cada valor de cor em 1 byte (8 bits).

2.2 Extração de Características e Data Augmentation

Extração de Características busca encontrar conhecimento a partir dos dados para uma melhor discriminação de classes. Características informativas são aquelas que melhoram a Generalização, pois além de conter informação relevante, não inclui características ruidosas, errôneas, enganadoras e inúteis. No caso das imagens tem como objetivo selecionar os dados de maior relevância e agrupá-los em um conjunto mais significativo, mais preciso e mais fácil de ser manipulado (GONZALEZ ; WOODS, 2010). O primeiro passo envolve a conversão de imagens coloridas em imagens com apenas um canal de cor (nível de cinza). Com base nos objetos detectados pelo processo de segmentação, diversas informações ou características podem ser obtidas a partir destes objetos, como histograma, tamanho, área, perímetro, textura, contornos, etc.

Data Augmentation é uma técnica que pode ser usada para expandir artificialmente o tamanho de um conjunto de treinamento, criando dados modificados a partir do existente, evitando também o overfitting em casos em que o conjunto de dados inicial é muito pequeno para treinar, ou mesmo se quiser extrair melhor desempenho de seu modelo.

Muitas vezes a obtenção de dados é difícil ou inviável por diversos motivos, e em decorrência disso, algumas práticas tornaram-se comuns para produzir novos exemplos a partir do conjunto de dados original. Na literatura existem diversos exemplos de práticas que costumam gerar bons resultados, entre elas estão: o espelhamento horizontal ou vertical da imagem; a rotação da imagem em torno do seu próprio eixo; a modificação das intensidades dos canais da imagem, isto é, incrementar ou diminuir os valores dos canais, ou mesmo misturar as intensidades entre os canais como mostra na figura 6.

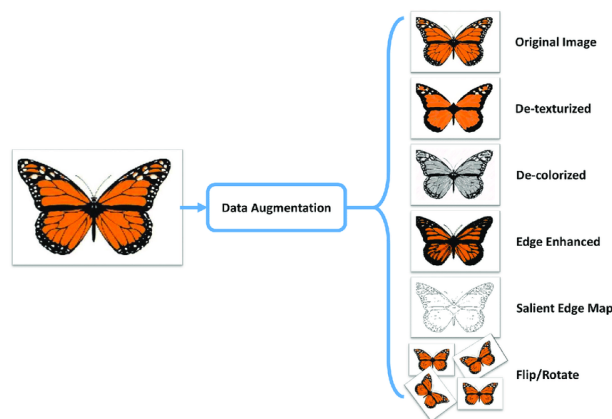


Figura 6 – Data Augmentation

2.3 Aprendizado de Máquina

Segundo [Mitchell et al \(1990\)](#), o aprendizado de máquina é definido por um sistema computacional que busca realizar uma tarefa T , aprendendo a partir de uma experiência E procurando melhorar uma performance P . Um algoritmo pode aprender a atingir um objetivo a partir de um grande volume de dados. Também esperamos que o computador acerte mais conforme lhe apresentamos mais dados, pois ele terá mais exemplos de padrões que pode generalizar para situações ainda não vistas. Assim quanto mais experiência, melhor será a nossa performance.

Aprendizado supervisionado Ocorre quando algoritmos são treinados com base de dados que possuem entradas e saídas bem definidos. Por exemplo, para classificar câncer de pele em uma imagem, é necessário ter a imagem (entrada) e o diagnóstico (saída).

Deep learning (aprendizado profundo) é uma classe de algoritmos de aprendizado de máquina que usa várias camadas para extrair progressivamente características de alto nível

da entrada. A maioria dos modelos modernos de aprendizado profundo é baseada em redes neurais artificiais, especificamente, Redes Neurais Convolucionais (CNN) ([APRENDIZADO DE MÁQUINA](#) , 2021).

Cada camada computacional da rede neural convolucional é composta por diferentes mapas de características, a primeira camada representacional pode abstrair os pixels e codificar bordas, a segunda camada pode compor e codificar arranjos de arestas, a terceira camada pode codificar um nariz e olhos, e a quarta camada pode reconhecer que a imagem contém um rosto. Fazendo com que seja possível lidar com variações, como distorção, rotação ou translação na imagem ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)).

2.3.0.1 Árvores de Decisão

As árvores são formadas por nós, que armazenam informação (perguntas). O nó raiz é o nó que possui maior nível hierárquico e, a partir dele, ramificam-se os nós filhos como mostra a figura 7. O nó que não possui filhos é conhecido como nó folha ou terminal.

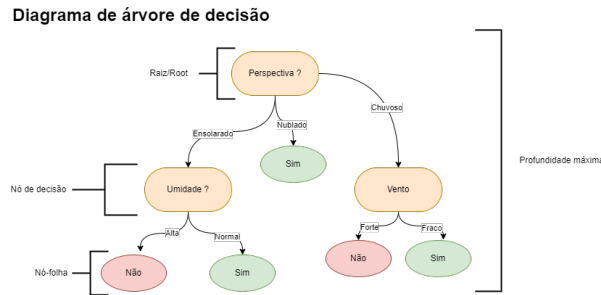


Figura 7 – Diagrama de árvore de decisão

Fonte: Elaborada pelo autor (2022)

Para se gerar um novo galho é necessário que ocorra uma separação dos dados no plano. No entanto, essa separação precisa melhorar a separação anterior, caso contrário não faz sentido separar mais os dados. Desse modo, utiliza-se alguns critérios para auferir a qualidade da separação. Para problemas de classificação, tem-se os critérios de Entropia.

Entropia Através da entropia o algoritmo verifica como os dados estão distribuídos nas variáveis preditoras de acordo com a variação da variável target. Quanto maior a entropia, maior a desordem dos dados, e quanto menor, maior será a ordem destes dados, quando analisados pela ótica da variável target. Dado um conjunto S , com instâncias pertencentes à classe i , com probabilidade p_i , temos:

$$Entropia = - \sum_{i=1}^n p(i) \log_2 p(i) \quad (2.1)$$

No exemplo apenas existem duas classes de classificação. Assim sendo, o valor da entropia varia de acordo com a figura 8:

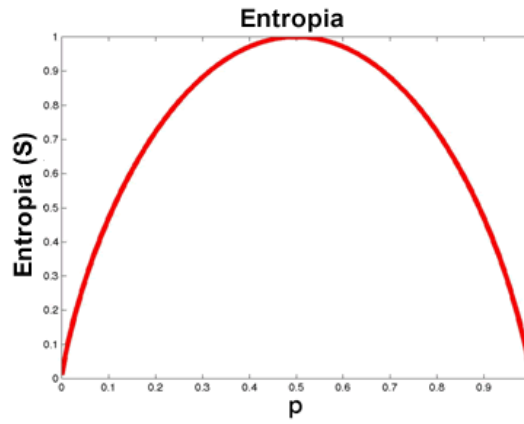


Figura 8 – Entropia

$$Entropia(s) = p_+ \log_2 p_+ - p_- \log_2 p_- \quad (2.2)$$

Onde:

- S é o conjunto de exemplo de treino;
- p_+ é a porção de exemplos positivos;
- p_- é a porção de exemplos negativos;

Suponha que S é uma coleção de 14 exemplos, incluindo 9 positivos e 5 negativos, Notação: $[9+, 5-]$. A entropia de S em relação a esta classificação booleana é dada por:

$$Entropy([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$$

Ganho de informação Representa a informação aprendida sobre os rótulos quando dividimos uma região do espaço em duas sub-regiões de acordo com ponto de corte. E é definida pela fórmula:

$$ganho(S, A) = entropia(S) - \sum_{v \in \text{valores}(A)} \frac{S_v}{S} entropia(S_v) \quad (2.3)$$

Onde:

- $|S_v|$ = Número de elementos em $|S_v|$.
- $|S|$ = Número de elementos em $|S|$.

- $\text{Valores}(A)$ = Conjunto dos valores possíveis de um atributo A.

Exemplo:

$$\text{ganho}(S, \text{Umidade}) = 0.940 - (7/14)0.985 - (7/14)0.592 = 0.151$$

Podagem Quando árvores de decisão são induzidas, muitas das arestas ou sub-árvores podem refletir ruídos ou dados ausentes. Ruídos referem-se a situações em que dois ou mais registros, composto por atributos que possuem os mesmos valores e que chegam a classes distintas, já dados ausentes, correspondem a registros que não possuem todos os valores dos atributos preenchidos. Em ambas as situações, os registros redundantes ou mal formados devem ser eliminados, ou modificados, de tal forma que tenham a mesma classe ou todos seus valores preenchidos, respectivamente. Uma maneira para detectar e excluir essas ramificações e subárvores é utilizando métodos de poda (pruning) da árvore, com o objetivo de melhorar a taxa de acerto do modelo para novos exemplos.

2.3.0.2 Support Vector Machine (SVM)

É uma técnica de classificação baseada em aprendizado de máquina na qual é utilizado o aprendizado supervisionado. Isso quer dizer que para seu treinamento deve se possuir uma base na qual as instâncias de cada classe que estejam previamente classificadas. O SVM tem o objetivo encontrar o hiperplano de separação ideal como mostra na figura 9, o qual maximiza a margem da base de treinamento buscando o equilíbrio entre ambos os erros, minimizando o excesso de ajustes (overfitting) e melhorando a capacidade de generalização.

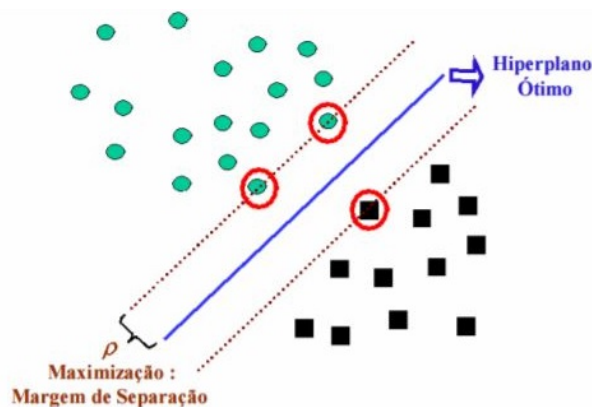


Figura 9 – O hiperplano ótimo separando os dados com a máxima margem ρ .

Um hiperplano de separação é uma linha que separa os pontos em classe distintas em um plano de dimensão N.

Hiperplano Ótimo Assumindo-se que o conjunto de treinamento é linearmente separável, o hiperplano ótimo é o hiperplano de separação com maior margem. O hiperplano ótimo é definido pela equação 2.4:

$$\langle w \cdot x_i \rangle + b = 0 \quad (2.4)$$

sendo w e b , o vetor peso e o bias respectivamente.

Considerando a restrição imposta pela Eq. abaixo, os classificadores lineares que separam um conjunto de treinamento possuem margem positiva. Ou seja, esta restrição afirma que não há nenhum dado entre $\langle w \cdot x_i \rangle + b = 0$ e $\langle w \cdot x_i \rangle + b \geq \pm 1$, sendo a margem sempre maior que a distância entre os hiperplanos $\langle w \cdot x_i \rangle + b = 0$ e $\langle w \cdot x_i \rangle + b = 1$. Devido a estas suposições as SVMs obtidas são normalmente chamadas de SVMs com margens rígidas (ou largas) como mostra a figura 10.

$$\begin{aligned} \langle w \cdot x_i \rangle + b &\geq +1, \text{ para } y_i = +1 \\ \langle w \cdot x_i \rangle + b &\leq -1, \text{ para } y_i = -1 \end{aligned}$$

Estas equações podem ser combinadas em:

$$y_i(\langle w \cdot x_i \rangle + b) \geq 1, i = \{1, 2, \dots, n\} \quad (2.5)$$

Seja $d_+(d_-)$ a distância euclidiana entre os vetores suporte positivos (negativos) e o hiperplano, definimos como margem ρ de um hiperplano de separação como sendo a maior margem geométrica entre todos os hiperplanos, podemos representar por $\rho = (d_+ + d_-)$. Denotaremos por $d_i(w, b; x_i)$, como a distância de um dado x_i ao hiperplano (w, b) , sendo calculado pela Eq 2.6.

$$d_i(w, b; x_i) = \frac{|\langle w \cdot x_i \rangle + b|}{\|w\|} = \frac{y_i(\langle w \cdot x_i \rangle + b)}{\|w\|} \quad (2.6)$$

levando em consideração a restrição imposta pela Eq 2.5, podemos escrever

$$d_i(w, b; x_i) \geq \frac{1}{\|w\|} \quad (2.7)$$

Com isso podemos identificar $\frac{1}{\|w\|}$ como o limite inferior da distância entre os vetores suporte x_i e o hiperplano de separação (w, b) , as distâncias d_+ e d_- ficam

$$d_+ = d_- = \frac{1}{\|w\|} \quad (2.8)$$

Como suposto anteriormente que a margem é sempre maior que a última instância, a minimização de $\|w\|$ leva a maximização da margem. A partir disto podemos definir a margem ρ através da Eq 2.9.

$$\rho = (d_+ + d_-) = \frac{2}{\|w\|} \quad (2.9)$$

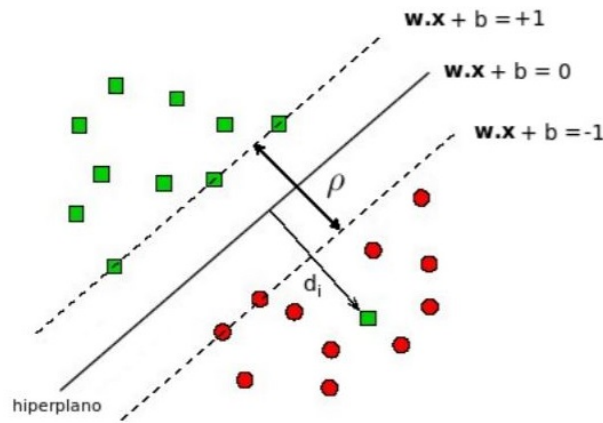


Figura 10 – mostra a distância entre hiperplanos e os vetores suporte.

O hiperplano ótimo é dado pela minimização da norma $\|w\|$, considerando a restrição da Eq 2.5.

SVM Kernel Os algoritmos SVM usam um conjunto de funções matemáticas que são definidas como kernel. A função do kernel é pegar dados como entrada e transformá-los na forma necessária. Diferentes algoritmos SVM usam diferentes tipos de funções de kernel. Essas funções podem ser de tipos diferentes. Por exemplo , linear, não linear, polinomial, função de base radial (RBF) e sigmoide. Algumas das funções Kernels mais utilizadas estão descritas na tabela abaixo.

Tabela 1 – Kernel

Tipo de Kernel	Expressão	Parâmetros
Polinomial	$K(x_i, x_j) = (\langle x_i \cdot x_j \rangle + 1)^p$	p
RBF	$K(x_i, x_j) = \exp\left(-\frac{\ x_i - x_j\ ^2}{2\sigma^2}\right)$	σ^2
Sigmoide	$K(x_i, x_j) = \tanh(\beta_0 \langle x_i \cdot x_j \rangle + \beta_1)$	β_0, β_1

Onde:

- A potencia p deve ser especificada pelo usuário, No caso de Kernels polinomiais, os mapeamentos também são funções polinomiais com complexidade crescente a medida que o indice p aumenta.

- A amplitude σ^2 e especificada pelo usuário, O Kernel Gaussiano corresponde a um espaço de características de dimensão infinita. Pode-se afirmar que quase todas formas de mapeamento podem ser implementadas por esta função em particular.
- Utilizado somente para alguns valores de β_0, β_1 , A utilização do Kernel Sigmoidal, por sua vez, permite explicitar uma Rede Neural Artificial do tipo Perceptron multicamadas.

2.3.0.3 Redes Neurais Artificiais

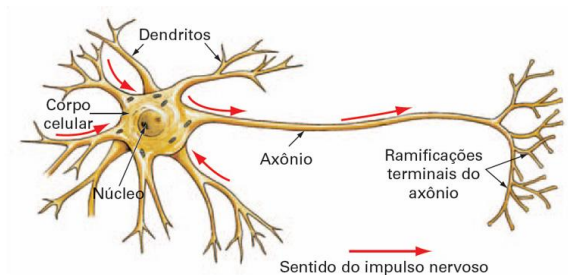


Figura 11 – Neurônios

Fonte: (Data Science Academy, 2021)

Os neurônios são as células que formam o nosso cérebro representado na figura 11. Elas são compostas basicamente por três partes: os detritos, que captam informações ou do ambiente ou de outras células, o corpo celular ou soma, responsável pelo processamento das informações, e um axônio, para distribuir a informação processada para outros neurônios ou células do corpo.

Assim como o nosso cérebro, uma rede neural artificial é um um sistema de processamento paralelo de informações constituído pela interconexão de unidades básicas de processamento, denominadas neurônios artificiais, que tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso Haykin, S. (2007). Todo conhecimento adquirido pela rede se da através de um algoritmo de aprendizagem, cuja função é modificar os pesos de conexões entre os neurônios da rede, conhecidos como pesos sinápticos, de forma ordenada a fim de alcançar o mapeamento desejado.

O Neurônio Artificial é a menor unidade de processamento de uma rede neural, que recebe sinais de entrada e produzem sinais de saída. O modelo de neurônio mais utilizado é o perceptron, representado na figura 12. Um único perceptron só pode resolver problemas linearmente separáveis. Basicamente, diz-se que um problema é linearmente separável se você pode classificar o conjunto de dados em duas categorias ou classes usando uma única linha.

Ele é composto por: m entradas $[x_1, \dots, x_m]$, m pesos sinápticos $[w_1, \dots, w_m]$, uma variável de deslocamento linear b (do inglês: bias) e uma saída y , que é descrita por:

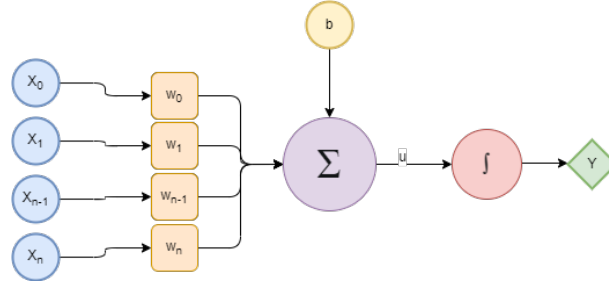


Figura 12 – O modelo do neurônio artificial perceptron

Fonte: Elaborada pelo autor (2022)

$$u = \sum_{i=1}^m w_i x_i + b = w_1 x_1 + w_m x_m + b \quad (2.10)$$

O termo adicional b provê um grau de liberdade a mais, que não é afetado pela entrada nessa expressão, correspondendo tipicamente ao "bias" (viés). O valor u passa então por uma função matemática de ativação $f(*)$, com a característica de ser não linear, responsável por limitar tal valor a um certo intervalo, produzindo o valor final de saída y do neurônio.

Algumas funções de ativação usadas são a degrau, sigmoide, tangente hiperbólica, Softmax e ReLU (Rectified Linear Unit). Ela é utilizada para adicionar uma não linearidade no sistema. Dentre as funções de ativação mais utilizadas estão a Sigmóide e a Relu, definidas pelas equações abaixo, respectivamente:

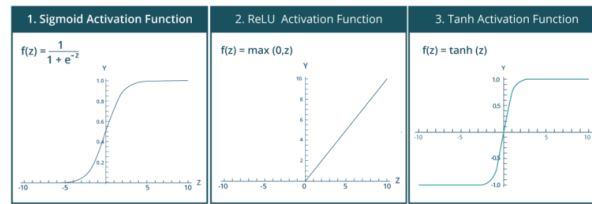


Figura 13 – funções de ativação

Arquitetura de uma RNA Para isso, a rede é dividida em três tipos de camadas: a de entrada, a escondida e a de saída. As camadas de entrada e saída são intuitivas e representam o número de entradas e saídas do problema em questão.

Redes neurais do tipo **Feedforward** (em português alimentação direta ou avante) são redes de múltiplas camadas na qual a informação só propaga em um sentido. No caso, os sinais provenientes dos neurônios de uma camada só podem estimular os neurônios da camada seguinte, não existindo realimentação. A figura 14 mostra uma rede neural artificial de múltiplas camadas do tipo Feedforward, com uma camada de entrada, com m entradas, k camadas escondidas e uma camada de saída com n saídas.

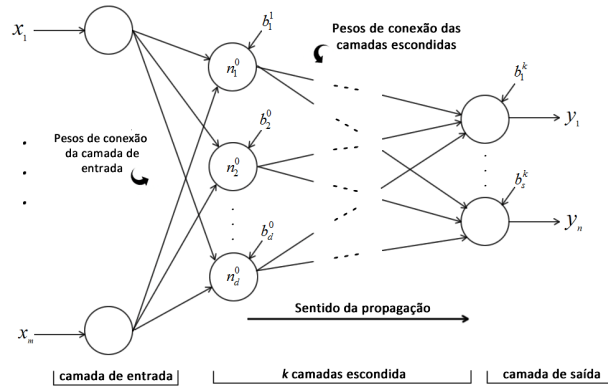


Figura 14 – Exemplo de rede neural artificial do tipo *feedforward* com k camadas escondidas

Algoritmo de treinamento é baseado no método dos mínimos quadrados, conhecido como Regra Delta. Devido a incapacidade do algoritmo de treinamento do Perceptron em gerar pesos que classifiquem dados não linearmente separáveis. O algoritmo utiliza o método do gradiente descendente com a intenção de diminuir o valor da função de erro possibilitando assim a convergência para um mínimo da função de erro.

$$w \leftarrow w + \eta(d^k - u)x^k \quad (2.11)$$

Sendo:

- $w = [\theta \ w_1 \ w_2 \ \dots \ w_n]^T$ é o vetor contendo o limiar e os pesos;
- $x^k = [-1 \ x_1^k \ x_2^k \ \dots \ w_n^k]^T$ é a k-ésima amostra de treinamento;
- d^k é o valor desejado para a k-ésima amostra de treinamento;
- u é o valor de saída do combinador linear; e
- θ é uma constante que define a taxa de aprendizagem da rede.

Backpropagation Dentre os algoritmos conhecidos, para solucionar esse tipo de problema, o mais utilizado é o **backpropagation**. A ideia do algoritmo é estimar os valores dos pesos e bias minimizando uma função de custo, que nada mais é do quanto a predição da rede está errando em relação a saída original do problema. Uma função de custo muito conhecida é o erro quadrático médio (MSE - do inglês, mean square error).

$$MSE = \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^n (d_j^i - y_j^i)^2 \quad (2.12)$$

Sendo p o número exemplos a ser utilizados no treinamento, n o número de saídas da rede e, finalmente, d e y as saídas desejadas e obtidas, respectivamente, para a entrada

em questão. A partir do erro calculado pela função de perda, o backpropagation utilizar uma regra de atualização para cada peso sináptico da rede, que é calculado pela seguinte equação:

$$W = W + \Delta W \rightarrow \Delta W = -\alpha \frac{\partial E}{\partial W} \quad (2.13)$$

Na qual, α é conhecido como taxa de aprendizado, que, resumidamente, indica o ‘tamanho do passo’ do gradiente rumo a minimização. O sinal negativo indica a busca por uma alteração no peso que reduza E .

2.3.0.4 Rede Neural Convolucional

Uma Rede Neural Convolucional (do inglês Convolutional Neural Network, CNN), é uma variação das redes Perceptrons de Múltiplas Camadas, tendo como inspiração o processo biológico da visão humana. Uma CNN é capaz de aplicar filtros em dados visuais, mantendo a relação de vizinhança entre os pixels da imagem ao longo do processamento da rede (VARGAS; PAES; VASCONCELOS, 2016).

Durante o processo de treinamento das CNN, os filtros são ajustados automaticamente para que as características mais relevantes sejam analisadas e mantenham de forma invariável as informações repassadas a cada nova camada.

Existem inúmeras variantes de arquiteturas da CNN na literatura. No entanto, seus componentes básicos são muito parecidos. A sua arquitetura consiste em três tipos básicos de camadas: 1 camada de convolução (Convolutional Layer), 2 camada de subamostragem (Subsampling Layers ou Pooling Layers) e 3 totalmente conectadas (Fully Connected Layers) como mostra a figura 15.

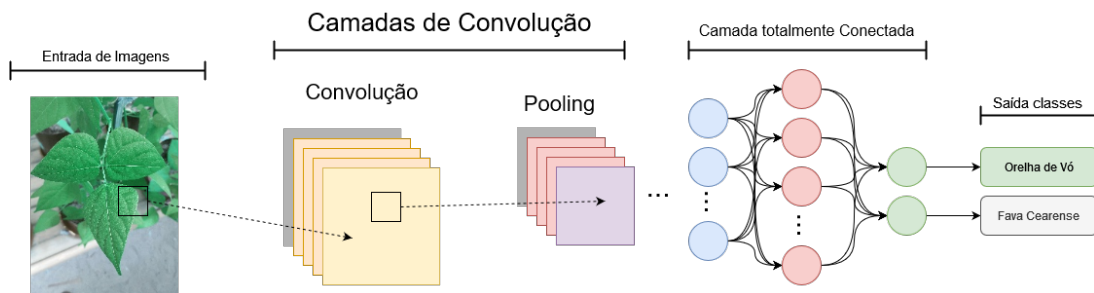


Figura 15 – Esquema de uma Rede Neural Convolucional.

Fonte: Elaborada pelo autor (2022)

2.3.0.4.1 Convolução

Uma convolução em uma imagem digital é uma operação matemática Eq 2.14, que basicamente aplica na matriz de pixels a multiplicação com outra matriz, chamado de filtro, máscara ou kernel, com intuito de extrair as características específicas.

$$g(x) = f(x) \otimes h(x) = \sum_{\forall k} h(x - k) \cdot f(k) \quad (2.14)$$

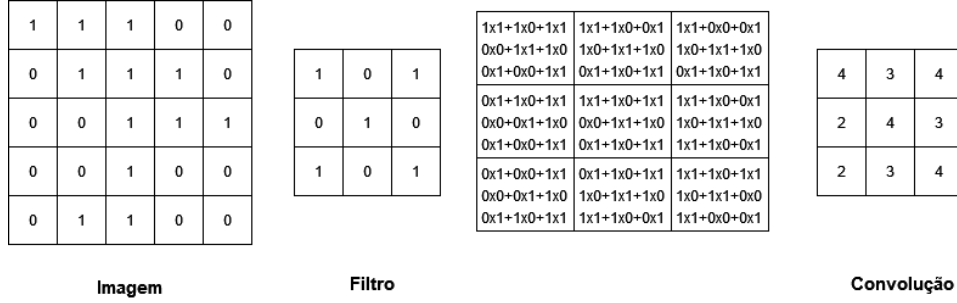


Figura 16 – convolução em uma imagem

2.3.0.4.2 Pooling

chamada de camada de pooling “pooling layer” tem como objetivo reduzir a quantidade de variáveis por meio de amostragem. Essa etapa ajuda a reduzir a quantidade de informação da rede e torna a manipulação mais fácil. Existem 3 operações diferentes de Pooling (*MaxPooling*, *SumPooling*, *AveragePooling*). Todas elas seguem o mesmo princípio e só se diferem na forma como calculam o valor final. O método *MaxPooling* que é o comum, que pega o maior valor dentro de um mapa de variáveis, Já a função *average pooling* utiliza o valor médio para atribuir como entrada à próxima camada, conforme demonstrado na figura 17.

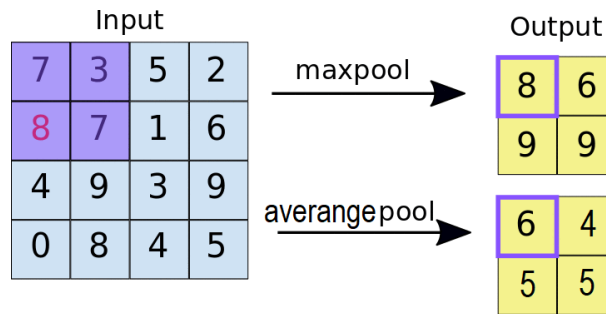


Figura 17 – Aplicação de max pooling/average pooling em uma imagem 4x4 utilizando um filtro 2x2

Dropout é uma técnica para minimizar overfitting proposta em [Srivastava et al. \(2014\)](#) que, na fase de treinamento e durante o passo de propagação dos dados pela rede, aleatoriamente desativa com probabilidade p a ativação de neurônios (em particular neurônios de camadas totalmente conectada). Esse p é um hiperparâmetro cujo o valor ideal pode depender dos dados, arquitetura da rede neural, etc.

2.3.0.4.3 Camada totalmente Conectada

Camada totalmente Conectada ou camada de classificação como próprio nome sugere, junta todas as informações coletadas em um único descritor que servirá de entrada para o processo de classificação da imagem. Uma rede neural convolucional pode ter várias sequências dessas camadas figura 14.

Flatten Uma vez que os mapas foram extraídos, aplica-se a técnica de flattening para transformação da matriz resultante das convoluções em um vetor com 1 coluna e N linhas, que dará entrada na camada densa da rede, responsável por classificar a imagem.

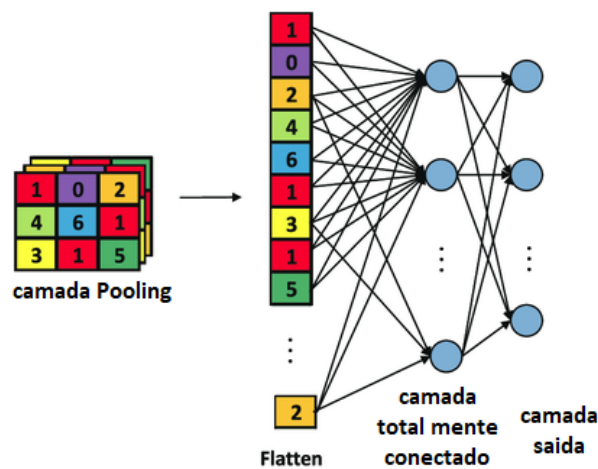


Figura 18 – Operação de Flatten da camada Totalmente conectada

Camada Densa Esta camada é uma rede neural que será treinada para classificar os vetores resultantes do processo de Flattening. Ela possuirá uma quantidade razoável de neurônios que receberão como entrada o vetor. Esta rede poderá possuir diferentes camadas figura 14.

2.3.0.5 VGG 16

É uma rede neural convolucional profunda que foi proposta por [Simonyan et al \(2014\)](#). VGG é um acrônimo para o nome do grupo, Visual Geometry Group, da Universidade de Oxford. Este modelo garantiu o 2º lugar na competição ([ILSVRC-201, 2022](#)). O VGG16 na figura 19 mostra sua arquitetura, onde é composta por 13 camadas convolucionais, 5 camadas de pool máximo e 3 camadas totalmente conectadas. Portanto, o número de camadas com parâmetros ajustáveis é 16 (13 camadas convolucionais e 3 camadas totalmente conectadas). Essa é a razão pela qual o nome do modelo é VGG16.

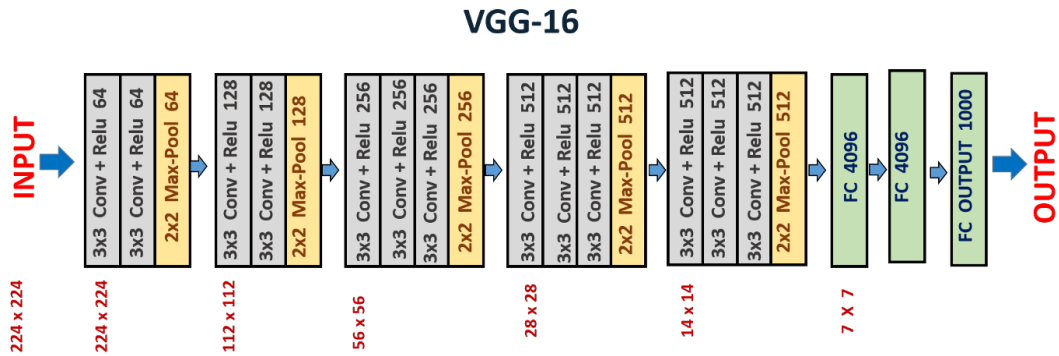


Figura 19 – A arquitetura da Rede

O número de filtros no primeiro bloco é 64, então este número é duplicado nos blocos posteriores até chegar a 512. Este modelo é finalizado por duas camadas ocultas totalmente conectadas e uma camada de saída. As duas camadas totalmente conectadas têm os mesmos números de neurônios que são 4096. A camada de saída consiste em 1000 neurônios correspondentes ao número de categorias do conjunto de dados Imagenet.

2.4 Métodos de Amostragem e Avaliação de Desempenho

2.4.1 Cross-Validation

A ideia por trás do cross-validation é simples. É um procedimento onde os dados são organizados em partições (no inglês conhecido como folders). É Ideal que cada partição deva conter a mesma quantidade de dados. No final das contas, podemos ter uma variação pequena entre as partições, o número de partições é um parâmetro que devemos definir de ante-mão conhecido como k . Na Figura 1 esse valor é definido como $k=4$ e por conta disso, seria denominado 4-fold cross-validation. O ponto central do CV é rotacionar as partições de treino e teste. A cada rodada, o modelo é treinado com $k-1$ partições e testado com uma. Na figura 20, isso é representado pelas cores verde e amarelo. Observe também, que existe um conjunto de teste final (em roxo) que é uma maneira de representar o desempenho do nosso modelo no mundo real. Dessa forma, o treinamento e a validação é realizado k vezes. Segundo Bishop (2006), este método tem uma grande desvantagem pois o número de interações no treino é multiplicado por um fator k , isso trás grandes problemas para modelos onde o treino é custoso do ponto de vista computacional.

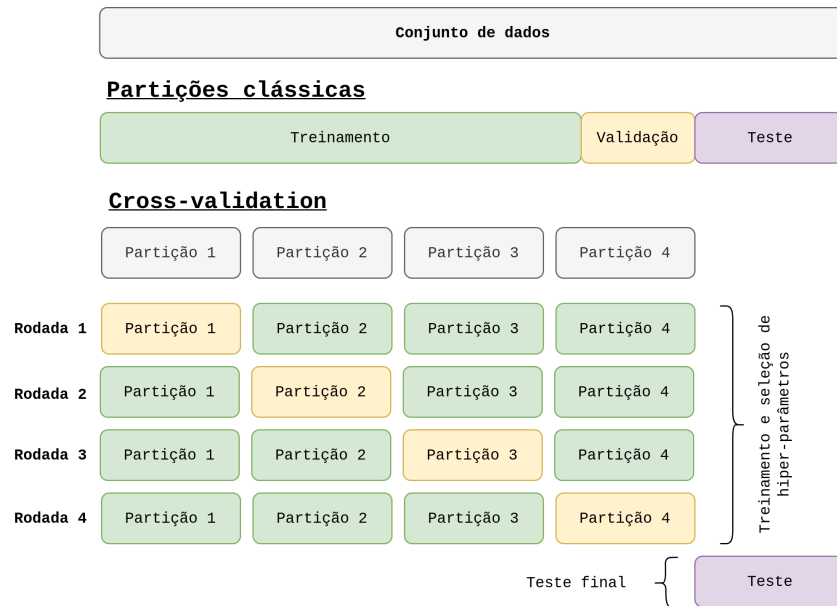


Figura 20 – Ilustração de cada tipo de partição, a versão clássica e o cross validation.

2.4.2 Hiperparâmetros

Os hiperparâmetros são valores que devem ser configurados antes de treinar o modelo para que se obtenham bons resultados na resolução de problemas. Existem algumas métodos para encontrar os parâmetros ideais para os modelos de previsão e elas são descritas adiante.

2.4.3 Pesquisa em grade e aleatória

A busca em grade e a busca manual são as estratégias mais usadas para otimização de hiperparâmetros (Bergstra, James and Bengio, Y., 2012). Dentro das pesquisas em grade à dois métodos que são busca em grade Grid Search como mostra a figura 21a, e a busca aleatória ou Random Search na figura 21b. A busca em grade analisa todas as combinações possíveis de hiperparâmetros e, por isso, é também conhecida como busca exaustiva (Klatzer and Pock, 2015). A pesquisa aleatória é uma métodos que usa o processo de escolha aleatória para avaliá-los. Isso faz com que nem todas as combinações sejam contempladas, ou seja, o espaço de busca é limitado (Bergstra, James and Bengio, Y., 2012).

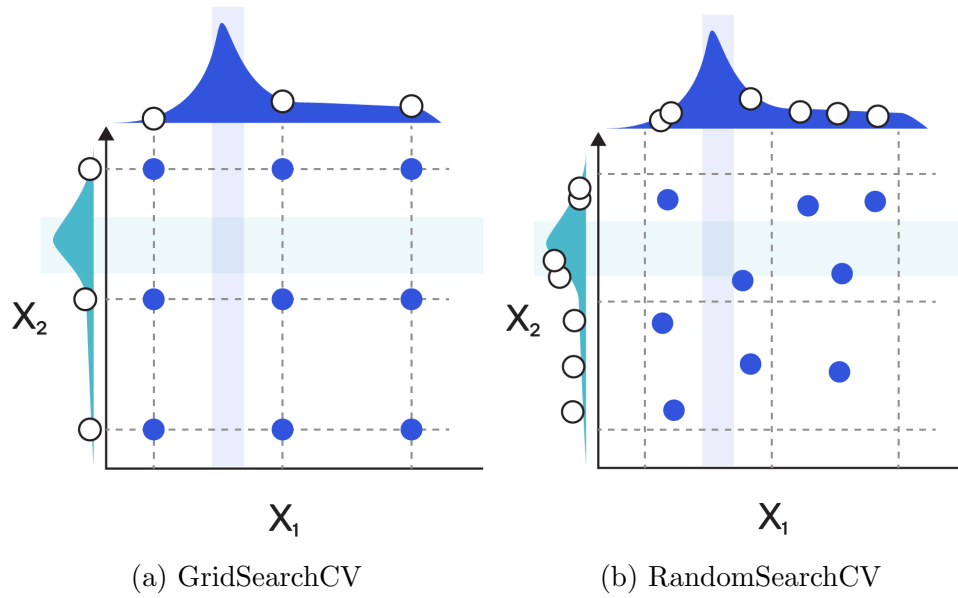


Figura 21 – Metodos de otimização dos hiper parâmetros

2.4.4 Medidas de Avaliação de Desempenho

A matriz de confusão é a matriz quadrada em que se compara os verdadeiros valores de uma classificação com os valores preditos através de algum modelo como mostrado na figura 22. Sua diagonal é composta pelos acertos do modelo e os demais valores são os erros cometidos. O caso binário, o mais comum, é representado pela seguinte matriz:

		Classe Predita	
		Positivo	Negativo
Classe Verdadeira	Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Figura 22 – Matriz de Confusão

A matriz de confusão é uma maneira de exemplificar esses resultados e ela nos traz as informações das frequências dos acertos e erros do modelo.

Onde:

- **Verdadeiro Negativo (VN)**: são as observações que o modelo previu como negativas e realmente eram negativas, ou seja, o modelo classificou corretamente.
- **Falso Positivo (FP)**: são as observações que o modelo previu como positivas, mas na realidade eram negativas. Ou seja, o modelo estimou errado a classe que temos interesse em estimar.

- **Falso Negativo (FN):** são as observações que o modelo identificou como negativas, mas eram positivas. Ou seja, as observações que o modelo estimou errado.
- **Verdadeiro Positivo (VP):** são as observações que são positiva e o modelo consegue classificar corretamente. Ou seja, as observações, da classe de interesse, que o modelo classificou corretamente.

Precision ou precisão, também conhecida como Valor Preditivo Positivo (VPP), é a métrica que traz a informação da quantidade de observações classificadas como positiva que realmente são positivas. Ou seja, entre todas as observações identificadas como positivas, quantas foram identificadas corretamente.

$$Precisão = \frac{VP}{VP + FP} \quad (2.15)$$

Recall cobertura ou Sensibilidade é a proporção dos Verdadeiros Positivos entre todas as observações que realmente são positivas no seu conjunto de dados. Ou seja, entre todas as observações que são positivas quantas o modelo conseguiu identificar como positiva. Representa a capacidade de um modelo em prever a classe positiva.

$$Sensibilidade = \frac{VP}{VP + FN} \quad (2.16)$$

F1-Score é a média harmônica entre o Sensibilidade e a precisão (Precisão). Utilizada quando temos classes desbalanceada.

$$F1-Score = 2 * \frac{sensibilidade * precisão}{sensibilidade + precisão} \quad (2.17)$$

A curva ROC (Receiver Operating Characteristic Curve) mostrada na figura 23 é a curva gerada pela taxa de verdadeiros positivos (sensibilidade) e pela taxa de falsos positivos (1 – especificidade) para diferentes pontos de cortes. Oferece uma visão geral de um classificador e pode ser utilizada para encontrar pontos de corte ideais. O corte que deixa a curva mais próxima do vértice (0, 1) maximiza a sensibilidade conjuntamente com a especificidade.

Uma medida decorrente da curva ROC é o AUC (Area Under the Curve), que nada mais é que a área abaixo da curva. O AUC varia entre 0 e 1 e quanto maior o AUC melhor o modelo.

Utilizada quando temos classes desbalanceada e sua principal vantagem é poder escolher o melhor ponto de corte para otimizar o desempenho do modelo.

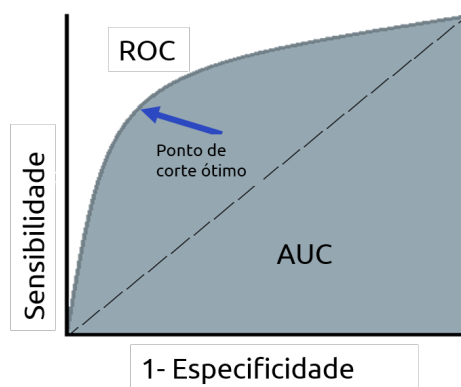


Figura 23 – Curva ROC

3 Trabalhos Relacionados

Em [Khan et al. \(2022\)](#), é proposta a detecção e classificação de doenças em imagens de folhas de pepino. Foram utilizadas as arquiteturas VGG-19 e o VGG-M pré-treinadas para extração de Deep Features, além de um método para seleção de características. Esses recursos refinados são finalmente enviados para a máquina de vetores de suporte multiclasse para classificação de doenças. Para provar a eficiência do método proposto, cinco doenças de folhas de pepino são consideradas e classificadas para alcançar uma precisão de classificação de 98.08% em 10,52 segundos. Adicionalmente, o método proposto também é comparado com as técnicas mais recentes para comprovar sua autenticidade.

Em [Hossain J. Amin \(2010\)](#) os autores utilizaram redes neurais probabilísticas (PNN) para classificar as plantas com folhas largas e planas. Neste algoritmo haviam poucos *selects*, ou seja, pontos em que o usuário precisa para identificar as folhas e um ponto de base com o qual a imagem é alinhada e comparada com as outras imagens com base em algumas características como área, excentricidade, etc. Nesse estudo, 1200 folhas de amostra pertencentes a 30 plantas diferentes foram utilizadas para treinar seu sistema. Este sistema também é semi-automático e com 91.41% de precisão média de reconhecimento.

Além de reportar a superioridade dos métodos tradicionais, os autores demonstraram aspectos positivos na utilização de redes neurais convolucionais como extratores de características Deep Features. Especificamente para a tarefa de identificação de espécies de plantas, em [Khan et al. \(2022\)](#), os autores demonstraram ótimos resultados com extração de features usando as redes VGG-19 e VGG-M e o classificador SVM. Contudo, não foram considerados procedimentos de otimização dos classificadores.

4 Metodologia

O método consiste em utilizar deep learning, mais especificamente uma rede neurais convolucional (CNN), para ajudar o agricultor na difícil etapa de classificar nos estágios iniciais as diferentes espécies de feijão-fava, cada espécie tem diferentes comprimento das sementes tempo de colheita, manejo. Assim foi abordadas três comparações utilizando CNN, descritas individualmente na sequência:

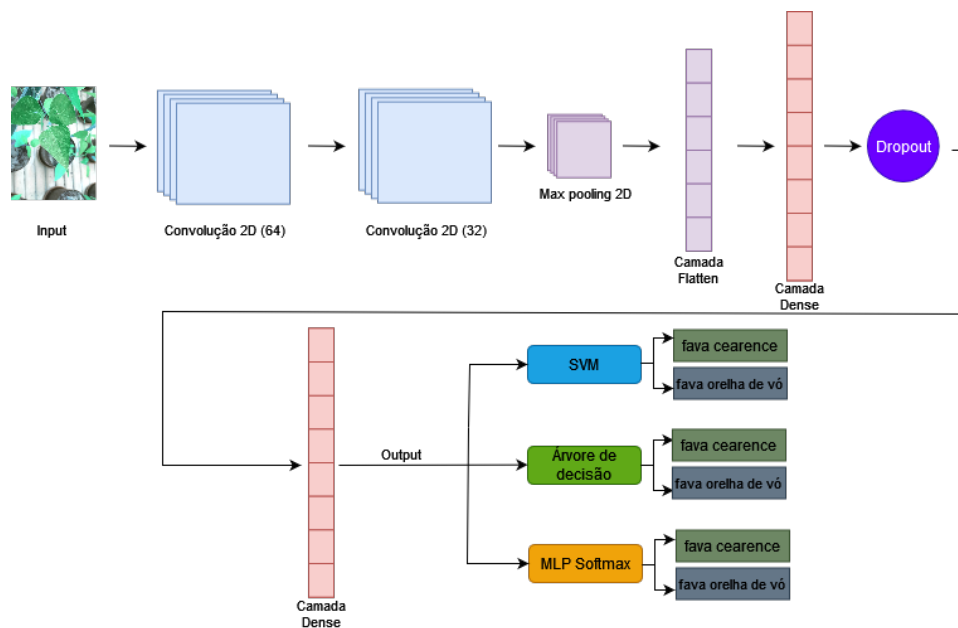


Figura 24 – Arquitetura proposta para a detecção automatizada de mudas *feijão-fava*

Fonte: Elaborada pelo autor (2022)

1. Nesta abordagem construímos um modelo de rede neural convolucional tradicional CNN sua arquitetura é mostrada na figura 27, foi treinada para classificar as instancias de feijão-fava, encontrando ótimos resultados com mínimo de erro, obtendo os melhores valores para os filtros no problema de classificação de feijão-fava.
2. Na segunda abordagem o modelo de CNN gerado na primeira abordagem foi utilizado como extrator de características, onde foi removida a camada de classificação softmax. Assim, os valores de ativação dos neurônios da rede que estariam ligados nesta camada são utilizados como entrada à Máquina de Vetores de Suporte (SVM) para a classificação como mostra na figura 24.
3. Na terceira e última abordagem é utilizado o modelo CNN gerado na primeira abordagem como extrator de características, foi removida a última camada a de classificação softmax. Assim, os valores de ativação dos neurônios da rede que

estariam ligados nesta camada são utilizados como entrada em uma árvore de decisão para a classificação como mostra na figura 24.

4.1 Aquisição das imagens e pré-processamento

As imagens foram disponibilizadas por Raphael Miller, na época era aluno do curso de agronomia da Universidade Federal Rural de Pernambuco, que apresentou o seguinte questionamento se era possível classificar as mudas de feijão-fava já que elas eram quase idênticas utilizando métodos de aprendizado computacional. O conjunto possui duas categorias de feijão-fava as Fava Cearense e Fava Orelha de Vó, totalizando 229 imagens, sendo 109 de Fava Cearense figura 25a, e 120 de Fava Orelha de Vó figura 25b.



(a) Cearense

(b) Orelha de Vó

Figura 25 – Favas

Devido a quantidade limitada de imagens utilizados nesse trabalho, as operações de acréscimo de dados (data augmentation), para aumentar a quantidade e a diversidade dos dados do conjunto de imagens, para evitar o overfitting da rede e garantir a sua generalização. Os métodos descritos foram implementados em Python, utilizando as bibliotecas Opencv (Open Source Computer Vision Library) ¹, uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto disponível em Python. As manipulação nas imagens foram: espelhamento horizontal e vertical, variação do nível de brilho, foi aplicado cinco transformações mais a copia da imagem original dando um total de seis gerada a partir de uma única imagem. Após esta etapa, o novo conjunto de

¹ <https://opencv.org/about/>

imagens que foi gerada ficou com 654 imagem de Fava Cearense e 720 imagens de Orelha de Vó, totalizando 1374 imagens ². A tabela 2 mostra total das imagens antes e depois do o acréscimo de dados (data augmentation).



(a) Imagem com variação no brilho (b) Variação no brilho e flip horizontal



(c) Imagem com variação no brilho (d) Variação no brilho e flip vertical

Técnica que ajuda na melhoria do modelo que é o acréscimo de dados (data augmentation) citada anteriormente. Todas as imagens possuem 3456 pixels de largura x 4608 pixels de altura, o que totaliza 47.775.744 atributos, que são provenientes dos três canais. Devido às limitações de hardware disponíveis, para que o treinamento dos algoritmos, foi necessário reduzir o tamanho da imagens em 200 pixels de largura x 267 pixels de altura, este é o menor valor obtido que não afeta a qualidade das imagens, o que resultou em 160.200 atributos para cada imagem, a figura 26c e figura 26d são imagens geradas com o método de acréscimo de dados onde seu tamanho foi reduzidos.

² https://github.com/ericoandre/reconhecimento_padroes

Conjunto	Fava Cearense	Orelha de Vó	Total
Imagens Originais	109	120	229
Imagens Aumentadas	654	720	1374

Tabela 2 – Distribuição dos conjuntos de imagens

4.2 Seleção de Modelos

Para o ambiente experimental utilizada para o desenvolvimento a linguagem Python. Para realizar a implementação dos experimentos e executar os algoritmos de Aprendizado de Máquina foram utilizadas as bibliotecas Scikit-Learn ³, Keras ⁴ e TensorFlow ⁵. No início, os experimentos foram executados no Colab em sua versão gratuita, em que é disponibilizado 12GB de memória RAM e GPU no modelo K80. Mais, suas limitações impediam a execução, sendo necessário utilizar o Google Colab ⁶, na sua versão Pro, o qual é fornecido 25GB de memória RAM, com GPU T4 ou P100.

Para otimizar a seleção dos modelos utilizou-se a busca aleatória *RandomSearchCV* para a escolha do melhor conjunto de hiperparâmetro, esse método possui um grande benefício significativo. De acordo com Bergstra, James and Bengio, Y. (2012), os parâmetros escolhidos utilizando a técnica de busca em grade aleatória é mais eficaz que as demais técnicas, pois não tenta todas as combinações hiperparâmetros. Ela é capaz de encontrar modelos tão bons ou melhores em uma pequena fração do tempo de computação. Esta é uma vantagem muito significativa devido às limitações dos recursos de hardware disponíveis para o experimento. Utiliza uma combinação de hiperparâmetros aleatória para treinar cada modelo, com o mesmo número de **k-folds** em $k=5$ para todas as abordagens, e utilizou $n=50$ esse n é o numero de combinações utilizadas na busca aleatória.

A tabela 3 mostra as distribuições dos hiperparâmetros que foram utilizados na busca aleatório.

³ <https://scikit-learn.org/>

⁴ <https://keras.io/>

⁵ <https://tensorflow.org/>

⁶ <https://colab.research.google.com/>

Tabela 3 – Hiperparâmetros

Algoritmos	Hiperparâmetros	Distribuição de Probabilidade
CNN MLP	dropout	uniforme em {0.5, 0.6, 0.7, 0.8, 0.9, 0.10, 0.11, 0.12}
CNN MLP	neurônios	uniforme em {32, 64, 128, 256}
CNN MLP	ativação	uniforme em {relu, sigmoid}
CNN MLP	épocas	uniforme em [50; 500]
CNN MLP	tamanho do batch	uniforme em {16, 32, 64, 128}
CNN MLP	otimizador	uniforme em {adam, sgd, rmsprop, nadam}
CNN SVM	C	uniforme em [0.0001; 100.00]
CNN SVM	kernel	uniforme em {poly, rbf, linear}
CNN SVM	Grau do polinômio	uniforme em [2; 5]
CNN SVM	γ	uniforme em [0.0001; 100.00]
CNN TREE	critério	uniforme em {gini, entropy}
CNN TREE	divisor	uniforme em {best, random}
CNN TREE	amos mín dividida	uniforme em [0; 1]
CNN TREE	amos mín por folha	uniforme em [0.00000; 0.5]
CNN TREE	máx de nós por folha	uniforme em [1; 1000]
CNN TREE	profundidade máx	uniforme em [1; 1000]
CNN VGG16	dropout	uniforme em {0.5, 0.6, 0.7, 0.8, 0.9, 0.10, 0.11, 0.12}
CNN VGG16	neurônios	uniforme em {32, 64, 128, 256}
CNN VGG16	ativação	uniforme em {relu, sigmoid}
CNN VGG16	épocas	uniforme em [50; 500]
CNN VGG16	tamanho do batch	uniforme em {16, 32, 64, 128}
CNN VGG16	otimizador	uniforme em {adam, sgd, rmsprop, nadam}

O modelo de rede neural convolucional utilizado como base na comparação com outros métodos tem a seguinte arquitetura mostrada na figura 27, e este mesmo modelo foi utilizado como extrator de características para os classificadores SVM e Árvore de decisão, após ter sido removida a camada de classificação softmax.

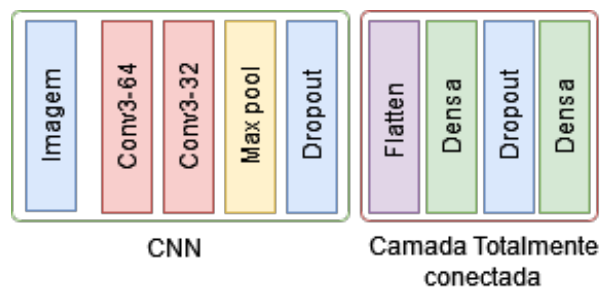


Figura 27 – Arquitetura da CNN

Fonte: Elaborada pelo autor (2022)

A extração das características usando a CNN pré treinada consiste inicialmente em carregar uma arquitetura de um modelo rede neural convolucional já treinada com o conjunto de imagens de feijão-lima, aproveitar o treinamento básico de suas características. Depois é removido seu classificador, para a saída da rede ser um vetor de características. Como mostra a figura 28, todo o fluxo dos dados que inicia na camada de convolução passa na camada de pooling onde é extraído os mapas de características que por sua vez passa por uma camada de flatter onde foi transformado em vetor de características é utilizado como entrada nos modelos de Máquina de Vetores de Suporte (SVM) e Árvore de Decisão.

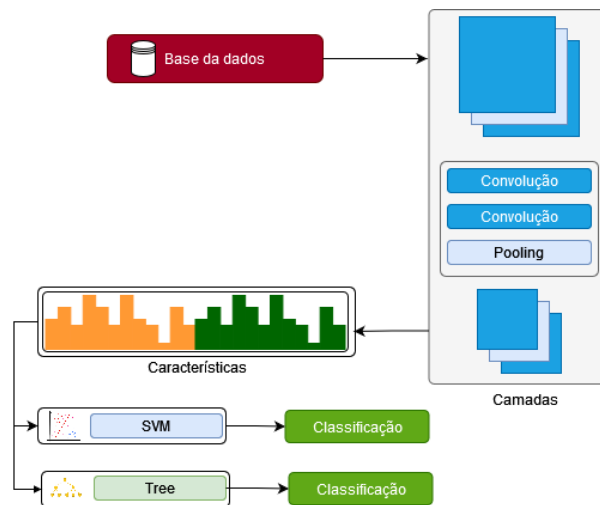


Figura 28 – Representação do método proposto nos experimentos

Fonte: Elaborada pelo autor (2022)

5 Resultados

Através da matriz de confusão onde a fava cearense representa classes verdadeira, e a fava orelha de vó representada pela classe negativas, foi estriadas todas as métricas apresentada, duas delas é de grande importância como cobertura e precisão, isto é se o valor para a precisão fosse de 90%, isto indicaria que a cada 100 mudas de fava classificadas como positivo, é esperado que apenas 90 seja de fava cearense. já a cobertura é razão entre a quantidade de exemplos classificados corretamente como positivos e a quantidade de exemplos que são de fato positivos, se o valor para a cobertura fosse de 95%, isto indicaria que a cada 100 mudas de fava que são de fato positivos, é esperado que apenas 95 sejam corretamente identificados como fava uma das classes.

5.0.1 CNN

Antes de utilizar o modelo como extrator de características, foi feito o treinamento de uma rede convolucional inteira. Ela é composta por duas camadas convolucionais, com filtros de tamanho 3×3 e 2×2 , seguidas de camadas max pooling e dropout. Por fim, existem uma camadas totalmente conectadas e a camada de classificação mostrado na figura 27. Foi separada uma sub amostra de 30% do conjunto de 1374 imagens totalizando 413 imagens, que foram utilizadas na validação dos modelos. Após isso, avaliação da rede convolucional pode ser vista na figura 29, através da matriz de confusão notamos que das 413 acertou 182 da classe fava Cearense e 212 de fava Orelha de vó e teve instâncias teve 19 erros obtendo uma precisão de 0.94, com uma cobertura de 0.96.

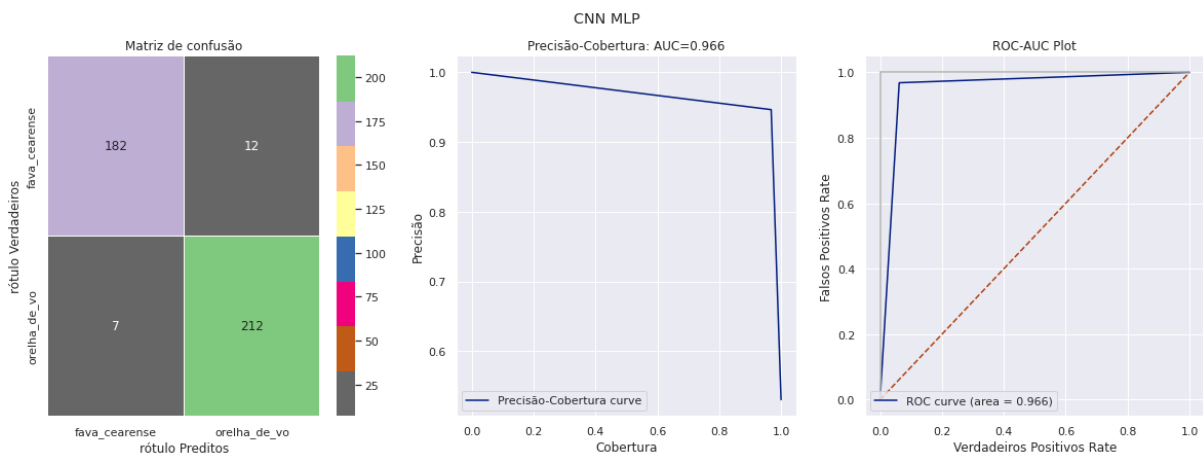


Figura 29 – CNN MLP Resultados

Na tabela 4, mostra os hiperparâmetros relacionado ao melhor modelo de CNN, obtidos através da seleção de modelos.

Tabela 4 – Hiper parâmetros dos Melhor Modelo CNN

Algoritmos	Hiperparâmetros	Variação
CNN MLP	épocasochs	364
CNN MLP	tamanho do batch	16
CNN MLP	otimizador	adam
CNN MLP	ativação	relu
CNN MLP	neurônios	128
CNN MLP	dropout	0.1

5.0.2 SVM

Utilizando CNN com extrator de característica, removendo sua última camada a de classificação softmax. Assim, os valores de ativação dos neurônios da rede que estariam ligados nesta camada são utilizados como entrada à Máquina de Vetores de Suporte (SVM) para a classificação como mostra na figura 28. Após isso, pode ser vista na figura 30, através da matriz de confusão, que das 413 instâncias acertou 197 da classe fava cearense e 201 de fava orelha de vô e teve 6 erros obtendo uma precisão de 0.98, com uma cobertura de 0.99, esse é o melhor das três abordagens.

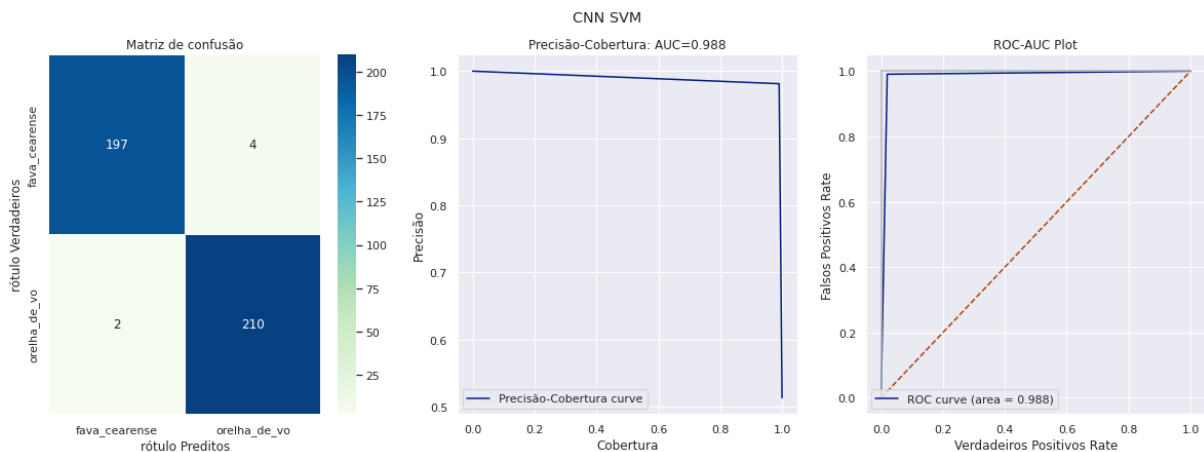


Figura 30 – CNN SVM Resultados

Na tabela 4, mostra os hiperparâmetros relacionado ao modelo que obteve os melhores resultados, obtidos através da seleção de modelos.

Tabela 5 – Hiperparâmetros do Melhor Modelo SVM

Algoritmos	Hiperparâmetros	Variação
CNN SVM	C	2.91×10^1
CNN SVM	kernel	poly
CNN SVM	Grau do polinômio	2
CNN SVM	γ	4.78×10^{-4}

Um dos fatores que levaram o SVM a se sobre sair são suas fronteiras otimizadas através do hipermetro C que responsável por aplicando um custo durante o treinamento para cada amostra classificada erroneamente.

5.0.3 Árvore de decisão

Da mesma forma, Utilizando CNN com extrator de característica, removendo sua última camada a de classificação softmax. Assim, os valores de ativação dos neurônios da rede que estariam ligados nesta camada são utilizados na entrada de uma Árvore de Decisão para a classificação como mostra na figura 28. Após isso, pode ser vista na figura 31, através da matriz de confusão, que das 413 instâncias acertou 186 da classe fava Cearense e 216 de fava Orelha de Vó e teve 11 erros obtendo uma precisão de 0.96, com uma cobertura de 0.98, essa abordagem foi o segundo melhor depois do SVM.

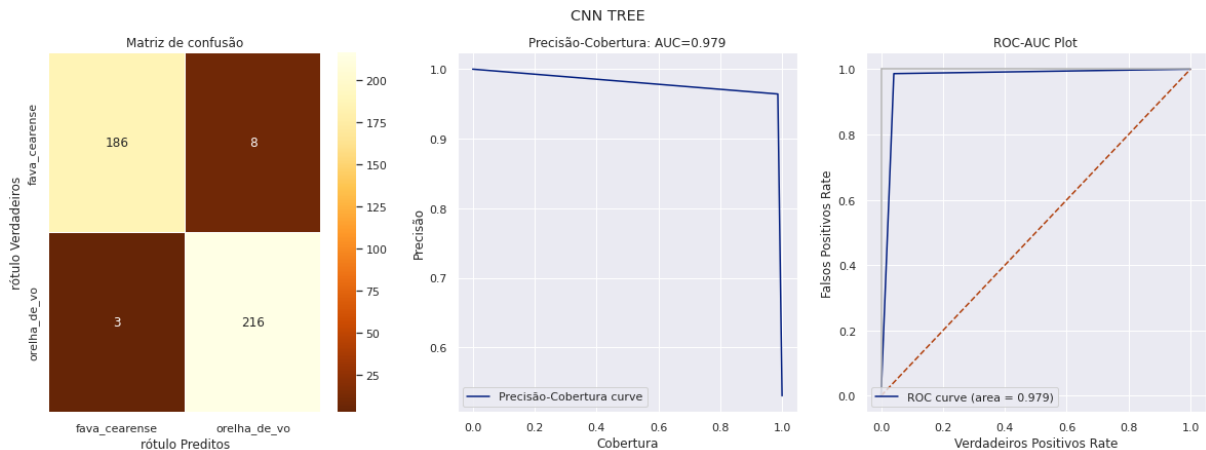


Figura 31 – CNN TREE Resultados

Na tabela 6, mostra os hiperparâmetros relacionado ao modelo que obteve os melhores resultados, obtidos através da seleção de modelos.

Tabela 6 – Hiperparâmetros do Melhor Modelo Arvore

Algoritmos	Hiperparâmetros	Variação
CNN TREE	criterion	gini
CNN TREE	max depth	436
CNN TREE	max leaf nodes	861
CNN TREE	min samples leaf	9.17×10^{-2}
CNN TREE	min samples split	7.79×10^{-1}
CNN TREE	splitter	best

5.0.4 VGGNet

Este experimento foi realizado de forma adicional, utilizando Transferência de Aprendizado (transfer learning) como arquitetura vgg16 treinados com os pesos da imagenet (ILSVRC-201, 2022), com extrator de característica, removendo a camada totalmente conectada e substituindo por uma arquitetura de MLP descrita na figura 32, a mesma configuração utilizada nas camadas totalmente conectadas da CNN já descritas.

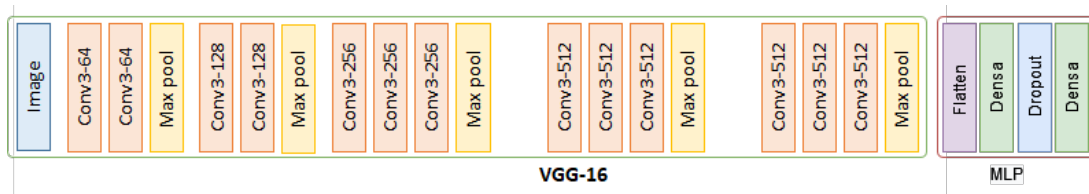


Figura 32 – Arquitetura VGG16 com MLP

Fonte: Elaborada pelo autor (2022)

Os Resultados obtido com está abordagem pode ser vista na figura 33, através da matriz de confusão, que das 413 instâncias acertou 186 da classe fava cearense e 212 de fava Orelha de Vó e teve 15 erros obtendo uma precisão de 0.95, com uma cobertura de 0.96, essa abordagem só foi melhor que a CNN.

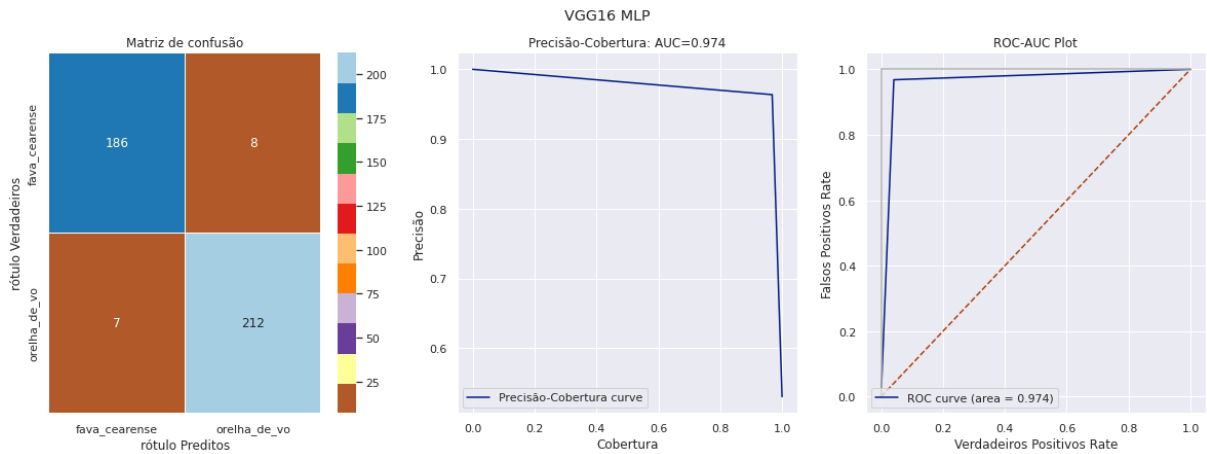


Figura 33 – CNN VGG16 Resultados

Na tabela 7, mostra os hiperparâmetros relacionado ao modelo que obteve os melhores resultados, obtidos através da seleção de modelos.

Tabela 7 – Hiperparâmetros do Melhor Modelo Arvore

Algoritmos	Hiperparâmetros	Variação
VGG16 MLP	épocas	166
VGG16 MLP	tamanho do batch	32
VGG16 MLP	otimizador	nadam
VGG16 MLP	ativação	relu
VGG16 MLP	neurônios	64
VGG16 MLP	dropout	0.1

6 Conclusões

Foi possível perceber que todos os métodos testados erraram poucas instância, indicando uma boa precisão nas classificações no experimento como mostra na tabela 8. De modo geral, os resultados com SVM foram superiores aos com o classificador baseado em Softmax e Árvore de decisão. Onde a máquina de vetores de suporte (SVM) obteve uma precisão de 0.98 com 6 erros apenas 6 instâncias foi onde se obteve os melhores resultados seguido pela Árvore de decisão que apresentou uma precisão de 0.96.

Tabela 8 – Contem as métricas precisão e Sensibilidade

Algoritimos	Precisão	Cobertura	AUC-ROC
CNN	0.946429	0.968037	0.966
CNN SVM	0.981308	0.990566	0.988
CNN TREE	0.964286	0.986301	0.979
VGG16 MLP	0.958762	0.963730	0.974

Os sistemas de visão computacional têm sido cada vez mais utilizados na agricultura facilitando a tomada de decisão dos agricultores. O presente trabalho apresentou um método para identificação de espécies e classificação de mudas de feijão-fava. Neste trabalho, quatro modelos foram comparados com a finalidade de determinar qual seria o melhor classificador. Como visto no capítulo anterior, a junção de SVM com as camadas de convolução na etapa de extração de características se mostrou eficaz, com apenas 6 erros obtendo uma precisão de 0.98 e cobertura de 0.99.

Como trabalhos futuros, pretendemos explorar mais variedades como boca-de-moça, branquinha, mororó, olho-de-ovelha, olho-de-peixe, raio-de-sol, rajada vermelha, rajada preta. Obter recursos de hardware mais robustos, para estender a seleção de modelos.

Referências

- APRENDIZADO DE MÁQUINA . *Aprendizado De Máquina*. 2021. <https://aprendizadodemaquina.com/artigos/o-que-e-deep-learning/>. Accessed: 2022-04-01. Citado na página 20.
- Bergstra, James and Bengio, Y. Random Search for Hyper-Parameter Optimization. *The Journal of Machine Learning Research*, v. 13, p. 281–305, 03 2012. Citado 2 vezes nas páginas 32 e 40.
- Bishop. *Pattern Recognition and Machine Learning*. [S.l.]: Springer. [S.l.: s.n.], 2006. ISBN 9780387310732. Citado na página 31.
- Data Science Academy. *Deep Learning Book*. 2021. <https://www.deeplearningbook.com.br>. Accessed: 2021-11-22. Citado na página 25.
- FIELDVIEW. *FieldView*. 2021. <https://www.cnabrazil.org.br/cna/panorama-do-agro>. Accessed: 2022-04-01. Citado na página 13.
- GONZALEZ ; WOODS. *Processamento digital de imagens. 3. ed.* [S.l.]: Pearson Prentice Hall, 2010. ISBN 9788576054016. Citado 3 vezes nas páginas 15, 16 e 18.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 20.
- HALLIDAY. *Fundamentos De Fisica - Volume 4 - Optica E Fisica*. [S.l.]: LTC, 2016. ISBN 9788521630388. Citado na página 17.
- Haykin, S. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: LTC Editora Rio de Janeiro, Brazil, 2007. Citado na página 25.
- HOSSAIN J. AMIN, M. A. Leaf shape identification based plant biometrics. *International conference on computer and information technology (ICCIT) IEEE*, v. 1, p. 458–463, 2010. Citado na página 36.
- ILSVRC-201. *ImageNet*. 2022. <https://www.image-net.org/challenges/LSVRC/2014/>. Accessed: 2022-04-01. Citado 2 vezes nas páginas 30 e 46.
- Ipma. *Área educativa - SATÉLITES METEOROLÓGICOS*. 2002. <https://www.ipma.pt/pt/educativa/observar.tempo/index.jsp?page=satelite03.xml&print=true>. Accessed: 2022-05-01. Citado na página 16.
- KHAN, M. A. et al. Cucumber leaf diseases recognition using multi level deep entropy-elm feature selection. *Applied Sciences*, v. 12, n. 2, 2022. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/12/2/593>. Citado na página 36.
- Klatzer and Pock. Continuous Hyper-parameter Learning for Support Vector Machines. *omputer Vision Winter Workshop - Paul Wohlhart, Vincent epetit (eds.)*, v. 20th, 9-11 2015. Citado na página 32.

Mitchell et al. Machine learning. Annual review of computer science, Annual Reviews 4139 El Camino Way. *PO Box 10139, Palo Alto*, v. 4, p. 417—433, 1990. Citado na página 19.

REVISTACULTIVAR. *Feijão-Fava (Phaseolus lunatus L.)*. 2015. <https://revistacultivar.com.br/artigos/feijao-fava-phaseolus-lunatus-l->. Accessed: 2022-04-01. Citado na página 13.

Simonyan et al. Very deep convolutional networks for large-scale image recognition. 2014. Citado na página 30.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, n. 56, p. 1929–1958, 2014. Disponível em: <http://jmlr.org/papers/v15/srivastava14a.html>. Citado na página 29.

VIEIRA, R. A cultura do feijão-fava. *Informe Agropecuário*, v. 16, n. 174, p. 30–37, 1992. Citado na página 13.

ZIMMERMANN, M. d. O.; TEIXEIRA, M. Origem e evolução. *ARAÚJO, R*, 1996. Citado na página 13.