



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**



Erico Andre da Silva

RECONHECIMENTO AUTOMÁTICO FOLHAS FAVA

Recife

2021

Resumo

Existem muitas soluções computacionais aplicados a área do agronegócio, que visam melhorar a produtividade, reduzir desperdícios e auxiliar na tomada de decisões. Em relação aos avanços tecnológicos uma das áreas que vem se destacando no setor é a de visão computacional, que ao longo do tempo tem se tornando mais acessível como tecnologia. Visão computacional é um conjunto de métodos e técnicas utilizadas para interpretar imagens, auxiliando na tomada de decisões, a partir de reconhecimento de padrões. Visando contribuir com esse cenário de desenvolvimento tecnológico no setor do agronegócio, este trabalho apresenta abordagem de identificação automática de plantas (favas sendo orelha de vó ou fava Cearense), auxiliando o especialista agrônomo.

Palavras-chave: Aprendizagem de máquina, Classificação de imagens, reconhecimento de imagens.

Sumário

1	INTRODUÇÃO	4
1.0.1	Justificativa	4
1.0.2	Objetivo	4
2	FUNDAMENTAÇÃO TEÓRICA	5
2.1	Processamento de Imagens	5
2.1.1	Imagem Digital	5
2.1.2	Cores	7
2.1.2.1	Padrão de Cores RGB	7
2.1.2.2	Padrão de Cores HSV	8
2.1.3	Segmentação	8
2.1.4	Extração de Características	9
2.1.4.1	Histograma	9
2.2	Data Augmentation	10
2.3	Reconhecimento de Padrões	11
2.3.1	Aprendizagem supervisionada	11
2.3.1.1	Naive Bayesian	11
2.3.1.2	K Vizinhos mais Próximos (KNN)	12
2.3.1.3	Árvores de Decisão	14
2.3.1.3.1	Entropia	14
2.3.1.3.2	Ganho de informação	15
2.3.1.3.3	Podagem	16
2.3.1.4	Support Vector Machine (SVM)	16
2.3.1.4.1	Hiperplano Ótimo	17
2.3.1.4.2	SVM Kernel	18
2.3.1.5	Redes Neurais Artificiais	19
2.3.1.5.1	O Neurônio Artificial	19
2.3.1.5.2	Arquitetura de uma RNA	20
2.3.1.5.3	Algoritmo de treinamento	21
2.3.1.6	Rede Neural Convolucional	22
2.3.1.6.1	Convolução	22
2.3.1.6.2	Pooling	23
2.3.1.6.3	Flatten	23
2.4	Cross-validation	24
2.4.1	Hiperparâmetros	24

2.5	Medidas de Desempenho	25
2.5.1	Matriz de Confusão	25
2.5.2	Acurácia	26
2.5.3	Precision	26
2.5.4	Recall	26
2.5.5	F1- Score	26
2.5.6	Área sob a Curva ROC	27
3	TRABALHOS RELACIONADOS	28
4	METODOLOGIA	29
4.1	Aquisição das imagens	29
4.2	Pré-processamento	29
4.2.1	Transformação do espaço de cores	30
4.2.2	Transformação geométrica	31
4.2.3	Extração de Características	32
4.3	Seleção de Modelos	32
5	RESULTADOS	35
5.0.1	Redes Neurais Artificiais CNN	35
5.0.2	Redes Neurais Artificiais MLP	36
5.0.3	Support Vector Machine	37
5.0.4	K Vizinhos mais Próximos	37
5.0.5	Árvores de Decisão	38
5.0.6	Naive Bayesian	38
6	CONCLUSÕES	40
	REFERÊNCIAS	41

1 Introdução

O uso mais intensivo das tecnologias da informação e comunicação no setor agrícola vem desde a década de 80, quando da introdução de microcomputadores no mercado a um preço mais acessível, utilizando planilhas eletrônicas e programas adaptados de administração e contabilidade (SOFTEX, 2016).

A agricultura faz parte do setor primário da economia sendo fornecedor de alimento e de matéria-prima. Independente do grau de desenvolvimento, do tamanho territorial de um país ou região. Com a modernização e profissionalização, o setor deu um salto de importância na economia nacional. Atualmente é responsável por quase um quarto do PIB (Produto Interno Bruto) e praticamente metade das exportações do país. De acordo com cálculos do Cepea (Centro de Estudos Avançados em Economia Aplicada), em 2020 o agronegócio brasileiro alcançou participação de 26,6% no Produto Interno Bruto do Brasil, contra 20,5% em 2019. Já a participação do agro na soma dos bens e serviços do país em 1970 era de 7,5%.

O processo de classificação de determinadas plantas, é realizado por profissionais de forma manual, onde demanda uma grande quantidade de tempo, para realizar. Neste contexto, a análise de imagens tem função valiosa ao transformar os dados coletados no campo em informações úteis ao especialista agrônomo.

1.0.1 Justificativa

A falta de agrônomos especialistas para efetuar classificação de determinadas plantas, e o fato de que demanda um bom tempo e experiência do profissional. A experiência é um fator que influencia na precisão da classificação, além de locais muito distante e de difícil acesso as vezes.

1.0.2 Objetivo

Este trabalho visa o estudo de uma abordagem de identificação automática de plantas (favas sendo orelha de vó ou fava Cearense), através de fotos tiradas das folhas, auxiliando no trabalho dos agrônomos.

2 Fundamentação Teórica

2.1 Processamento de Imagens

De acordo com [GONZALEZ, ; WOODS, Richard E \(2010\)](#), "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, ; WOODS, Richard E \(2010\)](#) 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, Richard E, 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 "pixels". (Figura 1).

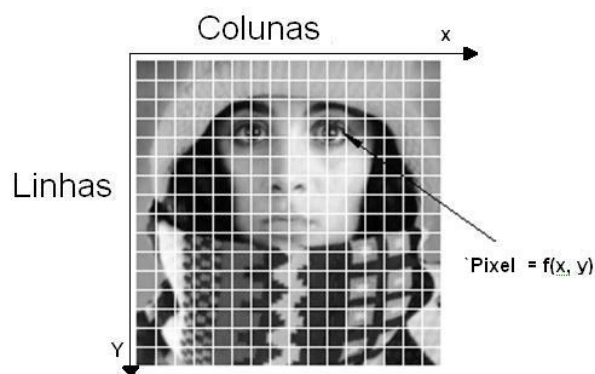


Figura 1 – Imagem Digital

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

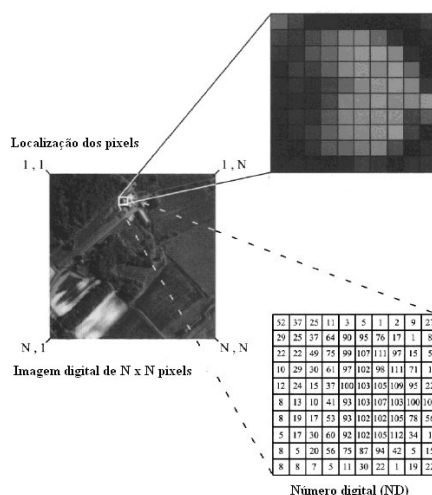


Figura 2 – Representação de imagem preta e branca

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).

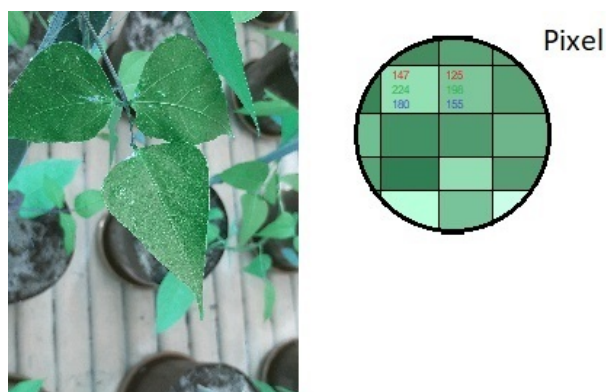


Figura 3 – Representação de imagem no Modelo RGB

2.1.2 Cores

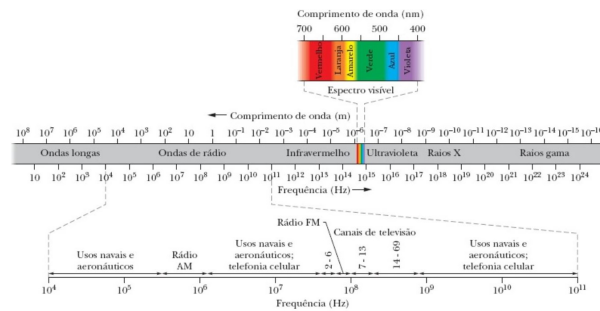


Figura 4 – O espectro eletromagnético.

O que vemos como cores em uma luz, é na verdade, uma diferença de comprimento de onda eletromagnética, 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).

2.1.2.1 Padrão de Cores RGB

O 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

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).

2.1.2.2 Padrão de Cores HSV

O modelo HSV/HSB é definido pelas grandezas de tonalidade (Hue), Saturação (Saturation) e valor/luminosidade (Value/Brightness).

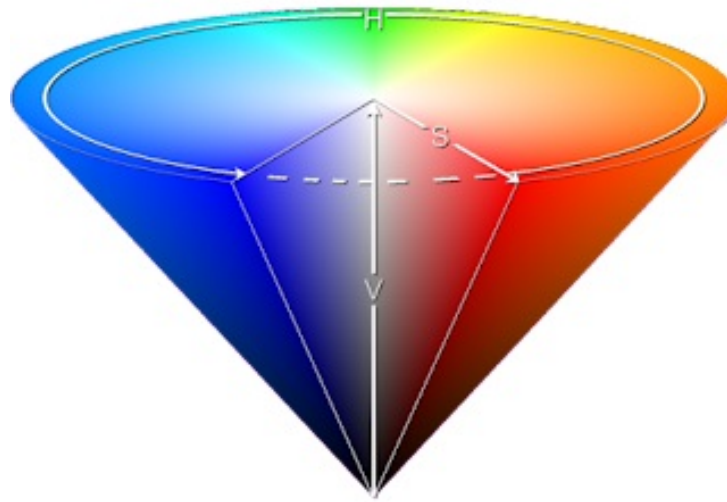


Figura 6 – Representação do Padrão de Cores HSV

A tonalidade ou matiz (Hue) é a cor pura com saturação e luminosidades máximas, por exemplo, amarelo, laranja, verde, azul, etc. A tonalidade permite fazer a distinção das várias cores puras e exprime-se num valor angular entre 0 e 360 graus.

A saturação (Saturation) indica a maior ou menor intensidade da tonalidade, isto é, se a cor é pura ou esbatida. Exprime-se num valor percentual entre 0 e 100%. O 0% significa a ausência de cor ou aproximação aos cinzentos e o valor 100% indica uma cor saturada ou pura.

O Valor/Luminosidade (Value/Brightness) traduz a luminosidade ou brilho de uma cor, isto é, se uma cor é mais clara ou escura, indicando a quantidade de luz que a mesma contém. Exprime-se num valor percentual entre 0 e 100%, sendo que o 0% indica que a cor é muito escura ou preta e o valor 100% indica que é saturada ou pura.

2.1.3 Segmentação

A segmentação é responsável por dividir a imagem em regiões de interesse segundo algum critério em particular. Quando o processo de segmentação é baseado em cor, sua

eficiência está relacionada ao fato da cor do objeto de interesse ser diferente da cor de fundo. A (Figura 7a) apresenta uma imagem contendo folhas, da qual se quer isolar (segmentar) as folhas do restante da imagem. Com o uso do filtro da média, essas regiões foram eliminadas e, então, o limiar colocado pode ser aplicado com sucesso, (Figura 7b).



(a) uma imagem original

(b) imagem segmentada

2.1.4 Extração de Características

A extração de características (feature extraction) é o processo que visa reduzir o tamanho das informações. É importante, pois imagens normalmente contém muita informação redundante que poderia vir atrapalha na classificação. 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, Richard E, 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.

2.1.4.1 Histograma

O histograma é considerado um gráfico ou plotagem relacionado à frequência de pixels em uma imagem em escala de cinza com valores de pixel (variando de 0 a 255). Além de fornece uma indicação da qualidade da imagem quanto ao contraste e intensidade luminosa.

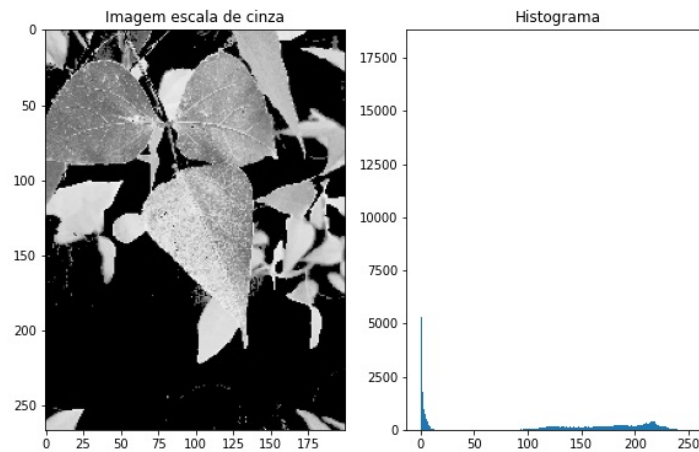


Figura 8 – Histograma

A (Figura 8) ilustra a imagem seu respectivo histograma. E extraído o histograma de cada imagem.

2.2 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. É uma boa prática usar se você quiser evitar overfitting, ou o conjunto de dados inicial for muito pequeno para treinar, ou mesmo se você 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 pixels de um canal, ou mesmo misturar as intensidades entre os canais.

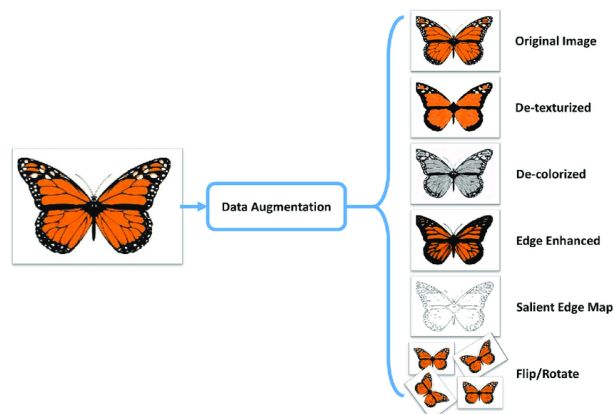


Figura 9 – Data Augmentation

2.3 Reconhecimento de Padrões

A capacidade do ser humano de reconhecer padrões e classificar objetos sempre impressionou cientistas dos mais diversificados campos de estudo. Tanto para as máquinas como para nós, humanos, o reconhecimento de objetos é proporcionado por suas características particulares. Para realizar essa tarefa, esses sistemas executam basicamente três procedimentos.

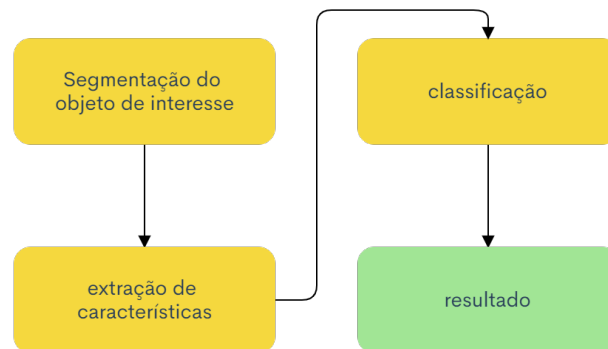


Figura 10 – Sistemas de Reconhecimento de Padrões

A segmentação do objeto e a extração de características são assuntos que foram abordados a cima. Com intuito de definir o objeto de interesse em uma determinada classe, vamos estudar como reconhecer padrões nessas características. Nesse contexto, uma classe pode ser definida como um conjunto de padrões que possuem as mesmas características.

O algoritmo que classifica o objeto a partir de suas características é conhecido como classificador. Para que os classificadores consigam designar objetos em classes, sem que tenham sido explicitamente programados, eles precisam aprender alguns critérios para executar essa tarefa. Existem dois métodos de aprendizagem de máquina bastante usados para ensiná-los: a aprendizagem supervisionada e a não supervisionada.

2.3.1 Aprendizagem supervisionada

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).

2.3.1.1 Naive Bayesian

O algoritmo "Naive Bayes" é um classificador probabilístico muito utilizado em machine learning, que assume o efeito de um determinado recurso em uma classe é independente de outros recursos. Por exemplo, um solicitante de empréstimo é desejável ou não, dependendo de sua renda, empréstimo anterior e histórico de transações, idade e localização. Mesmo que esses recursos sejam interdependentes, eles ainda são considerados

de forma independente. Essa suposição simplifica o cálculo e é por isso que é considerada ingênua. Essa suposição é chamada de independência condicional de classe, onde:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- $P(A)$: a probabilidade da hipótese A ser verdadeira (independentemente dos dados). Isso é conhecido como a probabilidade anterior de A .
- $P(B)$: a probabilidade dos dados (independentemente da hipótese). Isso é conhecido como probabilidade anterior.
- $P(A|B)$: a probabilidade da hipótese A dados os dados B . Isso é conhecido como probabilidade posterior.
- $P(B|A)$: a probabilidade do dado B dado que a hipótese A fosse verdadeira. Isso é conhecido como probabilidade posterior.

Exemplo de diagnóstico de uma nova doença. Após realizar testes, coletas e análises com 100 pessoas distintas, descobrimos que 20 pessoas possuíam a doença (20%) e 80 pessoas estavam saudáveis (80%). De todas as pessoas que possuíam a doença, 90% receberam Positivo no teste. Já 30% das pessoas que não possuíam a doença também receberam o teste positivo.

- $P(\text{doença}|\text{positivo}) = 20\% * 90\% = 0,2 * 0,9 = 0,18$.
- $P(\text{não doença}|\text{positivo}) = 80\% * 30\% = 0,8 * 0,3 = 0,24$.

Finalizado o cálculo inicial, é preciso normalizar os dados, para que a soma das duas probabilidades resulte 100% ou 1. Para gerar os dados normalizados dividimos o resultado pela soma das duas probabilidades.

- $P(\text{doença}|\text{positivo}) = 0,18/(0,18+0,24) = 0,4285$.
- $P(\text{não doença}|\text{positivo}) = 0,24/(0,18+0,24) = 0,5714$.
- $0,4285 + 0,5714 = 0,9999..$ (ou aproximadamente 1)

2.3.1.2 K Vizinhos mais Próximos (KNN)

O KNN é um algoritmo onde a estrutura do modelo será determinada pelo dataset utilizado. Este algoritmo também é conhecido como de aprendizado lento e não necessitam de dados de treinamento para se gerar o modelo, todos os dados obtidos no dataset serão

utilizados na fase de teste, resultando em um treinamento muito rápido e em um teste e validação lentos. Esse método de predição utiliza a distância entre a amostra atual e seus k vizinhos mais próximos no conjunto de treinamento para definir qual será o resultado de sua predição. Ele pode utilizar tanto para realização de classificação de instâncias como regressão.

O algoritmo de classificação utilizando kNN pode ser descrito da seguinte maneira. Dado um parâmetro k , e um grupo de amostras de treinamento $M_{x,y}$, nós descobrimos a classe de uma amostra teste t comparando a distância de t com todos as x amostras disponíveis em M .

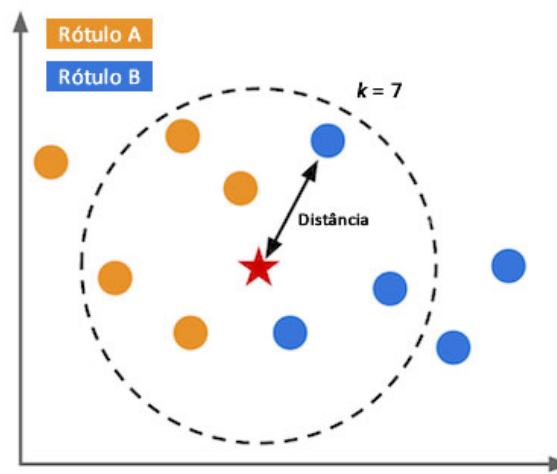


Figura 11 – exemplo de classificação do KNN com dois rótulos de classe e $k = 7$

Para determinar a classe de um elemento que não pertença ao conjunto de treinamento, o classificador KNN procura K elementos do conjunto de treinamento que estejam mais próximos deste elemento desconhecido, ou seja, que tenham a menor distância. Estes K elementos são chamados de K -vizinhos mais próximos.

Calcular a distância é fundamental para o KNN. Existem diversas métricas de distância. A mais utilizada é a distância Euclidiana.

$$D_E(p, q) = \sqrt{(p_1 - q_1)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Outros exemplos de distância, é a de Minkowsky:

$$D_M(p, q) = \left(\sum_{i=1}^n |p_i - q_i|^r \right)^{\frac{1}{r}}$$

E também, a distância de Chebyshev:

$$D_C(p, q) = \max_i(|p_i, q_i|)$$

Em relação a escolha do valor k , não existe um valor único para a constante, a mesma varia de acordo com a base de dados.

2.3.1.3 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. O nó que não possui filhos é conhecido como nó folha ou terminal.

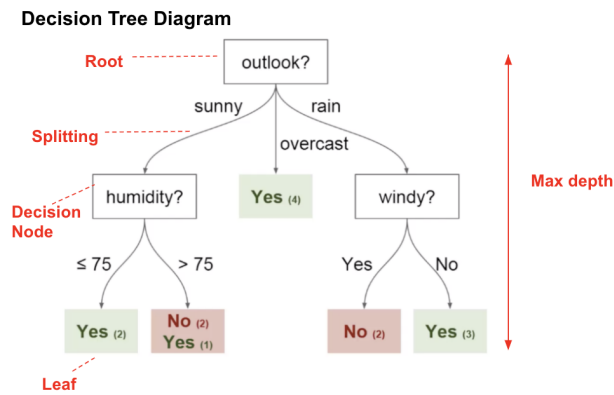


Figura 12 – Neurônios

Para se gerar um novo galho é necessário que ocorra um 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.

2.3.1.3.1 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 o gráfico:

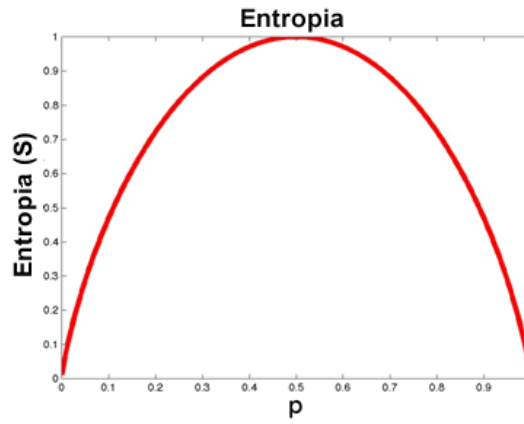


Figura 13 – Entropia

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

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:

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

2.3.1.3.2 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:

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

2.3.1.3.3 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.

2.3.1.4 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 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.

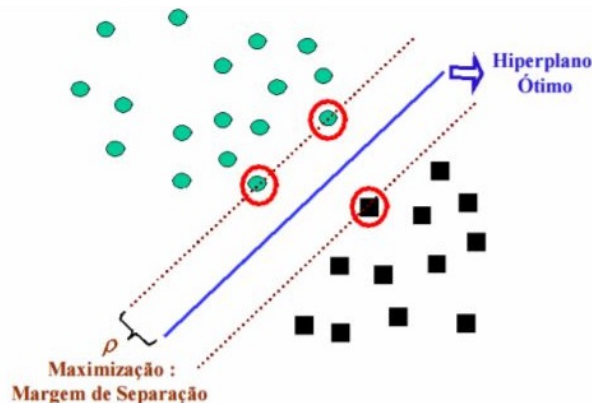


Figura 14 – O hiperplano ótimo separando os dados com a máxima margem ρ .

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

2.3.1.4.1 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 como:

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

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).

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

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.

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

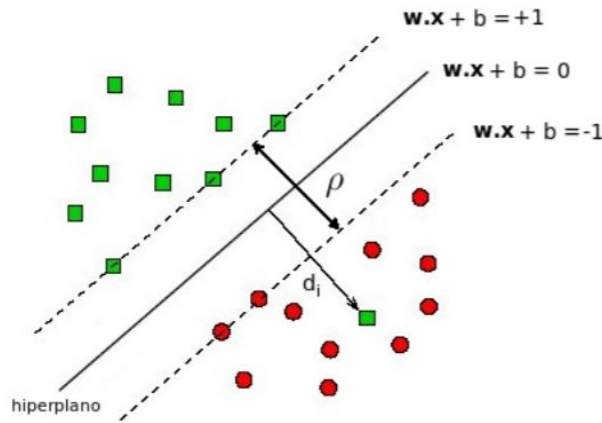


Figura 15 – mostra a distância entre hiperplanos e os vetores suporte.

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

2.3.1.4.2 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	Função $K(x_i, x_j)$	Tipo do Classificador
Polinomial	$(\langle x_i \cdot x_j \rangle + 1)^p$	Máquina de aprendizagem polinomial
RBF	$\exp\left(-\frac{\ x_i - x_j\ ^2}{2\sigma^2}\right)$	Rede RBF
sigmóide	$\tanh(\beta_0 \langle x_i \cdot x_j \rangle) + \beta_1$	Perceptron de duas camadas

2.3.1.5 Redes Neurais Artificiais

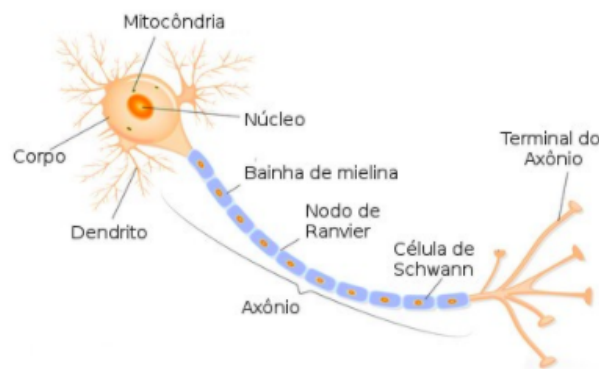


Figura 16 – Neurônios

Os neurônios são as células que formam o nosso cérebro. Elas são compostas basicamente por três partes: os dendritos, 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 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 dá 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.

2.3.1.5.1 O Neurônio Artificial

O neurônio artificial é a menor unidade de processamento de uma rede neural, que recebe sinais de entrada e produz sinais de saída. O modelo de neurônio mais utilizado é o perceptron, representado na Figura 17. 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:

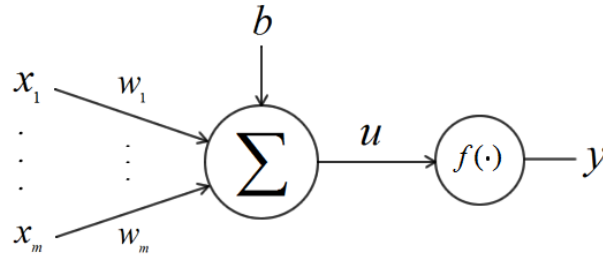


Figura 17 – O modelo do neurônio artificial perceptron

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

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 sigmoide e a relu, definidas pelas equações abaixo, respectivamente:

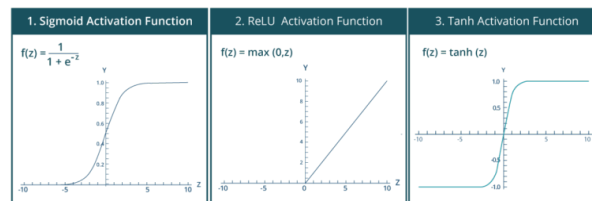


Figura 18 – funções de ativação

2.3.1.5.2 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 19 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.

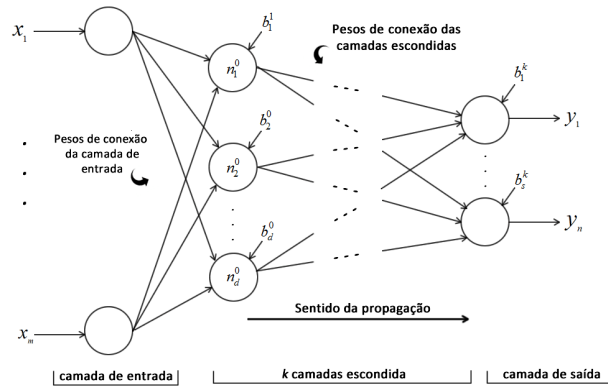


Figura 19 – Exemplo de rede neural artificial do tipo feedforward com k camadas escondidas

2.3.1.5.3 Algoritmo de treinamento

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).

$$MSE = \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^n (d_j^i - y_j^i)^2 \quad (2.11)$$

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:

$$W = W + \Delta W \rightarrow \Delta W = -\alpha \frac{\partial E}{\partial W} \quad (2.12)$$

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 .

2.3.1.6 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.

Uma CNN é composta por uma camada de convolução (Convolutional Layer), uma cada camada de subamostragem (Subsampling Layers ou Pooling Layers) uma camada chamada de RELU e uma camada totalmente conectada (Fully Connected Layers).

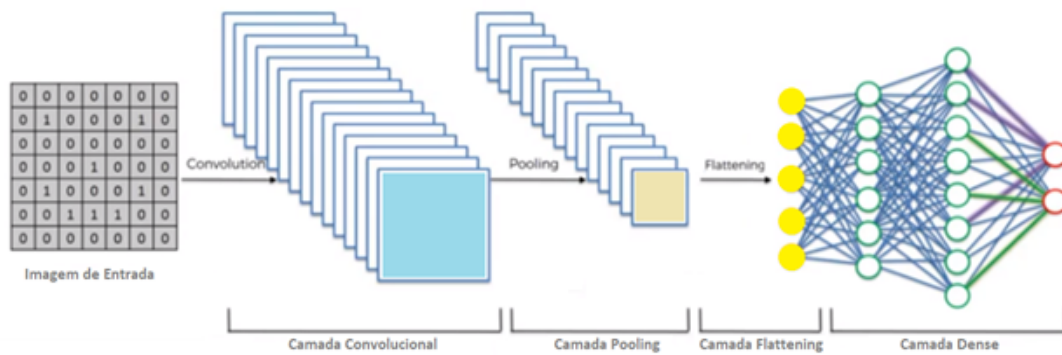


Figura 20 – Esquema de uma Rede Neural Convolucional.

2.3.1.6.1 Convolução

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

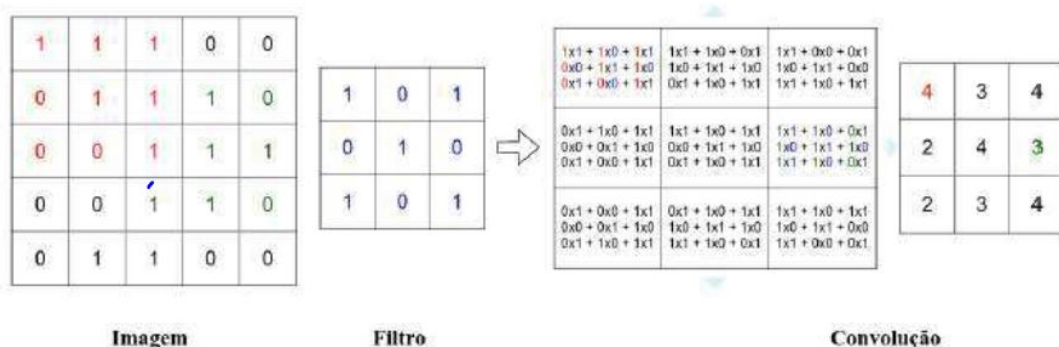


Figura 21 – convolução em uma imagem

2.3.1.6.2 Pooling

Na camada de pooling as saídas resultantes da camada de convolução tem as informações reduzidas em tamanho. A função desta camada é focar nos dados que são realmente importantes para a análise e tornar 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.

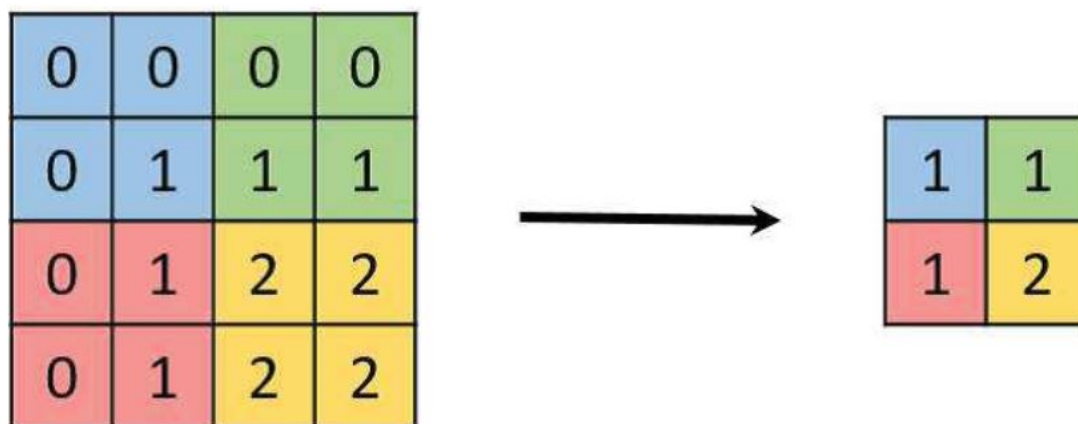


Figura 22 – Aplicação de max pooling em uma imagem 4x4 utilizando um filtro 2x2

2.3.1.6.3 Flatten

A camada flatten é normalmente utilizada na divisão das 2 partes da CNN (extração de características/rede neural tradicional). Ela basicamente opera uma transformação na matrix da imagem, alterando seu formato para um array. Por exemplo, uma imagem em grayscale de 28x28 será transformada para um array de 784 posições.

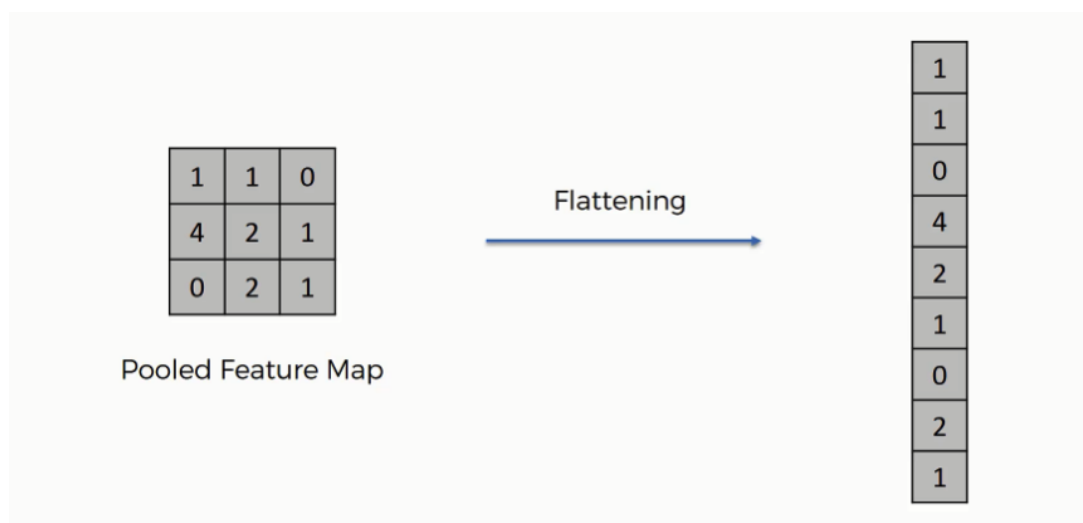


Figura 23 – Operação de Flatten

Na camada totalmente conectada como o 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 26)

2.4 Cross-validation

A ideia por trás do cross-validation é simples, porém muito efetiva. De maneira geral, ele nada mais é do que um procedimento na qual os dados são organizados. Como uma imagem vale mais do que mil palavras, vamos primeiro dar uma olhada na Figura 24 para termos uma ideia geral da técnica.

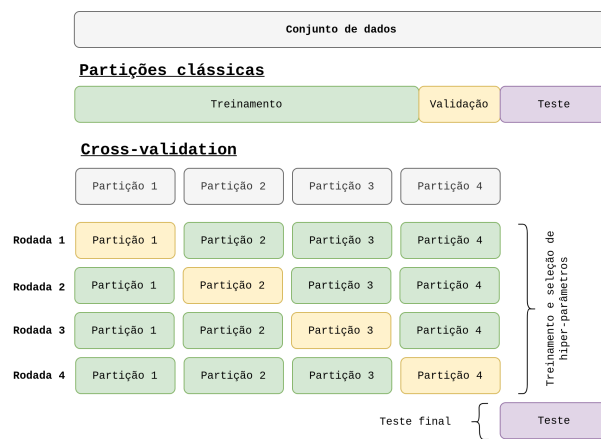


Figura 24 – Ilustração de cada tipo de partição, a versão clássica e o cross validation.

Os dados são divididos em partições (no inglês conhecido como folders). Idealmente, cada partição deve conter a mesma quantidade de dados. Porém, isso depende do dataset e a maneira de dividir os 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, 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.

2.4.1 Hiperparâmetros

Hiperparâmetros são os valores de configuração do modelo que um algoritmo de aprendizado usa. Esses valores não são aprendidos com os dados de treinamento. Esses parâmetros são importantes pois determinam propriedades do modelo, como complexidade, velocidade de aprendizagem, etc.

Para ajuste de hiperparâmetro na validação cruzada de K-fold, muitas combinações dos valores de hiperparâmetro são escolhidas a cada vez para realizar K iterações. Então, a melhor combinação é selecionada e testada. Portanto, para uma validação cruzada de 5 dobras para ajustar 5 parâmetros, cada um testado com 5 valores, 15625 iterações estão envolvidas. Há diversas estratégias como ManualSearch, GridSearchCV e RandomSearchCV. A principal diferença entre essas duas técnicas é a obrigação de experimentar todos os parâmetros. GridSearchCV tem que tentar TODAS as combinações de parâmetros, entretanto, RandomSearchCV pode escolher apenas algumas combinações 'aleatórias' de todas as combinações disponíveis.

2.5 Medidas de Desempenho

2.5.1 Matriz de Confusão

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. 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 25 – 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.

2.5.2 Acurácia

Acurácia é a métrica mais simples, ela representa o número de previsões corretas do modelo. Porém, não devemos utilizar a acurácia, quando temos classes desbalanceadas, causa uma falsa impressão de bom desempenho.

Se o modelo escolhido conseguir classificar corretamente todas as pessoas que não tem a doença e errar a classificação de todos os doentes, teremos uma acurácia de 95%. Dando uma falsa impressão que o modelo treinado tem uma ótima previsão. Porém, o modelo não consegue classificar corretamente a classe de interesse.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.13)$$

2.5.3 Precision

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 positiva. Ou seja, entre todas as observações identificadas como positivas, quantas foram identificadas corretamente.

$$Precision = \frac{VP}{VP + FP} \quad (2.14)$$

2.5.4 Recall

Recall 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.

$$Recall = \frac{VP}{VP + FN} \quad (2.15)$$

2.5.5 F1- Score

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

$$F1 = 2 * \frac{recall * precision}{recall + precision} \quad (2.16)$$

2.5.6 Área sob a Curva ROC

A curva ROC (Receiver Operating Characteristic Curve) 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.

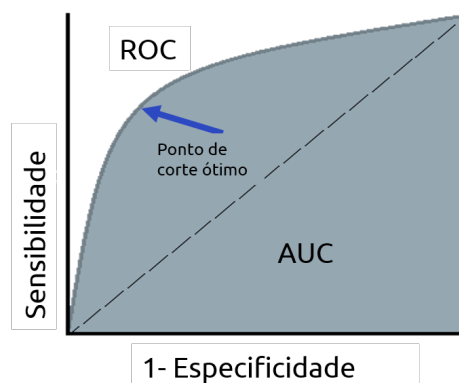


Figura 26 – Curva ROC

3 Trabalhos Relacionados

Em [Hossain, J. and Amin, M. A (2010)] os autores utilizaram 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 precisão de 91,41 %.

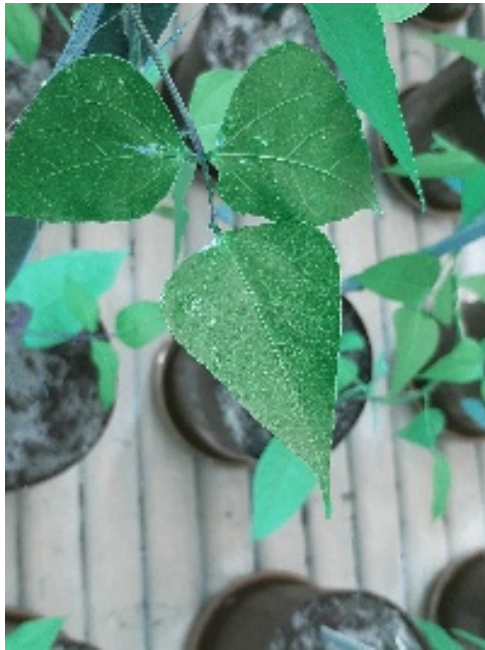
Em [Kumar, et al, Kamble, M., Pawar, S., Patil, P., Bonde, N. (2011)] é feita uma pesquisa de survey sobre os melhores classificadores para reconhecimento de plantas e folhas. Este autor determina que o classificador KNN (K-Nearest Neighbors) é o que possui o funcionamento mais simples, e tem seu uso mais específico para datasets pequenos. Apesar de ter um funcionamento simples, este obtém bons resultados e tem a vantagem de o seu tempo de computação para treinamento dos modelos de reconhecimento serem quase instantâneos.

No trabalho de [Satti, Vijay and Satya, Anshul and Sharma, Shanu (2013)] o KNN é utilizado, assim como, a classificação ANN (Artificial Neural Network) e seus resultados são comparados. Como resultado é verificado que a precisão do KNN é maior quando o dataset tem menos classes, porém quando o dataset aumenta, o ANN ultrapassa em precisão.

4 Metodologia

4.1 Aquisição das imagens

As imagens foram disponibilizadas por Raphael Miller e possui um total de 229 imagens, sendo 109 de Fava Cearense (Figura 27a), e 120 de Orelha de Vó (Figura 27b). Todas fotografadas com câmera digital.



(a) Fava Cearense



(b) Fava Orelha de Vó

Onde cada imagem irá se tornar uma instância do problema e cada atributo será o conjunto de pixel de cada imagem.

4.2 Pré-processamento

Com o intuito de obter um melhor desempenho, realizamos o processo de segmentação, nesta etapa de Segmentação baseada em cores, que realizamos no conjunto de imagens (Figura 28a), removendo todas as informações de cores que não faz parte do escopo do nosso problema, gerado assim um novo conjunto de imagens (Figura 28b), com isso os verdes são mantidos, bem como alguns tons de marrom e amarelo que as plantas das imagens possuem substituindo o restante dos pixels por zero dando uma aparência de imagens sem fundo.



(a) Com o fundo (original)

(b) Sem o fundo

para a remoção das informações de pixel segmentação por cor, foi feita utilizando o OpenCV (Open Source Computer Vision Library), uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto disponível em Python ([OpenCV, 2021](#)).

4.2.1 Transformação do espaço de cores

Aplicando esta transformação, usado um filtro no padrão HSL (Hue, Saturation, Lightness) também conhecido com HSV (Hue, Saturation, Value), onde as cores são definidas por meio de valores de tonalidade, saturação e brilho. O filtro analisa a imagem pixel a pixel e torna preto todo pixel que não está no intervalo definido. O intervalo de cor usado no filtro foi entre 25 de tonalidade, 52 de saturação e 72 de brilho até 102 de tonalidade, 255 de saturação, 255 e brilho (Figura 29a e Figura 29b).



(a) Imagem modificada 1

(b) Imagem modificada 2

4.2.2 Transformação geométrica

Este método também foi utilizado em conjunto com método citado à cima para criar um novo conjunto de dados, utilizando modificações de forma aleatórias de espelhamento horizontal, espelhamento vertical. Para cada imagem, foram gerada cinco imagens modificadas, usando de uma a todas as modificações supracitadas. Após a aplicação desta técnica, o novo conjunto de imagens aumentou para 654 imagem de Fava Cearense e 720 imagens de Orelha de Vó, totalizando 1374 imagens (Figura 30a e Figura 30b).



(a) Transformação geométrica 1

(b) Transformação geométrica 2

Aplicando a técnica de segmentação citada anteriormente e redimensionadas de 3456x4608 pixel tamanho para 200x267 pixel, mantendo a mesma proporção, nesse novo

conjunto de imagens resultando em um outro conjunto com fundo removido, mantendo as mesmas quantidade de imagens do conjunto de origem.

Conjunto	Fava Cearense	Orelha de Vó	Total
Imagens Aumentadas	654	720	1374
Imagens Aumentadas e Sem Fundo	654	720	1374

Tabela 2 – Distribuição dos conjuntos de imagens após o pré-processamento

Todos esses conjuntos de imagens estão disponível pelo link: [ericoandre \(2021\)](#)

4.2.3 Extração de Características

Nesta etapa é realizada a conversão de imagens coloridas em imagens com apenas um canal de cor (nível-de-cinza). Então para cada imagem do conjunto convertida e extraído a matriz de pixel de cada imagem convertida em um vetor o seu histograma, assim e criada uma instância do problema ou feature, é composta do vetor do histograma da imagem.

4.3 Seleção de Modelos

Para a seleção dos modelos utilizou-se a busca aleatória para escolha dos melhores hiperparâmetros em cada algoritmo de aprendizado de máquina. Foi utilizado o `RandomSearchCV` para automatizar essa busca pelos melhores hiperparâmetros. Esse metodo pusue um grande benefício significativo em comparação ao `GridSearchCV` é a redução do tempo de processamento pois ele não tentar todas as combinações hiperparâmetros como o `GridSearchCV`.

A Tabela 3 Mostra os hiperparâmetros utilizados no `RandomSearchCv`. A busca é realizado no qual pega de forma aleatória um conjunto de hiperparâmetros para treinar o algoritmo, até obter o melhor modelo com os melhores conjuntos de hiperparâmetros para então poder utilizá-lo na base de teste.

Tabela 3 – Hiperparâmetros

Algoritmos	Hiperparâmetros	Variação
BAYES	var smoothing	0.0000000001 , 0.000000000001, 0.000000000001
TREE	criterion	gini, entropy
TREE	splitter	best, random
TREE	min samples split	0 a 1
TREE	min samples leaf	0.000001 a 0.5
TREE	max leaf nodes	1 a 1000
TREE	max depth	1 a 1000
KNN	n neighbors	1 a 50
SVM	C	0.0001 a 100.00
SVM	kernel	poly, rbf, linear
SVM	degree	2 a 5
SVM	gamma	0.0001 a 100.00
MLP	learning rate init	Exp(0.000001 , 0.1)
MLP	max iter	1 a 1000
MLP	momentum	0.5 a 0.4999
MLP	solver	lbfgs, sgd, adam
CNN	epochs	30
CNN	batch size	16, 32
CNN	optimizer	adam, gdm

Na tabela 4, mostra os melhores hiperparâmetros selecionados para o problema.

Tabela 4 – Melhores Hiperparâmetros

Algoritmos	Hiperparâmetros	Variação
BAYES	var smoothing	0.0000000001
TREE	criterion	entropy
TREE	max depth	314
TREE	max leaf nodes	22
TREE	min samples leaf	0.003534152609858703
TREE	min samples split	0.023062425041415757
TREE	splitter	best
KNN	n neighbors	1
SVM	C	0.0013304651439322366
SVM	kernel	linear
SVM	degree	2
SVM	gamma	0.0016962756142947161
MLP	learning rate init	0.020262142283225707
MLP	max iter	460
MLP	momentum	0.8057654149280916
MLP	solver	lbfgs
MLP	activation	relu
MLP	hidden layer sizes	60
CNN	epochs	30
CNN	batch size	16
CNN	optimizer	adam

5 Resultados

Os resultados foram obtidos a partir dos melhores hiperparâmetros, para treinar os respectivos modelos. Nessa fase, utilizou-se o conjunto de testes, que representa 30% do conjunto total de dados utilizando técnica de k-folder com 5 folder. Os resultados dos testes são descritos a seguir.

Através da matriz de confusão é possível perceber que todos os modelos tiveram a quantidade de acertos superior à quantidade de erros. Por meio da matriz de confusão é possível avaliar o desempenho dos Algoritmos de classificadores, por meio das métricas de avaliação de desempenho descrita na Tabela 6.

Tabela 5 – Taxa de recall e Especificidade dos Algoritmos

Algoritmos	recall	Especificidade	Média
BAYES	0.62	0.72	0.67
TREE	0.85	0.69	0.77
KNN	0.97	0.89	0.93
SVM	0.94	0.92	0.93
MPL	0.96	0.93	0.94
CNN	0.97	0.98	0.97

Considerando a média entre sensibilidade e especificidade, o CNN é o modelo que possui os maiores scores, seguido do MLP e SVM, KNN, Árvore de Decisão e por último, Naive Bayes. No entanto, a sensibilidade e especificidade são duas medidas difíceis de conciliar, isso porque uma é complementar a outra, ou seja, enquanto uma aumenta, a outra diminui. Dessa forma, a Curva ROC (Receiver Operating Characteristic Curve), representa a sensibilidade em função da taxa de falsos positivos, assim, quanto mais a curva se aproxima do canto superior esquerdo, melhor é o classificador.

5.0.1 Redes Neurais Artificiais CNN

Na Figura 31 está dada a Curva ROC e a matriz de confusão da modelo CNN. Mesmo com limitações de hardware para treinar o modelo, este obteve melhores resultados, como mostra na tabela 6.

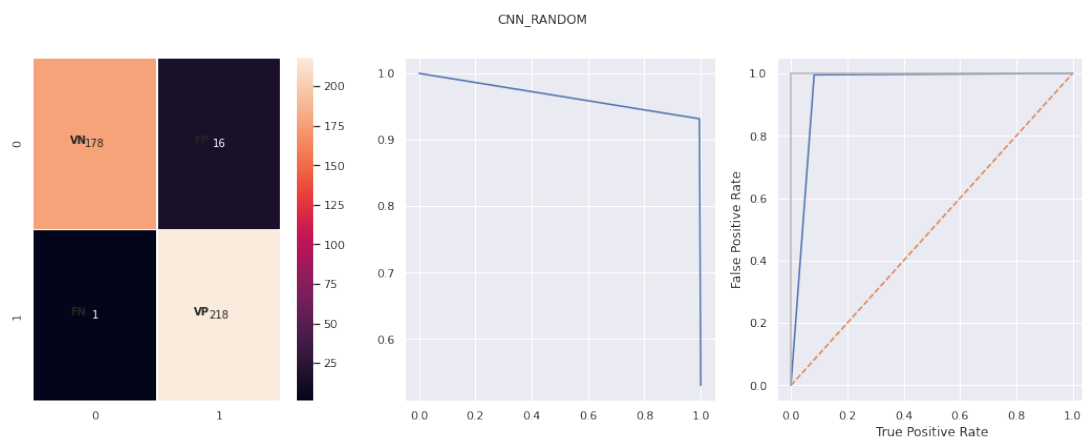


Figura 31 – Curvas ROC do CNN sobre o conjunto de teste para classificação de Fava Cearense e Orelha de Vó.

5.0.2 Redes Neurais Artificiais MLP

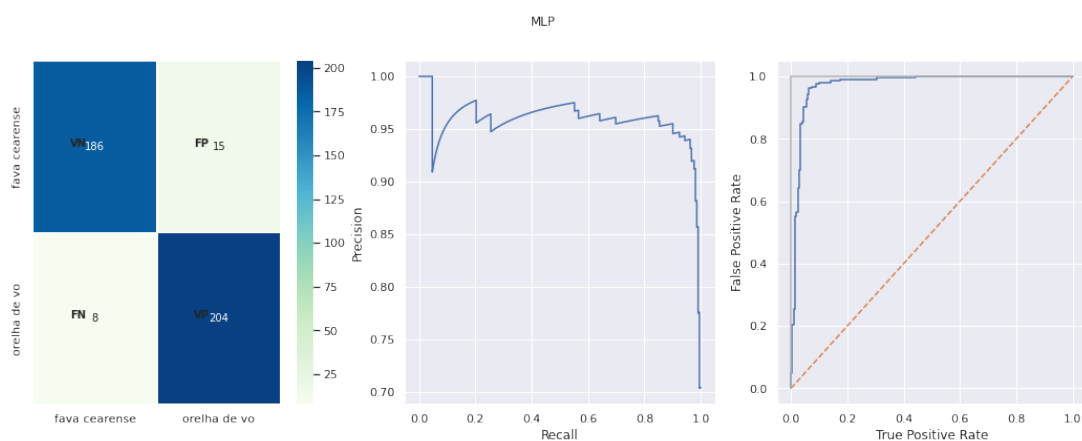


Figura 32 – Curvas ROC do MLP sobre o conjunto de teste para classificação de Fava Cearense e Orelha de Vó.

5.0.3 Support Vector Machine

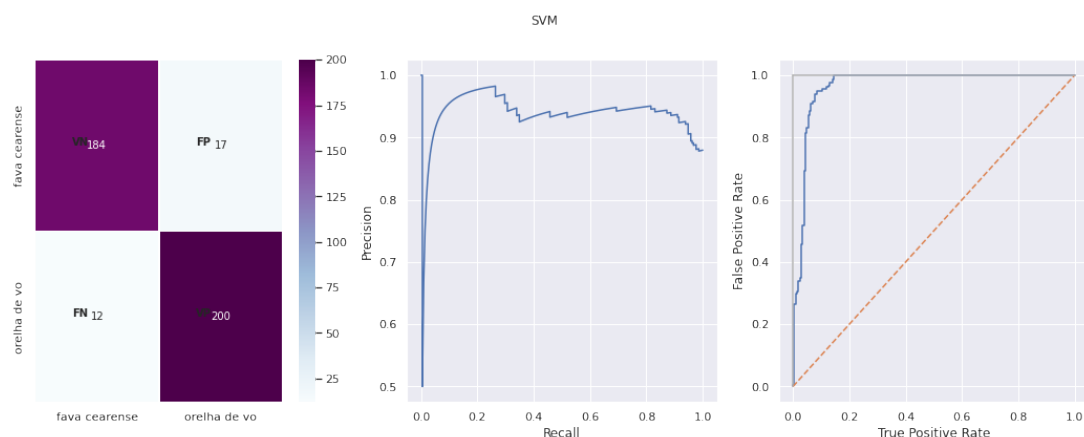


Figura 33 – Curvas ROC do SVM sobre o conjunto de teste para classificação de Fava Cearense e Orelha de Vó.

5.0.4 K Vizinhos mais Próximos

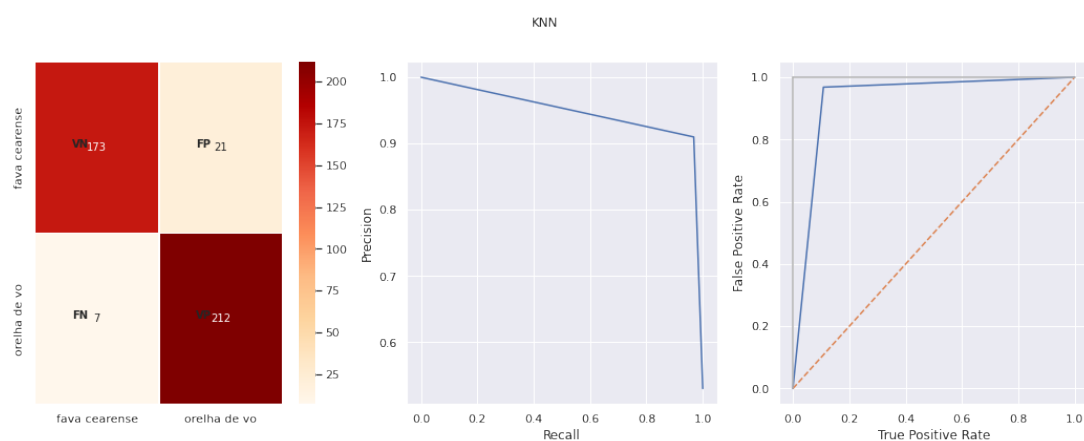


Figura 34 – Curvas ROC do KNN sobre o conjunto de teste para classificação de Fava Cearense e Orelha de Vó.

5.0.5 Árvores de Decisão

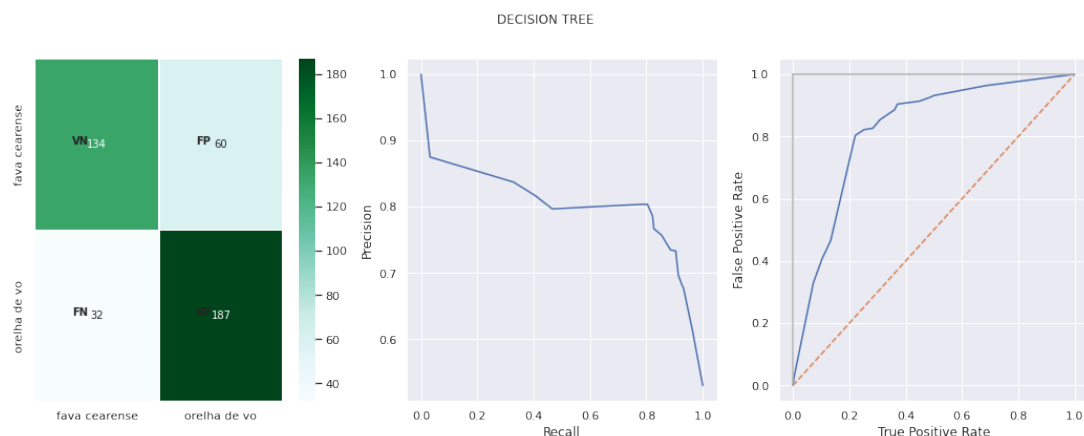


Figura 35 – Curvas ROC do TREE sobre o conjunto de teste para classificação de Fava Cearense e Orelha de Vó.

5.0.6 Naive Bayesian

O algoritmo Naive Bayesian foi testado com apenas um hiperparâmetros e o RandomSearchCv com três iterações, os demais algoritmos rodaram com 10 iterações, essa iterações são parâmetros da função RandomSearchCv.

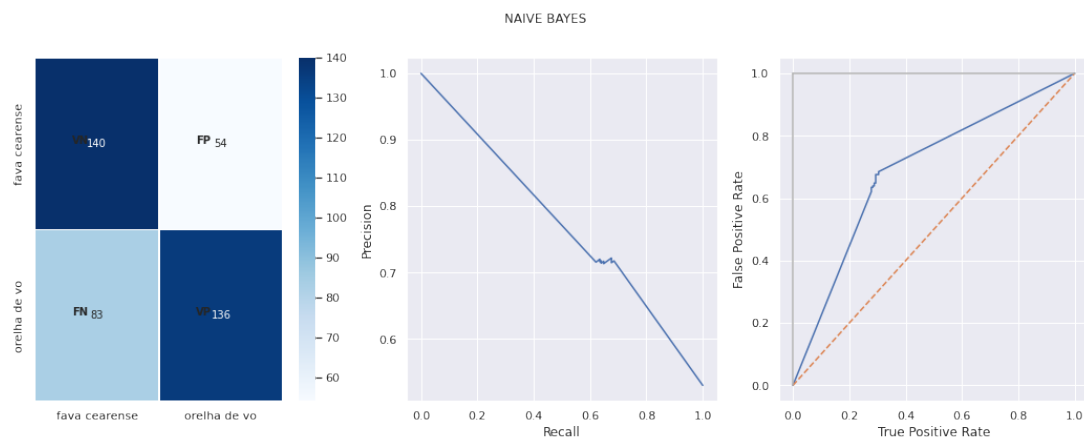


Figura 36 – Curvas ROC do Naive Bayesian sobre o conjunto de teste para classificação de Fava Cearense e Orelha de Vó.

Considerando as taxas de especificidade, sensibilidade e a curva ROC, o melhor modelo foi o CNN, e sendo o pior a Naive Bayesian.

Tabela 6 – Comparação dos resultados dos modelo

Algoritmos	Accuracia	ACC	Precision	Recall	F1-Scor	Matrix Conf
BAYES	0.668	0.775	0.716	0.621	0.665	140, 54, 83, 136
TREE	0.777	0.803	0.757	0.854	0.803	134, 60, 32, 187
KNN	0.932	0.947	0.910	0.968	0.938	173, 21, 7, 212
SVM	0.930	0.938	0.922	0.943	0.932	184, 17, 12, 200
MPL	0.944	0.958	0.932	0.962	0.947	186, 15, 8, 204
CNN	0.973	0.991	0.981	0.968	0.975	190, 4, 7, 212

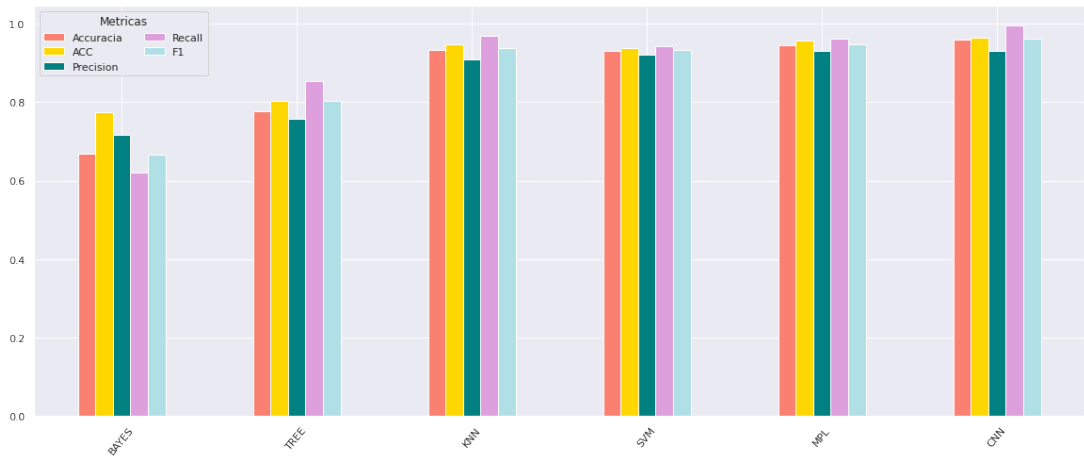


Figura 37 – Este gráfico e a representação visual da tabela 6

6 CONCLUSÕES

Atualmente inúmeras abordagens utilizando Redes Neurais com aprendizado profundo estão alcançando os melhores resultados em diversos problemas de Visão Computacional. Os métodos de CNN conseguem extrair características otimizadas da tarefa de classificação. Fato é que estes levam em consideração os pixel vizinhos, pois a posição de um pixel em relação a seu vizinho tem uma grande importância para reconhecer os padrões dos objetos nas imagens.

Referências

ericoandre. *Reconhecimento de Padrões*. 2021. https://github.com/ericoandre/reconhecimento_padroes. Citado na página 32.

GONZALEZ, ; WOODS, Richard E. *Processamento digital de imagens*. 3^a ed. Tradução de Cristina Yamagami e Leonardo Piamonte. [S.l.]: Pearson Prentice Hall, 2010. ISBN 9788576054016. Citado 2 vezes nas páginas 5 e 9.

Halliday. *Fundamentos De Física - Volume 4 - Optica E Física*. [S.l.]: LTC, 2016. ISBN 9788521630388. Citado na página 7.

Haykin, S. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: LTC Editora Rio de Janeiro, Brazil:, 2007. Citado na página 19.

Hossain, J. and 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 28.

Kumar, at all, Kamble, M., Pawar, S., Patil, P., Bonde, N. Survey on techniques for plant leaf classification. *International Journal of Modern Engineering Research*. v. 1, p. 538–544, 2011. Citado na página 28.

OpenCV. *OpenCV*. 2021. <https://opencv.org/about/>. Accessed: 2021-10-18. Citado na página 30.

Satti, Vijay and Satya, Anshul and Sharma, Shanu. An automatic leaf recognition system for plant identification using machine vision technology. *International Journal of Engineering Science and Technology (IJEST)*, v. 5, p. 874–879, 04 2013. Citado na página 28.

SOFTEX. *CADERNOS TEMÁTICOS DO OBSERVATÓRIO: TIC NO AGRONEGÓCIO. Associação para Promoção da Excelência do Software Brasileiro*. [S.l.]: Campinas SP, 2016. Citado na página 4.