

Narciso de Sousa Rodrigues
Orientador: Romuere Rodrigues Veloso e Silva

**Uma ferramenta automática para
reconhecimento de padrões e classificação de
dados**

Picos - PI
31 de Julho de 2023

Narciso de Sousa Rodrigues
Orientador: Romuere Rodrigues Veloso e Silva

Uma ferramenta automática para reconhecimento de padrões e classificação de dados

Trabalho de Conclusão de Curso de Bacharelado em Sistemas de Informação do Campus Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte dos requisitos para obtenção do Grau de Bacharel em Sistemas de Informação

Universidade Federal do Piauí
Campus Senador Heuvídio Nunes de Barros
Bacharelado em Sistemas de Informação

Picos - PI
31 de Julho de 2023

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

R696f Rodrigues, Narciso de Sousa

Uma ferramenta automática para reconhecimento de padrões e classificação de dados [recurso eletrônico] / Narciso de Sousa Rodrigues - 2023.

50 f.

1 Arquivo em PDF

Indexado no catálogo *online* da biblioteca José Albano de Macêdo-CSHNB

Aberto a pesquisadores, com restrições da Biblioteca

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do Piauí, Bacharelado em Sistemas de Informação, Picos, 2023.

“Orientador : Prof. Dr. Romuere Rodrigues Veloso e Silva”

1. Processamento de imagens. 2. Visão computacional. 3. Análise de algoritmos. 4. Ferramenta – reconhecimento de padrões. I. Silva, Romuere Rodrigues Veloso e. II. Título.

CDD 005.1

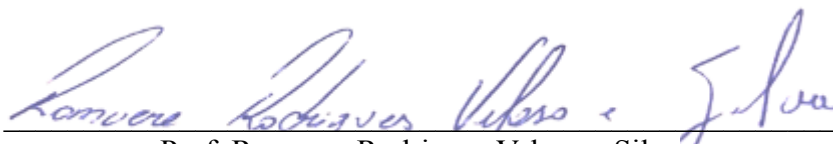
UMA FERRAMENTA AUTOMÁTICA PARA RECONHECIMENTO DE PADRÕES E
CLASSIFICAÇÃO DE DADOS

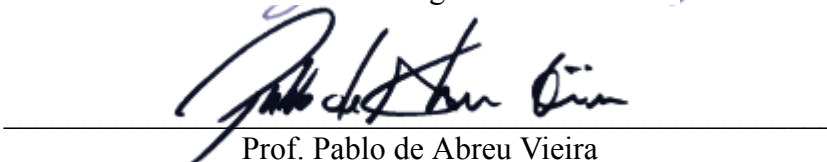
NARCISO DE SOUSA RODRIGUES

Monografia aprovada como exigência parcial para obtenção do grau de Bacharel em Sistemas
de Informação.

Data de Aprovação

Picos – PI, 14 de agosto de 2023


Prof. Romuere Rodrigues Veloso e Silva


Prof. Pablo de Abreu Vieira


Prof. Júlio Vitor Monteiro Marques

Agradecimentos

Primeiramente agradecer a Deus por me permitir a conclusão do curso que foi almejado desde sempre, um sonho desde que conheci o ramo da informática, desafiador porém muito gratificante.

Aos meus pais, João Cicero e Josefa Rodrigues, que são minha base, eles quem me sustentaram, não foi uma luta fácil de ser vencida. Faço aqui também os agradecimentos aos meus irmãos, Cicero que me ajudou também em tudo que podia e a Juliana a qual sempre nos divertimos quando eu tinha oportunidade de visitá-los em casa aos finais de semana, era muito renovador e gratificante.

Agradecer também ao Saul Rocha, amigo/irmão que o curso me deu, afinal foi devido ao curso que começamos morar juntos, e era ele um dos que eu via sempre, dentro ou fora da UFPI, sempre ajudamos um ao outro, desde incentivos até mesmo a ajuda com conteúdos e dúvidas.

Agradecer também ao Pedro Azevedo, outro irmão que o curso me proporcionou, quem diria que desde o primeiro dia de aula eu conheceria alguém tão especial nesta jornada, este de longe, outras culturas e outro sotaque ao qual sempre achei "muito massa, visse".

Aos demais moradores do, famoso, Apto 201 que se tornaram irmãos de consideração, a vida de estudante não é fácil, sem ajudas e incentivos esta formação não se tornaria possível.

Estendo meus agradecimentos, para o Elias Macedo, sua família e todos seus colaboradores da Expresso Macedo, afinal, meu curso se tornou menos dolorido devido à possibilidade de visitar meus pais, e essa possibilidade se tornou viável devido ao suporte dessa família maravilhosa, sempre fui admirador da humildade e carinho que essa família e a empresa Expresso Macedo tem com todos os alunos de nossa cidade de Acauã-PI.

Meus agradecimentos ao meu orientador, Prof. Dr. Romuere Rodrigues Veloso e Silva, foi como um pai no meu trajeto de curso, o mesmo quem me apresentou a visão computacional e que teve a paciência de me guiar em projetos e nesse trabalho de conclusão, sempre com paciência, positividade e paciência, nunca exitou em transmitir seu conhecimento.

Agradecer também à Universidade Federal do Piauí que forneceu toda estrutura física e de professores capacitados por todo meu período da graduação.

Por fim, não poderia deixar de agradecer à Maiara Macedo, minha noiva, companheira incrível que a vida me deu, sempre me incentivando e me acalmando nas horas de aperto, onde o estresse e a frustração queriam aparecer ela vinha com empatia, carinho e incentivos, me tranquilizando e fazendo erguer à cabeça para cada batalha.

Steve Jobs

A criatividade é a chave para a inovação.

Resumo

A visão computacional é uma área de atuação fundamental no campo da ciência da computação, com o objetivo de fornecer ao computador métodos para interpretar e extrair informações relevantes a partir de imagens e vídeos do ambiente circundante. Por meio dessa disciplina, as máquinas adquirem a capacidade de reconhecer, manipular e identificar padrões nos objetos presentes nas imagens.

O avanço tecnológico tornou possível o desenvolvimento de novos algoritmos de processamento de imagem, mas também tornou mais difícil a seleção do algoritmo adequado para um determinado problema. A etapa de análise dos algoritmos de processamento de imagem é importante para garantir que a solução proposta seja eficaz e eficiente.

Nesse contexto, este trabalho tem como objetivo desenvolver uma aplicação *desktop* que facilite a comparação dos algoritmos de visão computacional do estado da arte, possibilitando ao usuário uma escolha mais ágil e clara da melhor abordagem para um determinado problema. O sistema proposto será composto por: 1) algoritmos de extração de atributos baseados em redes neurais convolucionais; 2) algoritmos de classificação que utilizam os atributos calculados pelas redes neurais convolucionais como entrada; e 3) métricas de avaliação de desempenho que permitirão uma escolha quantitativa da melhor solução. Além disso, a aplicação permitirá a combinação de diversas metodologias simultaneamente.

O desenvolvimento dessa aplicação representa uma contribuição significativa para a área de visão computacional, pois oferecerá aos pesquisadores uma ferramenta eficiente e abrangente para a análise e comparação de algoritmos, reduzindo o tempo necessário para essa etapa e fornecendo resultados mais precisos na seleção da melhor solução para um determinado problema.

Palavras-chaves: visão computacional, processamento de imagens, ferramenta para reconhecimento de padrões.

Abstract

Computer vision is a fundamental area of activity in the field of computer science, with the aim of providing the computer with methods to interpret and extract relevant information from images and videos of the surrounding environment. Through this discipline, machines acquire the ability to recognize, manipulate and identify patterns in the objects present in the images.

Technological advancement has made it possible to develop new image processing algorithms, but it has also made it more difficult to select the appropriate algorithm for a given problem. The analysis step of image processing algorithms is important to ensure that the proposed solution is effective and efficient.

In this context, this work aims to develop a desktop application that facilitates the comparison of state-of-the-art computer vision algorithms, allowing the user to choose more quickly and clearly the best approach for a given problem. The proposed system will consist of: 1) feature extraction algorithms based on convolutional neural networks; 2) classification algorithms that use the attributes calculated by convolutional neural networks as input; and 3) performance evaluation metrics that will allow a quantitative choice of the best solution. In addition, the application will allow the combination of several methodologies simultaneously.

The development of this application represents a significant contribution to the field of computer vision, as it will offer researchers an efficient and comprehensive tool for analyzing and comparing algorithms, reducing the time required for this step and providing more accurate results in selecting the best solution for a certain problem.

Lista de ilustrações

Figura 1 – Arquitetura de uma Rede Neural Convolutacional.	14
Figura 2 – Arquitetura AlexNet.	16
Figura 3 – Arquitetura DenseNet.	17
Figura 4 – Arquitetura da VGG-16.	17
Figura 5 – Arquitetura da VGG-16 comparada a VGG-19.	18
Figura 6 – Diagrama de funcionamento Inception.	19
Figura 7 – Diagrama do atalho ResNet.	19
Figura 8 – Exemplo divisão de grupos SVM.	20
Figura 9 – Funcionamento do <i>random forest</i>	21
Figura 10 –Funcionamento do MLP.	21
Figura 11 –Diagrama UML demonstrando a metodologia do trabalho proposto.	26
Figura 12 –Tela <i>Convolution Neural Network</i>	31
Figura 13 –Seção <i>Setup Convolution Neural Network</i>	32
Figura 14 –Configurações de <i>Setup CNN</i>	33
Figura 15 –Seção <i>Architecture Convolution Neural Network</i>	34
Figura 16 –Seção <i>Input and Output Convolution Neural Network</i>	34
Figura 17 –Seção <i>Training and Test Setup Convolution Neural Network</i>	35
Figura 18 –Seção <i>Monitor Convolution Neural Network</i>	35
Figura 19 –Seção <i>Logs Convolution Neural Network</i>	36
Figura 20 –Tela <i>Classification</i>	36
Figura 21 –Seção <i>Datasets Classification</i>	37
Figura 22 –Seção <i>Models Classification</i>	38
Figura 23 –Seção <i>Results Classification</i>	38
Figura 24 –Seção <i>Logs Classificação</i>	39
Figura 25 –Tela <i>Metrics and Charts</i>	39
Figura 26 –Seção Modelos Salvos <i>Metrics and Charts</i>	40
Figura 27 –Seção Métricas <i>Metrics and Charts</i>	41
Figura 28 –Seção Resultados <i>Metrics and Charts</i>	41
Figura 29 –Gráfico do questionário, referente à usuários que tem conhecimento em visão computacional.	42

Lista de tabelas

Tabela 1 – Trabalhos Relacionados.	24
Tabela 2 – Requisitos Funcionais.	27
Tabela 3 – Requisitos Não Funcionais.	28
Tabela 4 – Questionário proposto por (BROOKE, 1996) no SUS.	29
Tabela 5 – Sistema de Pontuação do SUS.	30
Tabela 6 – Tabela com os resultados da pesquisa com usuários e questionário SUS.	44

Lista de abreviaturas e siglas

CNN	<i>Convolutional Neural Network</i>
VGG	<i>Visual Geometry Group</i>
ResNet	<i>Residual Network</i>
SVM	<i>Support Vector Machines</i>
MLP	<i>Multilayer Perceptron</i>

Sumário

1	Introdução	11
1.1	Justificativa	12
1.2	Objetivos	13
1.3	Organização do Trabalho	13
2	Referencial Teórico	14
2.1	Rede Neural Convolutacional (CNN)	14
2.1.1	AlexNet	15
2.1.2	DenseNet	15
2.1.3	Grupo de Geometria Visual (VGG)	16
2.1.4	Google Inception	18
2.1.5	ResNet (ResidualNetwork)	18
2.2	Classificadores	19
2.2.1	<i>Support Vector Machines (SVM)</i>	20
2.2.2	<i>Random Forest</i>	20
2.2.3	<i>Multilayer Perceptron</i>	21
2.3	Métricas Avaliativas	22
2.3.1	Acurácia	22
2.3.2	<i>Kappa</i>	22
2.3.3	Sensibilidade	23
2.3.4	Especificidade	23
2.3.5	F-Score	23
3	Trabalhos Relacionados	24
4	Aplicação Desenvolvida	26
4.1	Metodologia	26
4.2	Pesquisa Bibliográfica	26
4.3	Levantamento de requisitos	27
4.3.1	Requisitos Funcionais	27
4.3.2	Requisitos Não Funcionais	27
4.4	Definição do Ambiente de Testes	28
4.5	Definição dos Modelos	28
4.6	Definição das Métricas	28
4.7	Definição dos Classificadores	28
4.8	Avaliação dos Modelos	29

4.9	Validação	29
4.10	Avaliação da Ferramenta	29
4.11	Análise dos Resultados	30
5	Resultados e Discussões	31
5.1	Tela <i>Convolution Neural Network</i>	31
5.1.1	<i>Setup</i>	32
5.1.2	<i>Architectures</i>	33
5.1.3	<i>Input e Output</i>	33
5.1.4	<i>Training and Test Setup</i>	34
5.1.5	Monitor	34
5.1.6	<i>Logs</i>	35
5.2	Tela <i>Classification</i>	36
5.2.1	<i>Datasets</i>	36
5.2.2	<i>Models</i>	37
5.2.3	<i>Input e Output</i>	37
5.2.4	<i>Training and Test Setup</i>	37
5.2.5	Results	38
5.2.6	<i>Logs</i>	39
5.3	Tela <i>Metrics and Charts</i>	39
5.3.1	<i>Saved Models</i>	40
5.3.2	<i>Metrics</i>	40
5.3.3	<i>Input e Output</i>	40
5.3.4	<i>Results</i>	41
5.4	Avaliação dos Usuários	42
6	Conclusão	46
	Referências	47

1 Introdução

A visão computacional é uma ciência que vem da possibilidade do computador enxergar o mundo real de tal forma aos dos seres humanos, apesar de se tratar de uma ciência considerada recente.

Em virtude da visão computacional, surgiram as Redes Neurais Convolucionais(CNN). Uma CNN é um tipo de rede neural artificial que é especialmente eficaz para tarefas de processamento de imagem. As CNNs são inspiradas no cérebro humano, que usa um processo chamado visão computacional para processar informações visuais.

Apesar do gargalo nos primeiros passos de inserção da visão computacional à realidade, em tempos atuais se trata de um mercado que cresce de maneira elevada. O mercado global de visão computacional foi avaliado em cerca de US\$ 11,3 bilhões e tende a atingir a taxa de crescimento anual composta (CAGR) de 7,3% no período de 2021 à 2028 ([GVR, 2021](#)), se espalhando cada vez mais nas mais diversas áreas.

Hoje, o processamento de imagens, apesar de ser fruto da visão computacional e se tratar de uma ciência recente, já possui uma literatura vasta e muitos algoritmos para uso nas mais diversas áreas de processamentos, como: medicina, engenharia, segurança, entre outras. Desta maneira, a análise e tomada de decisão de qual algoritmo utilizar para resolução de um problema específico é uma tarefa crucial que deve ser executada com cautela, visto que o desempenho final de um trabalho de processamento está diretamente ligado ao algoritmo que o executa.

Em virtude da expansão da área da visão computacional e a dificuldade de um usuário sem algum conhecimento prévio sobre redes neurais tomar a decisão de qual arquitetura utilizar, foi enxergada a possibilidade de uma ferramenta para auxiliar o pesquisador que se encontre com esta problemática, porém não restringindo somente a estes, usuários que possuem mais conhecimentos podem utilizar da ferramenta para agilizar sua tomada de decisão.

Neste contexto, este trabalho visa a construção de uma ferramenta que englobe os módulos essenciais na diretriz de decisão de algoritmo ideal para um projeto de visão computacional. A plataforma que permite conduzir o treinamento de *Convolutional Neural Networks* (CNNs) em conjuntos de dados personalizados, direcionados aos problemas em questão selecionados pelo usuário. Além dessa capacidade, ela também viabiliza a classificação de amostras de dados específicas, tais como o conjunto de validação. Por fim, a plataforma disponibiliza a apresentação dos resultados das métricas previamente escolhidas a partir do modelo gerado.

1.1 Justificativa

Atualmente existem diversas arquiteturas de CNN disponíveis na literatura. Deste modo, uma arquitetura pode ser mais adequado a determinado problema que outra, isso geralmente vai estar associado à complexidade do problema e à quantidade de imagens disponíveis (PASZKE et al., 2019), porém não é somente esse o ponto, redes com filtros menores são melhores para tarefas que envolvem detecção de bordas, enquanto com filtros maiores são melhores para tarefas de detecção de objetos.

As redes neurais podem ser usadas para classificação de imagens, sejam somente elas, com suas camadas de classificação ou usando as para extração das *features*. Nesse formato, são extraídas características pelos blocos da CNN e antes da classificação e são utilizados classificadores já consolidados na literatura como *Support Vector Machine*, *Multilayer Perceptron*(MLP) ou *Random Forest*.

Outra problemática bastante usual para processamento de imagens com CNNs é o ramo da saúde, que visa a detecção de doenças ou patologias em imagens. Para a classificação de imagens histopatológicas de câncer de mama, por exemplo, no trabalho (FILONENKO; KURNIANGGORO; JO, 2017) foi demonstrado que a Inception V1 (HE et al., 2015) obteve melhor acurácia em todos os cenários propostos, o que não aconteceu em (PARVIN; HASAN, 2020), onde a Xception(CHOLLET, 2017) obteve os melhores resultados neste quesito.

Avaliando este cenário, onde há uma grande variedade de CNNs disponíveis hodiernamente e a gama de possibilidades de utilização destas, a ferramenta proposta para comparação direta entre várias arquiteturas vai facilitar o processo de análise e comparação das redes neurais, desta maneira auxiliando na decisão que melhor satisfaça a problemática proposta pelo usuário.

A escolha de qual arquitetura utilizar afeta diretamente a qualidade da solução proposta, deste modo o software proposto irá fornecer de forma clara uma comparação entre os principais algoritmos do estado da arte, bem como métodos de comparação entre eles, por meio de métricas, seja acurácia, *kappa*, especificidade, sensibilidade ou *f-score*.

Além de garantir uma escolha da CNN baseada em dados matemáticos o software possibilitará que a tomada de decisão da arquitetura a ser utilizada seja mais breve e embasada matematicamente, deste modo acelerando a etapa de escolha da melhor arquitetura para produção do trabalho por parte do usuário da ferramenta, pois o usuário executará os testes de maneira ágil comparará diretamente as informações obtidas por cada CNN.

1.2 Objetivos

Este trabalho tem como objetivo geral desenvolver um software que possa auxiliar os pesquisadores na área de visão computacional no Treino e teste das arquiteturas de CNNs do estado da arte. Dessa forma, o software poderá executar treinamento de redes neurais convolucionais e classificadores, além de comparações com métricas de avaliação.

Tem-se como objetivos específicos:

1. Desenvolvimento de um software *desktop* com interface gráfica;
2. Execução automática de diferentes arquiteturas de CNN para um determinado problema de visão computacional;
3. Avaliar o desempenho individual das arquiteturas selecionadas;
4. Realizar a avaliação do software de forma a realizar melhoria de acordo com o *feedback* de profissionais da área de visão computacional.

1.3 Organização do Trabalho

Esta monografia está disposta da seguinte maneira: Capítulo 1, este apresenta a introdução do trabalho. No Capítulo 2, estão apresentados conceitos básicos para melhor compreensão do trabalho. O Capítulo 3, lista alguns trabalhos relacionados. O Capítulo 4 descreve a metodologia aplicada para o desenvolvimento da ferramenta. No Capítulo 5, são demonstrados resultados obtidos a partir da ferramenta desenvolvida. Por fim, o Capítulo 6 apresenta as principais conclusões obtidas e sugestões de trabalhos futuros.

2 Referencial Teórico

Esta seção apresenta os conceitos necessários para a compreensão deste trabalho. A primeira parte trata das CNNs. A segunda apresenta os classificadores. A terceira parte trata das métricas avaliativas que serão dispostas para as comparações.

2.1 Rede Neural Convolucional (CNN)

Uma CNN é um algoritmo de aprendizado de nível aprofundado que pode captar uma imagem de entrada e atribuir níveis de importância para cada aspecto ou objeto contidos na imagem, desta forma sendo capaz de detectar e diferenciar umas das outras.

O etapa de pré-processamento necessário para uma CNN é muito menor se comparado diretamente a outros algoritmos de classificação. Enquanto uma CNN tem a capacidade de aprender de forma independente os filtros, nos métodos clássicos estes filtros necessitam definidos manualmente, de modo que o sistema se torna mais custoso em relação ao processamento e ao desenvolvimento.

A arquitetura de uma CNN, demonstrada é apresentada na Figura 1, é análoga à estrutura do padrão de conectividade dos neurônios do cérebro humano e foi diretamente inspirada na distribuição do córtex visual.

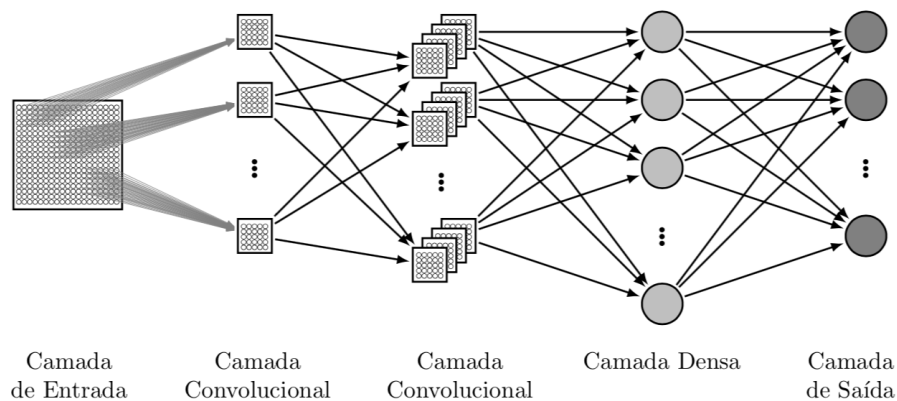


Figura 1 – Arquitetura de uma Rede Neural Convolucional.

Fonte: (SAKURAI, 2017)

As redes neurais podem resolver problemas diversos, como por exemplo: Diagnóstico médico: As redes neurais podem ser usadas para ajudar os médicos a diagnosticar doenças com mais precisão. O trabalho de (SILVER et al., 2016) lista diversos trabalhos empenhados nesta temática, onde redes são usadas para auxílio em diagnósticos de doenças diversas.

Prevedendo o comportamento do mercado financeiro: As redes neurais podem ser usadas para prever o comportamento do mercado financeiro. O Artigo produzido por (LECUN;

[BENGIO; HINTON, 2015](#)) discute o potencial das redes neurais para previsão do mercado financeiro, de modo que ajude investidores em tomadas de decisões.

Criando obras de arte: As redes neurais podem ser usadas para criar obras de arte. Por exemplo, uma rede neural foi desenvolvida que pode criar pinturas que são indistinguíveis das pinturas criadas por humanos, como exposto por ([GOODFELLOW et al., 2014](#)) em sua pesquisa.

O tema de CNN é bastante abrangente, e com isso diversas arquiteturas foram surgindo, variando desde o método de convolução, sendo simples ou paralela, até a quantidade de camadas de processamento convolucional. Dentre as várias arquiteturas temos: AlexNet, DenseNet, Grupo de Geometria Visual (VGG), Google Inception, e a ResNet, que são as redes disponíveis, todas estas são redes gratuitas, bem sucedidas e amplamente reconhecidas pela literatura.

2.1.1 AlexNet

O AlexNet([KRIZHEVSKY, 2014](#)) foi projetada por Alex Krizhevsky e ganhou destaque ao vencer o desafio *ImageNet Large Scale Visual Recognition Challenge*([RUSSAKOVSKY et al., 2014](#)). O artigo original descreveu que a profundidade do modelo foi crucial para alcançar um desempenho excepcional, embora isso exigisse um alto custo computacional. Com o uso de GPUs, esse problema pôde ser superado.

A arquitetura da AlexNet, como mostra a Figura 2 é composta por cinco camadas convolucionais, três camadas de *pooling* máximo, duas camadas de normalização, duas camadas totalmente conectadas e uma camada *softmax*. Cada camada convolucional aplica filtros convolucionais e utiliza a função de ativação *ReLU*. As camadas de *pooling* são responsáveis por realizar a operação de agrupamento máximo. O tamanho de entrada é fixo devido à presença das camadas totalmente conectadas. De maneira geral, a AlexNet possui cerca de 60 milhões de parâmetros.

Dessa forma, a AlexNet se destaca como uma arquitetura de referência na área de visão computacional, principalmente devido à sua profundidade considerável para a época em que foi desenvolvida e ao uso eficiente de GPUs para viabilizar o processamento computacionalmente intensivo necessário para obter altos resultados de desempenho.

2.1.2 DenseNet

DenseNets([HUANG et al., 2017](#)) são um tipo de CNN que utiliza conexões densas entre camadas. Isso significa que o resultado de um bloco de convolução é passado para todos os blocos de convolução subsequentes. Essa arquitetura se mostrou melhor na precisão das redes neurais em uma variedade de tarefas, como classificação de imagens, detecção de objetos e reconhecimento de fala. As DenseNets são muito eficientes em termos de

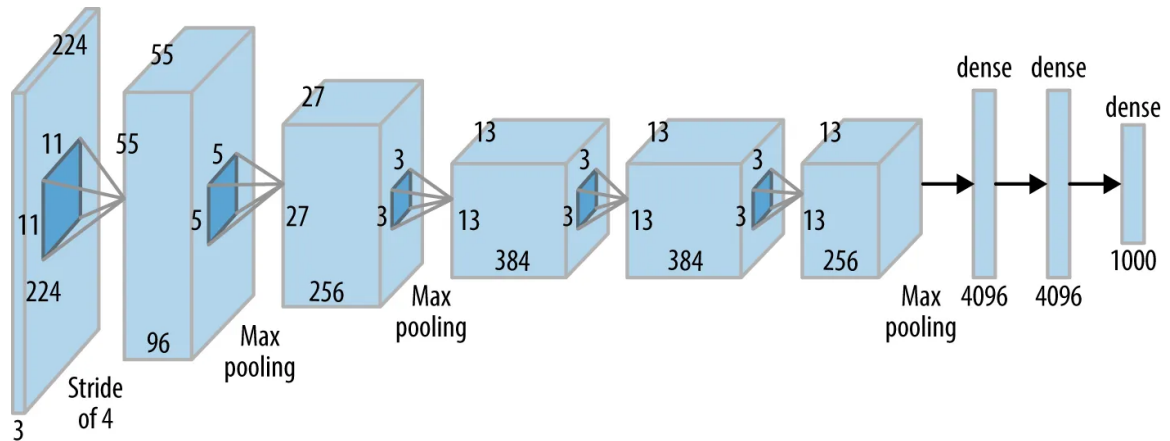


Figura 2 – Arquitetura AlexNet.
Fonte: (VARSHNEY, 2020)

computação e podem ser treinadas rapidamente, o que as torna uma opção atraente para o aprendizado de máquina.

A DenseNet é uma rede neural convolucional composta por uma sequência de blocos densos (Figura 3). Cada bloco denso é composto por uma sequência de camadas convolucionais, cada uma das quais é conectada a todas as camadas anteriores. Isso significa que cada bloco de convolução da DenseNet recebe entradas de todos os blocos anteriores. A saída da última camada da DenseNet é um mapa de recursos que representa os dados de entrada.

A DenseNet tem sido mostrado para melhorar a precisão em uma variedade de tarefas, como classificação de imagens, detecção de objetos e reconhecimento de fala. Ela também é muito eficiente em termos de computação, o que a torna uma opção atraente para o aprendizado de máquina.

2.1.3 Grupo de Geometria Visual (VGG)

Se trata de uma arquitetura de CNN profunda com padrão de várias camadas. O termo profundo se refere à quantidade de camadas que a CNN possui. Com a VGG-16, por exemplo, existem 13 camadas convolucionais e 3 camadas totalmente conectadas, enquanto a VGG-19 possui 16 camadas convolucionais e 3 camadas densas. O modelo VGG-16 trouxe a concepção de agrupar as camadas de convolução em blocos, tal estratégia se tornou um padrão amplamente adotado por modelos mais contemporâneos.

O modelo VGG-16 (SIMONYAN; ZISSERMAN, 2014) atinge quase 92,7% de precisão no conjunto de dados ImageNet (DENG et al., 2009), que se trata de um banco de imagens composto por mais de 14 milhões de imagens pertencente a quase 1000 classes distintas. A Figura 4 demonstra o mapa da arquitetura do VGG-16.

O conceito de modelo do VGG-19 é o mesmo do VGG-16, porém este suporta 19 camadas convolucionais, distribuídas em 16 camadas convolucionais com 3 camadas to-

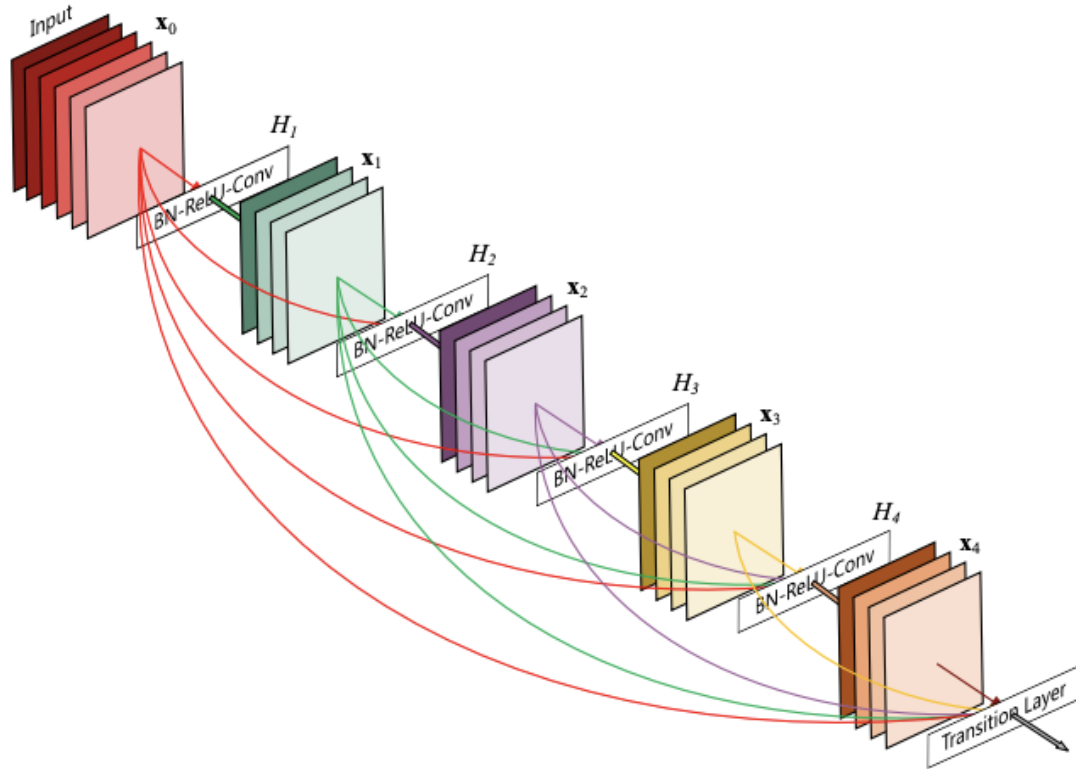


Figura 3 – Arquitetura DenseNet.
 Fonte: (HUANG et al., 2017)

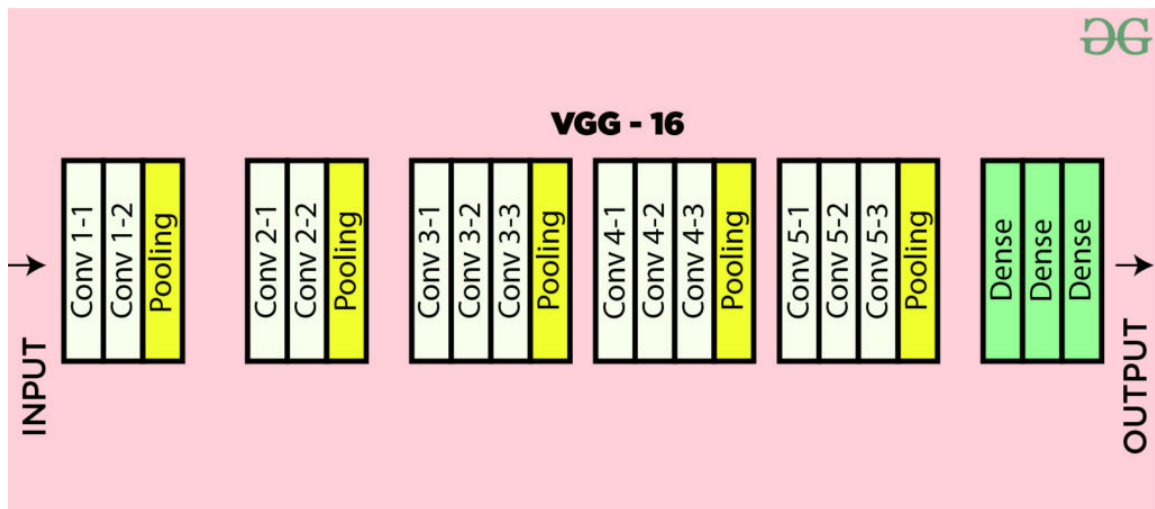


Figura 4 – Arquitetura da VGG-16.
 Fonte: (PAWANGFG, 2020)

talmente conectadas como mostra o mapa de sua arquitetura na Figura 5, ou seja, são 3 camadas de peso no modelo a mais que o seu precedente.

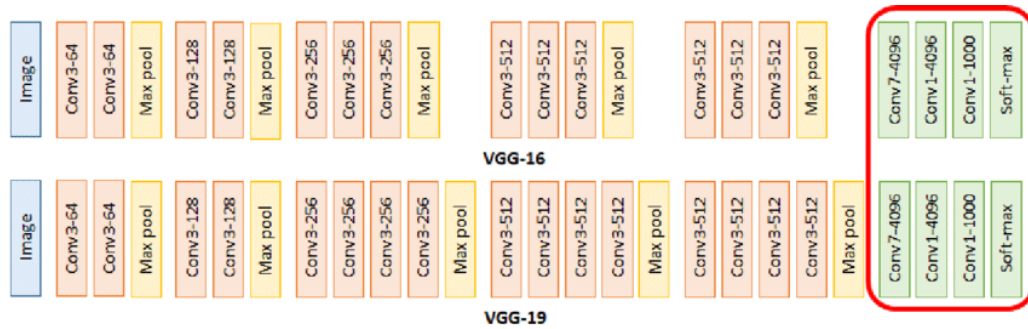


Figura 5 – Arquitetura da VGG-16 comparada a VGG-19.
 Fonte: (SAUMYATRIPATHI, 2020)

2.1.4 Google Inception

A Inception (SZEGEDY et al., 2014) é uma rede neural desenvolvida para auxiliar na análise de imagens e detecção de objetos, teve seu início como um módulo para o Googlenet. A Inception V1 foi premiada vencedora no ano de 2014 do ILSVRC (*Imagenet Large Scale Visual Recognition Challenge*), que se trata de um desafio de classificação de imagens em larga escala.

A arquitetura Inception tem como inovação chave a utilização dos múltiplos filtros 1x1, 3x3 e 5x5, bem como uma camada de *pooling* em paralelo. Esses diferentes filtros são concatenados em um único tensor, permitindo então que a rede capture informações detalhadas e abstratas de maneira eficiente. Outro ponto da rede é que com a ideia de usar camadas 1x1 como *bottlenecks*, basicamente esta característica possibilita a extração máxima do desempenho do computador, desta forma proporcionando uma redução nos custos computacionais da CNN.

Os módulos Inception são incorporados às redes neurais convolucionais como forma de reduzir o custo computacional. Direciona como uma CNN lida com uma vasta gama de imagens, com grande variação no conteúdo de destaque. Simplificadamente, como demonstrado na Figura 6, o módulo Inception funciona realizando uma convolução de uma entrada com três tamanhos diferentes (1x1, 3x3 e 5x5).

2.1.5 ResNet (ResidualNetwork)

A ResNet (HE et al., 2015) é uma estrutura de Rede Neural Convolutiva da Microsoft Research Asia que ganhou o primeiro lugar na ILSVRC-2015, ao atingir a taxa de erro top 5 de apenas 3,57% no banco de dados ImageNet.

Em casos de redes neurais profundas o treinamento se torna complexo devido ao problema de perda do gradiente, ou seja, quando empilhadas várias camadas em uma rede neural na teoria faria o erro de treinamento diminuir, porém na prática (ao tornar a rede mais profunda) o inverso acontece. A ResNet utiliza dos blocos definidos como residuais, são considerados blocos de construção do ResNet. No bloco residual, para a entrada x

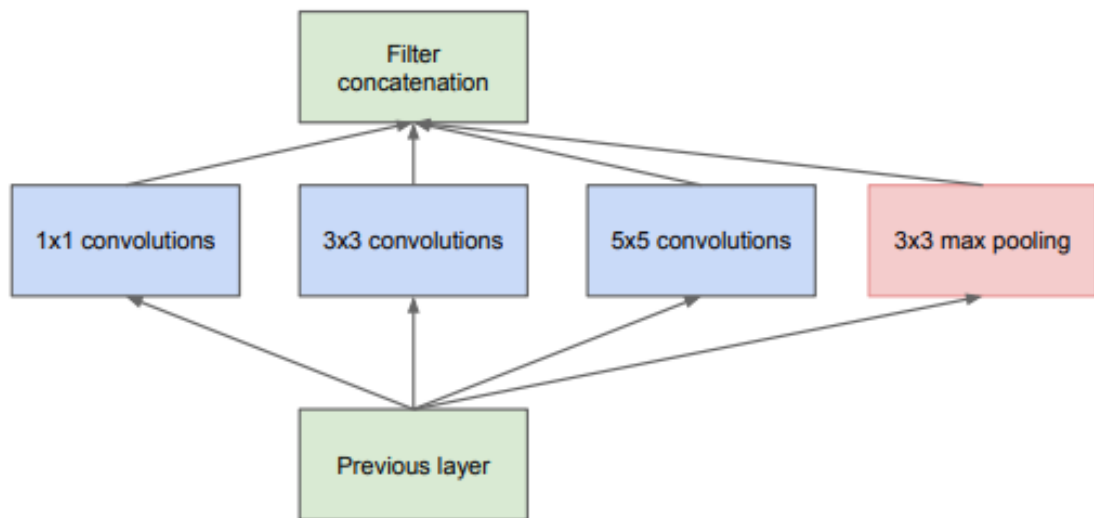


Figura 6 – Diagrama de funcionamento Inception.
 Fonte: (SZEGEDY et al., 2014)

é adicionado diretamente à saída (Figura 7) da rede, ou seja, $F(x) + x$, este caminho é conhecido como atalho.

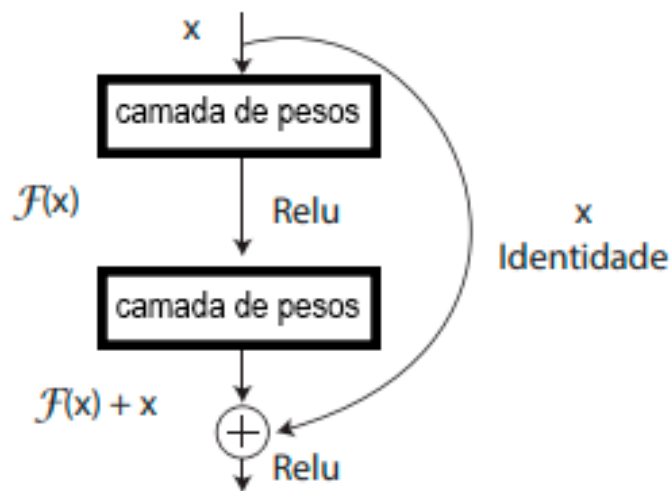


Figura 7 – Diagrama do atalho ResNet.

2.2 Classificadores

Um classificador possui um papel essencial para análise e categorização de objetos ou eventos desconhecidos com base em um conjunto de treinamento prévio. Ele utiliza um conjunto de dados conhecidos, chamado de conjunto de treinamento, composto por instâncias de cada uma das classes existentes. Com base nesse conjunto de treinamento, o classificador é capaz de aprender padrões e características distintivas de cada classe, a fim de determinar a classe provável de uma nova instância.

O classificador faz a classificação automática usando algoritmos e técnicas específicas. Ele extrai características pertinentes, como atributos numéricos ou descritivos, dos dados de treinamento e usa esses dados para criar um modelo capaz de distinguir entre as diferentes classes. Após a aplicação desse modelo em dados desconhecidos, o classificador pode usar os dados coletados durante o treinamento para determinar a classe mais provável para cada nova instância. Como reconhecimento de padrões, processamento de imagens, análise de texto e uma variedade de outros campos onde a classificação de dados é necessária, os classificadores são amplamente utilizados.

As redes neurais já possuem seus classificadores, porém este trabalho visa a utilização de classificadores já consolidados na literatura, para que desta forma as redes neurais sejam analisadas de maneiras equânimes, pois os classificadores serão os mesmos para quantas redes o usuário necessite testar.

2.2.1 *Support Vector Machines (SVM)*

O SVM(CORTES; VAPNIK, 1995) é um modelo de aprendizado de máquina supervisionado linear que usa algoritmos de classificação para problemas onde busca uma linha de separação entre duas classes distintas como demonstra a Figura 8. Este modelo pode ser usado para desafios de regressão ou propriamente classificação.

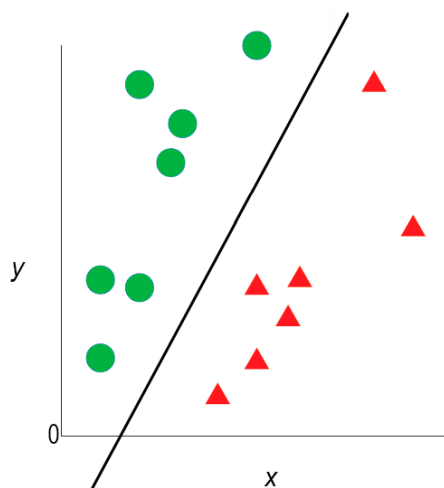


Figura 8 – Exemplo divisão de grupos SVM.

2.2.2 *Random Forest*

O *Random Forest*(BREIMAN, 2001) é um algoritmo de aprendizado de máquina supervisionado amplamente utilizado em problemas de classificação e regressão. Ele constrói árvores de decisão em diferentes amostras e leva a maioria dos votos para classificação e média em caso de regressão.

Como já mencionado o funcionamento do *Random Forest* se dá por meio da criação de várias árvores de decisão, e utilizando de algoritmos de entropia por exemplo é feita a seleção de cada nó até a obtenção do valor ao fim daquela árvore para junção ao fim com demais árvores como mostra a Figura 9.

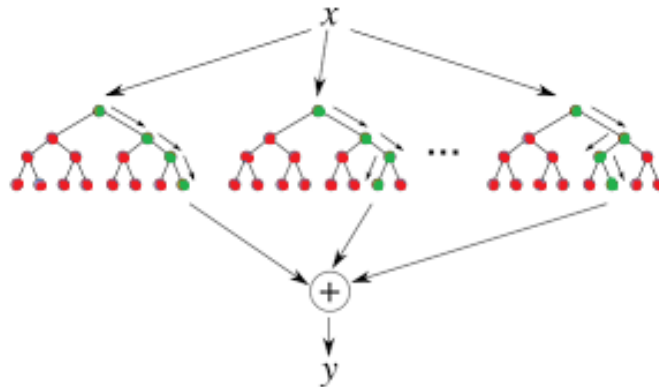


Figura 9 – Funcionamento do *random forest*.

2.2.3 Multilayer Perceptron

O MLP (HAYKIN, 1994), é considerado um incremento à rede neural de alimentação direta. Ele consiste em três tipos de camadas - camada de entrada, camada de saída e camada oculta (intermediária), como demonstrado na Figura 10.

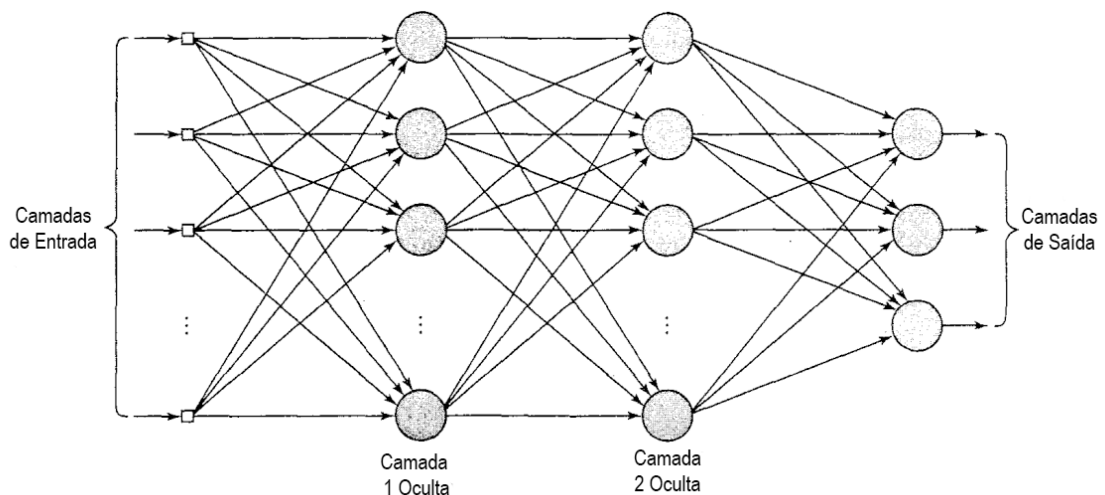


Figura 10 – Funcionamento do MLP.
Fonte: (BREIMAN, 2001)

A camada de entrada é responsável por receber o sinal a ser processado e sua classificação é executada pela camada de saída. Um número aleatório de camadas ocultas são adicionadas entre as camadas de entrada e saída. A camada oculta recebe este nome pois não é possível prever a saída desejada desta(s) camada(s).

2.3 Métricas Avaliativas

As métricas avaliativas desempenham um papel fundamental no processo de seleção da melhor configuração de redes neurais convolucionais para classificação de imagens, conforme proposto neste estudo de monografia. Essas métricas possuem como objetivo possibilitarem uma avaliação de maneira objetiva e matematicamente fundamentada das redes convolucionais, permitindo ao pesquisador comparar e identificar a configuração mais adequada para o problema em questão.

Ao final de cada processamento das redes neurais selecionadas pelo usuário, as métricas avaliativas desempenharão um papel crucial ao fornecer medidas de desempenho e permitir a comparação entre os modelos classificadores propostos. Essas métricas, de natureza quantitativa, serão essenciais para avaliar a capacidade de cada modelo em classificar corretamente os resultados e para identificar diferenças significativas de desempenho entre eles.

Cada métrica tem seus próprios pontos fortes e fracos, e a melhor métrica para usar depende do problema específico que você está tentando resolver, neste contexto a ferramenta dispõe de: Acurácia, *Kappa*, Sensibilidade, Especificidade e F-Score, melhores detalhadas na sequência.

2.3.1 Acurácia

A acurácia é a aproximação de um resultado com seu devido valor de referência real. Deste modo, quanto maior o nível da acurácia, mais próximo da referência ou valor real é o resultado localizado. No termo CNN quanto maior a acurácia encontrada mais próximo da realidade são os valores obtidos de forma automatizada.

A Equação 2.1 demonstra a forma matemática para cálculo da acurácia.

$$\frac{VP + VN}{VP + VN + FP + FN}, \quad (2.1)$$

onde, VP é Verdadeiro Positivo, FN é Falso Negativo, VN é Verdadeiro Negativo e FP é Falso Positivo.

2.3.2 *Kappa*

O coeficiente de concordância de *kappa* é utilizado para descrever a concordância entre dois ou mais meios de processamento quando realizam uma avaliação nominal ou ordinal de uma mesma amostra. O coeficiente *kappa* é calculado pela Equação 2.2

$$\frac{P0 - Pe}{1 - Pe}. \quad (2.2)$$

onde, P0: concordância relativa observada pelos avaliadores e Pe: Probabilidade hipotética de concordância

2.3.3 Sensibilidade

A sensibilidade é uma medida estatística que avalia a capacidade de um teste ou diagnóstico em identificar corretamente os casos positivos em uma população. Ela representa a porcentagem de resultados positivos que realmente correspondem a casos positivos na realidade. Em outras palavras, a sensibilidade indica a taxa de acerto dos casos positivos pelo teste em relação à condição ou característica em questão. A sensibilidade é calculada por meio de uma fórmula estatística, como a Equação 2.3, que leva em consideração o número de verdadeiros positivos e a soma dos verdadeiros positivos com os falsos negativos.

$$\frac{VP}{VP + FN} \quad (2.3)$$

2.3.4 Especificidade

A especificidade é uma medida estatística que avalia a capacidade de um teste ou diagnóstico em identificar corretamente os casos negativos em uma população. Em outras palavras, ela representa a porcentagem de resultados negativos que realmente correspondem a casos negativos na realidade. É uma medida importante, pois indica a precisão do teste em descartar corretamente os casos que não possuem a condição ou característica em questão. A especificidade é calculada por meio de uma fórmula estatística, conhecida como Equação 2.4, que leva em consideração o número de verdadeiros negativos e a soma dos verdadeiros negativos com os falsos positivos.

$$\frac{VN}{VN + FP} \quad (2.4)$$

2.3.5 F-Score

O *F-Score*, ou *F-Measure*, se trata de uma medida de precisão de um teste. Este é calculado diante da precisão e *recall* do teste, de modo que a precisão é o número de resultados positivos, adicionando aqueles que não foram identificados corretamente, e o *recall* é a quantidade de resultados positivos verdadeiros dividido pelo número de todas as amostras que deveriam ter sido detectadas como positivas. Seu cálculo se dá pela Equação 2.5.

$$\frac{VP}{VP + \frac{1}{2}(FP + FN)} \quad (2.5)$$

3 Trabalhos Relacionados

Nesta seção, serão apresentados outros trabalhos relacionados ao tema em discussão. A Tabela 1 agrupa esses trabalhos com base em critérios específicos, como o uso de técnicas de *Deep Learning*, *Machine Learning* e a comparação dos métodos utilizados. Essa abordagem de agrupamento permitirá uma visão abrangente e organizada dos estudos relevantes, facilitando a compreensão das contribuições e resultados obtidos por cada trabalho.

Tabela 1 – Trabalhos Relacionados.

Trabalho	<i>Deep Learning</i>	<i>Machine Learning</i>	Comparação de métodos
WEKA (LANG et al., 2019)	Sim	Sim	Sim
Orange (DEMsAR et al., 2013)	Não	Sim	Sim
KNIME (BERTHOLD et al., 2007)	Não	Sim	Sim
Apache Mahout (VOHRA, 2016)	Não	Sim	Não
Rattle GUI (WILLIAMS, 2009)	Não	Sim	Sim
Tanagra (RAKOTOMALALA, 2005)	Não	Sim	Sim
Este Trabalho	Sim	Sim	Sim

O WEKA (LANG et al., 2019) é um software voltado para a mineração de dados e aprendizado de máquinas de modo gráfico, o software possui limitações para o uso por pessoas que não possuam conhecimento em mineração de dados, deste moto restringindo o seu grupo de uso.

A ferramenta Orange (DEMsAR et al., 2013) se trata de mais um software para mineração e análise de dados, uma ferramenta de fácil utilização para algumas atividades, porém em determinados casos requer conhecimentos mais aprofundados para melhor desempenho da ferramenta, outro ponto é sua instalação que em alguns sistemas operacionais é um pouco complexa devido suas dependências.

O software KNIME (BERTHOLD et al., 2007) também é uma aplicação para análise, mineração e processamento de dados, possui interface gráfica, um ponto a ser mencionado é a curva de aprendizado para o uso da ferramenta que, caso o usuário não tenha conhecimentos prévios em ferramentas semelhantes pode vir a ser um pouco alongada.

Apache Mahout (VOHRA, 2016) é uma biblioteca focada em algoritmos de aprendizado de máquinas, deste modo sem foco direto em mineração e análise de dados, sem fornecer função direta de comparação de modelos, claro que ainda há a possibilidade da comparação manual.

Rattle GUI (WILLIAMS, 2009) é uma ferramenta desenvolvida em R para análise de dados e estatísticas, também de fácil utilização, devido a sua linguagem o mesmo não é tão rápido quanto outras ferramentas bem como não é, também, tão flexível.

O Tanagra (RAKOTOMALALA, 2005) é uma ferramenta de código aberto para análise de dados e mineração de dados. Ele é desenvolvido em Java e foi projetado para ser uma plataforma acessível e amigável. Uma limitação é que ela é uma ferramenta de análise descritiva, o que significa que pode ser usada para descrever dados, mas não para prever o futuro. Para isso, é necessário usar uma ferramenta de análise preditiva.

O critério *Deep Learning* é uma característica importante abordada neste trabalho, que se trata da tecnologia utilizada pelos métodos de processamentos para classificação de imagens dispostos por este trabalho. Apenas o trabalho WEKA(LANG et al., 2019) possibilita, a partir de uma extensão, o uso desta tecnologia.

O aspecto *Machine Learning*, é referente a tecnologia de processamento utilizada pelos métodos abordados nos trabalhos (DEMsAR et al., 2013), (DEMsAR et al., 2013), (BERTHOLD et al., 2007), (VOHRA, 2016), (WILLIAMS, 2009), (RAKOTOMALALA, 2005), (CHEN; GUESTRIN, 2016). Apesar de não se tratar da mesma tecnologia de processamento, serve como base para comparação.

Este trabalho apresenta uma ferramenta para facilitar o processamento de dados por usuários leigos no ramo da visão computacional, de modo que o este não necessite conhecimentos específicos e profundos para comparar modelos de redes neurais convolucionais, um dos pilares da aplicação é facilitar a entrada de novos usuários à pesquisa de modo que ajude estes a decidir quais redes neurais utilizar em suas pesquisas e trabalhos, por meio da comparação entre as redes de acordo com as métricas disponibilizadas na ferramenta.

4 Aplicação Desenvolvida

Este capítulo aborda aspectos relevantes ao desenvolvimento das funcionalidades da ferramenta em questão pelas quais foram desenvolvidas no decorrer deste trabalho.

4.1 Metodologia

Nesta seção está descrito o planejamento para o desenvolvimento da ferramenta proposta. A metodologia usada por este trabalho está disposto em: pesquisa bibliográfica, levantamento de requisitos, definição dos modelos, definição das métricas, desenvolvimento do ambiente de testes, avaliação, validação, desenvolvimento do ambiente de resultados e análise dos resultados. A Figura 11 apresenta o diagrama de UML com a metodologia deste trabalho.

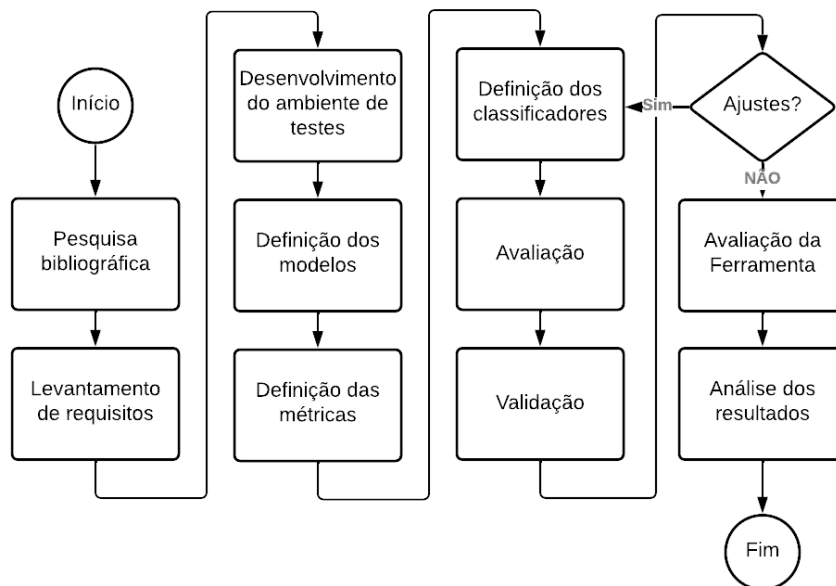


Figura 11 – Diagrama UML demonstrando a metodologia do trabalho proposto.

4.2 Pesquisa Bibliográfica

Esta etapa consiste no estudo da literatura para o tema do trabalho proposto com intuito de encontrar problemas e trabalhos relacionados. Além disso, será verificada a existência de trabalhos ou softwares com a mesma proposta, a possibilidade de melhorar a proposta já existente.

4.3 Levantamento de requisitos

Esta etapa é designada à compreensão dos comportamentos da ferramenta proposta, qual fluxo de dados e retorno de resultados. Como por exemplo, a aplicação proposta para este projeto necessita que exista a possibilidade de comparação de métodos *Deep Learning* para que ao fim o usuário escolha do melhor método para aquele caso.

4.3.1 Requisitos Funcionais

Os Requisitos Funcionais (RF) de um software descrevem as funcionalidades que o mesmo deve possuir, ou seja, diz a respeito do seu comportamento: a como ele deve reagir de acordo com aquela entrada, como irá se comportar em determinadas situações. A Tabela 2 apresenta os identificadores, descrições e dependências dos RF da ferramenta proposta.

Tabela 2 – Requisitos Funcionais.

Identificador	Descrição	Dependência
RF01: Inserção do <i>Dataset</i>	O usuário insere o seu banco de informações para que seja feito o processamento.	-
RF02: Seleção dos métodos CNN	O usuário selecionará as arquiteturas CNN a serem executadas para obtenção/classificação de dados e detecção de padrões	-
RF03: Seleção das Métricas	O usuário selecionará as métricas para o cálculo e validação dos resultados ao fim da execução das arquiteturas.	-
RF04: Exibição dos resultados	O software exibirá os resultados de forma iterativa e de fácil compreensão, bem como exibirá em forma de comparação os resultados, em gráficos e tabelas.	-

4.3.2 Requisitos Não Funcionais

Os Requisitos Não Funcionais (RNF) se referem a uma "não funcionalidade", ou seja, trata de algo que não é a funcionalidade mas que precisa ser realizado para que o software atenda seu propósito. Eles descrevem desde restrições, requisitos de interface entre outras características. A Tabela 3 apresenta as identificações, descrições e categorias dos RNF do software proposto.

Tabela 3 – Requisitos Não Funcionais.

Identificador	Descrição	Categoria
RNF01	O software deve possuir design intuitivo e de fácil utilização.	Usabilidade
RNF02	O software deve ser multiplataforma, sendo executável em Linux e Windows.	Compatibilidade
RNF03	O software deve ser desenvolvido em módulos de maneira que a alteração de algum componente tenha um impacto mínimo nos outros componentes.	Modularidade
RNF04	Os módulos devem seguir os mesmos padrões de desenvolvimento para melhorias e/ou incrementações de mais módulos.	Reusabilidade

4.4 Definição do Ambiente de Testes

Esta etapa é responsável pela aquisição de *dataset* para testes e configurações das arquiteturas que serão dispostas na ferramenta, de modo que possibilite a execução para os futuros testes e validação de resultados.

4.5 Definição dos Modelos

Para que seja possível a comparação entre modelos de CNN, se faz necessário o levantamento dos métodos que serão dispostos pela ferramenta para a comparação entre os tais, deve-se escolher estrategicamente os principais e mais abrangentes em tempos atuais.

4.6 Definição das Métricas

As métricas escolhidas são diretamente responsáveis pelo resultado do trabalho, pois se trata das informações que o modelo deve ser capaz de retornar a partir dos parâmetros propostos. No caso deste trabalho, as métricas serão: Acurácia, *Kappa*, Sensibilidade, Especificidade e *F-Score*.

4.7 Definição dos Classificadores

Etapa fundamental no projeto, que se refere à definição dos classificadores, estes serão responsáveis pela qualidade da classificação e obtenção de resultados, visto que as CNN serão treinadas a partir dos classificadores.

4.8 Avaliação dos Modelos

Depois que os modelos estiverem prontos, devem ser escolhidas as formas de avaliação dos resultados obtidos pelo trabalho, propondo mais de um método para uma melhor comparação, tal validação poderá ser feita mesmo com as métricas foram decididas na etapa Definição das Métricas.

4.9 Validação

Para a validação dos modelos propostos, a partir dos resultados expostos, estes devem ser comparados com os de um sistema real. O Ambiente de testes desenvolvido deve ser usado como referência de sistema real para o modelo. Se os modelos propostos estiverem muito distantes da realidade estes devem ser reajustados.

4.10 Avaliação da Ferramenta

Esta etapa trata da avaliação da usabilidade da ferramenta, para validar será utilizado o *System Usability Scale* (SUS) (BROOKE, 1996), de modo que são feitas as perguntas apresentadas na Tabela 4, com respostas variando de 1 (Discordo Completamente) até 5 (Concordo Fortemente).

Tabela 4 – Questionário proposto por (BROOKE, 1996) no SUS.

Questão	Pergunta
1	Eu acho que gostaria de usar esse sistema com frequência ?
2	Eu acho o sistema desnecessariamente complexo ?
3	Eu achei o sistema fácil de usar ?
4	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema ?
5	Eu acho que as várias funções do sistema estão muito bem integradas ?
6	Eu acho que o sistema apresenta muita inconsistência ?
7	Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente ?
8	Eu achei o sistema difícil de usar ?
9	Eu me senti confiante ao usar o sistema ?
10	Eu precisei aprender várias coisas novas antes de conseguir usar o sistema ?

Logo após o uso do sistema, o questionário é aplicado. Diante das respostas do questionário é necessário seguir 3 regras de avaliação definidas por (BROOKE, 1996), mostradas a seguir:

1. As respostas das perguntas ímpares (1,3,5,7 e 9), subtraia 1 da resposta selecionada;

2. As respostas das perguntas pares (2,4,6,8 e 10), subtraia de 5 o valor da resposta;
3. Por último, some todas as respostas e multiplique o resultado por 2,5.

O resultado obtido com o questionário varia entre 0 a 100. Com resultado quantitativo gerado, se faz necessária a conversão para qualitativo, seguindo este contexto (BROOKE, 1996) propôs a Tabela 5 para que se tornasse possível a realização da classificação do produto avaliado.

Tabela 5 – Sistema de Pontuação do SUS.

Pontuação Quantitativa	Resultado Qualitativo
0 até 25	Pior que o imaginável.
26 até 40	Fraco.
41 até 50	Mediano.
51 até 70	Bom.
71 até 85	Muito Bom.
86 até 100	Melhor que o imaginável.

4.11 Análise dos Resultados

É a última fase do projeto, nesta fase a ferramenta desenvolvida será testada e avaliada por meio do questionário anteriormente exposto, os resultados obtidos serão analisados e serão feitas as devidas conclusões sobre a ferramenta proposta. Os resultados devem ser apresentados por meio de gráficos e tabelas para melhor compreensão.

5 Resultados e Discussões

Esta seção tem como objetivo apresentar a ferramenta desenvolvida, juntamente com seus resultados de acordo com a avaliação SUS obtidas mediante pesquisa após o uso da ferramenta. A aplicação é composta por três telas principais e duas telas de configuração. A ferramenta inclui a Tela *Convolution Neural Network*, a Tela *Classification* e a Tela *Metrics and Charts*. Além disso, há telas de configuração para Novos Pesos (*New Weights*) e Ajuste Fino (*Fine-Tuning*).

5.1 Tela *Convolution Neural Network*

A Figura 12 apresenta a Tela *Convolution Neural Network*, que corresponde à tela principal da ferramenta desenvolvida. Essa tela carrega a aplicação diretamente, uma vez que é a partir dela que são obtidos os dados necessários para utilizar as demais funções da aplicação, esta dispõe de diversas seções para configuração geral antes do processamento.

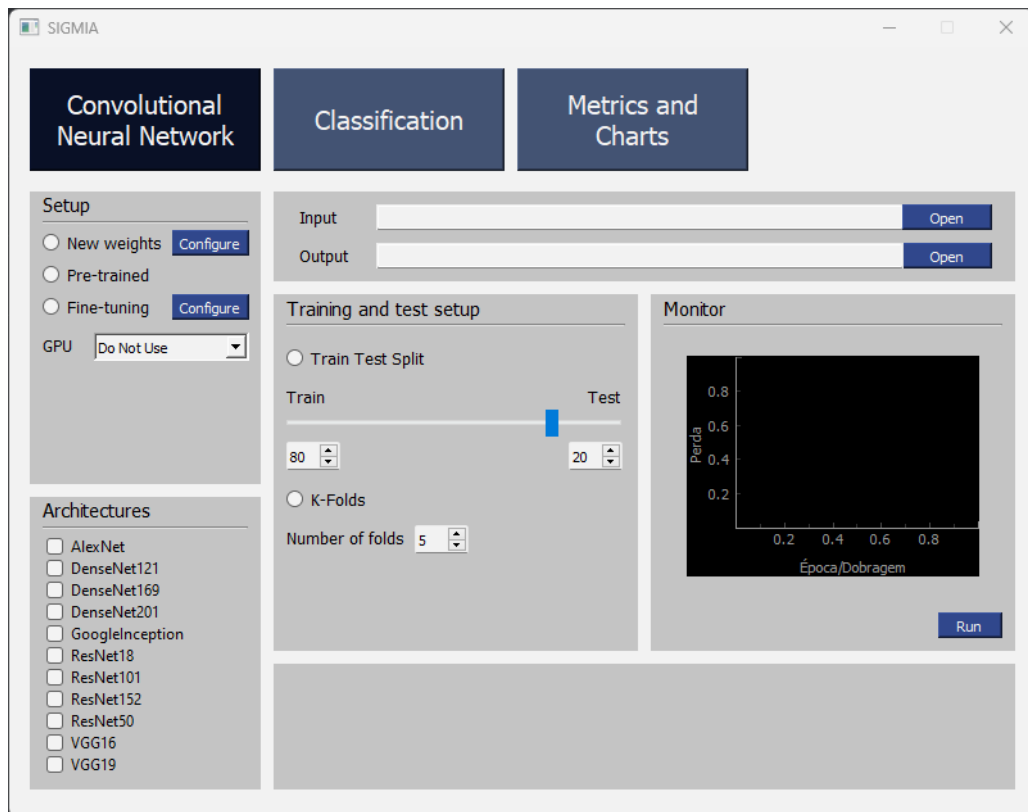


Figura 12 – Tela *Convolution Neural Network*.

5.1.1 Setup

O *Setup* (Figura 13) dispõe de 3 opções de configuração para rede neural e selecionar caso necessário a GPU para o processamento, acerca das configurações da CNN temos: *New-weights*, *Pre-trained* e *Fine-tuning*, se selecionado *New-weights* ou *Fine-tuning* precisa que seja feita a configuração conforme ilustrado nas Figuras 14a e 14b respectivamente, são semelhantes e requerem especificações adicionais. Ao selecionar uma dessas configurações de processamento, é necessário definir as seguintes configurações do compilador:

- *Loss functions*: determina qual função de perda o modelo deve executar, estão disponíveis na ferramenta: Entropia Cruzada (*Cross Entropy Loss*) e Perda de Entropia Cruzada Binária com *Logits* (*BCEWithLogitsLoss*).
- *Optimizer*: representa o otimizador usado pelo modelo, os otimizadores disponíveis para processamento são: Adam e SGD (Descida de Gradiente Estocástico).
- *Metrics*: é a métrica a ser usada pelo modelo, a princípio, as métricas disponíveis são: Acurácia e *Kappa*.

Além das configurações do compilador é preciso definir também as configurações de treinamento também:

- *Epochs*: representa a quantidade de épocas para o treinamento.
- *Batch Size*: é o tamanho do lote de imagens que será usado por iteração durante o treinamento do modelo.

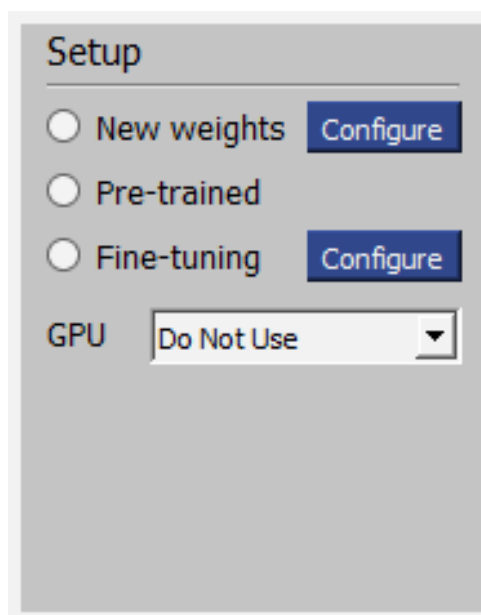
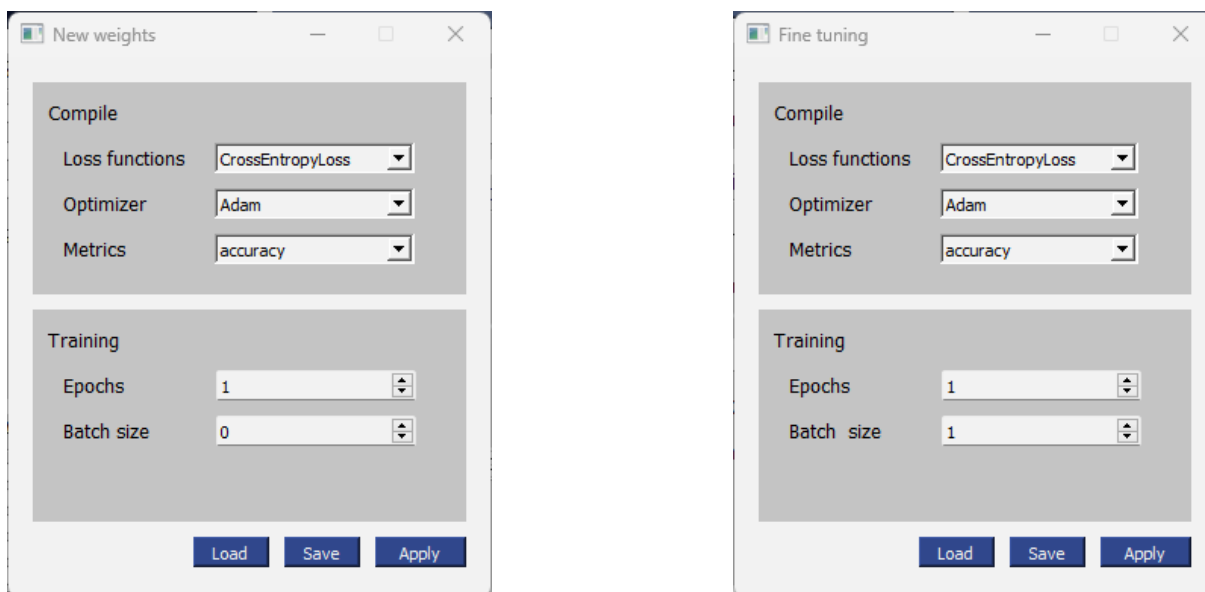


Figura 13 – Seção *Setup Convolution Neural Network*.

(a) Tela *New Weights*.(b) Tela *Fine Tuning*.Figura 14 – Configurações de *Setup CNN*.

Essas configurações permitem ajustar o modelo de acordo com as necessidades específicas do problema em questão, garantindo a melhor performance durante o treinamento.

5.1.2 Architectures

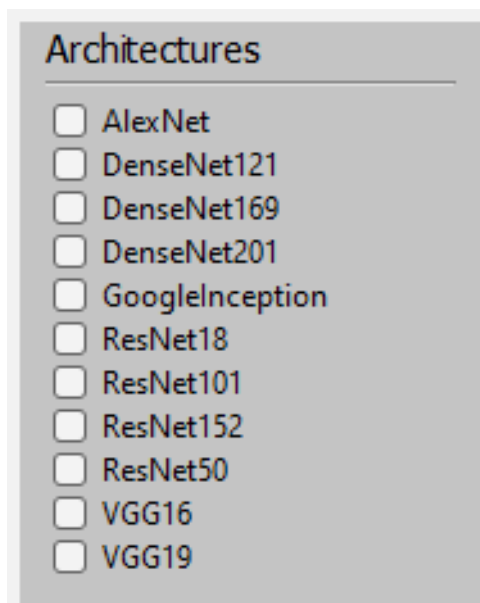
Esta é a parte responsável por seleção das CNNs que serão executadas, o critério para o funcionamento é a seleção de pelo menos uma arquitetura das disponibilizadas: AlexNet, DenseNet, Inception, ResNet e VGG, como visto na Figura 15. Estas são amplamente reconhecidas e comprovadas em diversos desafios de visão computacional. Suas contribuições inovadoras, como conexões residuais (ResNet), conectividade densa (DenseNet) e módulos Inception (Inception), tornam-nas escolhas sólidas para diferentes tarefas de aprendizado profundo. Essas redes têm demonstrado resultados de alta precisão e eficiência, podendo ser selecionadas N Redes Neurais.

5.1.3 Input e Output

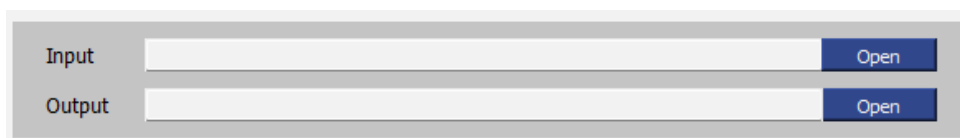
A seleção da pasta de entrada das imagens (*Input*), que estão organizadas em: dataset/classes/imagens, em resumo é uma pasta principal com subpastas que são as classes com suas imagens, a pasta de entrada deve ser *dataset*.

A saída (*Output*) será uma pasta com a nomenclatura das configurações definidas antes do processamento, dentro dessa pasta será salvo o modelo treinado, uma subpasta determinada de *train* que são as *features* obtidas do processamento das redes neurais, no caso denominadas de *features*.

Estas seleções de pastas serão possíveis mediante o clique no botão open referente a seleção que o usuário deseja, como visto na Figura 16, após o clique, abre-se o sistema de

Figura 15 – Seção *Architecture Convolution Neural Network*.

gerenciamento de arquivos do sistema operacional para que seja selecionado a pasta, tanto de entrada quanto saída, ao selecionar o caminho da pasta selecionada ficará exposto no formulário para facilitar a compreensão e haver uma validação pelo usuário que a pasta correta foi selecionada.

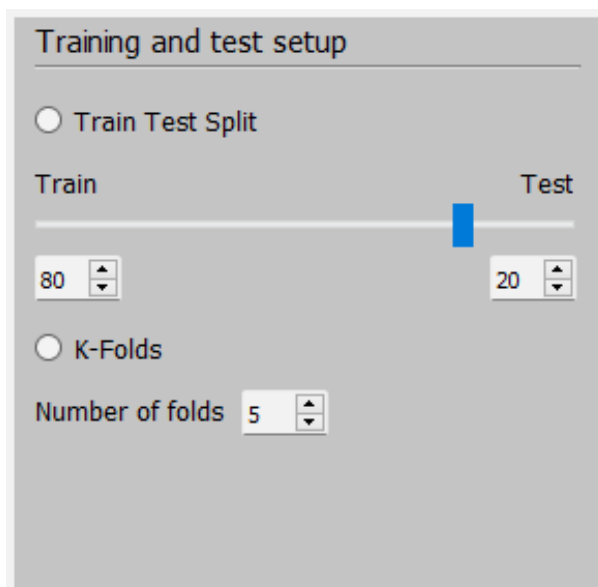
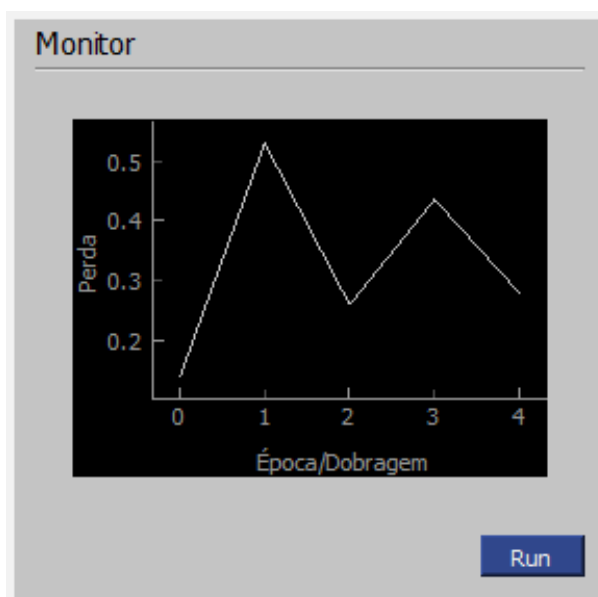
Figura 16 – Seção *Input and Output Convolution Neural Network*.

5.1.4 *Training and Test Setup*

Etapa responsável pelo ajuste do treinamento do modelo, onde será configurado a divisão do conjunto de dados. Existem duas opções disponíveis, possibilitando selecionar apenas uma destas, além disto há a possibilidade da customização da divisão, sejam as porcentagens de treino e teste ou o número de *folds* como demonstra a Figura 17.

5.1.5 *Monitor*

O Monitor (Figura 18) exibe o gráfico com o erro de acordo com cada época durante o treinamento onde pode-se acompanhar o desempenho do treinamento da CNN selecionada, deste modo fica compreensível se a rede está aprendendo ou não, bem como é possível notar que o algoritmo está em execução.

Figura 17 – Seção *Training and Test Setup Convolution Neural Network*.Figura 18 – Seção *Monitor Convolution Neural Network*.

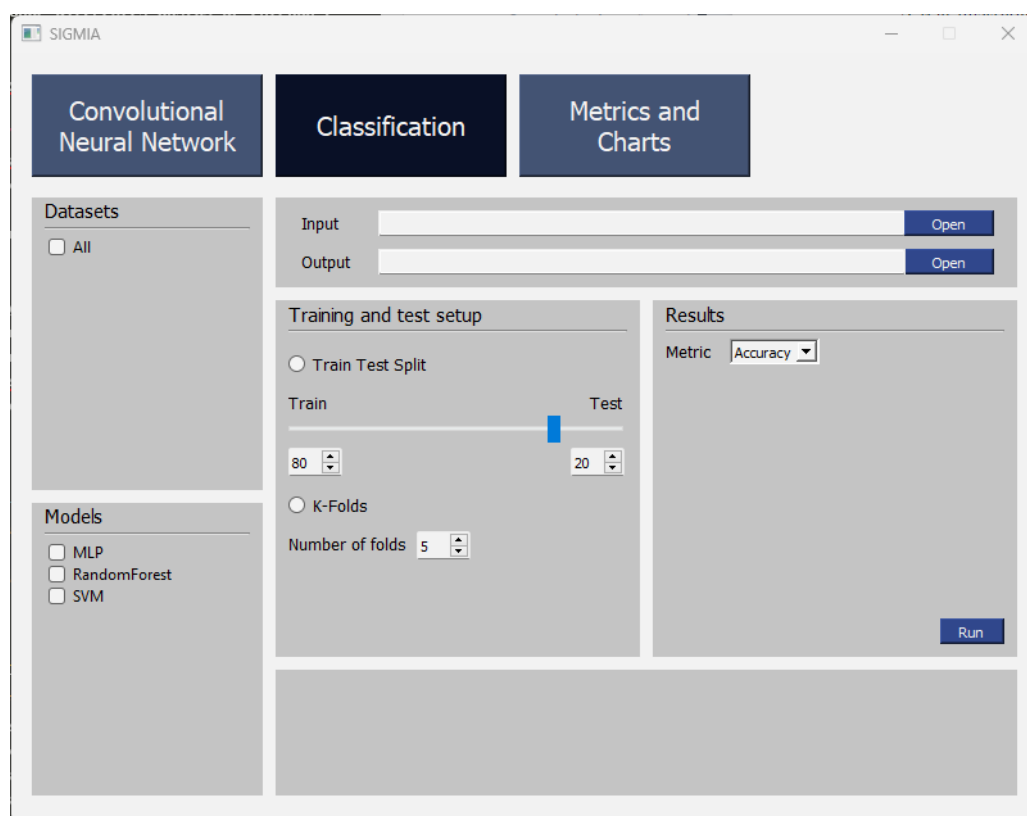
5.1.6 Logs

A parte designada à exibição de informações, exposto na Figura 19 que é denominada de seção de *Logs*, como tempo de processamento de cada época podem ser úteis ao usuário além da exibição de alguma mensagem de erro que for encontrada, esta parte da tela do sistema é focada a exibição de mensagens para facilitar o uso e compreender melhor como está a execução da ferramenta como um todo. A princípio essa seção vem vazia pois as informações irão sendo adicionadas mediante o decorrer do processamento pelo usuário, é uma partição de caráter informativa.

Figura 19 – Seção *Logs Convolution Neural Network*.

5.2 Tela *Classification*

A tela ilustrada na Figura 20, desempenha um papel crucial na configuração da classificação. Esta tela apresenta uma estrutura hierárquica, na qual a seleção do conjunto de dados obtido a partir da execução da rede neural só é possível se a pasta de origem tiver sido previamente especificada durante o processamento de extração das características pela CNN. Essa abordagem hierárquica visa fornecer maior controle e precisão na seleção do conjunto de dados para a classificação, garantindo assim uma configuração adequada e mais eficiente para o processo de análise.

Figura 20 – Tela *Classification*.

5.2.1 *Datasets*

A seção responsável pela seleção do *dataset* (Figura 21) a ser classificado mostrará somente as bases de dados após a seleção da pasta de entrada. Essa pasta deve conter a pasta

gerada pelo processamento da CNN, que é a pasta *Output* do processamento anterior. Todos os *datasets* aparecerão com o nome da pasta, possibilitando assim a identificação de qual e/ou quais o usuário se interessa em classificar.

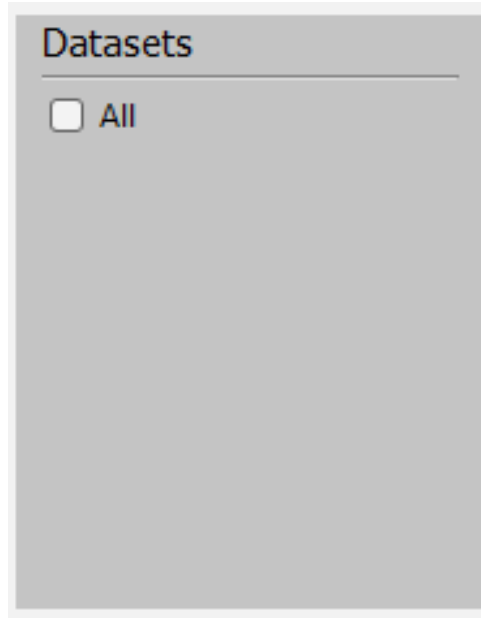


Figura 21 – Seção *Datasets Classification*.

5.2.2 *Models*

A partição responsável pela seleção do classificador (Figura 22) de interesse para testes e classificação das informações, permite que o usuário selecione um ou n classificadores para o processamento, a execução será de forma sequencial e os dados armazenados.

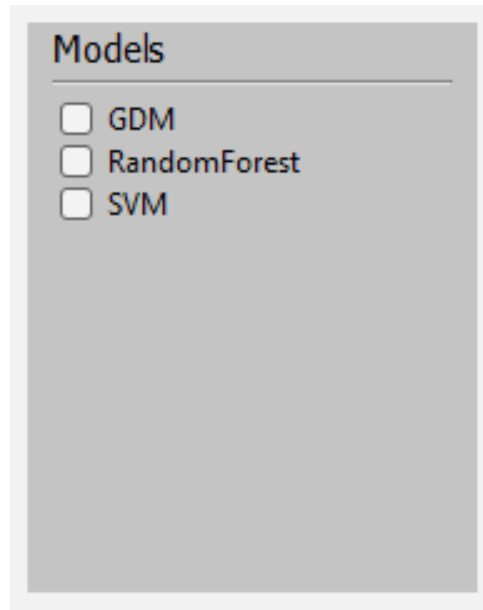
Os classificadores disponíveis: MLP, *Random Forest* e SVM. Oferecem uma combinação abrangente de capacidade de aprendizado não linear (MLP), resistência ao *overfitting* e conjuntos de dados grandes (*Random Forest*) e eficácia em espaços de alta dimensão (SVM), garantindo resultados sólidos em diversas situações.

5.2.3 *Input e Output*

Essa seção é idêntica a da tela CNN (Figura 16), permite que o usuário selecione as pastas de entrada e saída. Geralmente, a pasta de entrada é a mesma da saída, então a seleção de uma pasta de entrada carrega automaticamente a pasta de saída. No entanto, o usuário pode selecionar diferentes pastas de entrada e saída.

5.2.4 *Training and Test Setup*

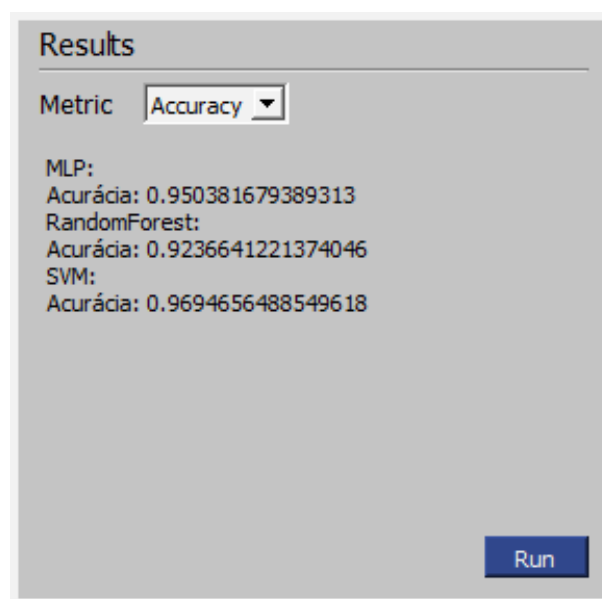
Esta seção também é idêntica a seção *Training and Test Setup*, demonstrada na Figura 17, da tela CNN, porém esta etapa visa a divisão do *dataset* para classificação, ou seja

Figura 22 – Seção *Models Classification*.

a divisão das *features* obtidas pela CNN, deste modo é feita a configuração também por Treino e Teste ou *K-Fold*, podendo ser personalizada a configuração independentemente do método escolhido.

5.2.5 Results

A partição *Results*(Figura 23) é bem semelhante à Monitor, porém esta esta exibe as informações em forma de mensagens em lista e não gráfica como o mencionado, outra diferença é que *Results* a seleção de exibição dos resultados, se serão exibidos os resultados da Acurácia ou do *kappa* de cada processamento.

Figura 23 – Seção *Results Classification*.

5.2.6 Logs

Também faz-se necessária a parte designada a *logs* dos classificadores (Figura 24) e informações ao sistema, para facilitar a usabilidade do usuário, caso o mesmo esqueça de selecionar algo essencial para o funcionamento uma mensagem de alerta será exposta para que o mesmo corrija e então consiga continuar com seu trabalho e consiga obter os seus resultados desejados.



Figura 24 – Seção *Logs* Classificação.

5.3 Tela *Metrics and Charts*

A tela de métricas e resultados (Figura 25) exibe os resultados de cada processamento. O usuário pode selecionar o(s) *dataset(s)* que deseja visualizar e as métricas que deseja exibir para cada um deles. A partir dessa tela, o usuário pode analisar os resultados e tirar conclusões sobre qual modelo é mais adequado para a problemática em questão.

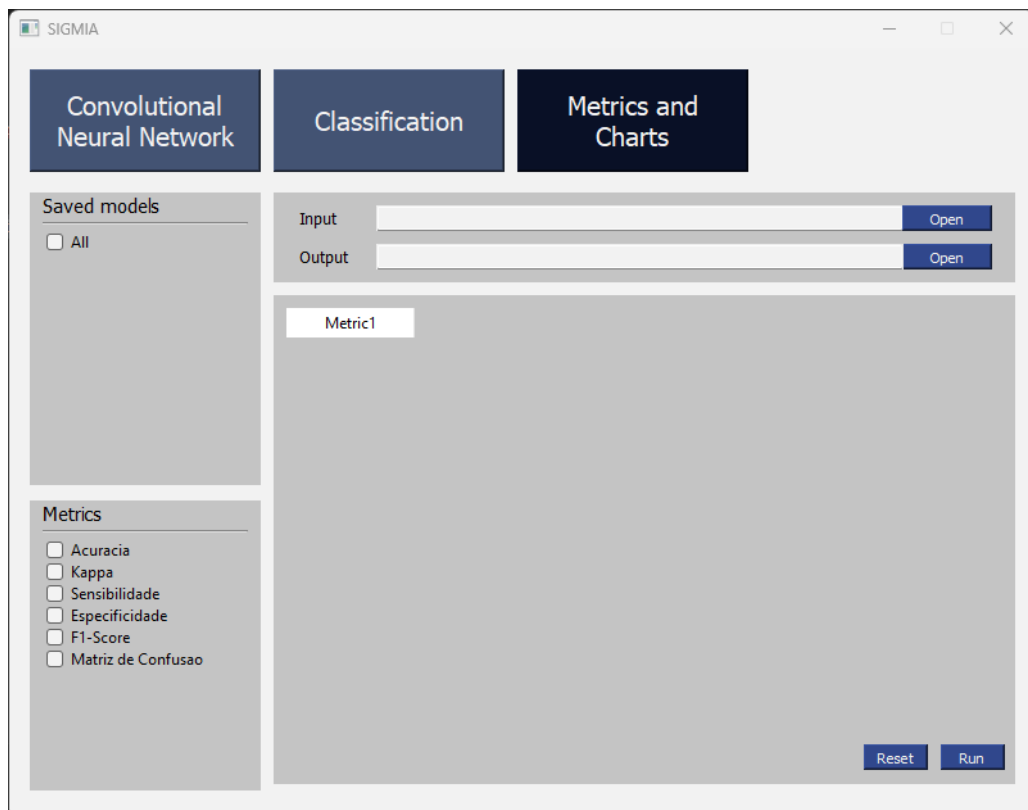


Figura 25 – Tela *Metrics and Charts*.

5.3.1 *Saved Models*

Esta seção, demonstrada na Figura 26 é preenchida com os modelos que foram processados e retornaram os resultados na etapa anterior (processamento dos classificadores). Desta forma, a seção em questão é diretamente dependente da seção Seleção de Entrada e Saída. Após o carregamento da entrada, o algoritmo exibe automaticamente os resultados que estão aptos a exibir os resultados.



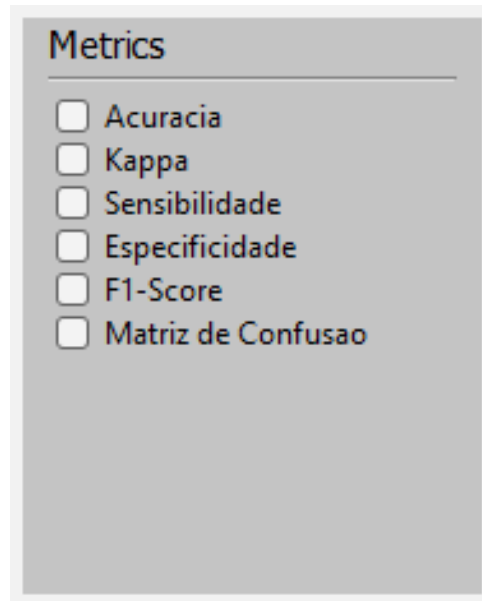
Figura 26 – Seção Modelos Salvos *Metrics and Charts*.

5.3.2 *Metrics*

Esta seção (Figura 27) disponibiliza as métricas: Acurácia, *Kappa*, Sensibilidade, Especificidade, *F-Score* e Matriz de Confusão avaliam diferentes aspectos do desempenho de um modelo de classificação. Acurácia é a taxa geral de acertos, enquanto *kappa* considera o acordo além do acaso. Sensibilidade e Especificidade avaliam a capacidade do modelo de detectar corretamente as classes positivas e negativas. O *F-Score* é uma medida balanceada entre precisão e *recall*. A Matriz de Confusão fornece uma visão detalhada das classificações corretas e incorretas. Estas serão exibidas por meio de tabela, em exceção da matriz de confusão, pois será exibido a foto da matriz de confusão.

5.3.3 *Input e Output*

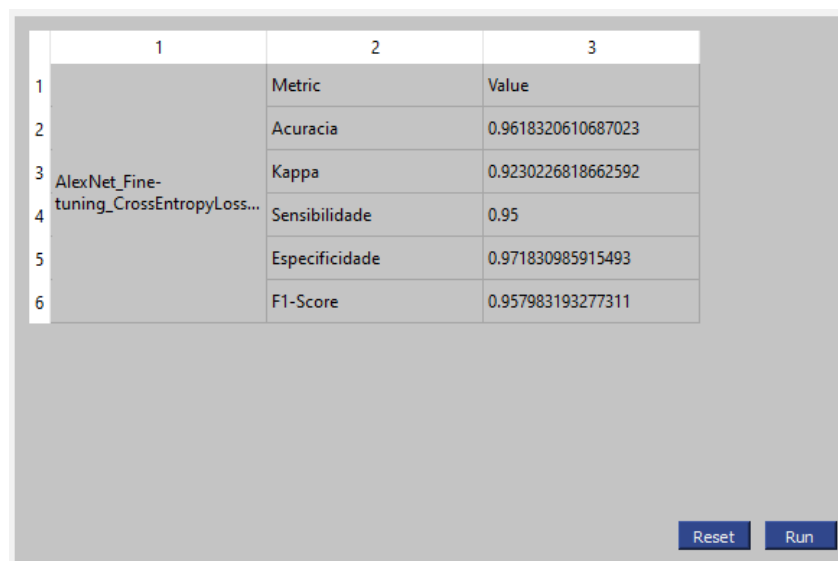
Esta partição é idêntica às telas CNN e Classificador, citadas nos tópicos 5.1.3 e 5.2.3 respectivamente, demonstrado pela Figura 16. É responsável pela seleção de onde estão os resultados, as pastas com resultados obtidos anteriormente no processamento dos classificadores. Desta forma, a entrada deste processo será a saída do processo anterior. A

Figura 27 – Seção Métricas *Metrics and Charts*.

saída pode ser tanto a mesma entrada como uma pasta terceira, esta será onde será salva a tabela com os resultados selecionados e os gráficos que foram escolhidos para exibição.

5.3.4 Results

Esta é a parte principal da tela *Metrics and Charts*, exposta na Figura 28, pois é a partição responsável por expôr os resultados dos resultados obtidos nos processamentos anteriores e selecionados a serem exibidos. Somente as tabelas serão exibidas nessa etapa, e os gráficos serão exibidos de maneira avulsa, para melhor compreensão.



	1	2	3
1		Metric	Value
2		Acuracia	0.9618320610687023
3	AlexNet_Fine-tuning_CrossEntropyLoss...	Kappa	0.9230226818662592
4		Sensibilidade	0.95
5		Especificidade	0.971830985915493
6		F1-Score	0.957983193277311

Figura 28 – Seção Resultados *Metrics and Charts*.

5.4 Avaliação dos Usuários

Mediante a aplicação desenvolvida aplicou-se o formulário SUS (BROOKE, 1996), este questionário permite a avaliação qualitativa da ferramenta. Foi disponibilizado o questionário para alunos do Bacharelado em Sistemas de Informação e também de outros cursos, como Análise e Desenvolvimento de Sistemas e Curso Integrado ao Médio em Informática, com foco em informática. Todos que responderam conheciam a visão computacional, porém em 33,7% atestaram apenas ter ouvido falar sobre o tema, outros 26,7% possuíam conhecimento básico, que pode-se traduzir em já ter estudado algo sobre, porém sem prática, em contrapartida 20% afirmaram possuir conhecimentos intermediários e os outros 20% conhecimento avançado, estes podem ser compreendidos como quem já processou algo, ou quem mesmo trabalha/pesquisa no ramo, respectivamente. como demonstrado na Figura 29.

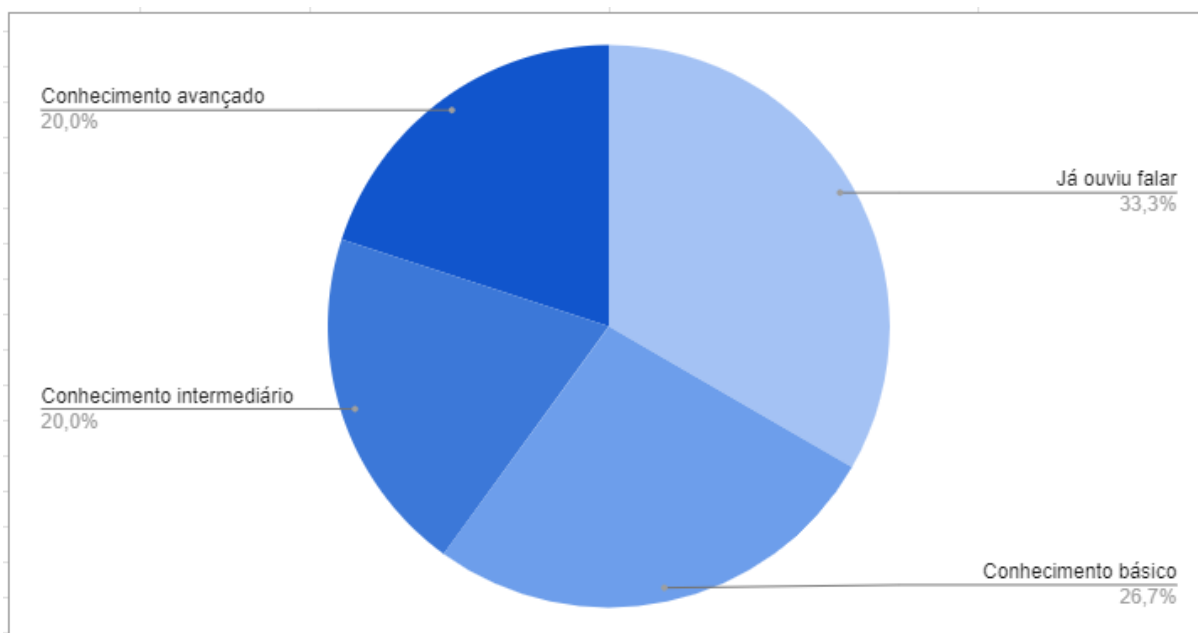


Figura 29 – Gráfico do questionário, referente à usuários que tem conhecimento em visão computacional.

Todos os pesquisados foram expostos ao questionário abaixo de 10 perguntas, onde todos retornaram respostas inteiras de 1 a 5, onde 1 quer dizer 'Discordo completamente' e 5 'Concordo completamente'. As perguntas foram as seguintes:

1. Eu acho que gostaria de usar esse sistema com frequência.
2. Eu acho o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.

5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8. Eu achei o sistema difícil de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Os resultados, expostos na Tabela 6 demonstram que foi alcançado com êxito a satisfação dos usuários, pois pode-se perceber que 53,3% dos pesquisados concordam que gostariam de usar o sistema com frequência, além disso, outros 33,3% concordam completamente com a questão, enquanto 6,7% não concordam nem discordam demais 6,7% discordam da afirmativa. Em consenso com o item anterior os usuários 46,7% dos usuários discordaram completamente que o sistema é desnecessariamente complexo, outros 26,7% discordaram, e 26,7% não discordaram nem concordaram.

Acerca da facilidade do uso do sistema, 60% concordam que o sistema é de fácil utilização, outros 33,3% concordam completamente, enquanto apenas 6,7% nem concordam nem discordam da afirmativa. Além disso, 33,3% dos usuários discordam completamente que necessitou ajuda de alguém com conhecimento técnico para usar o sistema, outros 53,3% discordam da afirmativa, demais 13,3% nem concordaram nem discordaram da afirmativa.

Em relação à integração das funções da ferramenta foi constatado que 53,3% dos usuários concordam completamente que várias funções do sistema estão muito bem integradas, enquanto outros 33,3% concordam também, os 13,3% restantes optaram por não concordarem nem discordarem. Além disso, 53,3% dos usuários discordam completamente que o sistema possui muita inconsistência, enquanto 26,7% discordam porém 20% nem concordam nem discordam.

Referente a rapidez de aprendizagem no uso da ferramenta, 33,3% dos usuários concordam completamente que as pessoas aprenderão como usar o sistema rapidamente, enquanto 53,3% dizem concordar com a afirmativa, enquanto 13,3% optaram por não concordar nem discordar. Em contrapartida, 53,3% dos usuários discordam que o sistema é difícil de usar, outros 26,7% discordam completamente disso, porém 20% nem concordam nem discordam da afirmativa.

Por fim, foi obtido que 33,3% dos usuários concordam completamente que se sentiram confiantes ao usar o sistema, 53,3% concordam e outros 13,3% nem concordam nem discordam com a afirmativa. Em contrapartida 46,7% dos usuários discordam completamente que precisaram aprender várias coisas novas para usar a ferramenta, outros 40% discordam enquanto 13,3% nem concordam nem discordam.

Tabela 6 – Tabela com os resultados da pesquisa com usuários e questionário SUS.

Pergunta	Discordo Completamente	Discordo	Nem concordo, nem discordo	Concordo	Concordo Completamente
Eu acho que gostaria de usar esse sistema com frequência.	0%	6,7%	6,7%	53,3%	33,3%
Eu acho o sistema desnecessariamente complexo.	46,7%	26,7%	26,7%	0%	0%
Eu achei o sistema fácil de usar.	0%	0%	6,7%	60%	33,3%
Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	33,3%	53,3%	13,3%	0%	0%
Eu acho que as várias funções do sistema estão muito bem integradas.	0%	0%	13,3%	33,3%	53,3%
Eu acho que o sistema apresenta muita inconsistência.	53,3%	26,7%	20%	0%	0%
Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	0%	0%	13,3%	53,3%	33,3%
Eu achei o sistema difícil de usar.	26,7%	53,3%	20%	0%	0%
Eu me senti confiante ao usar o sistema.	0%	0%	13,3%	53,3%	33,3%
Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	46,7%	40%	13,3%	0%	0%

Após a coleta dos dados avaliativos dos usuários é necessário executar o cálculo para obtenção geral da avaliação do sistema, onde subtrai-se 1 nas respostas de questões ímpares e subtrai-se 5 nas pares, lembrando que o resultado é o módulo, não tem resultado negativo. Deste modo o resultado geral do aplicativo segundo a avaliação dos usuários com o questionário SUS(BROOKE, 1996) foi de 80,833. Este resultado como visto na Tabela 5 retorna que a ferramenta alcançou a marca de Muito Bom, segunda melhor possibilidade qualitativa de acordo com a escala SUS.

Mediante o teste da ferramenta, foi obtido também comentários sobre sugestões de melhorias para a aplicação, de modo que facilitaria o uso e melhoraria a experiência. As sugestões foram:

- Gerar um *pop-up* ou barra de progresso do processamento de cada rede neural;
- Melhorar o gráfico do monitor, para que seja mais intuitivo;
- Ajustar as cores da ferramenta para que se adapte a cor tema do sistema operacional;
- Melhorias na tabela dos resultados;

6 Conclusão

Este trabalho expôs uma ferramenta *desktop* desenvolvida para facilitar a manipulação das redes neurais. A ferramenta permite aos usuários comparar diferentes redes neurais e classificadores com base em uma série de critérios, como a precisão, a eficiência e a facilidade de uso. Isso pode ser útil para usuários que estão em dúvida sobre qual rede neural usar, visto que existem várias opções disponíveis. A ferramenta fornece um apoio matemático conciso e seguro para a escolha da melhor rede para o caso em questão.

A ferramenta tem o potencial de facilitar a entrada de pessoas no ramo da pesquisa, contribuindo com o avanço desta. A ferramenta também pode ajudar os pesquisadores a tomar decisões mais concisas sobre qual melhor CNN para o problema do pesquisador.

A ferramenta não visa alcançar o melhor retorno possível de cada rede neural, mas sim que seja possível de maneira prática e totalmente gráfica que o pesquisador consiga executar testes com seu *dataset* para que veja qual rede retorna os melhores resultados de acordo com o interesse do usuário. Desse modo, o pesquisador pode aprofundar seu conhecimento diretamente no melhor método, o poupando tempo e deixando-o mais seguro.

A ferramenta desenvolvida tem algumas limitações, desde o hardware do usuário, também o limite de redes neurais, classificadores, métricas e otimizadores disponíveis, pois somente se executa com os que estão disponíveis na ferramenta, não há possibilidade que o usuário adicione alguma dessas configurações.

Como sugestões futuras para a ferramenta, cabem melhorias como:

- Possibilitar a adição de redes neurais, métricas, classificadores e otimizadores pelo próprio usuário;
- Geração de gráficos dinâmicos dos resultados, o que facilitaria mais a exibição e compreensão dos resultados;
- Barra de processamento que demonstraria uma estimativa da porcentagem do processamento atual.
- Inserir informações adicionais para compreensão do usuário sobre o impacto das configurações nos resultados.
- Adicionar explicações breves nas funcionalidades do sistema para auxiliar o usuário.
- Adicionar explicações sobre as métricas e suas divergências, afinal uma pode ser muito positiva enquanto outra não.

Referências

- BERTHOLD, M. R. et al. KNIME: The Konstanz Information Miner. In: *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. [S.l.]: Springer, 2007. ISBN 978-3-540-78239-1. ISSN 1431-8814. Citado 2 vezes nas páginas 24 e 25.
- BREIMAN, L. Random forests. *Machine Learning*, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <<http://dx.doi.org/10.1023/A%3A1010933404324>><http://dx.doi.org/10.1023/A%3A1010933404324>. Citado 2 vezes nas páginas 20 e 21.
- BROOKE, J. "SUS-A quick and dirty usability scale." *Usability evaluation in industry*. CRC Press, 1996. ISBN: 9780748404605. Disponível em: <<https://www.crcpress.com/product/isbn/9780748404605>>. Citado 5 vezes nas páginas 7, 29, 30, 42 e 45.
- CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016. (KDD '16), p. 785–794. ISBN 978-1-4503-4232-2. Disponível em: <<http://doi.acm.org/10.1145/2939672-2939785>><http://doi.acm.org/10.1145/2939672-2939785>. Citado na página 25.
- CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: . [S.l.: s.n.], 2017. p. 1800–1807. Citado na página 12.
- CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 20.
- DEMSAR, J. et al. Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, v. 14, p. 2349–2353, 2013. Disponível em: <<http://jmlr.org/papers/v14/demsar13a.html>><http://jmlr.org/papers/v14/demsar13a.html>. Citado 2 vezes nas páginas 24 e 25.
- DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255. Citado na página 16.
- FILONENKO, A.; KURNIANGGORO, L.; JO, K.-H. Comparative study of modern convolutional neural networks for smoke detection on image data. In: *2017 10th International Conference on Human System Interactions (HSI)*. [S.l.: s.n.], 2017. p. 64–68. Citado na página 12.
- GOODFELLOW, I. J. et al. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. Citado na página 15.
- GVR. Computer vision market size. In: . [S.l.]: Grand View Research, 2021. Citado na página 11.
- HAYKIN, S. *Neural networks: a comprehensive foundation*. [S.l.]: Prentice Hall PTR, 1994. Citado na página 21.

HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>><http://arxiv.org/abs/1512.03385>. Citado 2 vezes nas páginas 12 e 18.

HUANG, G. et al. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2017. Citado 2 vezes nas páginas 15 e 17.

KRIZHEVSKY, A. One weird trick for parallelizing convolutional neural networks. *CoRR*, abs/1404.5997, 2014. Disponível em: <<http://arxiv.org/abs/1404.5997>><http://arxiv.org/abs/1404.5997>. Citado na página 15.

LANG, S. et al. Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. *Knowledge-Based Systems*, v. 178, p. 48–50, 2019. ISSN 0950-7051. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950705119301789>>. Citado 2 vezes nas páginas 24 e 25.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, p. 436–444, 2015. Citado na página 15.

PARVIN, F.; HASAN, M. A. M. A comparative study of different types of convolutional neural networks for breast cancer histopathological image classification. In: *2020 IEEE Region 10 Symposium (TENSYP)*. [S.l.: s.n.], 2020. p. 945–948. Citado na página 12.

PASZKE, A. et al. Pytorch: An imperative style, high-performance deep learning library. In: WALLACH, H. et al. (Ed.). *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: <<http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>><http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>. Citado na página 12.

PAWANGFG. *VGG-16 / CNN model*. [S.l.], 2020. Disponível em: <<https://www.geeksforgeeks.org/vgg-16-cnn-model/>>. Acesso em: 23 mar. 2022. Citado na página 17.

RAKOTOMALALA, R. Tanagra : un logiciel gratuit pour l’enseignement et la recherche. In: . [S.l.: s.n.], 2005. v. 2, p. 697–702. Citado 2 vezes nas páginas 24 e 25.

RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. 2014. Cite arxiv:1409.0575Comment: 43 pages, 16 figures. v3 includes additional comparisons with PASCAL VOC (per-category comparisons in Table 3, distribution of localization difficulty in Fig 16), a list of queries used for obtaining object detection images (Appendix C), and some additional references. Disponível em: <<http://arxiv.org/abs/1409.0575>><http://arxiv.org/abs/1409.0575>. Citado na página 15.

SAKURAI, R. *Implementando a estrutura de uma Rede Neural Convolutacional utilizando o MapReduce do Spark*. Brasil, 2017. Disponível em: <<http://www.sakurai.dev.br/cnn-mapreduce/>><http://www.sakurai.dev.br/cnn-mapreduce/>. Acesso em: 23 mar. 2022. Citado na página 14.

SAUMYATRIPATHI. *Face Recognition using Vgg-16*. [S.l.], 2020. Disponível em: <<https://medium.com/@saumya3006tripathi/face-recognition-using-vgg-16-21a23791d5e9>>. Acesso em: 23 mar. 2022. Citado na página 18.

- SILVER, D. et al. Mastering the game of go with deep neural networks and tree search. *Nature*, Nature Publishing Group, v. 529, p. 484–489, 2016. Citado na página 14.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <<http://arxiv.org/abs/1409.1556>><http://arxiv.org/abs/1409.1556>. Citado na página 16.
- SZEGEDY, C. et al. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. Disponível em: <<http://arxiv.org/abs/1409.4842>><http://arxiv.org/abs/1409.4842>. Citado 2 vezes nas páginas 18 e 19.
- VARSHNEY, P. *Alexnet Architecture: A complete guide*. Kaggle, 2020. Disponível em: <<https://www.kaggle.com/code/blurredmachine/alexnet-architecture-a-complete-guide>>. Citado na página 16.
- VOHRA, D. Apache mahout. In: _____. [S.l.: s.n.], 2016. p. 377–414. ISBN 978-1-4842-2198-3. Citado 2 vezes nas páginas 24 e 25.
- WILLIAMS, G. J. Rattle: A Data Mining GUI for R. *The R Journal*, v. 1, n. 2, p. 45–55, 2009. Disponível em: <<https://doi.org/10.32614/RJ-2009-016>>. Citado 2 vezes nas páginas 24 e 25.



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- () Tese
- () Dissertação
- (**x**) Monografia
- () Artigo

Eu, **Narciso de Sousa Rodrigues**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação **Uma ferramenta automática para reconhecimento de padrões e classificação de dados** de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 23 de agosto de 2023.

Narciso de Sousa Rodrigues

Assinatura