

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

THIAGO GONÇALVES LIRIO

**ESTUDO E IMPLEMENTAÇÃO DE TÉCNICAS DE AUMENTO DE
DADOS PARA APLICAÇÕES DE REDES NEURAIS
CONVOLUCIONAIS**

FLORIANÓPOLIS, 2022.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

THIAGO GONÇALVES LIRIO

**ESTUDO E IMPLEMENTAÇÃO DE TÉCNICAS DE AUMENTO DE
DADOS PARA APLICAÇÕES DE REDES NEURAIS
CONVOLUCIONAIS**

Trabalho de Conclusão de Curso
submetido ao Instituto Federal de
Educação, Ciência e Tecnologia de Santa
Catarina como parte dos requisitos para
obtenção do título de Engenheiro em
Mecatrônica.

Orientador:
Maurício Edgar Stivanello, Dr. Eng.
Coorientador:
Raimundo Ricardo Matos da Cunha, Dr.
Eng.

FLORIANÓPOLIS, 2022.

Ficha de identificação da obra elaborada pelo autor.

Lirio, Thiago

Estudo e implementação de técnicas de aumento de dados para aplicações de redes neurais convolucionais / Thiago Lirio; orientação de Mauricio Stivanello; coorientação de Raimundo Cunha. - Florianópolis, SC, 2022. 2022 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Florianópolis. Bacharelado em Engenharia Mecatrônica. Departamento Acadêmico de Metal Mecânica. Inclui Referências.

1. Redes Neurais Artificiais. 2. Redes Neurais Convolucionais. 3. Aumento de Dados. 4. Banco de Imagens.
I. Stivanello, Mauricio. II. Cunha, Raimundo. III. Instituto Federal de Santa Catarina. IV. Estudo e implementação de técnicas de aumento de dados para aplicações de redes neurais convolucionais.

ESTUDO E IMPLEMENTAÇÃO DE TÉCNICAS DE AUMENTO DE DADOS PARA APLICAÇÕES DE REDES NEURAIS CONVOLUCIONAIS

THIAGO GONÇALVES LIRIO

Este trabalho foi julgado adequado para obtenção do título de Engenheiro em Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 21 de Julho, 2022.

Banca Examinadora:

Maurício Edgar Stivanello, Dr. Eng.
Instituto Federal de Santa Catarina

Delcino Picinin Júnior, Dr. Eng.
Instituto Federal de Santa Catarina

Gregory Chagas da Costa Gomes, Me. Eng.
Instituto Federal de Santa Catarina

AGRADECIMENTOS

Deixo aqui meu profundo agradecimento primeiramente aos meus pais que sempre fizeram tudo que podiam para a minha educação e formação. Além de todo o apoio emocional e acolhimento que sempre tive durante toda a minha caminhada. Gostaria de agradecer também ao meu irmão por todo o companheirismo e amizade que tivemos e que me ouviu desabafar em alguns momentos dessa jornada. Outro agradecimento especial é a todos que fazem parte da minha família por inteiro que sempre me incentivaram a fazer tudo aquilo que sempre quis.

Para a minha namorada e companheira Eduarda Faber queria deixar muito mais que um agradecimento, pois ela me acompanha desde a metade do curso e sabe o quanto foi importante em diversos momentos difíceis durante este trajeto.

Outro agradecimento importante é aos meus colegas, amigos e professores do Ensino Médio que sem eles nada disso seria possível. Assim como um agradecimento especial a todos aqueles amigos, colegas e professores durante o curso.

Um abraço e um agradecimento muito mais que especial ao meu professor orientador Mauricio Stivanello, assim como ao meu co-orientador Raimundo Ricardo Matos, que me apoiaram, me compreenderam e fizeram o melhor deles durante o desenvolvimento do presente trabalho.

Por fim, agradecer a todos aqueles que de certa forma contribuíram e me ajudaram a conquistar este importante objetivo.

“Se você focar nos obstáculos, irá esquecer dos seus sonhos e acabará desistindo deles, se focar nos seus sonhos, irá ganhar resistência e coragem para enfrentar todos seus problemas.”
Pablo Paucar.

RESUMO

As redes neurais artificiais (RNA's), são sistemas computacionais cujos modelos são baseados nas estruturas do cérebro humano. Dentre os tipos de RNA's existem as redes neurais convolucionais, também conhecidas como CNN's. As Redes Neurais Convolucionais são treinadas a partir de imagens de conjuntos de dados de treinamento. A rede neural é tão boa quanto os dados que você alimenta. Porém, às vezes não é possível ter a quantidade de imagens suficientes para se obter uma precisão e acurácia julgadas necessárias para aplicação. Pensando em criar uma ferramenta que auxilie a melhorar os *Datasets* já existentes, foi criado um *software* com interface gráfica para gerar mais imagens utilizando as técnicas de aumento de dados. É um sistema executável que realiza transformações geométricas e adiciona ruído gaussiano. Esse sistema foi testado em relação às suas funcionalidades, tempo de geração de imagens e consumo de memória e capacidade de processamento. Nestes testes foi possível confirmar que utilizar este tipo de sistema é mais rápido, do que realizar a programação do aumento de dados durante a implementação de uma rede neural.

Palavras-chave: Redes Neurais Artificiais. Redes Neurais Convolucionais. Aumento de Dados. Banco de Imagens.

ABSTRACT

Artificial neural networks (ANNs) are computational systems whose models are based on the structures of the human brain. Among the types of ANNs there are convolutional neural networks, also known as CNNs. Convolutional Neural Networks are trained from images of training datasets. The neural network is only as good as the data you feed it. However, sometimes it is not possible to have the amount of images sufficient to obtain a precision and accuracy deemed necessary for the application. Thinking about creating a tool that helps to improve the existing Datasets, a software with a graphical interface was created to generate more images using the techniques of data augmentation. It is an executable system that performs geometric transformations and adds Gaussian noise. This system was tested in terms of functionality, imaging time, memory consumption and processing power. In these tests it was possible to confirm that using this type of system is faster than performing the data increase programming during the implementation of a neural network.

Keywords: Artificial Neural Network. Convolutional Neural Network. Data Augmentation. Dataset.

LISTA DE FIGURAS

Figura 1 – Comparação entre um neurônio biológico e um artificial.....	18
Figura 2 – Interligação dos nós das redes neurais artificiais.....	18
Figura 3 – Arquitetura Rede Artificial LeNet5.....	20
Figura 4 – Imagem como uma matriz de pixel.....	21
Figura 5a – Imagem original, com marcador da matriz de pixels.....	22
Figura 5b – Valores do filtro da matriz de pixel.....	22
Figura 5c – Multiplicação do filtro pela matriz de pixels da imagem original.....	22
Figura 6 – Camada de pooling.....	22
Figura 7 – Maneiras de organizar o <i>Dataset</i>	26
Figura 8 – Atuação do <i>Data Augmentation</i> em relação a ordem de aplicação.....	28
Figura 9 – Cameraman.....	29
Figura 10 – Rotação de um ponto no plano.....	30
Figura 11 – Equação matricial de rotação.....	31
Figura 12 – Data Augmentation In Play.....	31
Figura 13a – Cameraman rotacionado 45°.....	32
Figura 13b – Cameraman rotacionado 135°.....	32
Figura 13c – Cameraman rotacionado 225°.....	32
Figura 14 – Espelhamento de um ponto em relação ao eixo X.....	32
Figura 15 – Equação matricial do espelhamento em relação ao eixo X.....	33
Figura 16 – Cameraman espelhado na vertical.....	33
Figura 17 – Espelhamento de um ponto em relação ao eixo Y.....	34
Figura 18 – Equação matricial do espelhamento em relação ao eixo Y.....	34
Figura 19 – Cameraman espelhado na vertical.....	34
Figura 20 – Escala de uma imagem no plano.....	35
Figura 21 – Equação matricial da escala.....	36
Figura 22a – Cameraman com escala 2 na horizontal.....	36
Figura 22b – Cameraman com escala 2 na vertical.....	36
Figura 22c – Cameraman com escala 0,5 na horizontal.....	36
Figura 22d – Cameraman com escala 0,5 na vertical.....	36
Figura 23 – Translação de um ponto no plano.....	37
Figura 24 – Equação matricial de translação.....	37
Figura 25 – Resultado da translação de uma figura no plano.....	37
Figura 26a – Cameraman transladada 20% para direita.....	38
Figura 26b – Cameraman transladada 20% para baixo.....	38
Figura 26c – Cameraman transladada 20% para esquerda.....	38
Figura 26d – Cameraman transladada 20% para cima.....	38
Figura 27 – Cisalhamento de uma imagem.....	39
Figura 28 – Equação matricial do cisalhamento.....	39
Figura 29a – Cameraman Cisalhamento 45°.....	40
Figura 29b – Cameraman Cisalhamento -45°.....	40

Figura 30a – Cameraman corte realizado 1.....	40
Figura 30b – Cameraman corte realizado 2.....	40
Figura 30c – Cameraman corte realizado 3.....	40
Figura 31 – Transformações fotométricas.....	41
Figura 32 – Ruídos gaussiano (b) e impulsivo (c) aplicados sobre imagem modelo (a).....	42
Figura 33 – Função gaussiana.....	43
Figura 34 – Cameraman com desvio padrão 1 e tom de cinza 255.....	43
Figura 35 – Diagrama de caso de uso do sistema de aumento de dados.....	49
Figura 36 – Diagrama de classes para o sistema de aumento de dados.....	50
Figura 37 – Diagrama de sequência do sistema de aumento de dados.....	51
Figura 38 – Tela home Data augmentation System.....	52
Figura 39 – Tela de transformações geométricas.....	53
Figura 40 – Tela de corte e cisalhamento do sistema de aumento de dados.....	53
Figura 41 – Tela de adição de ruído do sistema de aumento de dados.....	54

LISTA DE TABELAS

Tabela 1 – Quantidade de imagens em <i>dataset</i> de exemplo.....	25
Tabela 2 – Exemplo da Quantidade De Imagens Equilibradas em um Dataset.....	25
Tabela 3 – Transformações da Biblioteca <i>Imgaug</i>	45
Tabela 4 – Parâmetros de utilização do sistema.....	54
Tabela 5 – Teste das transformações realizados através do sistema no banco de dados de Rebites.....	56
Tabela 6 – Teste das transformações realizadas no banco de dados de Dogs vs. Cats.....	58
Tabela 7 – Teste das transformações realizados através do sistema no banco de dados de Rebites.....	59
Tabela 8 – Tempo médio consumido.....	60

LISTA DE ABREVIATURAS E SIGLAS

Asirra	Animal Species Image Recognition for Restricting Access
CNN, ConvNet	Convolutional Neural Network (Rede Neural Convolucional)
DA	Data Augmentation (Aumento de Dados)
GUI	Graphical User Interface (Interface Gráfica do Usuário)
HIP	Human Interactive Proof (Prova Interativa Humana)
IA	Inteligência Artificial
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
IFSC	Instituto Federal de Santa Catarina
IoT	Internet of Things (Internet das Coisas)
MLP	Multilayer Perceptron (Perceptron de Múltiplas Camadas)
RAM	Random Access Memory (Memória de Acesso Aleatório)
RGB	Red, Green and Blue (Vermelho, Verde e Azul)
RNA	Rede Neural Artificial

SUMÁRIO

1. Introdução.....	15
1.1 Contextualização.....	15
1.2 Justificativa.....	16
1.3 Objetivo Geral	17
1.4 Objetivos Específicos	17
2. Fundamentação Teórica	18
2.1 Redes Neurais Artificiais	18
2.2 Redes Neurais Convolucionais.....	20
2.3 Datasets/Imagesets	24
2.4 Data Augmentation	27
2.4.1 Transformações Geométricas	28
2.4.1.1 Rotação	29
2.4.1.2 Espelhamento	32
2.4.1.3 Escala.....	35
2.4.1.4 Translação	36
2.4.1.5 Cisalhamento	38
2.4.1.6 Corte	40
2.4.2 Transformações Fotométricas	40
2.4.2.1 Adição de Ruído	41
2.4.2.2 Ruído Gaussiano	42
2.5 Ferramentas Úteis.....	43
2.5.1 Ambiente de Desenvolvimento e Software.....	44
2.5.2 Linguagem de Programação.....	44
2.5.3 Bibliotecas Utilizadas	45
2.5.3.1 OpenCV.....	45
2.5.3.2 Imgaug	45
2.5.3.3 Tkinter	46
2.6 Trabalhos Correlatos	47
3. Desenvolvimento de um sistema de criação de bases de imagens através de aumento de dados.....	48
3.1 Discussão sobre os requisitos do sistema	48
3.2 Ferramentas selecionadas para atender cada requisito.....	48
3.3 Descrição da arquitetura do sistema	49
3.3.1 Interface gráfica.....	51
3.3.2 Descrição da utilização do sistema	54
4. Apresentação dos Resultados	56
4.1 Análise e discussão dos resultados.....	60

5. Considerações Finais	62
5.1 Sugestões para trabalhos futuros	62
Referências.....	64

1 Introdução

Neste capítulo será dada uma breve introdução sobre os principais assuntos que o presente trabalho se baseou.

1.1 Contextualização

Redes neurais artificiais (RNA's) são sistemas computacionais cujos modelos são baseados nas estruturas do cérebro humano. Organizados por nós interconectados e por camadas. Tais técnicas são utilizadas para reconhecer padrões e relações nos dados de entrada, agrupando e classificando. (SAS, 2019).

Dentre os tipos de RNA's existem as redes neurais convolucionais, também conhecidas como CNN's ou ConvNet's. Tais redes vêm sendo cada vez mais utilizadas com visão computacional. Por estarem obtendo resultados na interpretação de imagens superiores aos obtidos pelas técnicas clássicas em diversas aplicações como na Detecção e classificação de equipamentos em torres de telecomunicação por OSHIRO, 2022. Assim como na detecção da Covid-19 através de imagens de Raio-x, por RICCI, SANTOS, OTONI, 2021. (DATA SCIENCE ACADEMY, 2021).

Além das aplicações citadas acima, as indústrias também vêm utilizando cada vez mais soluções oriundas da indústria 4.0. Sendo uma delas a utilização de redes neurais, incluindo a CNN.

As Redes Neurais Convolucionais são treinadas a partir de imagens de conjuntos de dados de treinamento, também conhecidos como *Datasets* ou *Imagesets*. A seleção do conjunto de dados é uma das tarefas mais importantes na utilização de uma rede neural convolucional, “A rede neural é tão boa quanto os dados que você alimenta”. (GANDHI, 2021).

Os *datasets* são de modo geral composto por muitas imagens. Entretanto, em alguns casos não se tem um conjunto de dados bons o suficientes para gerar uma boa precisão e acurácia nas redes neurais. É necessário realizar um “pré-processamento” ao treinamento da CNN, gerando mais imagens a partir das já existentes. Este pré-processamento consiste em aumentar o conjunto de dados. Tal

técnica é conhecida por aumento de dados, ou *Data Augmentation*. Esse aumento de dados é dado a partir de transformações geométricas e/ou fotométricas. Para uma boa rede neural artificial, é necessário de um dataset completo, com grandes variações de todas as classes de saída. Para realizar esse pré-processamento da rede neural convolucional é um processo trabalhoso, lento e que demanda bom conhecimento de programação.(GANDHI, 2021).

1.2 Justificativa

Atualmente é possível observar grande utilização nas indústrias de CNN's. Esse aumento é devido a grande velocidade e precisão das redes neurais convolucionais. Para obter grande precisão e/ou acurácia, uma CNN necessita de um banco de imagens de entrada bom o suficiente, para a rede conseguir associar corretamente as imagens de entrada com as classes de saída. Na grande maioria das vezes os *Datasets* utilizados não são balanceados de forma correta, dificultando o aprendizado das mesmas. (ALVES, 2020).

Ter um conjunto de dados, contendo apenas uma classe de saída em grande quantidade e as outras classes com poucas imagens não é recomendado. Nessas condições a rede conseguiria classificar de forma correta somente um tipo de saída, tendo um baixo índice de acerto nas outras classes de saída. Tal situação é conhecida como *Dataset* desbalanceado. (ROSA, 2019).

O tamanho de um dataset ideal depende muito da aplicação. Atualmente existem datasets na ordem de milhões de imagens. O ideal para aplicações mais simples é encontrar banco de imagens na ordem de centenas, porém o mais encontrado hoje em dia são datasets na ordem de milhares. (GANDHI, 2021).

Pensando em auxiliar a indústria foi pensado em criar uma ferramenta que auxilie a melhorar os Datasets já existentes com interface gráfica. Esse sistema irá gerar mais imagens utilizando as técnicas de aumento de dados. Com o intuito de melhorar a precisão e a acurácia das redes já utilizadas. Facilitaria a geração de dados sem uma codificação, somente definindo parâmetros e selecionando as transformações desejadas. Além de realizar esse aumento de dados de forma mais

rápida em comparação a um pré-processamento realizado diretamente no código da rede.

1.3 Objetivo Geral

Desenvolver uma ferramenta com interface gráfica na qual facilite a aplicação de técnicas de aumento de dados. Gerando variações das imagens existentes do Dataset, produzindo assim um banco de dados de treinamento satisfatório.

1.4 Objetivos Específicos

Com o intuito de completar o objetivo geral, foram definidos diversos objetivos específicos, abaixo citados, facilitando assim a conclusão do presente trabalho.

- a) Estudo teórico das redes neurais artificiais;
- b) Estudo teórico das técnicas de aumento de dados;
- c) Projeto e implementação das técnicas de aumento de dados;
- d) Desenvolvimento de interface gráfica;
- e) Integração da interface gráfica com a integração do algoritmo de aumento de dados;
- f) Validação do projeto;
- g) Documentação dos resultados obtidos.

2 Fundamentação Teórica

Neste capítulo serão apresentados e discutidos os conceitos teóricos, na qual serviram para o desenvolvimento da ferramenta proposta nos objetivos do trabalho.

2.1 Redes Neurais Artificiais

As redes neurais artificiais, ou RNA's, "são um subconjunto de aprendizado de máquinas e estão no núcleo dos algoritmos de deep learning." (IBM CLOUD EDUCATION, 2020).

A primeira proposta de uma rede neural artificial surgiu em 1943 com Warren McCulloch e Walter Pitts. Tinha como objetivo solucionar problemas conforme o cérebro humano baseado em modelos matemáticos. Somente em 1958 foi posto em prática um primeiro modelo com o pesquisador Frank Rosenblatt, na qual denominou de Perceptron uma rede desenvolvida pelo próprio. Constituída por neurônios MLP arranjados em forma de rede. Já em 1969 um artigo publicado por Minsky e Papert, ilustrou diversas adversidades. Fazendo com que não avançassem nos estudos sobre RNA's. Somente a partir de 1982 foram retomadas pesquisas com relação a essa tecnologia com a publicação do trabalho de Hopfield. Desde então o campo de redes neurais artificiais só vem crescendo e ganhando força. (DATA SCIENCE ACADEMY, 2021).

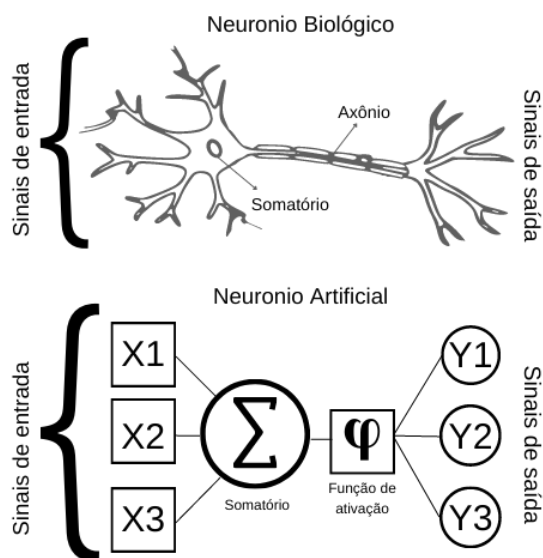
As RNA's de forma muito simplificada são sistemas computacionais, onde são utilizados nós conectados. Esses nós simulam as estruturas dos neurônios do cérebro humano, através de modelos matemáticos. (SAS, 2019).

A Figura 1 ilustra a comparação entre um neurônio biológico ao modelo matemático de uma rede neural.

A rede neural é composta por diversos neurônios artificiais que são conhecidos como nós.

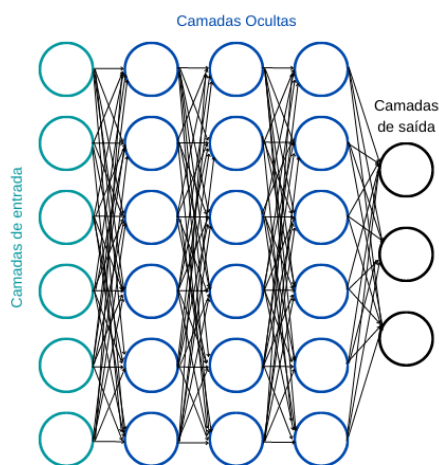
Os nós são organizados em forma de camadas, nas quais podem ser de diferentes tipos de dados. Incluindo uma camada de entrada, podendo ter diversas camadas ocultas e uma camada de saída, assim como ilustrado na Figura 2. (ALVES, 2018).

Figura 1 – Comparação entre um neurônio biológico e um artificial.



Fonte: Do Autor, 2022.

Figura 2 – Interligação dos nós das redes neurais artificiais.



Fonte: Do Autor, 2022.

O modelo matemático de uma rede neural artificial foi pensado para ser uma regressão linear. Cada nó tem o seu próprio modelo matemático, dados de entrada, pesos e saída. O valor de saída de cada nó é definido através de um limiar e ao final de todas as camadas obtém-se um número binário de saída. A partir deste número binário é determinado a qual classe de saída o dado de entrada representa. (IBM CLOUD EDUCATION, 2020).

Com o passar do tempo, foi percebendo-se que as redes neurais artificiais poderiam ser utilizadas para tarefas específicas. Atualmente as redes neurais

específicas auxiliam em diversas tarefas do cotidiano como no reconhecimento de fala, filtros das redes sociais, jogos de videogame, diagnóstico medicinal, na visão computacional, entre diversos outros. (SAS, 2019).

As RNA's podem ser divididas em diferentes tipos de redes. As redes neurais recorrentes, na qual utilizam informações sequenciais, onde os dados de entrada são relacionados e dependentes um do outro, normalmente aplicadas em análises de texto. As redes neurais feedforward, onde cada perceptron é conectado ao perceptron da camada seguinte, não tendo retorno ou feedback para própria, utilizado para se obter previsões mais rapidamente. As redes neurais autoencoder tem como objetivo aumentar a sensibilidade do que é relevante e diminuir do que é irrelevante, normalmente utilizadas como classificadores lineares. Por fim existem as redes neurais convolucionais na qual se trata o escopo deste trabalho, sendo descrito de forma mais detalhada abaixo.(SAS, 2019).

2.2 Redes Neurais Convolucionais

As redes neurais convolucionais, conhecidas como ConvNet ou CNN, “são arquiteturas biologicamente inspiradas capazes de serem treinadas e aprenderem representações invariantes a escala, translação, rotação e transformações afins” (LECUN; KAVUKCUOGLU; FARABET, 2010, Tradução nossa).

A história das CNN's caminha junto com as RNA's, pois a partir do desenvolvimento da primeira rede neural artificial, surgiu o problema de reconhecimento e classificação de imagens. Tal problema começou a ser discutido no meio acadêmico nos anos 70. Porém somente em 1998 foi publicado o artigo “Gradient-based learning applied to document recognition“, de Yann LeCun, Léon Bottou, Yoshua Bengio e Patrick Haffner, estabelecendo o conceito de redes neurais convolucionais como conhecido atualmente. Com o passar do tempo foi-se desenvolvendo CNN's mais complexas e profundas como a AlexNet e a GoogleNet chegando a conter 4 milhões de parâmetros. (DATA SCIENCE ACADEMY, 2021).

Uma Rede Neural Convolucional por ser um tipo de RNA é também um algoritmo de Deep Learning, captando imagens de entrada, atribuindo pesos e

importâncias aos objetos sendo assim capaz de identificá-los. (DATA SCIENCE ACADEMY, 2021).

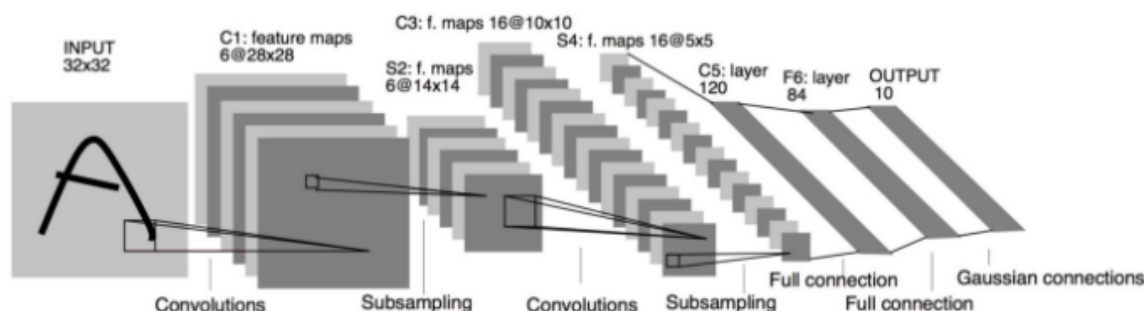
Um dos grandes desafios para uma Rede neural artificial é o reconhecimento de padrões, com maior dificuldade em imagens. Tal tarefa se torna ainda mais desafiadora quando os dados de entrada correspondem a imagens de objetos, visto que as mesmas podem variar de acordo com a iluminação, orientação, pose, foco, entre diversas outras variáveis. (JURASZEK, 2014).

A constituição de uma imagem é hierárquica, logo:

“[...]uma boa representação interna deve ser hierárquica, onde os pixels formam arestas, as arestas formam padrões, os padrões combinados dão origem a partes, as partes combinadas formam os objetos e os objetos formam cenas.” (LECUN; KAVUKCUOGLU e FARABET, 2010, TRADUÇÃO NOSSA).

Com a afirmação acima observa-se que uma imagem forma uma cadeia hierárquica, onde o topo seria a saída. Essa cadeia é mostrada na Figura 3. De forma geral os modelos de CNN, passam por camadas de convolução, pooling e totalmente conectadas, formando a cadeia de nós interconectadas. (JURASZEK, 2014).

Figura 3 - Arquitetura Rede Artificial LeNet5.

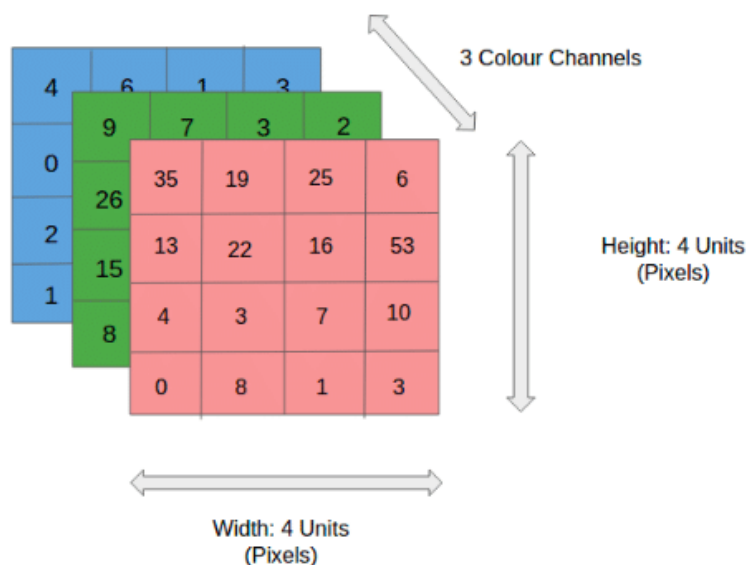


Fonte: Alves, 2018.

A partir da afirmação de LECUN; KAVUKCUOGLU e FARABET acima, é possível compreender que o sistema primeiramente busca a identificação dos pixels, porém é preciso compreender como uma máquina interpreta uma imagem.

Uma imagem é uma composição de matrizes. Quando uma imagem é em escala de cinza, contém somente 1 canal, ou seja, contém somente 1 matriz. Enquanto uma imagem colorida é tridimensional, por conter três canais. Usualmente esses canais são representados pelo sistema de cor RGB. Assim a máquina interpreta que uma imagem é uma matriz de pixels, conforme ilustra a Figura 4. (ALVES, 2018).

Figura 4 – Imagem como uma matriz de pixel.



Fonte: Alves, 2018.

A partir do reconhecimento da imagem como matriz de pixels, as redes neurais convolucionais aplicam um filtro, também chamado de *kernel*, essa etapa é chamada de camada de convolução.

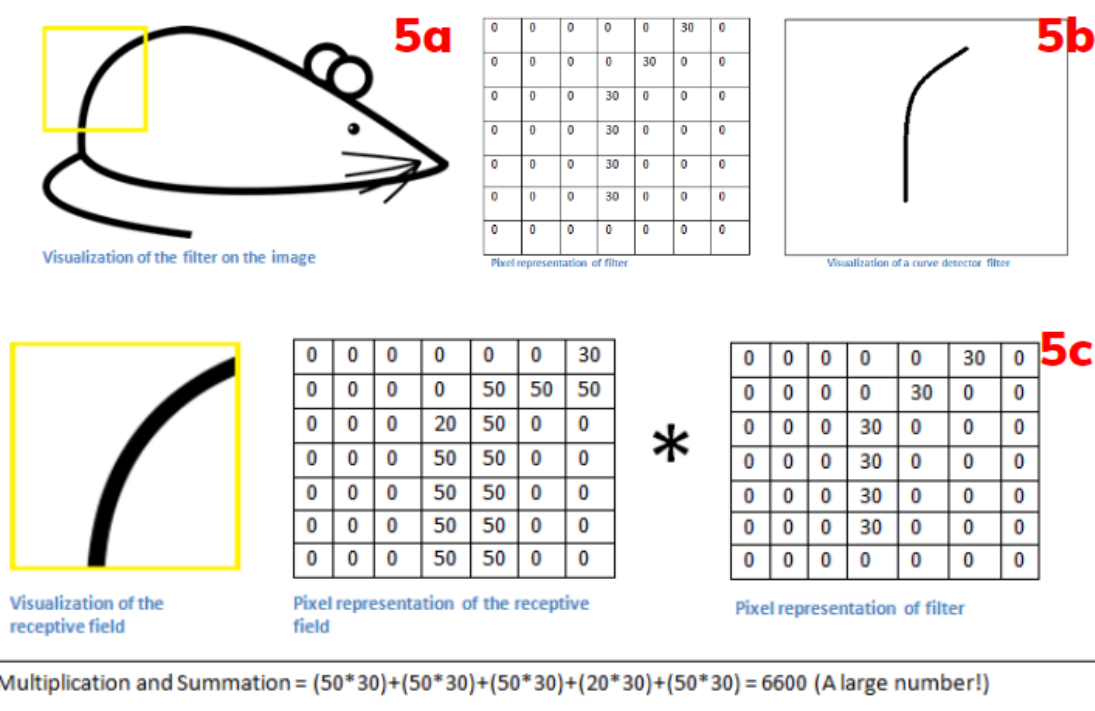
Esses filtros passam por toda a imagem em todos os canais da mesma, captando os traços, ou as arestas como dito anteriormente. Inicialmente esse kernel é formado por pesos aleatórios, atualizando a cada entrada, formando valores com o modelo matemático de regressão linear, fazendo isso quantas vezes necessárias para passar às camadas de pooling. (ALVES, 2018).

As Figuras 5a, 5b, 5c, abaixo ilustram o funcionamento deste filtro, supondo a imagem do desenho de um rato. A Figura 5a é a imagem original com o desenho de um pequeno rato, com a marcação quadrada de uma parte do desenho.

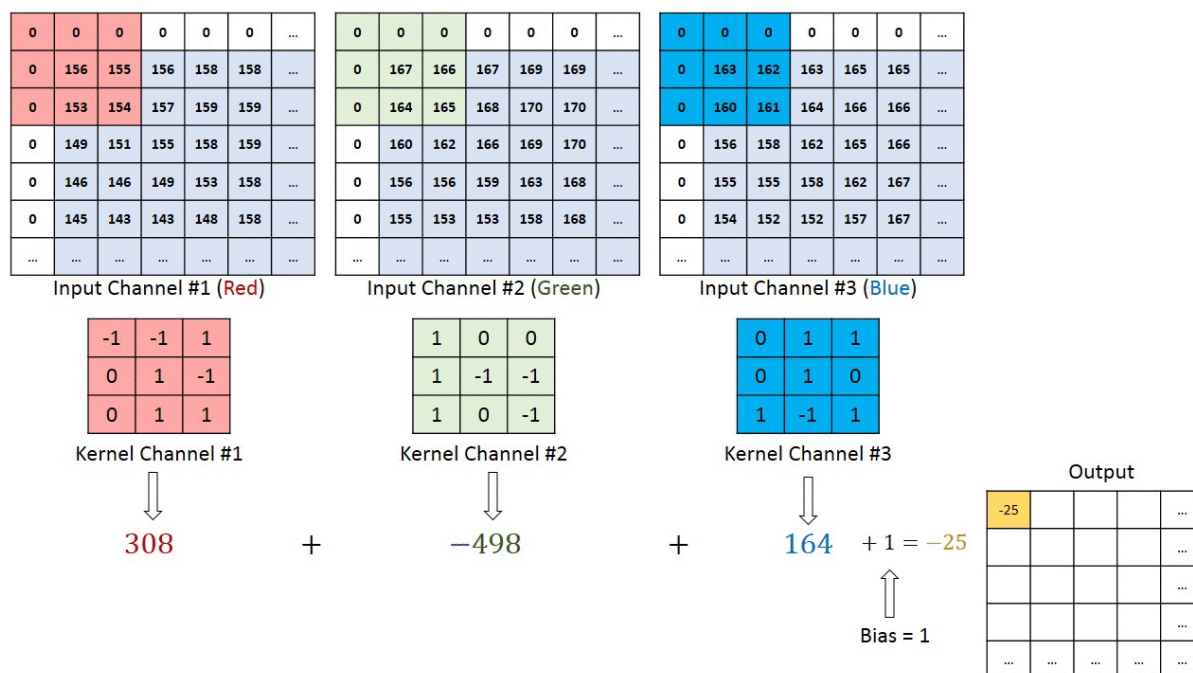
A Figura 5b é a matriz de pixels comparada com o pedaço do desenho retirado. A Figura 5c é a multiplicação do filtro com a matriz de pixels.

O filtro aplicado multiplica os valores dos pixels, passando por cada pixel e em cada canal para assim gerar uma matriz de saída, conforme a figura 6. (ALVES, 2018).

Figura 5a – Imagem original, com marcador da matriz de pixels. 5b - Valores do filtro na matriz de pixel. 5c - Multiplicação do filtro pela matriz de pixels da imagem original

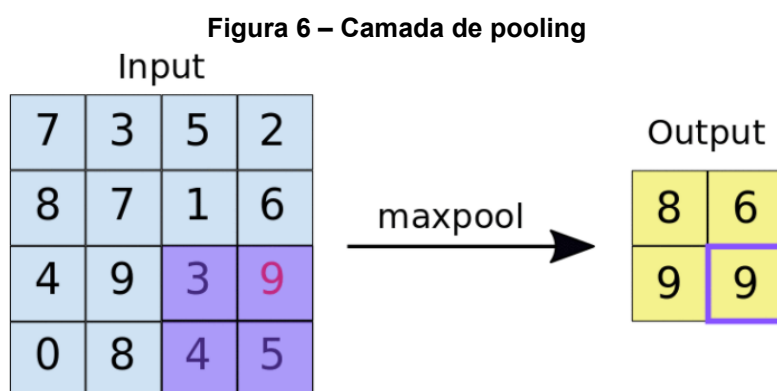


Fonte: Alves, 2018.



Fonte: Alves, 2018.

Após a aplicação do *kernel*, a matriz de saída é do mesmo tamanho da de entrada, logo é necessário passar pela camada de Pooling. Essa camada é utilizada para simplificar a informação. Assim como a camada anterior, a camada de pooling é necessário escolher a área e o método de pooling. Atualmente o mais utilizado é o *maxpooling*, pois ele seleciona o maior valor da unidade para indicar a saída, conforme a Figura 6. (ALVES, 2018).



Fonte: Alves, 2018.

Por fim, depois de passar por várias camadas de pooling e de *kernel* a imagem passa pela camada chamada totalmente conectada. Após essa etapa é finalizada a classificação. A saída desta camada é a quantidade de nós correspondentes a quantidade de classes de saída. (ALVES, 2018).

Com esse funcionamento o pré-processamento de uma CNN é menor em comparação com outros algoritmos de classificação. Normalmente nas outras redes os filtros são feitos à mão, enquanto as CNN's têm a capacidade de aprender todas as características e filtros necessários no processo. (ALVES, 2018).

Para ter essa capacidade de aprender características e filtros, as redes neurais convolucionais necessitam de uma boa base de dados de entrada. Esses dados devem estar balanceados e com boa quantidade, então uma das coisas mais importantes em uma CNN é ter um *dataset* com uma quantidade o suficiente de imagens e equilibrado. (ALVES, 2018).

2.3 Datasets/Imagesets

Os *datasets* são conjuntos de dados de entrada para o aprendizado das máquinas. No treinamento de uma CNN, um conjunto de imagens de entrada com gabarito é utilizado. A qualidade e quantidade desses dados têm papel determinante no nível de sucesso da rede (ULKU; AKAGUNDUZ, 2020, TRADUÇÃO NOSSA).

“No caso de redes para classificação de imagens, esse treinamento depende de imagens anotadas pixel a pixel. Classificar manualmente esse tipo de imagem é bastante demorado, frustrante e comercialmente caro. Daí a importância dos chamados ImageSets, ou banco de imagens.” (Cunha, 2020).

Os *Datasets* têm dois problemas principais: desbalanceamento e quantidade ideal de imagens. Os *Datasets* são considerados desbalanceados quando existe um conjunto de dados de apenas uma classe de saída em grande quantidade e os outros com poucas. A falta de balanceamento dificulta o aprendizado e a classificação das CNN's, pois a máquina saberá classificar bem somente um tipo de saída tendo um baixo índice de acerto nas outras classes de saída. (ROSA, 2019).

Na Tabela 1 tem-se um exemplo de classificação, onde o *dataset* de entrada estaria desbalanceado. Imaginando que se queira classificar animais em imagens e se tem as quantidades de arquivos dentro do Imageset.

Tabela 1 – Quantidade de imagens em *dataset* de exemplo

Animal	Quantidade de Imagens
Cachorro	1780
Gato	1200
Passarinho	390
Peixe	780

Fonte: Do Autor, 2022.

É possível observar através da tabela 1 que existem mais imagens de cachorro do que das outras classes. Percebe-se que o *Dataset* está desbalanceado. Isso faz com que surjam problemas na hora da utilização da rede neural. No momento da classificação a CNN, por exemplo, acabaria classificando peixes como passarinhos. Este erro ocorreria pela rede não ter sido treinada com uma quantidade de exemplos suficientes para tais classes. Ainda considerando tal exemplo, uma situação próxima do ideal seria como a representada pela Tabela 2, onde a quantidade de imagens de cada classe é muito parecida.

Tabela 2 – Exemplo da quantidade de imagens equilibradas em um Dataset

Animal	Quantidade de Imagens
Cachorro	1780
Gato	1700
Passarinho	1790
Peixe	1780

Fonte: Do Autor, 2022.

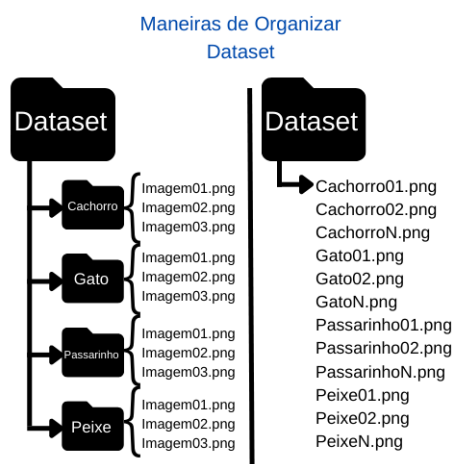
Como mencionado anteriormente, outro problema existente em relação aos *Datasets* é a quantidade ideal de dados necessários para uma boa classificação. O tamanho de um dataset ideal depende muito da aplicação. Atualmente existem datasets na ordem de milhões de imagens. É possível encontrar para aplicações simples banco de imagens na ordem de centenas, porém o mais encontrado hoje em dia são datasets na ordem de milhares. (GANDHI, 2021).

Porém estão sendo realizadas diversas pesquisas para entender a quantidade de imagens ideal para o treinamento de uma rede neural. Entre um dos estudos realizados está o desenvolvido por Shahinfar, Meek e Falzon, 2020, em universidades australianas. Foram realizados diversos testes utilizando *Datasets* balanceados, para analisar a relação entre o número de imagens de treinamento e medidas de precisão. A conclusão desta pesquisa é que essa relação tem uma tendência logarítmica. Porém ainda não chegaram a uma conclusão matemática da quantidade ideal.

A organização dos datasets pode ser feita de diversas formas, sempre associando o nome do arquivo de imagem a classe de saída. Os dados podem ser organizados a partir do nome de arquivo, onde o mesmo tem como prefixo a denominação de sua classificação.

Outro modo de organização de *Datasets* é separar cada classe com uma pasta separada dentro de uma pasta-mãe. A Figura 7 exemplifica esses dois modos de organização.

Figura 7 – Maneiras de organizar o Dataset.



Fonte: Do Autor, 2022.

Como mencionado anteriormente os problemas de *Datasets*, fazendo com que a CNN, não tenha boa precisão e acurácia. Com isso foram desenvolvidas técnicas de pré-processamento para o aumento do banco de imagens.

Para realizar o aumento do *Dataset* primeiramente deve-se analisar se realmente é necessário a criação de novas imagens de acordo com a aplicação. Por exemplo, a rede neural deseja classificar carros andando na rua, então não faz sentido rotacionar a imagem em 180°.

Vendo a necessidade de gerar mais imagens, para melhorar a rede neural, é realizado as técnicas de aumento de dados.

2.4 Data Augmentation

As redes neurais convolucionais tem uma propriedade chamada de invariância, o que significa que elas podem classificar objetos em diversas orientações, tamanhos, iluminação entre outras características. (DATA SCIENCE ACADEMY, 2021).

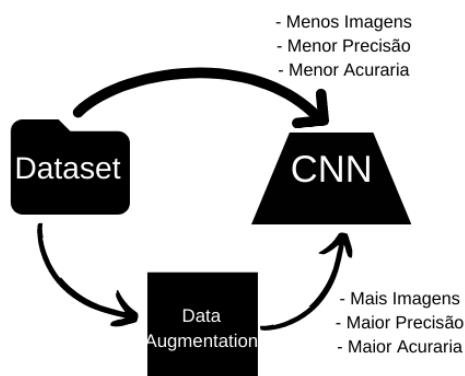
A partir dessa propriedade das CNN's, o aumento de dados é uma técnica que serve para criar artificialmente variações de imagens já existentes para expandir um *Dataset*. (CLARO; VOGADO; SANTOS, VERAS, 2019).

“O *Image Augmentation* cria um conjunto rico e diversificado de imagens a partir de um pequeno conjunto de imagens para classificação de imagens, detecção de objetos ou segmentação de imagens”. (KHANDELWAL, 2019, TRADUÇÃO NOSSA).

Com essa técnica podemos manipular as imagens do *dataset* de diversas maneiras, como colocar zoom, espelhar verticalmente e horizontalmente, criar ruídos, cortar, cisalhar entre outros, sendo possível combinar as técnicas utilizadas.

Imaginando o funcionamento da rede neural, o Data Augmentation atua anterior à entrada das imagens, conforme a Figura 8, ilustra abaixo.

Figura 8 – Atuação do *Data Augmentation* em relação a ordem de aplicação.



Fonte: Do Autor, 2022.

A técnica de aumento de dados é dividida em duas categorias: as transformações geométricas e as transformações fotométricas. No desenvolvimento deste trabalho foram utilizadas em sua grande maioria transformações geométricas. Além da utilização de ruído gaussiano como transformação fotométrica.

2.4.1 Transformações Geométricas

As transformações geométricas são realizadas de forma mais simples que as transformações fotométricas, atuando somente na posição dos pixels. Porém essas transformações são de grande importância. Normalmente esse tipo de transformação é mais utilizada, devido a sua facilidade de compreensão e implementação. Essas transformações atuam diretamente na forma geométrica, ou seja, na orientação, tamanho e/ou formas de objetos, das imagens. (CLARO; VOGADO; SANTOS, VERAS, 2019).

De forma geral, todas as mudanças são realizadas através de valores das coordenadas de cada pixel. Este tipo de transformação é chamado de operação pontual. Nas sessões seguintes serão explicados com detalhes as transformações consideradas no desenvolvimento do trabalho.

Utilizou-se uma imagem clássica de processamento de dados. Esta imagem foi utilizada pela primeira vez em um artigo sobre a transformada Fuzzy para remover ruído de imagens. Nesse artigo a imagem “Cameraman” é considerada a original. A Figura 9 ilustra a imagem original.

Figura 9 – Cameraman.



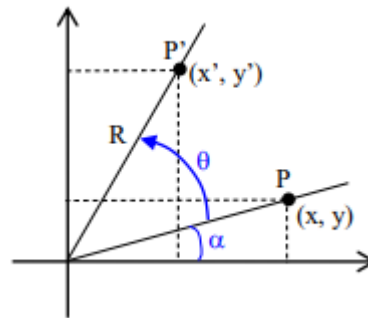
Fonte: ABU-SAMAN; AL-ZERAI, 2016 .

2.4.1.1 Rotação

Assumindo que cada pixel da imagem seja um ponto correspondente da mesma, a rotação basicamente rotaciona cada ponto através das equações abaixo.

A rotação pode ocorrer em relação à origem ou em relação a um ponto arbitrário. Porém nas transformações realizadas com as técnicas de aumento de dados aplicadas é realizado rotações somente a partir da origem. A Figura 10 ilustra a rotação a partir da origem.

Figura 10 – Rotação de um ponto no plano.



Fonte: ARAKAKI, 2018.

A partir do gráfico e de equações provenientes na álgebra, obtemos as Equações 1 e 2 para o eixo x e as Equações 3 e 4 para o eixo y.

$$x = R \cos \alpha \quad (1)$$

$$x' = R \cos(\alpha + \theta) \quad (2)$$

$$y = R \sin \alpha \quad (3)$$

$$y' = R \sin(\alpha + \theta) \quad (4)$$

A partir das propriedades de seno e cosseno para valor do eixo x, temos para o ponto rotacionado x' as Equações 5 e 6:

$$x' = R (\cos \alpha \cdot \cos \theta) - R (\sin \alpha \cdot \sin \theta) \quad (5)$$

$$x' = x \cdot \cos \theta - y \cdot \sin \theta \quad (6)$$

Realizando as mesmas operações do eixo x para o eixo y, obtém-se as seguintes Equações 7 e 8:

$$y' = R (\sin \alpha \cdot \cos \theta) + R (\cos \alpha \cdot \sin \theta) \quad (7)$$

$$y' = x \cdot \sin \theta + y \cdot \cos \theta \quad (8)$$

É possível também escrever essas transformações de forma matricial, onde já é empregada uma matriz de transformação de rotação, proveniente da álgebra linear. É possível visualizar através da Figura 11.

Figura 11 – Equação Matricial de Rotação.

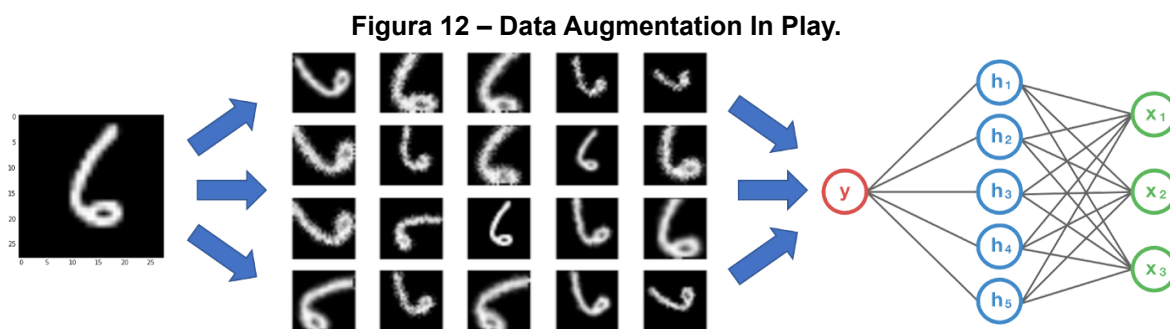
$$P' = P * R$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} * \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

Fonte: Autor, 2022.

A rotação é a técnica onde é possível rotacionar a imagem original em um ângulo arbitrário, aumentando assim a qualidade dos dados. Para aplicar a rotação em uma imagem é preciso analisar qual a faixa relevante de ângulos a serem utilizados, frente aos requisitos da aplicação. (GANDHI, 2021).

Um exemplo a ser citado é na identificação de números manuscritos. Caso seja necessário aplicar um aumento de dados, deve-se atentar aos números 6 e 9. Não é interessante aplicar uma rotação muito grande a esses números. Tendo em vista que se rotacionarem próximos a 90°, eles se confundem. Fazendo com que atrapalhe no treinamento da rede. É possível observar na Figura 12, a rotação do número 6 tendo um limite, para não confundir com o número 9. As Figuras 13a, 13b e 13c ilustram a Figura 9 rotacionada com ângulos de 45°, 135° e 225°, respectivamente.



Fonte: GANDHI, 2021.

Figura 13a – Cameraman rotacionado 45°. 13b - Cameraman rotacionado 135°. 13c - Cameraman rotacionado 225°.



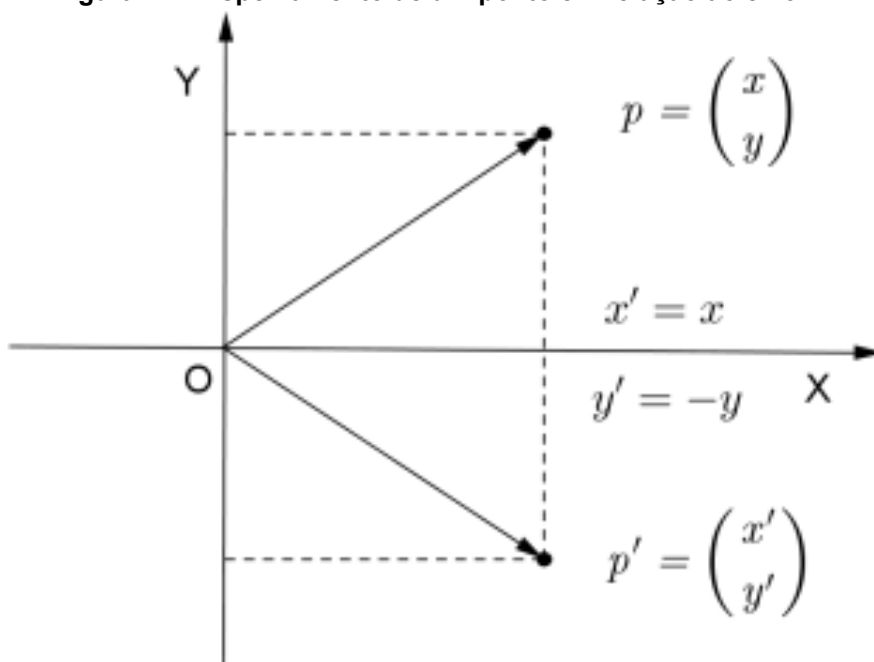
Fonte: Adaptado, ABU-SAMAN.

2.4.1.2 Espelhamento

O Espelhamento ou *flip* ou *flipping* é uma técnica que basicamente inverte a imagem na horizontal ou na vertical.

A transformação de espelhamento pode ser aplicada de diversas maneiras, porém no projeto foram aplicados somente no eixo x e y. O espelhamento em relação ao eixo x, ou na vertical é dado a partir da Figura 14. A matriz de uma transformação no eixo x, definida na álgebra linear, é observado na Figura 15.

Figura 14 – Espelhamento de um ponto em relação ao eixo X.



Fonte: ARAKAKI, 2018.

Figura 15 – Equação matricial do espelhamento em relação ao eixo X.

$$Fx = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$P' = P * Fx$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ -y \end{bmatrix}$$

Fonte: Autor, 2022.

Conclui-se então que para cada pixel da imagem original é invertido o valor `y` do pixel. Espelhando a imagem na vertical, conforme mostra a Figura 16.

Figura 16 – Cameraman espelhado na vertical.

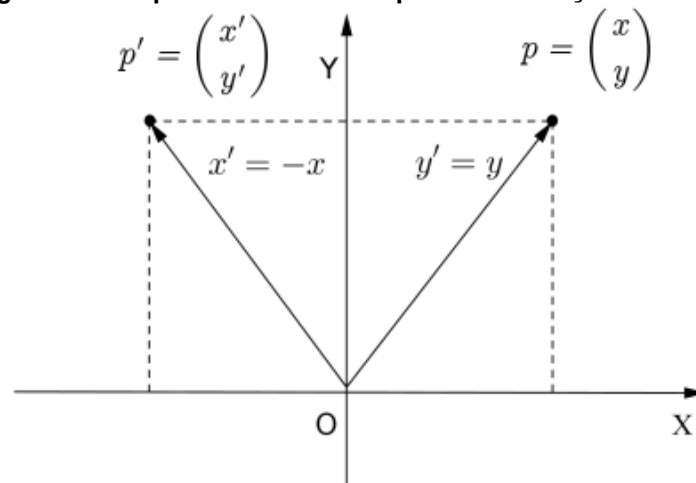


Fonte: Adaptado, ABU-SAMAN.

Já o espelhamento em relação ao eixo `y`, ou na horizontal é dado a partir das Figuras 17 e 18. De maneira geral é muito parecido com o espelhamento no eixo `x`.

Conclui-se então que para cada pixel da imagem original é invertido o valor da coordenada `x` do pixel. Espelhando a imagem na vertical, conforme mostra a Figura 19.

Figura 17 – Espelhamento de um ponto em relação ao eixo Y.



Fonte: ARAKAKI, 2018

Figura 18 – Equação matricial do espelhamento em relação ao eixo Y.

$$F_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$P' = P * F_y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} * \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -x \\ y \end{bmatrix}$$

Fonte: Autor, 2022.

Figura 19 – Cameraman espelhado na horizontal.



Fonte: Adaptado, ABU-SAMAN.

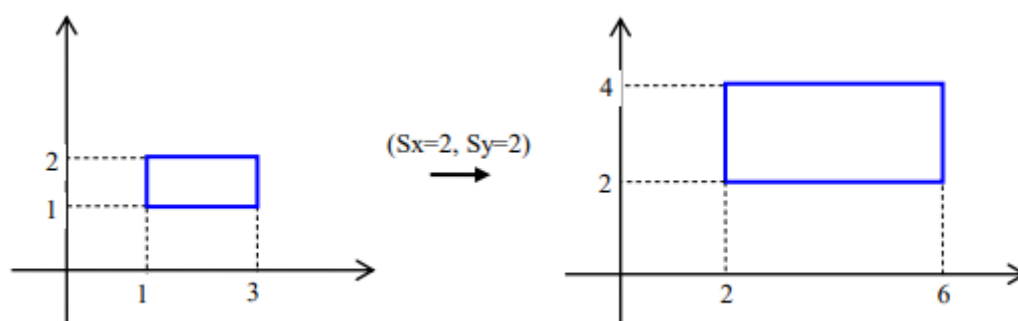
É mais comum encontrar aplicações que utilizam o flip do eixo horizontal do que do eixo vertical, por depender muito da aplicação. Muitas das vezes o espelhamento vertical não é possível utilizá-lo. (CLARO; VOGADO; SANTOS, VERAS, 2019).

Foi mencionado sobre essa técnica o seguinte: “Para alguns objetos, não devemos invertê-los verticalmente, pois a imagem pode mudar completamente”. Essa técnica além de ser de grande importância para o aumento de dados, também é uma de mais simples implementação. (BALA, 2020).

2.4.1.3 Escala

A escala é a técnica do Data Augmentation que amplia ou reduz a imagem original. De forma matemática a escala multiplica o tamanho da imagem original por um fator em 'x' e em 'y'. Conforme mostra a Figura 20.

Figura 20 – Escala de uma imagem no plano.



Fonte: ARAKAKI, 2018.

Ou seja, cada pixel multiplica a sua posição pela matriz de escala, conforme mostra a Figura 21.

Se esse fator 'Sx' e/ou 'Sy' for decimal diminui a imagem e se for valor inteiro amplia a imagem.

Figura 21 – Equação Matricial da Escala.

$$P' = S * P$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x * x \\ S_y * y \end{bmatrix}$$

Fonte: Autor, 2022.

Quando a escala aumenta o tamanho das imagens, as estruturas automaticamente cortam para ficar do mesmo tamanho da original. Porém, quando reduz a figura, existem duas opções: a imagem gerada diminuir de tamanho, ou complementar a imagem gerada com outras informações para ficar do tamanho original. (GANDHI, 2021).

“A escala adiciona novos valores de pixel ao redor da imagem ou interpola os valores de pixel, dependendo do que se deseja. A quantidade de escala é colocada de forma uniforme e aleatória para cada dimensão (largura e altura) separadamente.” (BROWNLEE, 2019, TRADUÇÃO NOSSA).

As Figuras 22a, 22b, 22c e 22b ilustram o resultado da imagem que passou pelo método de escala para o aumento de dados. Essa escala pode ser aplicada na horizontal, ou na vertical, ou em ambos.

Figura 22a – Cameraman com escala 2 na horizontal. 22b - Cameraman com escala 2 na vertical. 22c - Cameraman com escala 0.5 na horizontal. 22d - Cameraman com escala 0.5 na vertical.



Fonte: Adaptado, ABU-SAMAN.

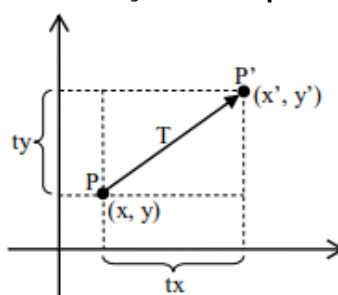
2.4.1.4 Translação

A translação basicamente reposiciona o objeto no plano. Cada pixel altera o valor da sua posição em `x` e em `y`, através de um vetor de translação `T` e a

Figura 23, demonstra como é feita a transformação de forma matemática. (ARAKAKI, 2018).

É realizada a soma de cada coordenada do vetor 'T' a sua correspondente do valor original. Também é possível realizar através da equação matricial, conforme mostra as equações na Figura 24. O resultado da translação é possível ser visualizado na Figura 25.

Figura 23 – Translação de um ponto no plano.



Fonte: ARAKAKI, 2018.

Figura 24 – Equação matricial de translação.

$$P' = P + T$$

$$x' = x + Tx$$

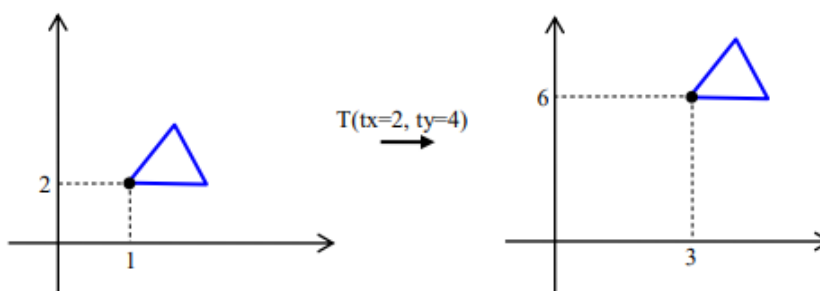
$$y' = y + Ty$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} Tx \\ Ty \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + Tx \\ y + Ty \end{bmatrix}$$

Fonte: Autor, 2022.

Figura 25 – Resultado da translação de uma figura no plano.



Fonte: ARAKAKI, 2018.

A translação é definida assim:

“Translação é o ato de transferir para outro lugar, ou seja, a técnica de translação muda as imagens para a esquerda, direita, para cima ou para baixo pode ser uma transformação muito útil para evitar distorções posicionais nos dados. Por exemplo, se todas as imagens em um conjunto de dados estiverem centralizadas, o que é comum em conjuntos de dados de reconhecimento de face, isso exigiria que o modelo também fosse testado em imagens perfeitamente centralizadas.

Como a imagem original é traduzida em uma direção, o espaço restante pode ser preenchido com um valor constante, como 0 ou 255, ou pode ser preenchido com ruído aleatório ou gaussiano. Esse preenchimento preserva as dimensões espaciais da imagem após-aumento.” (SHORTEN e KHOSHGOFTAAR, 2019. TRADUÇÃO NOSSA).

Esse método de aprimoramento é muito útil, pois a maioria dos objetos pode ser localizada em praticamente qualquer lugar da imagem. Isso força a rede neural convolucional a procurar em todos os lugares da imagem. (CLARO; VOGADO; SANTOS, VERAS, 2019).

As Figuras 26a, 26b, 26c e 26d ilustram a imagem do cameraman transladadas em 20% em cada sentido.

Figura 26a – Cameraman transladada 20% para direita. 26b - Cameraman transladada 20% para baixo. 26c - Cameraman transladada 20% para esquerda. 26d - Cameraman transladada 20% para cima.

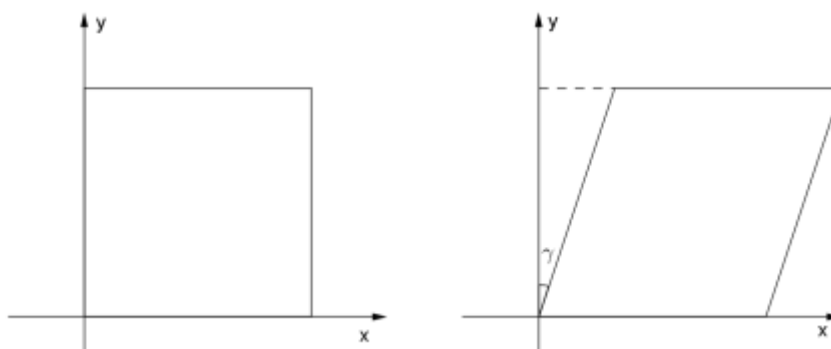


Fonte: Adaptado, ABU-SAMAN.

2.4.1.5 *Cisalhamento*

A equação matemática da transformação de cisalhamento se dá a partir da Figura 27.

Figura 27 – Cisalhamento de uma imagem.



Fonte: ARAKAKI, 2018.

Onde γ é o ângulo de deslocamento em relação ao eixo y e a matriz de cisalhamento proveniente da álgebra. A equação matricial do cisalhamento é vista na Figura 28.

Figura 28 – Equação matricial do cisalhamento.

$$C = \begin{bmatrix} 1 & \operatorname{tg} \gamma \\ 0 & 1 \end{bmatrix}$$

$$P' = C * P$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \operatorname{tg} \gamma \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + y \operatorname{tg} \gamma \\ y \end{bmatrix}$$

Fonte: Autor, 2022.

Para definir a transformação de cisalhamento

“A transformação de cisalhamento inclina a forma da imagem. Diferentemente da rotação no sentido de que, na técnica de cisalhamento, fixamos um eixo e esticamos a imagem em um determinado ângulo conhecido como ângulo de cisalhamento.” (CLARO; VOGADO; SANTOS; VERAS, 2019).

Isso cria um tipo de alongamento na imagem, diferentemente do que ocorre na rotação. Para melhor visualizar o funcionamento desse tipo de transformação, as Figuras 29a e 29b ilustram a imagem Cameraman no processo de cisalhamento de 45° .

Figura 29a – Cameraman Cisalhamento 45°. 29b - Cameraman Cisalhamento -45°.



Fonte: Adaptado, ABU-SAMAN.

2.4.1.6 Corte

O corte é uma técnica de DA, na qual foi criado um subconjunto aleatório da imagem original, ajudando o modelo porque os objetos de interesse nem sempre estão totalmente visíveis na imagem ou na mesma escala. (NELSON, 2021, TRADUÇÃO NOSSA).

“Ao contrário da escala e da translação, o corte retira um pedaço da imagem e amplia deixando do tamanho da imagem original.” (RAJ, 2018, TRADUÇÃO NOSSA).

A seguir as Figuras 30a, 30b e 30c mostram o corte realizado na imagem do cameraman. É possível observar que essa técnica realmente corta um pedaço da figura original. O corte é possível observar que é um pouco diferente da translação e da escala, pois são retirados os pixels que existem e aumenta ao tamanho original da imagem.

Figura 30a – Cameraman corte realizado 1. 30b - Cameraman corte realizado 2. 30c - Cameraman corte realizado 3.



Fonte: Adaptado, ABU-SAMAN.

2.4.2 Transformações Fotométricas

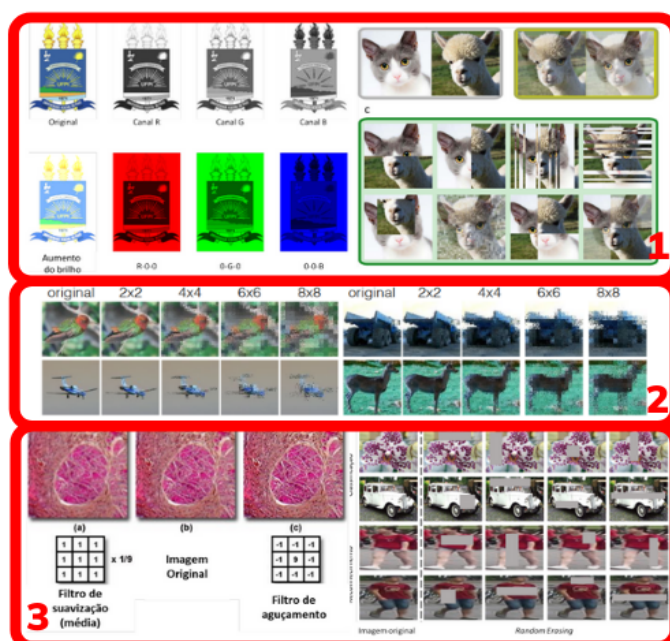
Como mencionado anteriormente, tem-se dois tipos de transformações possíveis de realizar em uma imagem: as geométricas e as fotométricas. Diferentemente das transformações geométricas, as transformações fotométricas atuam sobre a composição pictórica.

Essas transformações são aplicadas de diversas maneiras, conforme mencionado a seguir.

[...]são aplicadas sobre o valor armazenado nos pixels e em alguns casos a substituição parcial de conteúdo são encontradas em abordagens como adição de ruído, mudanças no espaço de cor e processamentos de imagem, filtragem com kernel, combinação de imagens e random erasing. (CLARO; VOGADO; SANTOS; VERAS, 2019).

Neste projeto foi utilizado somente adição de ruído, logo abaixo é possível observar na Figura 31 algumas das possibilidades de transformações fotométricas. A parte 1 da figura ilustra as mudanças nos canais de cor RGB, aumento de brilho e também a união de duas imagens. A segunda parte mostra como atuam dois tipos de ruídos. Na parte 3 mostra a técnica de *random_erasing*, na qual apaga de forma aleatória uma parte da imagem original e também as de borramento e aguçamento.

Figura 31 - Transformações Fotométricas.



Fonte: Adaptado, CLARO; VOGADO; SANTOS; VERAS.

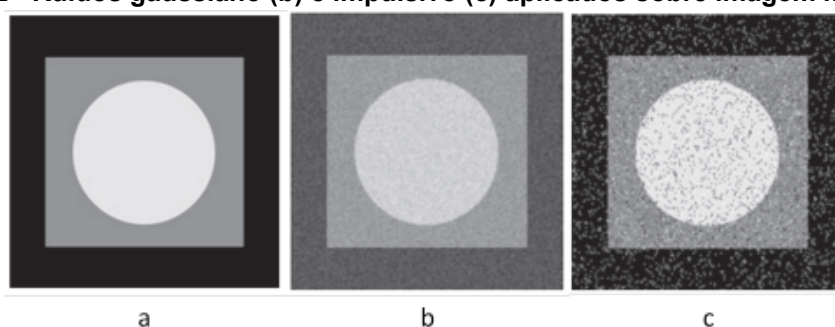
2.4.2.1 *Adição de Ruído*

A definição de adição de ruído está abaixo.

Ruídos em imagens são caracterizados por valores aleatórios em pixels. Geralmente são provocados por fatores relacionados ao processo de obtenção da imagem. Em imagens analógicas, ruídos podem ser causados pelo próprio filme (aspereza da superfície), já em imagens digitais, variações de temperatura, falhas no sensor, entre outros, podem causar esse efeito. (CLARO; VOGADO; SANTOS; VERAS, 2019).

A adição de ruído em imagens pode ajudar as CNNs a aprender características mais robustas. Existem diversos tipos de ruído como o gaussiano, sal e pimenta, mancha, impulsivo entre diversos outros. A Figura 32 exemplifica dois tipos de ruídos comuns: o gaussiano e o impulsivo.

Figura 32 - Ruídos gaussiano (b) e impulsivo (c) aplicados sobre imagem modelo (a).



Fonte: CLARO; VOGADO; SANTOS; VERAS, 2019.

2.4.2.2 *Ruído Gaussiano*

De forma geral e resumida o ruído gaussiano, é um ruído estatístico cuja função densidade de probabilidade é igual a da distribuição normal. Uma imagem com ruído gaussiano, basicamente soma os valores de pixel original mais um valor aleatório dentro da função gaussiana. O ruído branco normalmente é o mais utilizado em diversas aplicações. (LENDAVE, 2021).

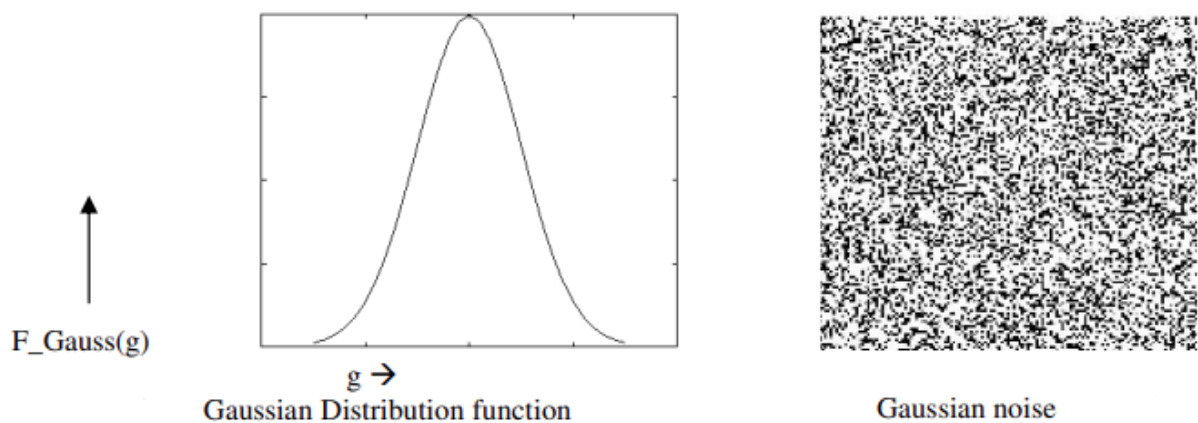
A função densidade P de uma variável aleatória gaussiana z é dada pela Equação 9:

$$P(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (9)$$

Sendo, “ μ ” a média da distribuição normal, “ σ ” o desvio padrão e “ z ” o tom de cinza. Essas variáveis é onde o software irá definir os valores para aplicar assim o ruído gaussiano.

A Figura 33 ilustra a função gaussiana e a representação dos pixels do ruído gaussiano. E em seguida a Figura 34, ilustra a distorção causada pela adição de ruído, com desvio padrão 1 e tom de cinza igual a 255.

Figura 33 – Função Gaussiana.



Fonte: LENDAVE, 2021.

Figura 34 – Cameraman com desvio padrão 1 e tom de cinza 255.



Fonte: Adaptado, ABU-SAMAN.

2.5 Ferramentas Úteis

Nesse tópico são apresentadas as ferramentas utilizadas para auxiliar no desenvolvimento do projeto.

2.5.1 Ambiente de Desenvolvimento e Software

Foi utilizado para realizar o desenvolvimento deste projeto o ambiente de desenvolvimento Pycharm.

Segundo o próprio site da empresa JetBrains desenvolvedora do Pycharm “O Pycharm é uma IDE multiplataforma com versões para todos os sistemas operacionais.” (JET BRAINS, [entre 2000 e 2022].).

Desenvolvido pela empresa JetBrains o software foi feito para de todas as formas ajudar e auxiliar o desenvolvedor. Entre suas principais características estão: assistência inteligente em Python, Frameworks de desenvolvimento online, ferramentas científicas, desenvolvimento multitecnologia, recursos de compartilhamento e desenvolvimento a distância e tem grande coleção de ferramentas já incorporadas. (JET BRAINS, [entre 2000 e 2022].).

2.5.2 Linguagem de Programação

A definição da linguagem de programação *Python*, além de suas vantagens, é vista abaixo.

“[...]Python é uma linguagem de programação de alto nível dinâmica, interpretada, modular, multiplataforma e orientada a objetos. Por ser uma linguagem de sintaxe relativamente simples e de fácil compreensão, vem crescendo em popularidade entre engenheiros, matemáticos, cientistas de dados, pesquisadores, entre outros. Uma das suas principais vantagens e que atrai mais pessoas, é possuir um grande número de bibliotecas nativas ou de terceiros, tornando-a bastante difundida e útil em vários setores como no desenvolvimento web, análise de dados, *machine learning* e IA.[...]” (ROVEDA, 2020).

Além das vantagens citadas acima, a linguagem *Python* é bastante utilizada por ser possível encontrar diversas bibliotecas para resolver praticamente

todos os desafios. É possível encontrar diversas bibliotecas comumente utilizadas nas CNN's, como por exemplo o *Pytorch* e *Pandas*. (INSIGHT, 2019).

2.5.3 Bibliotecas Utilizadas

Durante o desenvolvimento observou-se a necessidade de utilizar algumas bibliotecas da linguagem *Python*, pois com elas facilitam e simplificam a codificação. Neste tópico é apresentado e explicado sobre como funcionam cada uma das bibliotecas utilizadas.

2.5.3.1 *OpenCV*

A OpenCV é uma biblioteca de código aberto que inclui várias centenas de algoritmos de visão computacional. Esta biblioteca possui as funções necessárias para realizar abertura e salvamento de imagens nos formatos comumente utilizados como PNG, BMP e JPG. Na plataforma da própria biblioteca é possível observar os diversos métodos e funções que a mesma contém. (OPENCV TEAM, 2022).

2.5.3.2 *Imgaug*

A biblioteca é conceituada de forma geral abaixo.

“[...] *imgaug* é uma biblioteca para aumento de imagens em experimentos de aprendizado de máquina. Ele suporta uma ampla gama de técnicas de aumento, permite combiná-las facilmente e executá-las em ordem aleatória ou em vários núcleos de CPU, possui uma interface estocástica simples e poderosa e pode não apenas aumentar imagens, mas também pontos-chave / marcos, caixas delimitadoras, mapas de calor e mapas de segmentação.”

De forma geral são com os métodos dessa biblioteca que é realizado as técnicas de Data Augmentation. Na tabela 3 é possível verificar a função da biblioteca e seus parâmetros para cada transformação aplicada.

Tabela 3 – Transformações da Biblioteca *Imgaug*

Transformação aplicada	Método	Ação desempenhada	Parâmetros
Sequência de métodos	<i>iaa.Sequential</i>	Lista de outros métodos.	Parâmetro importante <i>random_order</i> .
Espelhar Imagem	<i>iaa.Flip</i>	<i>iaa.Fliplr</i> espelha imagem horizontalmente.	p - probabilidade de espelhar a imagem
		<i>iaa.Flipud</i> espelha a imagem verticalmente.	
Cortar Imagem	<i>iaa.Crop</i>	Extraí sub imagens de tamanho menor de determinadas imagens de entrada de tamanho normal.	<i>percent</i> - indica o número de pixels a serem cortados em cada lado da imagem dado como uma fração da altura/largura da imagem.
Adicionar Ruído Gaussiano	<i>iaa.AdditiveGaussianNoise</i>	Adicionar Ruído Gaussiano	loc - equivalente ao “ μ ” da fórmula do ruído gaussiano. <i>scale</i> - desvio padrão.
Aplicar transformações geométricas.	<i>iaa.Affine</i>	Aplica Escala	<i>scale</i> - valor em decimal para cada eixo
		Aplica translação	<i>translate_percent</i> - valor em decimal para cada eixo
		Aplica rotação	<i>rotate</i> - menor e o maior valor do ângulo
		Aplica Cisalhamento	<i>shear</i> - menor e o maior valor do ângulo

Fonte: Do Autor.

2.5.3.3 Tkinter

Tkinter é a biblioteca de interface padrão da linguagem *Python* e é definida pela própria empresa como.

“O pacote Tkinter é a interface padrão do Python para o kit de ferramentas Tcl/Tk GUI. Executar linhas de comando Tk pode desenvolver uma interface simples. Por mais que não tenha um toque fino, mas adiciona uma boa quantidade de sua própria lógica para tornar uma experiência mais nítida.” (Python Software Foundation, [entre 2001 - 2022], TRADUÇÃO NOSSA).

Essa biblioteca tem duas principais vantagens. Por ser um pacote nativo *Python* tem diversos recursos. Somado a isso é possível verificar sua facilidade de implementação por ter somente que incluir a biblioteca e utilizá-la. Além dessas duas principais vantagens é também um *framework* multiplataforma.

2.6 Trabalhos Correlatos

É possível observar que diversos trabalhos e artigos exploram essa técnica de aumento de dados de diversas maneiras. Porém é encontrado redes neurais onde tenha implementado junto a técnica de aumento de dados. Não contendo um sistema desenvolvido focado para o aumento de dados.

De forma geral são encontradas diversas aplicações utilizando a técnica de aumento de dados, como para detecção de Covid-19 em imagens de raio-x desenvolvidos por RICCI, SANTOS, OTTONI, 2021. Assim como para inspeção do vagão ferroviário por ROCHA, 2020. Ou também no Desenvolvimento de uma Base de Dados de Sinais de Libras para Aprendizado de Máquina: Estudo de Caso com CNN 3D, desenvolvido por CASTRO et. al. 2019. Além da aplicação desenvolvida na monografia desenvolvida por OSHIRO, 2022. Na qual utilizou as técnicas de data augmentation para a Detecção e Classificação de Equipamentos em Torres de Telecomunicação Através de Redes Neurais Convolucionais.

Também é possível observar a existência do *Journal of Imaging*, que segundo o próprio é “O Journal of Imaging é um periódico internacional, multi/interdisciplinar, revisado por pares e de acesso aberto de técnicas de imagem, publicado online mensalmente pelo MDPI.” De modo geral este jornal traz artigos acadêmicos sobre tratamentos de imagem. (MDPI, JOURNAL OF IMAGING, [entre 2015 e 2022]. TRADUÇÃO NOSSA)

Por fim, existem artigos que comparam diferentes abordagens de aumento de dados como o de Comparação entre diferentes abordagens de aumento de dados em imagens, por NANNI et al. 2021. Neste artigo é comparado 2 abordagens de transformadas de imagens a de Gabor e a Wavelet. Em cada caso

foram utilizados 11 métodos distintos de aumento de dados aplicados a uma rede ResNet pré-treinada, com bastante *overfitting*.

3 Desenvolvimento de um sistema de criação de bases de imagens através de aumento de dados

Neste capítulo é apresentado o processo de desenvolvimento de um sistema de criação de bases de imagens através do aumento de dados.

3.1 Discussão sobre os requisitos do sistema

A primeira etapa para realizar o desenvolvimento do sistema foi definir requisitos funcionais que deveriam ser atendidos. Com os requisitos definidos foi possível pensar em técnicas e ações para atender cada um deles. Os requisitos funcionais do sistema foram levantados com base nas demandas identificadas em trabalhos da literatura e ficaram definidos de tal forma:

- Gerar imagens através de transformações geométricas;
- Gerar imagens através da adição de ruído;
- Gerar imagens através de corte e cisalhamento.

Além dos requisitos funcionais foram definidos os seguintes requisitos não funcionais:

- O sistema deve possuir uma interface gráfica com o usuário para que a sua utilização seja facilitada;
- O sistema deve ser parametrizável de modo que o usuário possa especificar a natureza das transformações a serem realizadas na geração de imagens.

3.2 Ferramentas selecionadas para atender cada requisito

Com o objetivo de gerar imagens através de transformações geométricas e fotométricas, foi selecionada a linguagem de programação Python com a biblioteca OpenCV e Imgaug. Tais bibliotecas disponibilizam estruturas de dados e funções para a manipulação de imagens e realização de transformações, sem que as mesmas precisem ser implementadas e, desta forma, que o trabalho de desenvolvimento pudesse ser concentrado na criação da ferramenta com interface

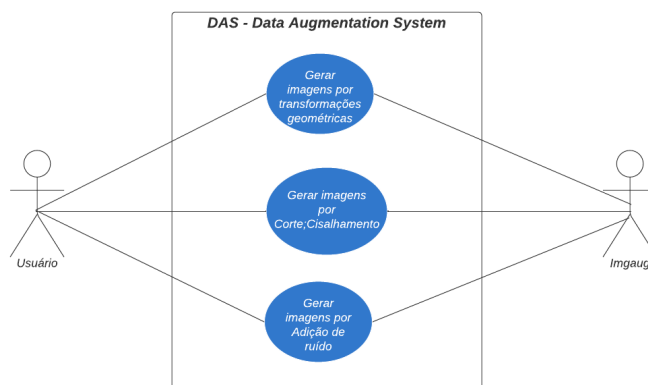
gráfica do usuário. A biblioteca OpenCV, descrita na seção 2.5.1.1, manipula os arquivos das imagens e a biblioteca Imgaug, descrita na seção 2.5.1.2, realiza de fato as transformações.

Logo em seguida, com a compreensão das técnicas a serem utilizadas para manipular as imagens, foram avaliadas possíveis soluções para criar uma interface simples, onde o usuário informe os parâmetros, aperte um botão de geração e sejam então geradas e salvas as imagens no diretório informado pelo mesmo. Para a criação da interface gráfica com o usuário foi selecionada a biblioteca Tkinter. Tal biblioteca permite a criação do layout de tela contendo os controles gráficos que permitem ao usuário definir os parâmetros envolvidos. Além disso, também permite que sejam implementados os eventos que disparam o processamento das imagens.

3.3 Descrição da arquitetura do sistema

O sistema foi desenvolvido utilizando uma organização dos módulos baseada em orientação a objetos. Diagramas de caso de uso são utilizados para descrever os aspectos estáticos e dinâmicos principais do sistema. Na Figura 35 é possível visualizar um diagrama de caso de uso, que visa mostrar de modo geral como ocorre a interação do usuário com o sistema. Entende-se que o mesmo deve solicitar a realização das transformações, enquanto o sistema deve se comunicar com a biblioteca e gerar as imagens com a transformação escolhida.

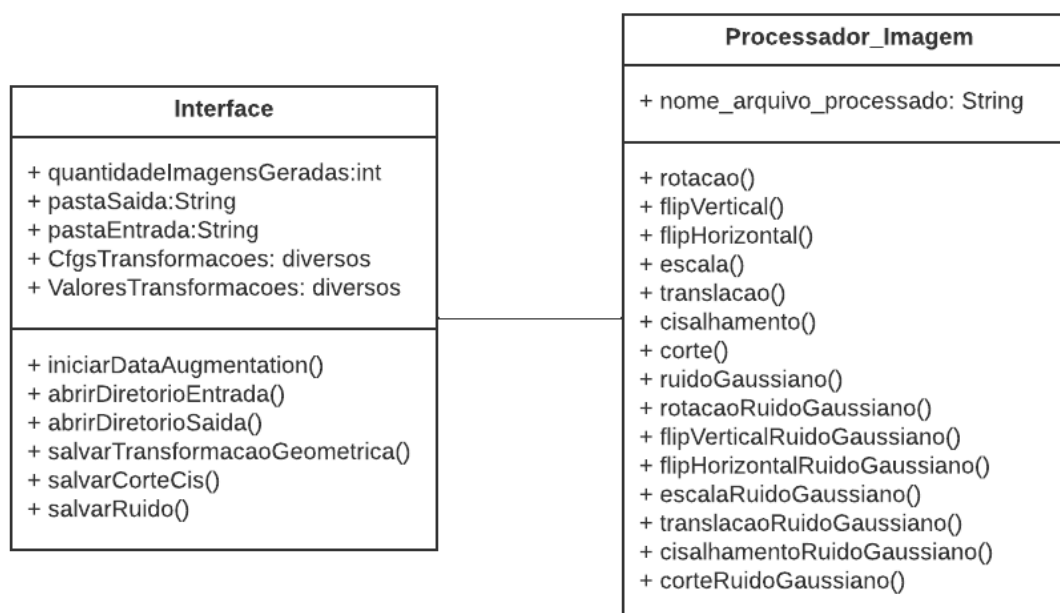
Figura 35 – Diagrama de caso de uso do sistema de aumento de dados.



Fonte: Do Autor. (2022).

De modo geral foram consideradas para realizar o desenvolvimento duas classes principais do sistema. Essas duas classes são Interface Gráfica e Processador de Imagem. Na classe Interface Gráfica são implementadas todas as rotinas e especificações da interface gráfica com o usuário. Por sua vez, a classe Processador de Imagem possui a implementação das rotinas de transformações de imagens. O desacoplamento entre Interface e Processamento das imagens permite que no futuro seja possível migrar mais facilmente o sistema para outro tipo de interface, como por exemplo, Web. Com a definição das classes foi desenvolvido um diagrama de classe ilustrado na Figura 36 para melhor visualização das variáveis do sistema.

Figura 36 – Diagrama de classes para o sistema de aumento de dados.



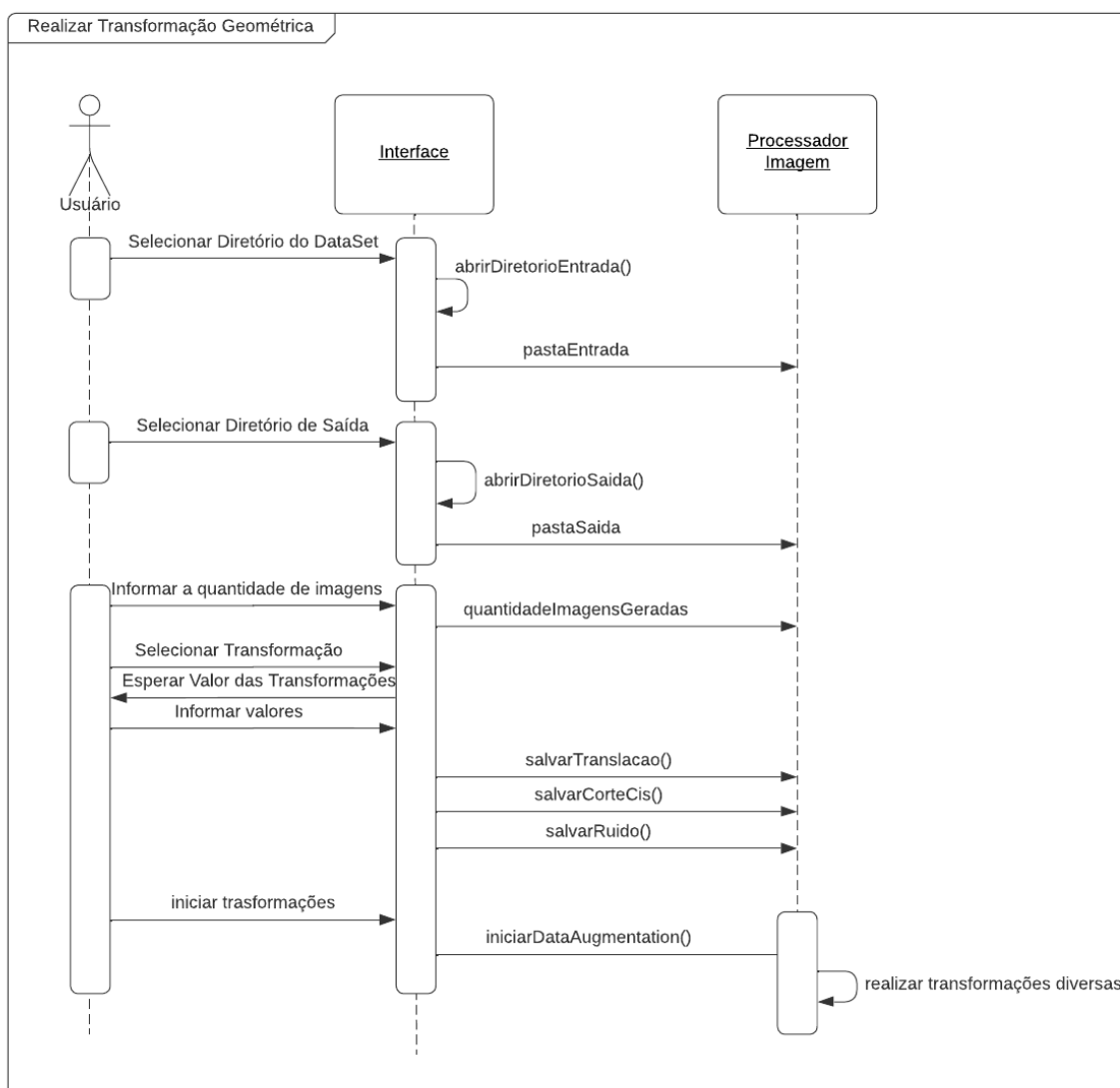
Fonte: Do Autor. (2022).

Para cumprir os requisitos do projeto de maneira geral, são necessárias diversas variáveis e funções. Na interface é onde são selecionadas as configurações das transformações, enquanto o processador de imagem executa a função de realizá-las.

Na Figura 37, é apresentado o diagrama de sequência da rotina principal do sistema, associada a geração da base de imagens aumentada. Inicialmente, o usuário define na interface os parâmetros. Tal informação propagada do objeto da interface ao objeto de processador de imagem. O Processador de Imagens aloca em

memória cópias das imagens originais presentes na pasta de origem e aplica em sequência cada uma das transformações selecionadas. Em seguida tais imagens são salvas na pasta de destino previamente definida.

Figura 37 – Diagrama de sequência do sistema de aumento de dados.



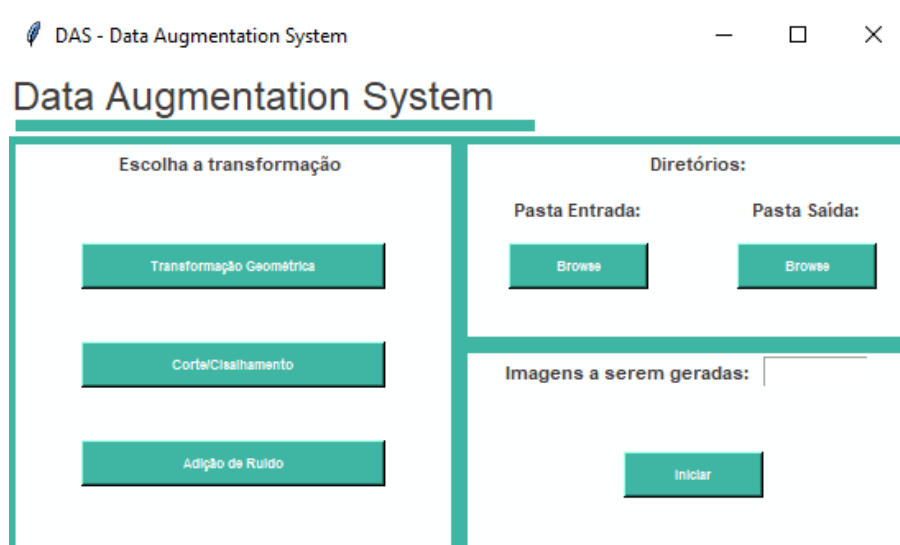
Fonte: Do Autor.

3.3.1 Interface gráfica

A interface gráfica do usuário foi organizada em várias telas para simplificar sua utilização. Dentro destas telas existem subdivisões de forma que os campos a serem preenchidos pelo usuário estejam agrupados em funções das transformações possíveis. A Figura 38 ilustra a tela Home, onde são solicitadas as informações gerais, como quantidade de imagens a serem geradas, pasta onde fica

o DataSet de Entrada e o pasta de Saída. Além disso, o usuário deve informar quantas imagens serão geradas a partir de cada uma existente dentro do diretório. É possível visualizar também uma outra região da tela onde existem três botões: Transformações geométricas, Corte/Cisalhamento e Adição de ruído. Esses botões abrem as telas onde são parametrizadas as transformações a serem realizadas.

Figura 38 – Tela home Data augmentation system.



Fonte: Do Autor.

A Figura 39 mostra a tela de transformações geométricas. Nessa tela é possível encontrar 4 versões diferentes uma para cada transformação. Para cada transformação existe um tipo de configuração específica. Para o espelhamento existem duas caixas de seleção para especificar se a transformação vertical ou horizontal, um para vertical e outro para horizontal, e no restante das transformações são encontrados botões para realizar uma escolha única uma dessas transformações. Quando selecionado a margem muda de cor e mostra onde devem ser preenchidos os parâmetros.

Na tela de Corte/Cisalhamento ilustrada na Figura 40 é possível ver mais duas opções que seriam a de corte e de cisalhamento. Assim como na tela de transformações é possível selecionar somente uma opção. Com a opção selecionada é possível preencher com os valores desejados para a transformação escolhida.

Figura 39 – Tela de Transformações Geométricas.

Seleção a transformação:

☐ Rotação Ângulo Mínimo: 0 Ângulo Máximo: 0

☐ Flip Vertical ☐ Flip Horizontal

☐ Escala Mínimo Escala (X): 0 Máximo Escala (X): 0
Mínimo Escala (Y): 0 Máximo Escala (Y): 0

☐ Translação Mínimo Transl. (X): 0 Máximo Transl. (X): 0
Mínimo Transl. (Y): 0 Máximo Transl. (Y): 0

Fonte: Do Autor.

Figura 40 – Tela de corte e cisalhamento do sistema de aumento de dados

Corte e Cisalhamento

☐ Cisalhamento Ângulo Mínimo: 0 Ângulo Máximo: 0

☐ Corte Relação Altura/Largura: 0

Fonte: Do Autor.

Já na tela de Adição de Ruído exibida na Figura 41 encontra-se uma caixa de seleção onde é possível escolher se deseja ou não a adição de ruído gaussiano nas imagens a serem geradas. Por fim, pode-se preencher os valores de desvio padrão e o tom de cinza entre 0 e 100% de 255.

Então é possível obter, uma análise combinatória com 96 casos diferentes de transformações, devido a possibilidade de escolha de cada parâmetro existente. Para cada uma dessas possibilidades são realizadas as funções escolhidas na classe Processadora de imagem.

Figura 41 – Tela de adição de ruído.

The screenshot shows a window titled 'DAS - Data Augmentation System'. Inside, there's a section titled 'Data Augmentation System' with a subtitle 'Adição de Ruído'. Below this, there's a checkbox labeled 'Ruído Gaussiano'. To its right are two input fields: 'Desvio Padrão:' with the value '0' and 'Valor Ruído:' with the value '0'.

Fonte: Do Autor.

3.3.2 Descrição da utilização do sistema

Para utilizar o sistema basta abrir o arquivo executável do mesmo. Na tela inicial, é necessário selecionar as pastas de entrada dos arquivos de imagem e de saída através dos botões ilustrados na Figura 38, do tópico anterior. Em seguida deve-se selecionar as transformações desejadas e preencher os valores dos parâmetros conforme a Tabela 4.

Tabela 4 – Parâmetros das transformações de utilização do sistema.

Transformação Aplicada	Parâmetro	Explicação	Dados Aceitáveis
Rotação	Ângulo Mínimo Ângulo Máximo	Ângulo mínimo de rotação da imagem. Ângulo máximo de rotação da imagem	Valores Inteiros.
Espelhamento Vertical	Ocorre sempre que selecionado	-	-
Espelhamento Horizontal	Ocorre sempre que selecionado.	-	-
Escala	Min X. Max X Min Y. Max Y.	Min X e Min Y fator de multiplicação mínima de escala em X e em Y em porcentagem. Max X e Max Y fator de multiplicação máximo de escala em X e em Y em porcentagem.	Conjunto dos números Naturais. Sendo valores de 0 a 100 valores que diminuem a imagem e valores de 101 em diante ampliadores da imagem
Translação	Min X Max X Min Y Max Y	Min X e Min Y valor mínimo e máximo do vetor de translação em porcentagem nos	Conjunto dos números inteiros. Sendo os valores negativos deslocam a imagem

		eixos. Máx X e Máx Y valor máximo e máximo do vetor de translação em porcentagem nos eixos.	para a esquerda no eixo X e para baixo no eixo Y e valores positivos no sentido contrário. Recomenda-se a utilização de valores entre -100 e 100.
Cisalhamento	Ângulo Min: -45 Ângulo Max: 45	Ângulo mínimo de cisalhamento da imagem. Ângulo máximo de cisalhamento da imagem	Conjunto dos números inteiros.
Corte	Relação Altura/Largura: 10	Proporção de corte entre Altura e Largura da imagem	Conjunto dos números naturais
Ruído Gaussiano	Desvio Padrão: 1 Valor do Ruído: 1	Desvio Padrão da curva do ruído gaussiano. O valor do ruído corresponde a porcentagem do tom de cinza desejado.	Conjunto dos números Naturais até 100. Esse valor multiplica por 255 que é o quanto varia o tom de cinza

Fonte: Do Autor

Com as transformações preenchidas é necessário escolher a quantidade de imagens a serem geradas a partir de cada imagem já existente na base original. A partir das informações preenchidas o sistema aplica as transformações de forma aleatória dentro dos valores desejados. Os arquivos suportados para realização do aumento de dados são: png, jpg, jpeg, bmp.

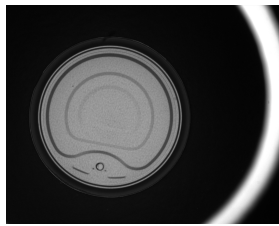
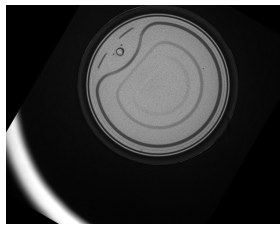
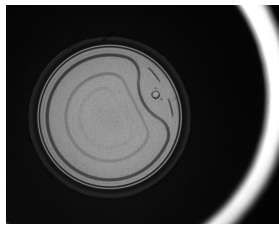
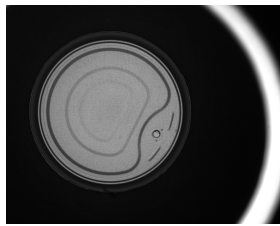
4 Apresentação dos Resultados

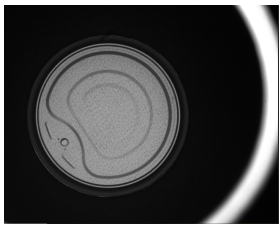
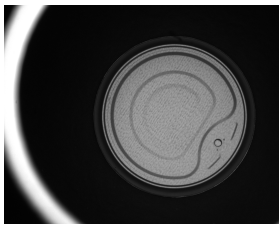
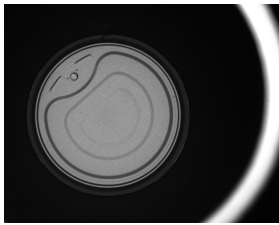
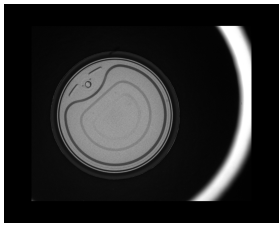
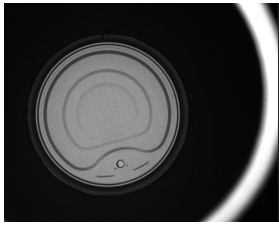
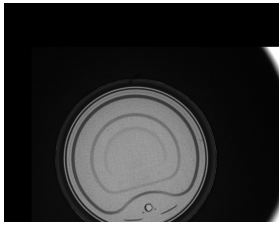
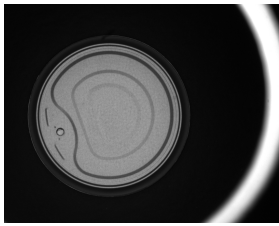
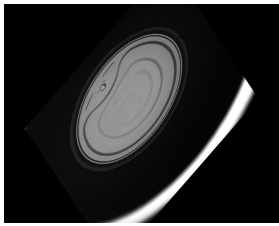
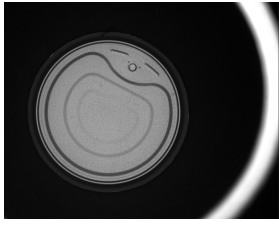
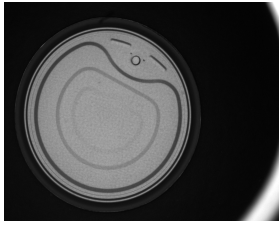
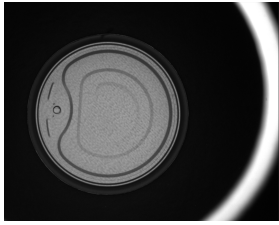
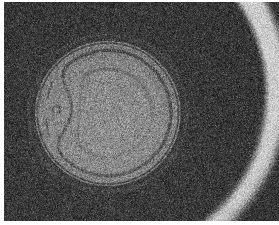
Foram realizados testes utilizando o sistema para verificar se o mesmo cumpriu os requisitos que orientaram o seu desenvolvimento.

4.1 Testes e resultados

Esses testes foram realizados inicialmente com o *Dataset* de Rebites usado na indústria. Tal base de imagens deriva de um projeto de pesquisa onde o objetivo principal foi detectar rebites defeituosos presentes em uma linha de produção de enlatados. A base é composta por 158 imagens do fundo das tampas de latas de metal. (STIVANELLO; MARCELLINO, 2019). Na Tabela 5 é ilustrado o resultado das transformações propostas em relação à imagem original. Também é possível realizar uma combinação entre algumas das transformações. O sistema foi configurado para gerar uma base de imagens aumentadas. Partindo das 158 imagens da base original foram geradas 2370 imagens através das transformações disponibilizadas pelo sistema combinando as transformações.

Tabela 5 – Teste das transformações realizadas através do sistema no banco de dados de Rebites.

Transformações Realizadas			
Transformação Aplicada	Parâmetros Utilizados	Imagem Original	Imagem Transformada
Rotação	Min: 45° Máx: 135°		
Espelhamento Vertical	100% das vezes irá ocorrer a transformação.		

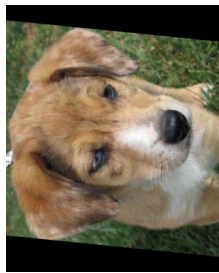
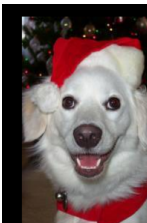
Espelhamento Horizontal	100% das vezes irá ocorrer a transformação.		
Escala	Min X: 80% Max X: 80% Min Y: 80% Max Y: 80%		
Translação	Min X: 10% Max X: 10% Min Y: 20% Max Y: 20%		
Cisalhamento	Ângulo Min: -45° Ângulo Max: 45°		
Corte	Relação Altura/Largura: 10%		
Ruído Gaussiano	Desvio Padrão: 1 Porcentagem do tom de cinza: 100%		

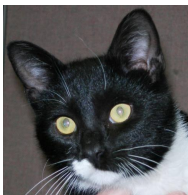
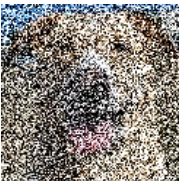
Fonte: Do Autor

Além da base de dados de rebites, foi utilizada a base de dados, bastante conhecida da literatura de visão artificial, Dogs vs. Cats. Esta base de dados tem por objetivo estudo de visão artificial para validar um HIP (Prova Interativa Humana), utilizando o próprio computador. Disponibilizado pelo ASIRRA, organização para

reconhecimento de imagem de espécies animais para restrição de acesso, o banco de dados contém 12500 de imagens de cachorros e gatos em diversas raças e posições. A partir do sistema desenvolvido, foram geradas 25000 imagens aplicando todas as transformações disponibilizadas no sistema. Abaixo, a Tabela 6 ilustra algumas das transformações aplicadas.

Tabela 6 – Teste das transformações realizadas no banco de dados de Dogs vs. Cats.

Transformações Realizadas			
Transformação Aplicada	Parâmetros Utilizados	Imagem Original	Imagem Transformada
Rotação	Min: -90° Máx: 90°		
Espelhamento Vertical	100% das vezes irá ocorrer a transformação.		
Espelhamento Horizontal	100% das vezes irá ocorrer a transformação.		
Escala	Min X: 50% Max X: 200% Min Y: 50% Max Y: 200%		
Translação	Min X: - 20% Max X: 20% Min Y: - 20% Max Y: 20%		

Cisalhamento	Ângulo Min: -45° Ângulo Max: 45°		
Corte	Relação Altura/Largura: 20%		
Ruído Gaussiano	Desvio Padrão: 1 Porcentagem do tom de cinza: 100%		

Fonte: Do Autor

Utilizando a função `time()` da linguagem Python, foi estimado o tempo de processamento demandado para a geração de uma base de dados aumentada. Esse ensaio foi executado em um notebook Windows 10, com memória RAM de 4GB, placa de vídeo Intel Graphics 5500 e SSD de 240GB e processador Intel I3 5ª geração. O teste foi realizado utilizando uma imagem de tamanho HD de dimensões 1280 x 1024 pixels, considerado o tamanho padrão de imagens adquiridas por uma câmera atualmente. Assim como realizado individualmente cada transformação possível e por fim com o maior número de transformações possíveis que são 5: Translação, Espelhamento Vertical, Espelhamento Horizontal, Corte e Adição de Ruído. Foi realizado dessa maneira a fim de saber o tempo estimado de processamento máximo de execução do sistema. Foram geradas 470 imagens para cada uma das condições descritas acima. Conforme observado na Tabela 7 foi calculada uma média para cada uma das transformações.

Tabela 7 – Tempo médio consumido

Transformação Aplicada	Tempo (s)
Rotação	0,69
Espelhamento Vertical	0,63
Espelhamento Horizontal	0,71
Escala	1,04

Translação	0,88
Cisalhamento	0,87
Corte	1,08
Ruído Gaussiano	5,59
5 Transformações (Translação, espelhamento vertical, horizontal, corte, ruído gaussiano)	6,57

Fonte: Do Autor

Outro ensaio realizado foi a verificação de uso de memória e capacidade de processamento do computador. Realizado sob as mesmas condições que o ensaio anterior. Utilizando a função psutil da linguagem Python o quanto de memória RAM o sistema de aumento de dados utiliza, enquanto é executado. Estes dados podem ser observados através da Tabela 8. A memória fica próxima de 3,5 GB, para todas as transformações, com pequenas variações.

Tabela 8 – Utilização da memória RAM pelo sistema de aumento de dados.

Transformação Aplicada	Memória Absoluta Utilizada (GB)	Memória Relativa Utilizada (%)
Rotação	3,59	89,8
Espelhamento Vertical	3,52	88,0
Espelhamento Horizontal	3,58	89,5
Escala	3,51	87,6
Translação	3,54	88,4
Cisalhamento	3,49	87,2
Corte	3,44	86,0
Ruído Gaussiano	3,45	86,2
5 Transformações (Translação, espelhamento vertical, horizontal, corte, ruído gaussiano)	3,39	84,7

Fonte: Do Autor

4.2 Análise e discussão dos resultados

Os testes realizados mostraram que o sistema de aumento de dados realiza suas funções básicas de gerar imagens com transformações realizadas a

partir de um conjunto já existente. É possível observar um bom número de transformações realizadas, das quais são as mais utilizadas quando se pensa em aplicar essa técnica. Além disso, é possível realizar combinações entre elas fazendo com que esse sistema se aplique a praticamente qualquer situação. Porém vale ressaltar que há outras transformações que podem ser aplicadas, principalmente as transformações fotométricas.

A utilização do sistema é bem mais rápida do que comparado quando necessário implementar a técnica de aumento de dados, durante a implementação de uma rede neural. O sistema depende da transformação aplicada, porém é possível observar que uma imagem, independente das transformações desejadas, não demora mais de 10 segundos a sua geração. Além de mais veloz do que a implementação de um código, também é muito mais fácil. Através de alguns parâmetros é possível começar a gerar imagens sem um conhecimento amplo desta técnica.

O sistema desenvolvido atendeu a todos os requisitos que orientam seu desenvolvimento. Em relação aos requisitos funcionais, é possível gerar uma base aumentada incluindo transformações geométricas e fotométricas. Em relação aos requisitos não funcionais a solução final é parametrizável e de fácil utilização.

5 Considerações Finais

As redes neurais artificiais, de modo geral, precisam ser treinadas a partir de um banco de dados já existente. Porém, para muitas das aplicações não são encontrados bancos de dados abrangentes e de tamanho suficiente, para se obter uma boa precisão e acurácia das redes neurais. No presente trabalho foi desenvolvido um sistema de fácil utilização para realizar a geração de bases de dados aumentadas voltadas para utilização de redes neurais convolucionais.

Foram analisadas e estudadas as transformações mais comumente utilizadas. De modo geral foram encontradas diversos tipos de transformações possíveis de realizar com uma imagem. A partir deste estudo foram definidas quais seriam as transformações a serem aplicadas no desenvolvimento do sistema.

O sistema foi criado combinando uma interface gráfica amigável com rotinas de geração de imagens. Esta interface foi desenvolvida utilizando a biblioteca Python, Tkinter. Já para as rotinas de geração de imagens foram utilizadas principalmente duas bibliotecas, a de manipulação de imagens OpenCv e a de realização de transformações, *Imgaug*.

Pelos testes realizados é possível afirmar que o sistema cumpriu os requisitos a serem atendidos. É disponibilizado ao usuário a opção de gerar bases de imagens aumentadas, incluindo diversas transformações através de uma interface gráfica amigável e de fácil utilização.

5.1 Sugestões para trabalhos futuros

Como trabalhos futuros são sugeridas melhorias a serem implementadas no sistema como:

- permitir fazer as transformações via script *python*, passando os parâmetros desejados;
- possibilidade de salvar um projeto de geração de imagens contendo configurações anteriores;
- a partir das configurações o próprio sistema calcular a quantidade de imagens a serem geradas;

- possibilidade de interromper a execução de uma rotina de geração já iniciada;
- realizar transformações em imagens provenientes de uma câmera;
- realizar transformações em vídeo;
- realizar outras transformações, principalmente fotométricas;
- possibilidade de realizar todas as transformações desejadas de forma paralela;
- permitir fazer as transformações via um script python, passando os parâmetros desejados;
- gerar interface gráfica Web, a fim de possibilitar a utilização de qualquer usuário de forma remota.

Todas estas melhorias agregariam ao sistema e o deixariam mais completo e com mais funcionalidades, flexibilizando ainda mais o seu uso.

Referências

ABU-SAMAN, Awni M.; AL-ZERAI, Huda S. Using F-Transform to Remove Image Noise. Journal of Mathematical and Computational Science, [s. l.], 28 ago. 2016. Disponível em: https://www.researchgate.net/publication/307138664_USING_F-TRANSFORM_TO_REMOVE_IMAGE_NOISE. Acesso em: 21 fev. 2022.

ALVES, Gisely. Entendendo Redes Convolucionais (CNNs). [S. l.], 8 out. 2018. Disponível em: <https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>. Acesso em: 14 dez. 2021.

ALVES, Priscila Mello. Inteligência Artificial e Redes Neurais. [S. l.], 23 dez. 2020. Disponível em: <https://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/106-inteligencia-artificial-e-redes-neurais>. Acesso em: 10 dez. 2021.

BALLA, Niteesha. Five Popular Data Augmentation Techniques in Deep Learning. [S. l.], 31 ago. 2020. Disponível em: <https://dataaspirant.com/data-augmentation-techniques-deep-learning/>. Acesso em: 24 mar. 2022.

BROWNLEE, Jason. How to Configure Image Data Augmentation in Keras. [S. l.], 12 abr. 2019. Disponível em: <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>. Acesso em: 5 abr. 2022.

CUNHA, Leonardo Cardoso da. Redes Neurais Convolucionais e Segmentação de Imagens - Uma Revisão Bibliográfica. Orientador: Profa. Dra. Luciana Gomes Castanheira. 2020. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) - Universidade Federal de Ouro Preto, [S. l.], 2020. Disponível em: https://www.monografias.ufop.br/bitstream/35400000/2872/6/MONOGRRAFIA_Redес_NeuraisConvolucionais.pdf. Acesso em: 22 mar. 2022

DATA SCIENCE ACADEMY. Deep Learning Book: Capítulo 40 – Introdução as Redes Neurais Convolucionais. [S. l.]: Data Science Academy, 2021. Disponível em: <https://www.deeplearningbook.com.br/introducao-as-redes-neurais-convolucionais/>. Acesso em: 22 nov. 2021.

INSIGHT. Por que o Python é a linguagem mais adotada na área de data science? UNIVERSIDADE FEDERAL DO CEARÁ (Fortaleza, CE). Insight. [S. l.], 18 set. 2019. Disponível em: <https://insightlab.ufc.br/por-que-o-python-e-a-linguagem-mais-adotada-na-area-de-data-science>. Acesso em: 24 fev. 2022.

GANDHI, Arun. Data Augmentation: How to use Deep Learning when you have Limited Data - Part 2. [S. l.], 2021. Disponível em:

<https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/>. Acesso em: 11 nov. 2021.

Giulia Zanon de Castro et al. Desenvolvimento de uma Base de Dados de Sinais de Libras para Aprendizado de Máquina: Estudo de Caso com CNN 3D. In: ANAIS DO 14º SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 2019, Ouro Preto. Anais eletrônicos... Campinas, Galoá, 2019. Disponível em: <<https://proceedings.science/sbai-2019/papers/desenvolvimento-de-uma-base-de-dados-de-sinais-de-libras-para-aprendizado-de-maquina--estudo-de-caso-com-cnn-3d>> Acesso em: 10 Maio. 2022.

JET BRAINS. PyCharm: O IDE Python para desenvolvedores profissionais. [S. l.], [entre 2000 e 2022]. Disponível em: <https://www.jetbrains.com/pt-br/pycharm/>. Acesso em: 18 jan. 2022.

JURASZEK, Guilherme Defreitas. Reconhecimento de produtos por imagem utilizando palavras visuais e redes neurais convolucionais. Orientador: Alexandre Gonçalves Silva e André Tavares da Silva. 2014. Dissertação (Mestre em Computação Aplicada) - Universidade do Estado de Santa Catarina - Centro de Ciências Tecnológicas, [S. l.], 2014. Disponível em: https://www.researchgate.net/profile/Guilherme-Juraszek-2/publication/275958578_Image_Recognition_of_Products_Using_Bag_of_Visual_Words_and_Convolutional_Neural_Networks/links/554b6b7e0cf29752ee7c7cda/Image-Recognition-of-Products-Using-Bag-of-Visual-Words-and-Convolutional-Neural-Networks.pdf. Acesso em: 20 jan. 2022.

KHANDELWAL, Renu. Data Augmentation techniques in python: A sneak peek into data augmentation techniques in Python using imgaug. [S. l.], 10 dez. 2019. Disponível em: <https://towardsdatascience.com/data-augmentation-techniques-in-python-f216ef5eed69>. Acesso em: 16 fev. 2022.

LENDAVE, Vijaysinh. Um guia para diferentes tipos de ruídos e métodos de eliminação de ruídos de imagem. [S. l.], 25 set. 2021. Disponível em: <https://analyticsindiamag.com/a-guide-to-different-types-of-noises-and-image-denoising-methods/>. Acesso em: 13 abr. 2022.

MDPI. Journal of Imaging. [S. l.], [entre 2015 e 2022]. Disponível em: mdpi.com/journal/jimaging. Acesso em: 8 apr. 2022.

NANNI, Loris; PACI, Michelangelo; BRAHNAM, Sheryl; LUMINI, Alessandra. Comparison of Different Image Data Augmentation Approaches. [S. l.], 12 nov. 2021. Disponível em: https://www.researchgate.net/publication/356667599_Comparison_of_Different_Image_Data_Augmentation_Approaches. Acesso em: 12 abr. 2022.

NELSON, Joseph. Why and How to Implement Random Crop Data Augmentation. [S. l.], 21 fev. 2020. Disponível em: <https://blog.roboflow.com/why-and-how-to-implement-random-crop-data-augmentation/>. Acesso em: 12 abr. 2022.

OPENCV TEAM. OpenCV. [S. l.], 2022. Disponível em: <https://opencv.org/about/>. Acesso em: 8 mar. 2022.

OSHIRO, Gabriel Eike. Detecção e Classificação de Equipamentos em Torres de Telecomunicações Através de Redes Neurais Convolucionais. Orientador: Prof. Dr. Eng. Francisco Rafael Moreira da Mota. 2022. TCC (Bacharelado em Engenharia Mecatrônica) - Instituto Federal de Santa Catarina, [S. l.], 2022.

PYTHON SOFTWARE FOUNDATION. Tkinter: Python interface to Tcl/Tk. [S. l.], [entre 2001 e 2022]. Disponível em: <https://docs.python.org/3/library/tkinter.html>. Acesso em: 15 mar. 2022.

RAJ, Bharath. Data Augmentation | How to use Deep Learning when you have Limited Data — Part 2. [S. l.], 11 abr. 2018. Disponível em: <https://blog.roboflow.com/why-and-how-to-implement-random-crop-data-augmentation/>. Acesso em: 14 abr. 2022.

REDES neurais. [S. l.], 17 ago. 2020. Disponível em: <https://www.ibm.com/br-pt/cloud/learn/neural-networks>. Acesso em: 18 nov. 2021.

RICI, Pedro; SANTOS, Samara Oliveira Silva; OTTONI, André Luiz Carvalho. Análise da seleção de hiperparâmetros de Data Augmentation na detecção de Covid-19 em imagens de raio-x com Deep Learning. Artigo - Universidade Federal do Recôncavo Baiano, [S. l.], 2021. Disponível em: https://sbic.org.br/wp-content/uploads/2021/09/pdf/CBIC_2021_paper_22.pdf. Acesso em: 02 mar. 2022.

ROCHA, Rafael de Lima. Redes neurais convolucionais aplicadas à inspeção de componentes do vagão ferroviário. Orientador: Prof. Dr. Cleison Daniel Silva. Coorientadora: Dra. Ana Claudia da Silva Gomes. 2020. Dissertação (Pós Graduação em Computação Aplicada) - Universidade Federal do Pará, [S. l.], 2020. Disponível em: http://repositorio.ufpa.br/jspui/bitstream/2011/12790/1/Dissertacao_RedesNeuraisConvolucionais.pdf. Acesso em: 4 abr. 2022.

ROSA, Ayrton Lima Da. Classificação de Imagens de Frutas utilizando Aprendizado de Máquina. Orientador: Prof. Danilo Silva, Ph.D. 2019. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica) - Universidade Federal de Santa Catarina - Centro Tecnológico - Departamento de Engenharia Elétrica e Eletrônica, [S. l.], 2019. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/197598/TCC-Ayrton-Lima.pdf?isAllowed=y&sequence=1>. Acesso em: 16 nov. 2021.

ROVEDA, Ugo. O que é python, para que serve e por que aprender?. [S. l.], 13 out. 2020. Disponível em: <https://kenzie.com.br/blog/o-que-e-python/#:~:text=Python%20%C3%A9%20uma%20linguagem%20de%20programa%C3%A7%C3%A3o%20de%20alto%20n%C3%ADvel%20%E2%80%94%20ou,maior%20controle%20e%20estabilidade%20de>. Acesso em: 9 fev. 2022.

SAS INSTITUTE INC. (USA). Redes Neurais: O que são e qual sua importância?. [S. l.], 2019. Disponível em: https://www.sas.com/pt_br/insights/analytics/neural-networks.html. Acesso em: 25 nov. 2021.

SHAHINFAR; Saleh., MEEK, Paul.; FALZON, Greg. "How many images do I need?" Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. *Ecological Informatics*, Volume 57, 2020. Disponível em: <https://doi.org/10.1016/j.ecoinf.2020.101085>. Acesso em: 12 Fev. 2021.

STIVANELLO, Mauricio Edgard; MARCELLINO, Kleber Juan. A Machine Vision System for Online Metal Can-End Rivet Inspection. *Journal of Physics: Conference Series*, [S. l.], p. 0 - 12, 27 ago. 2019. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/1335/1/012002/pdf>. Acesso em: 30 maio de 2022.

ULKU, Irem; AKAGÜNDÜZ, Erdem. A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D Images. *Journal of Applied Artificial Intelligence*, [S. l.], Çankaya University, Turquia, mar. 2022. Disponível em: <https://arxiv.org/abs/1912.10230>. Acesso em: 20 Abr. 2022.

UTILIZAÇÃO de Técnicas de Data Augmentation em Imagens:. *In*: CLARO, Maíla; VOGADO, Luis; SANTOS, Justino Santos and Rodrigo Veras; VERAS, Rodrigo. *Utilização de Técnicas de Data Augmentation em Imagens:: Teoria e Prática*. [S. l.: s. n.], 2020. Disponível em: <https://sol.sbc.org.br/livros/index.php/sbc/catalog/download/48/217/455-1?inline=1>. Acesso em: 16 fev. 2022.