UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ
INSTITUTO DE GEOCIÊNCIAS E ENGENHARIAS
Faculdade de Engenharia da Computação
Bacharelado em Engenharia da Computação

Trabalho de Conclusão de Curso

SEGMENTAÇÃO DE LÁBIOS POR IMAGENS DE RESSONÂNCIA MAGNÉTICA UTILIZANDO DEEP LEARNING

Lucas Keley Sousa de Jesus

Marabá-PA
2023

Lucas Keley Sousa de Jesus

SEGMENTAÇÃO DE LÁBIOS POR IMAGENS DE RESSONÂNCIA MAGNÉTICA UTILIZANDO DEEP LEARNING

Trabalho de Conclusão de Curso, apresentado à Universidade Federal do Sul e Sudeste do Pará, como parte dos requisitos necessários para obtenção do Título de Bacharel em Engenharia da Computação.

Orientador:
Prof. Dr. Haroldo Gomes Barroso Filho

Marabá-PA
2023

Lucas Keley Sousa de Jesus

SEGMENTAÇÃO DE LÁBIOS POR IMAGENS DE RESSONÂNCIA MAGNÉTICA UTILIZANDO DEEP LEARNING

Trabalho de Conclusão de Curso, apresentado à Universidade Federal do Sul e Sudeste do Pará, como parte dos requisitos necessários para obtenção do Título de Bacharel em Engenharia da Computação.

Marabá: 03 de Abril de 2023

BANCA QUALIFICADORA:

Prof. Dr. Haroldo Gomes Barroso Filho
(Orientador - UNIFESSPA)

Prof. Dr. Adam Dreytron Ferreira dos Santos
(Membro da Banca - UNIFESSPA)

Prof. Me. Cláudio de Castro Coutinho Filho
(Membro da Banca - UNIFESSPA)

Marabá-PA
2023

Este trabalho é dedicado especialmente aos meus pais, José Nilson Braga de Jesus e Loidilene Sousa de Jesus, além de todos aqueles que compartilham dos meus sonhos e ideais.

AGRADECIMENTOS

Acredito que devo toda minha caminhada acadêmica que certamente está apenas no começo, a diversas pessoas que estiveram comigo até aqui, que certamente sem elas não poderia ter chegado até onde estou, também estou ciente que não conseguirei citar todos, mais se de alguma forma você me inspirou, ajudou ou acreditou em mim, se sinta representados pelos nomes dispostos nas próximas linhas desse agradecimento.

Meus agradecimentos sinceros a toda minha família, que sempre foram o motivo pelo qual não desisti, aqui representados pelos meus pais, Jose Nilson Braga de Jesus, e Loidilene Sousa de Jesus e meus irmãos, Luanderson Sousa de Jesus e Luyane Sousa de Jesus.

Agradecer a pessoa que me acompanhou em todo o trajeto de execução desse trabalho, quem esteve ao meu lado me ajudando em momentos que eu deixei de acreditar que conseguiria dar mais um passo, muito obrigado a Lais Aguiar Carvalho, que venha ainda mais conquistas ao seu lado.

Meu muito obrigado a Anderson da Silva Neves, por nesses longos anos de amizade me motivar a seguir meus objetivos.

Meus agradecimentos a quem me ensinou meus primeiros comandos em um computador, Felipe Ribeiro Lima, (Ph).

Aos professores que me inspiraram, desde minha educação infantil, que me fizeram acreditar no meu potencial, que me ajudaram a chegar até a universidade, representados aqui pela Profª. Alessandra, Prof. Andre Luiz Pereira da Silva, Profª. Clarice Lima, Prof. Everaldo Marinho e Prof. Sílvio Rogério Pinto Gomes.

A todos os professores que fizeram parte da minha formação acadêmica em Engenharia da Computação, que me ensinaram e me inspiraram, representados aqui pela Prof. Ma. Aline Farias Gomes de Sousa, Prof. Dra. Cindy Stella Fernandes, Prof. Me. Cláudio de Castro Coutinho Filho, Prof. Dra. Franciane Silva de Azevedo, Prof. Dr. João Victor Costa Carmona, Prof. Dr. José Carlos da Silva, junto ao meu orientador Haroldo Gomes Barroso Filho.

Agradecer aqueles que tornaram o caminho da graduação algo mais leve e descontraído, aos amigos que eu fiz nessa caminhada, aqui representados por, Amanda Fiel Savino, Bruno Borges Guerra, Bryan Frankin Sena de Sousa, Cristina Vitória Leal Leite, Lêda Maria Alves Fiel, José Ademir Pinto Rodrigues, Jefferson Yure Silva Pereira e Lucas Vinícius Silva Idelfonso.

Um agradecimento especial a todos os meus alunos, que me ensinaram tantas

coisas, entre elas ser alguém mais confiante diante de uma multidão.

Meu muito obrigado aqueles que vieram antes de mim, a Condessa de Lovelace Augusta Ada Byron King, (Ada Lovalec), e cientista Charles Babbage e o pai da computação Alan Turing.

Eu acredito que às vezes são as pessoas que ninguém espera nada que fazem as coisas que ninguém consegue imaginar.

(Alan Turing)

RESUMO

A segmentação de imagem visa simplificar uma imagem de interesse, para melhor analisá-la e compreendê-la. O objeto do projeto é a segmentação de lábios de imagens de ressonância magnética, através da implementação da arquitetura *mask R-CNN* em uma execução usando um ambiente em nuvem, o modelo foi treinado com um conjunto de dados com 1050 imagens de ressonância magnética e testado em vários conjuntos de dados, desde conjuntos com indivíduos para qual o modelo não foi pré treinado, até em imagens com adição de filtros como, *gaussian bluir*, ruído RGB e inversão de cores. Os resultados obtidos foram otimistas em vários conjuntos conjuntos de dados, como no conjunto de teste 02 sem adição de filtro com uma precisão de 0,94.

Palavras-chave: Segmentação. Imagem de Ressonância Magnética. Mark R-CNN.

ABSTRACT

Image segmentation aims to simplify an image of interest, to better analyze and understand it. The object of the project is the segmentation of lips from magnetic resonance images, through the implementation of the mask R-CNN architecture in a run using a cloud environment, the model was trained with a dataset with 1050 magnetic resonance images and tested in several data sets, from sets with individuals for which the model was not pre-trained, even in images with the addition of filters such as gaussian blur, RGB noise and color inversion. The obtained results were optimistic in several sets of data, as in test set 02 without filter addition with a precision of 0.94.

Keywords: Segmentation. Magnetic Resonance Imaging. Mark R-CNN.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

R-CNN	*Region-based Convolutional Neural Network*
Colab	*Colaboratory*
GPUs	*Graphics Processing Units*
MRI	*Magnetic Resonance Imaging*
IRM	*Imagens de Ressonância Magnética*
UMCL	*University Medical Center Ljubljana*
MR	*Magnetic Resonance*
FLAIR	*Fluid Attenuated Inversion Recovery*
BraTS	*Brain Tumor Segmentation*
IEEE	*Institute of Electrical and Electronics Engineers*
VIA	*Vgg Image Annotator*
VP	*Verdadeiro Positivo*
VN	*Verdadeiro Negativo*
FP	*Falso Positivo*
FN	*Falso Negativo*
TR	*Tempo de Repetição*
TE	*Tempo de Eco*
RF	*Radiofrequência*
Kbps	*Kilobits por segundo*
FPS	*Frames Por Segundo*
RGB	*Red, Green, Blue*
FAIR	*Facebook AI Reserach*
FPN	*Feature Pyramid Network*
RPN	*Region Proposal Network*
FG	*Foreground*
BG	*background*
ROI	*Region of Interest*

SUMÁRIO

1 INTRODUÇÃO

O processamento de imagem teve inicio no seculo XX, com a implementação do sistema *bartlane* de transmissão, que ligava por cabos submarinos, Londres e Nova York, esse sistema era usado para transmitir imagens digitalizadas para jornais entre os dois países, onde no transmissor a imagens era codificada e transmitida e logo após reconstruída no receptor, após essa implementação, o tempo de transmissão caiu de 10 dias para apenas 3 horas. Entretanto houvesse a preocupação do tratamento dessas imagens no receptor, usando técnicas tradicionais de processamento de imagens.

Porém as técnicas de processamento de imagem, incluindo a segmentação de imagem, vieram possuir atenção apenas após seu uso no processamento de imagens lunares, com o intuito de remover distorções nas imagens da lua extraídas pela sonda espacial *ranger 7* em 1964.

Desde então o processamento de imagem cresce gradativamente, possuindo aplicações em diversas áreas, atualmente podemos encontrar aplicações dessas técnicas em veículos autômatos, ou seja, veículos que não precisam de um motorista humano para se dirigir de um ponto a outro, isso é possível graças ao uso da segmentação de imagens, junto a técnicas de detecção de objetos e *deep learning*, para detectar objetos em um ambiente de trânsito, segmentá-los e assim aprender de que modo deve-se prosseguir com uma ação de se dirigir até um determinado local.

A análise de imagens médicas como tomografias, mamografias, MRI – *Magnetic resonance imaging*,e entre outras,é uma área de grande importância na medicina, porém além da necessidade de um especialista, vários fatores como a variabilidade das imagens, exaustão humana, grau de experiência do analista, complexidade das imagens, podem levar a análises menos confiáveis e diferentes resultados de diagnósticos. Com isso o uso de técnicas de visão computacional, que envolvem *deep learning*, e segmentação, ganham cada vez mais espaço em aplicações médicas. Tendo em vista que a segmentação possui como um de seus objetivos a simplificação de imagens, como uma maneira de facilitar sua compreensão, o uso de tal técnica pode ser aplicada para detectar variações em tamanhos de tumores após tratamento, busca de anomalias em imagens médicas, entre outros auxílios ao profissional da área da saúde.

Os avanços atuais no poder computacional, além do cada vez maior interesse em usar as técnicas de *deep learning*, como uma ferramenta de auxílio nas diversas áreas de atuação, impulsionam cada vez mais cientistas, acadêmicos e entusiastas, a buscarem compreender, estudar e produzir conteúdo na área de visão computacional.

1.1 Justificativa

A segmentação de imagem no contexto de visão computacional possui como objetivo a decomposição de uma imagem para posteriormente analisá-la, podendo ser decomposta como uma classificação pixel a pixel. O ato de segmentar uma imagem usando *deep learning* se justifica pelo fato da possibilidade do uso de uma grande variedade de dados, superando algoritmos superficiais de *machine learning* na maioria dos casos referentes a dados de imagens e vídeos, (LeCun et al. 2015). Onde na ocasião deste trabalho consistindo os dados como imagens de RMI de trato vocal humano.. Além dos demais fatores:

- Volume de dados: a possibilidade de processar volumes gigantescos de dados através de computadores se dá devido ao fato das grandes melhorias de *hardware* dos últimos tempos, que escalonam as capacidades de armazenamento e potencializam o processamento de tarefas;
- Redução de Erros: a aplicação de *deep learning* reduz a margem de erro que podem passar despercebidos por algoritmos convencionais não focados em aprendizado profundo, maximizando assim a assertividade das análises;
- Redução de Tempo: o alto grau de processamento de computadores e recursos atuais, proporcionam a possibilidade de um algorítimo executar um enorme volume de tarefas simultaneamente, viabilizando o uso de modelos de *deep learning* como uma abordagem que reduz o tempo gasto em análise de imagens quando usado de maneira correta e com auxílio de especialistas.

1.2 Objetivo Geral

Analisar o desempenho e eficiência do algoritmo de segmentação de imagem no trato vocal, podendo assim ser usado como uma opção de baixo custo para potencializar resultados do trato vocal e auxiliar especialistas em múltiplas profissões da saúde.

1.3 Objetivos Específicos

Nos objetivos específicos serão retratados os seguintes pontos com o objetivo de agregar e alcançar o objetivo geral:

- Teste do algoritmo *Mask R-CNN* no contexto sugerido;
- Testar a performance do algoritmo *Mask R-CNN*;
- Aplicação e teste de filtros nos dados através do algoritmo *Mask R-CNN*;

- Análise de Impacto de filtros nos resultados;
- Analisar as dificuldades enfrentadas pela tipo de abordagem;
- Segmentação da Imagem pelo algoritmo *Mask R-CNN* e consequentemente auxiliar especialista;

2 TRABALHOS CORRELATOS

Os trabalhos aqui apresentados possuem alguns aspectos relacionados ao projeto proposto, tendo como objetivo agregar embasamento e boas referências ao projeto.

2.1 *Automatic detection of multiple sclerosis lesions using Mask R-CNN on magnetic resonance scans*

Os autores (SÜLEYMAN, Mehmet. Yıldırım. 2020), (DANDIL, Emre. 2020), realizaram o trabalho aqui apresentado com o objetivo de segmentar lesões de esclerose múltipla em imagem de ressonância magnética, através da *Mask R-CNN*.

Para a execução do trabalho, foram utilizados duas bases de dados públicas, sendo elas, a base de dados de esclerose múltiplas do *eHealth Laboratory* e a base de dados do *University Medical Center Ljubljana* (UMCL) para detecção de lesões de esclerose múltiplas. O conjunto de dados do *eHealth Laboratory* fornecido pelo laboratório da Universidade de Chipre para estudos científicos, possui um total de 1838 imagens de Ressonância Magnética de 38 pacientes, onde em 667 dessas imagens possuem 1777 lesões de esclerose múltipla, que foram detectadas por médicos especialistas em primeiro exame e exames periódicos entre 6 a 12 meses. Já o conjunto de dados da*University Medical Center Ljubljana* (UMCL) é composto por imagens de ressonância magnética de 30 pacientes, obtidas através do sistema *3T Siemens Magnetom Trio MR*, tendo como resultado imagens em 2D de MR e 3D FLAIR, formando um conjunto de 600 Imagens.

O treinamento da rede neural foi executado em sistema operacional linux, usando a arquitetura de código aberto da *Mask R-CNN*, os recursos de *hardware* para o projeto são descritos na tabela abaixo:

Hardware	Descrição
Unidade Central de Processamento	Intel Core i9–9900 K @ 5 GHz (8 núcleos/16 threads)
Memoria RAM	32 GB (DDR4 2666 MHz)
Placa-Mãe	ASUS WS Z390 PRO
GPU (x2)	NVIDIA GeForce RTX 2080Ti 11 GB GDDR6
driver de disco rígido	256 GB SSD HDD + 3 TB SATA 6 Gb 3,5 HDD

Tabela 1 – Recursos de *hardware* para a implementação da segmentação de lesões de esclerose múltipla

2.2 *Automated glioma grading on conventional MRI images using deep convolutional neural networks*

O trabalho dos autores em questão (ZHUGE, Ying. 2020), (NING, Holly. 2020), (CHENG, Jason Y. 2020), (KRAUZE, Andra V. 2020), (CAMPHAUSEN, Kevin. 2020) e

(MILLER, Robert W. 2020), tem como proposta dois métodos não invasivos e automáticos para a distinção de glioma, usando redes neurais convolucionais profundas.

Ambos os métodos possuem duas etapas, a primeira etapa consiste na segmentação tridimensional (3D) do tumor cerebral usando como base uma modificação do popular modelo U-Net, e a segunda etapa consiste na classificação do tumor cerebral segmentado. A base de dados para implementação da rede neural foi obtida através de um conjunto de duas bases de dados pré operatório e anotados por médicos especialistas, sendo eles, o conjunto de dados BraTS - *Brain Tumor Segmentation* 2018, que incluem dados de 285 pacientes e o conjunto *The Cancer Genome Atlas Low Grade Glioma Collection* (TCGA-LGG), de onde foram selecionados 30 pacientes entre o total.

O treinamento e a execução de testes foram implementados em sistema operacional linux em sua distribuição ubuntu 18.04, usando como arquitetura a *Mask R-CNN*, já os recursos de hardware são descritos na tabela abaixo:

Hardware	Descrição
Modelo	DELL PRECISION TOWER T7910
Unidade Central de Processamento	Xeon de 2,20 GHZ e 20 núcleos
Memoria RAM	64GB
Placa-Mãe	ASUS WS Z390 PRO
GPU (x2)	NVIDIA Titan Xpcom 12 GB

Tabela 2 – Recursos de *hardware* para a implementação da classificação automatizada de glioma em imagens

2.3 *Automatic Detection and Segmentation of Breast Cancer on MRI Using Mask R-CNN Trained on Non–Fat-Sat Images and Tested on Fat-Sat Images*

O trabalho possui o objetivo a implementação de um modelo para a detecção automática e segmentação de lesões suspeitas de câncer de mama em imagens de ressonância magnética (apud ZHANG. 2020).

Para a concepção do conjunto de dados foram selecionados apenas pacientes que possuíam lesão de massa unilaterais em uma única mama, com o intuito de usar a mama contralateral como referência baseada na simetria. O conjunto de dados usado na implementação possui duas partes, a primeira é obtido de um grupo de 241 pacientes com idade média de 49 anos, na faixa etária de 30 a 80 anos, cujo dados foram usados como base de treinamento, já a segunda parte do conjunto de dados, composto por um grupo de 98 pacientes possuindo idade média de 49 anos, em uma faixa etária de 22 a 67 anos, foi usado como base de teste.

A implementação foi executada usando a linguagem *python 3.6*, biblioteca *Ten-*

sorFlow 1.4 de código aberto, a arquitetura *Mask R-CNN*, entretanto a única configuração de *hardware* disposta no trabalho é o uso de uma de GPU com quatro placas NVIDIA GeForce GTX Titan X (12 GB, arquitetura Maxwell).

2.4 *A Novel Deep Learning Method for Recognition and Classification of Brain Tumors from MRI Images*

O estudo possui o intuito de demonstrar uma nova abordagem para a segmentação e a detecção de tumores cerebrais em imagens de ressonância magnética, (apud MASOOD. 2021).

Para a composição do conjunto de dados necessário para a concepção do trabalho, os autores usaram duas bases de dados públicas, sendo elas, a base de dados *brain tumor dataset* disposta no acervo online *figshare*, e a base de dados *Brain MRI Images for Brain Tumor Detection* disposta no acervo online *kaggle*. Os conjuntos de dados foram organizados de modo que 70% dos dados foram usados como imagens na base de treinamento e 30% dos dados foram usados como base de teste.

A concepção do modelo de treinamento foi feito através da arquitetura *mask R-CNN*, para a geração de pesos das redes neurais foi usado o modelo pré-treinado MS-COCO, dispensando assim a implementação de pesos de um ponto zero. Entretanto as informações dos recursos de *hardware* não foram dispostas no trabalho.

2.5 Correlação entre os trabalhos

Os trabalhos correlatos e o projeto do autor possuem suas similaridades, sendo a principal delas o tipo de abordagem usando a arquitetura *Mask R-CNN* para a segmentação de imagens de ressonância magnética.

Trabalho	Ano de Publicação	Arquitetura	Objeto de Segmentação	Precisão em Melhor Caso
Trabalho correlato 2.1	2020	Mask R-CNN	Lesões de esclerose múltipla	0.8703
Trabalho correlato 2.2	2020	Mask R-CNN	Glioma	0.9710
Trabalho correlato 2.3	2020	Mask R-CNN	Lesões de mama	0.8600
Trabalho correlato 2.4	2021	Mask R-CNN	Tumores cerebrais	0.9834
Autor	2023	Mask R-CNN	Labios	0.9421

Tabela 3 – Correlação entre os trabalhos

Na tabela 3, podemos notar os resultados de precisão de melhor caso para cada trabalha, que consiste em um desempenho de precisão acima de 85% em todos os casos, entre os trabalhos o que mais se destacam são, trabalho 2.2 *(Automated glioma grading on conventional MRI images using deep convolutional neural networks)*, tranalho 2.4 *(A Novel Deep Learning Method for Recognition and Classification of Brain Tumors from MRI Images)* juntamente com o o trabalho do autor, com precisão, , 0.9710, 0.9834 e 0.9421, respectivamente.

Para os casos implementados a *Mask R-CNN* se mostrou bastante promissora, levando em consideração que os trabalhos implementados são recentes, sendo todos eles implementados entre 2020 a 2023, demostrando que tais abordagens sugerem um futuro otimista para novas contribuições na área da segmentação de imagens de ressonância magnética, tendo em visto os bons resultados adquiridos nos trabalhos selecionados.

3 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo constam os conceitos e fundamentações que serviram de base para o desenvolvimento do trabalho do autor.

3.1 Imagem de Ressonância Magnética

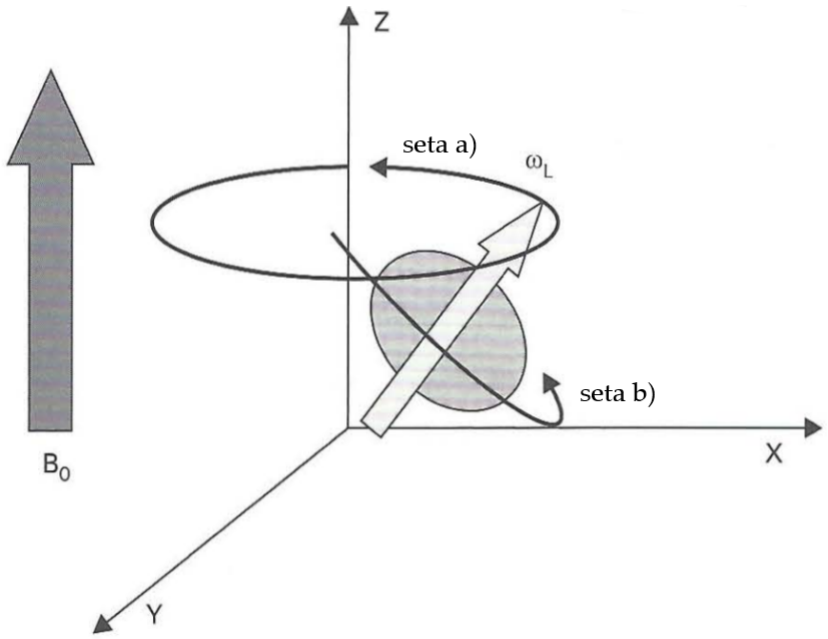
A ressonância magnética é a propriedade física exibida por núcleos de determinados elementos que, quando excitados por ondas de rádio em frequência de Larmor e submetidos a um campo magnético forte, emitem rádio sinal, o qual pode ser captado por uma antena receptora e transformado em imagem. (FERRARINI, Maria. 2009; IWASAKI, Masao. 2009). A imagem de Ressonância Magnética é um método não invasivo, usado idealmente para análise de partes moles, como órgãos, gorduras e massas.

Em 1970 Raymond Damadian, observou em ratos de laboratório que os tecidos saudáveis e os tecidos que possuíam tumores malignos produziam diferentes respostas a quando ambos eram expostos a um pulso de radiofrequência ressonante, onde cada um emitia um sinal diferente à medida que os momentos dos dipolos magnéticos de cada tecido se equilibram. Os sinais recebidos continham variações em suas características dependendo diretamente do tipo de tecido, sendo ele saudável ou não, já que uma célula saudável é menos permeável ao fluxo da água em comparação a uma célula não saudável, com um movimento mais bruscos, as taxas de relaxamento são mais curtas. Pois uma célula doente é maior que uma célula saudável e possui uma membrana mais fina. Teoricamente o fluxo de entrada e saída da água é geralmente livre e lento, proporcionando taxas de relaxamento mais longas. Entretanto, essa teoria ainda não é amplamente aceita pela comunidade científica. Com tal experiência, Damadian pode chegar a conclusão que a estrutura da água se faz fundamental para a obtenção da imagem por ressonância magnética. Onde as características de contraste de imagem são definidas não pela saúde do tecido, mas sim pela diferença de hidratação dos tecidos.(FERRARINI, Maria. 2009; IWASAKI, Masao. 2009).

Para compreendermos a física da ressonância magnética, podemos utilizar o modelo físico clássico, para uma abordagem analógica.

Se compararmos o movimento de um núcleo de hidrogênio com a ausência de um campo magnético ao movimento de um peão, nota-se que ao estabiliza-se em um ângulo de 90°, entre o plano e eixo do pião, o mesmo continuaria girando, através da conservação de movimento angular, que podemos denominar como L, sendo a grandeza física associada rotação e translação de um corpo. A partir do momento que esse ângulo torna-se menor que 90°, o produto vetorial do peso do objeto produz um movimento chamado de precessão, ilustrado na Figura 1.

Figura 1 – Movimento de Precessão de um Núcleo em um campo magnético externo



Fonte: Adaptação de Vugman & Herbst

A Figura 1, demonstra a movimentação de precessão de um núcleo, quando o mesmo está em uma campo magnético externo, as setas demonstram seus dois movimentos, seta a) o movimento translacional denominado de precessão, e a seta b) demonstra a movimentação rotacional. Em um núcleo atômico, este possui um *spin*, que o faz se movimenta como um pião, e como esse núcleo possui próton, produz um campo magnético em direção ao eixo de rotação. O movimento magnético μ é o comportamento magnético de um conjunto de átomos, que apenas pode assumir valores que são impostos pela quantização do momento angular em que γ é a razão giromagnética, conhecida como uma constante diferente para cada núcleo e reflete a relação entre o momento angular e o momento magnético. Como mostra a equação (1).(FERRARINI, Maria. 2009; IWASAKI, Masao. 2009).

$$\mu = \mathbf{L} \times \gamma \tag{1}$$

Portanto, o momento magnético de um núcleo, é compreendido como um vetor, que quando está em presença de um campo magnético, coincide com um torque T que tende a realinha-lo à direção do campo, por meio da equação (2):

$$\mathbf{T} = \mu \times B_0 \tag{2}$$

logo, com o surgimento do torque, manifesta-se também o movimento de precessão

a frequência angular ω_0, com isso, em uma condição de equilíbrio, a associação entre torque e momento angular pode ser demostrada na equação(3).

$$\mu \times \mathbf{B_0} = \omega_\mathbf{0} \times \mathbf{L} \tag{3}$$

concluindo, ao relacionar as equações anteriores, temos a equação(4).

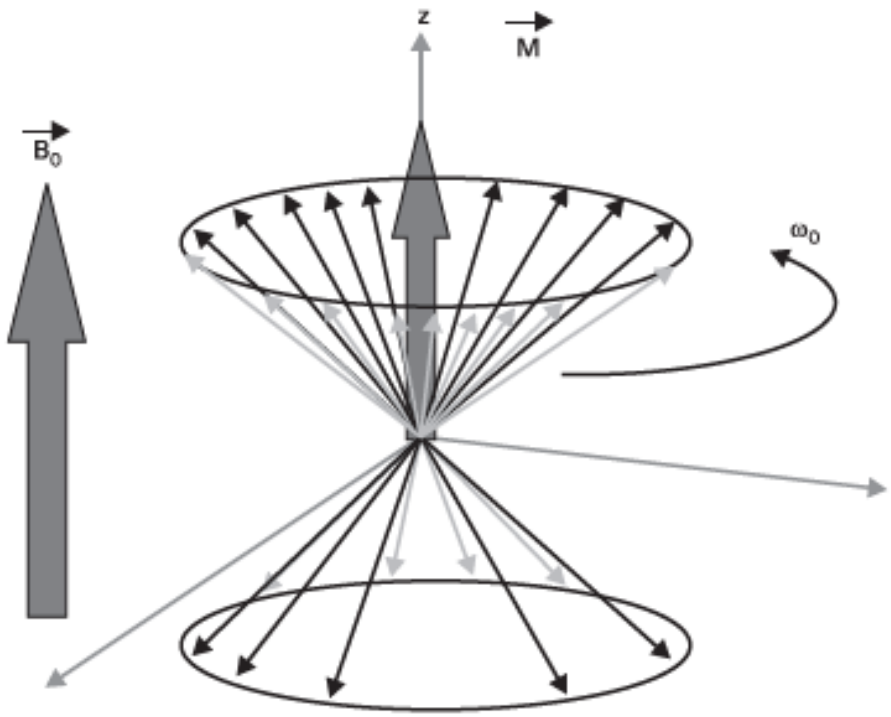
$$\omega_0 = (\mu/\mathbf{L}) \times |\mathbf{B}| \tag{4}$$

Como demostrado na equação(1), a razão entre as grandezas vetoriais e a constante **y**. Ao substituir a equação(4) temos a equação(5):

$$\omega_\mathbf{0} = \gamma \times \mathbf{B_0} \tag{5}$$

Ao analisar a equação(5), nota-se que o sentido da precessão é o mesmo do campo magnético. Este fenômeno foi demostrado pelo físico Irlandês Joseph Larmor, sendo ω_0 representando à frequência de Larmor em megahertz (MHz). Ilustrado na Figura 2.

Figura 2 – Precessão de Larmor



Fonte: Adaptação de Vugman & Herbst

Procedendo os conceitos quânticos, a direção do campo magnético estático, sendo o plano **z** do sistemas de coordenadas {x, y, z}, a direção a qual está definida a quantificação do

sistemas de *spins*. Sendo assim o módulo de suas projeções na direção da coordenada **z** adota valores quantizados. Sendo os módulos dos momentos magnéticos também quantizados, o ângulo formado por eles com o eixo **z** assume um conjunto de valores constantes dependendo apenas do número quântico de *spin*. A princípio não há uma orientação definida no plano **xz**, por tanto, consegue assumir qualquer orientação possível, e o conjunto dos vetores momentos magnéticos determina um conjunto de superfícies cônicas coaxiais cujo número é igual ao de estados quânticos possíveis de *spin*. Na hipótese de *spin* $\frac{1}{2}$ a precessão acontece ao longo de duas superfícies cônicas opostas pelo vértice, com eixo de precessão na direção **z**.

A soma vetorial das componentes dos momentos magnéticos no plano **xz** possui como resultado o valor nulo. Já o resultado das componentes na direção **z**, que estiverem no mesmo sentido que o campo magnético aplicado, deverão se manter em um estado de energia menor do que aquelas que se encontram em sentido oposto ao campo. Em estado de equilíbrio, ocorre a chamada distribuição de Boltzmann, onde um número maior de momentos magnéticos ocupa o estado de menor energia, possuindo como consequência uma magnetização que aponta no mesmo sentido do campo magnético. Em resumo a distribuição de Boltzmann estabelece que para qualquer sistema que esteja em equilíbrio térmico T a probabilidade para se encontrar um estado a uma energia particular é proporcional a $e^{-E/kT}$, onde k é a constante de Boltzmann. Nota-se na equação(6), A diferença de energia ΔE entre os subníveis desdobrados pelo efeito Zeeman é igualmente proporcional à frequência $h\upsilon_0$:

$$\Delta E = h\upsilon_0 \tag{6}$$

Ainda sobre a equação(6), temos que:

- h é a constante de Planck em J/s
- $\upsilon_0 = \omega_0/2\pi$

Um agrupamento de *spins* de natureza diversa determinados pelos valores de γ sobre a presença de um campo magnético B_0 irá absorver energias de frequências desiguais que são localizados na faixa de radiofrequência - RF. No momento em que as frequências de oscilação forem semelhantes à respectiva frequência de precessão, ocorrerá o fenômeno descrito como ressonância magnética. No entanto, devido cada núcleo possuir um dado γ e logo, uma determinada frequência angular, o que permite um estudo direcionado. Por haver um número de maior de núcleos com *spins* alinhados paralelamente ao campo magnético, por tanto, sempre haverá um curto excesso nessa direção que gera um momento magnético resultante, cujo é denominado de vetor de magnetização. A ressonância magnética depende diretamente da manipulação do vetor da magnetização efetiva, quando em maior magnitude

em altos campos magnéticos, ocasiona um melhor sinal, que consequentemente provê uma qualidade de imagem superior.

Em uma ressonância magnética, no decorrer de um pulso de radiofrequência, os prótons absorvem energia em frequência específica. Logo após o pulso de radiofrequência, os prótons expelem uma energia na mesma frequência, sendo chamada de frequência de Larmor, que por sua vez, proporciona uma mudança no seu alinhamento paralelo já descrito. Desse modo, para ser possível a ressonância magnética é necessário aplicar ao meio magnético, um pulso de radiofrequência igual a frequência de Larmor do hidrogênio. Por ser bastante sensível ao campo magnético devido ao grande valor de sua razão giromagnética γ e ter um *spin* $\frac{1}{2}$, o uso dos parâmetros de frequência do hidrogênio torna-se uma escolha adequada.

3.1.1 Ponderação de Imagem

Imagens com ponderação em diversos tecidos podem ser adquiridos através de uma imagens de ressonância magnética - IRM, essa ponderação irá depender do Tempo de Repetição - TR sendo o intervalo entre um pulso de 90° e o pulso seguinte, juntamente do Tempo de Eco - TE, sendo o tempo entre o pulso de RF de 90° e a recepção do sinal pela bobina recaptura com sinal em sua amplitude máxima.

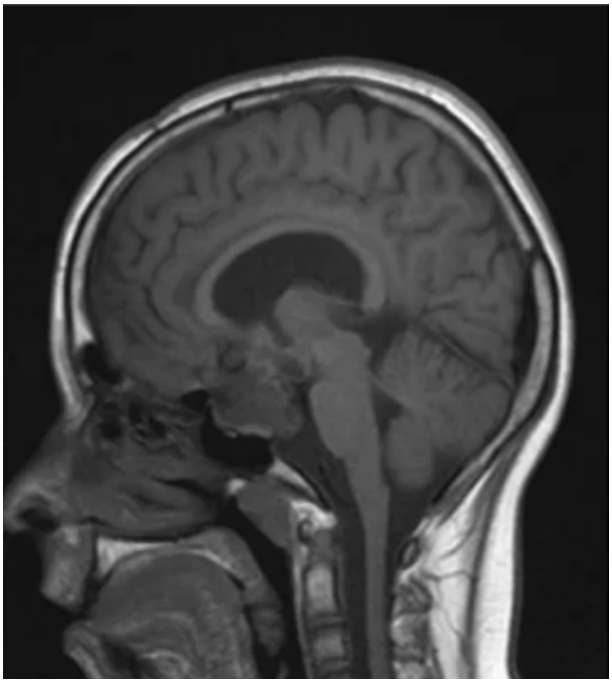
A água e a gordura possuem uma leve diferença na frequência de precessão, embora as duas possuam em sua composição os átomos de hidrogênio, sendo essa diferença entre as precessões que garantem o contraste nas IRM.

A gordura tem seu tempo de precessão menor em relação a água, além do seu tempo de recuperação do eixo longitudinal ser mais rápido. O TR determina o grau de ponderação T1 e TE determinar o grau de ponderação que irá ser demonstrado na imagem, por tanto, para uma imagem ponderada em T1, precisa-se dos seguintes parâmetros:

- $\text{TR} < 700\text{ms}$
- $5\text{ms} \leq \text{TE} \leq 25\text{ms}$

Com o Tempo de Repetição curto possibilitamos que a água não possa ter tempo o suficiente para recuperar sua magnetização longitudinal total, da mesma forma, com o Tempo de Eco curto minimiza-se os efeitos de ponderação, dando contraste para o tecido rico em gordura. Usando como ponto de referência a água em uma imagem T1, podemos analisar na Figura 3 estruturas ricas em água, com um baixo contraste, como é o caso do córtex cerebral, porém se analisarmos a substância branca que é rica em mielina, por ser um tipo de gordura pode-se notar um contraste maior.

Figura 3 – Imagem Ponderada em T1



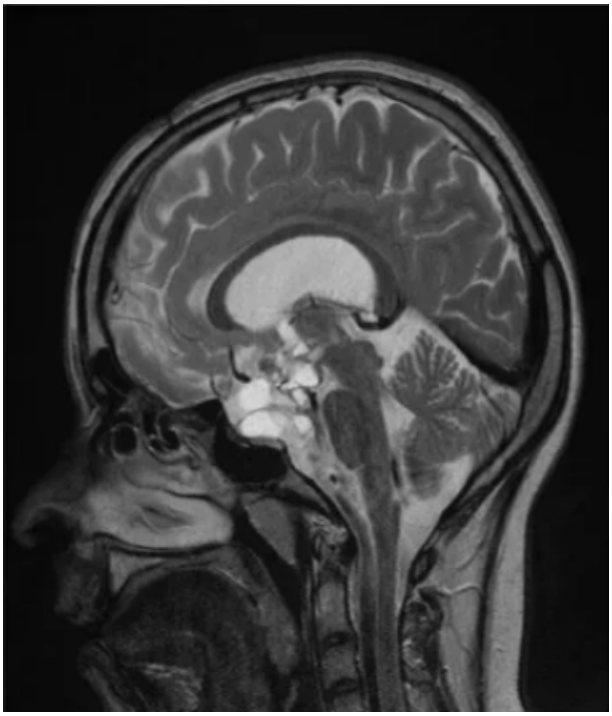
Fonte: Herculys Douglas

Para imagens ponderadas em T2, o TR necessita ser bem mais alto, em comparação ao TR de uma imagem ponderada em T1, para que assim a água tenha tempo o suficiente para recuperação de sua magnitude longitudinal e possa absorver o máximo de energia do pulso seguinte de radiofrequência, sendo os parâmetros:

- TR $>$ 2000ms
- TE $>$ 60ms

Em imagens ponderadas em T2, temos tecidos e substancias ricas em água, com um contraste maior, devido ao sinal de ressonância da água no ponderamento T2, que pode ser notado na figura 4, com o córtex da substancia cinzenta no encéfalo emitindo contraste.

Figura 4 – Imagem Ponderada em T1



Fonte: Fonte: Herculys Douglas

A tabela a seguir mostra alguns tecidos e substancias que podem ter seu sinal de ponderação em T1 e T2 classificados como hipersinal (quando possui tonalidade clara) e hipossinal (quando possuem tonalidade escura).

Tecido / Substância	Ponderação em T1	Ponderação em T2
Água	Hipossinal	Hipersinal
Ar	Hipossinal	Hipossinal
Calcificação	Hipossinal	Hipossinal
Cistos c/ líquido proteináceo	Hipersinal	Hipossinal
Edemas	Hipossinal	Hipersinal
Esclerose	Hipossinal	Hipersinal
Gordura	Hipersinal	Hipossinal
Hemangioma	Hipersinal	Hipersinal
Hematoma Agudo	Hipossinal	Hipossinal
Hemorragia Subaguda	Hipersinal	Hipersinal
Infecções	Hipossinal	Hipersinal
Limpoma Intra-ósseo	Hipersinal	Hipossinal
Melanina	Hipersinal	Hipossinal
Osso Cortical	Hipossinal	Hipossinal
Sangue Alto Fluxo	Hipossinal	Hipossinal
Sangue em Baixo Fluxo	Hipersinal	Hipersinal

Tabela 4 – Tabela de Hipossinal e Hipersinal de Tecidos

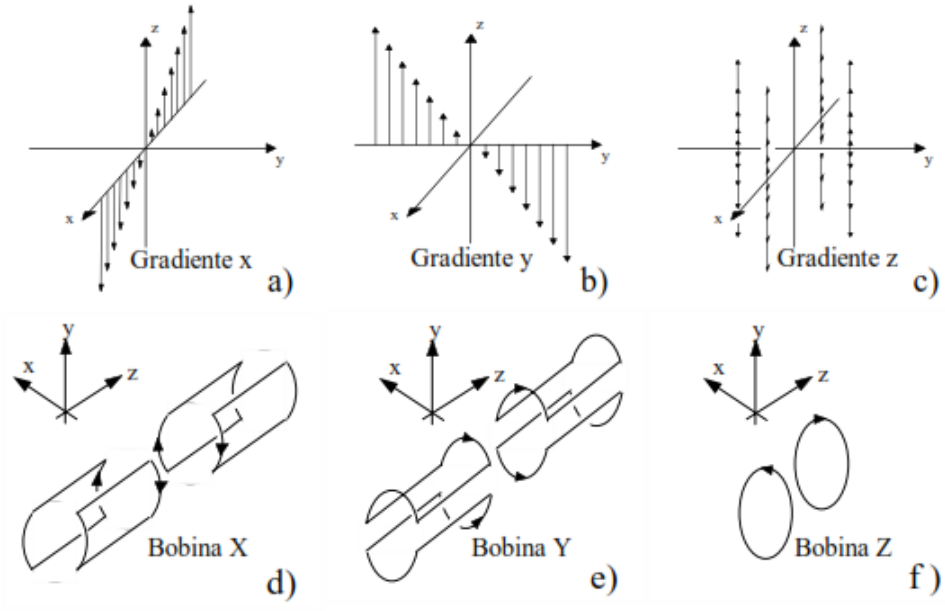
3.1.2 Plano Anatômico e Gradientes

O plano anatômico pode ser compreendido como uma representação em três eixos da anatomia humana ou animal, com o intuito de descrever a localização de estruturas e movimentos. Na aplicação em Ressonância Magnética podemos levar em consideração o sistema de bobinas gradiente de campo de ressonância, que produz um gradiente de campo magnético que percorre o corpo em questão.

De uma forma simplificada entende-se que dentro de um aparelho de ressonância magnética, possui 3 bobinas, sendo **x, y** e **z**, cada uma relacionada ao seu respectivo eixo. os gradientes são responsáveis pela codificação em fase e codificação em frequência, com o intuito da localização espacial ao longo eixo mais curto, tanto na localização espacial de um eixo ao longo do eixo vertical da anatomia, respectivamente. Além de ser responsável pela localização de um corte. Ao analisar a Figura 5, podemos compreender que:

- gradiente x: altera a potência do campo magnético em relação ao eixo x horizontal, proporcionando a seleção dos cortes sagitais.
- gradiente y: altera a potência do campo magnético em relação ao eixo y vertical, proporcionando a seleção dos cortes coronais
- gradiente z: altera a potência do campo magnético em relação ao eixo z, proporcionando a seleção dos cortes axiais.

Figura 5 – Gradientes Gx, Gy, Gz

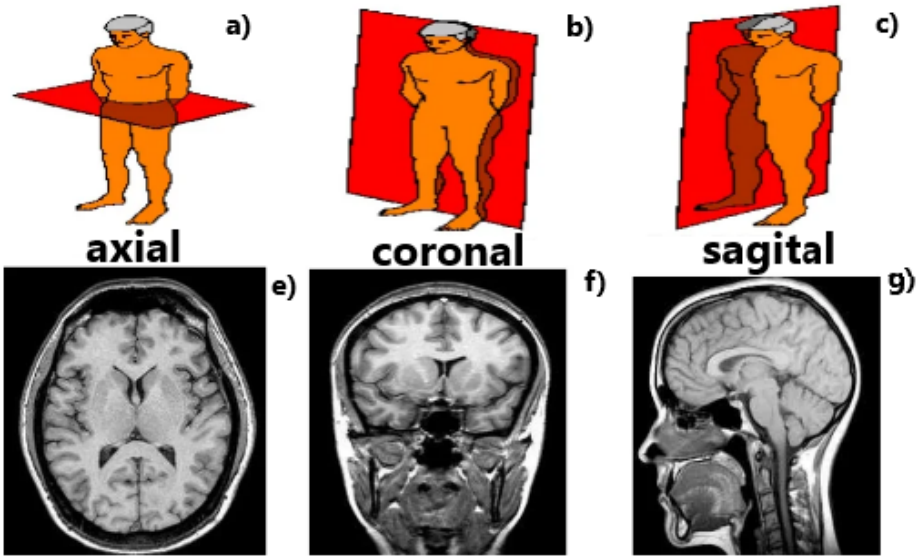


Fonte: Autor. André Luis Bonfim Bathista e Silva

Como exemplo de resultado de uma extração de imagens de ressonância magnética de um crânio humano, nos três cortes, temos a figura 6.

- plano axial: podendo ser visto na Figura 6 (a), sendo o plano usado como referência para a obtenção da imagem de ressonância magnética da Figura 6 (e).
- plano coronal: podendo ser visto na Figura 6 (b), sendo o plano usado como referência para a obtenção da imagem de ressonância magnética da Figura 6 (f).
- plano sagital: podendo ser visto na Figura 6 (c), sendo o plano usado como referência para a obtenção da imagem de ressonância magnética da Figura 6 (g).

Figura 6 – Representação de Diferentes Cortes em RM



Fonte: Adaptado de Herculys Douglas

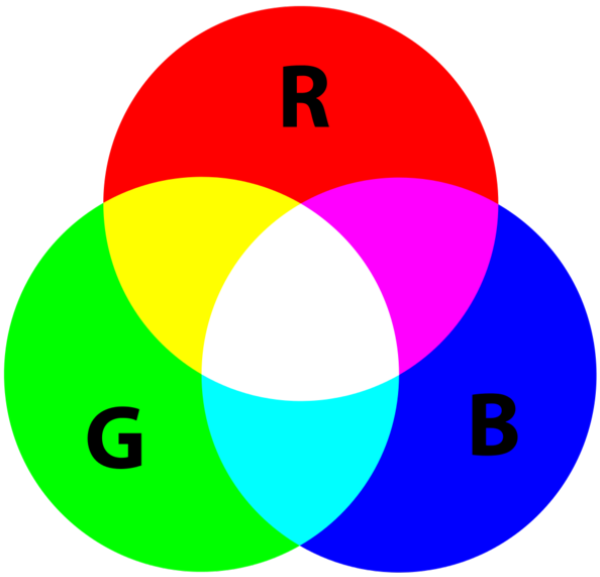
3.2 Imagem Digital

Uma imagem digital é uma representação em conjunto de valores discretos através de um conjunto de *pixels*, podendo através de sua representação digital o armazenamento, tratamento, reprodução, transferência e edição. Diversos valores podem ser atribuídos a esses *pixels*, e dependendo dos valores dado a cada pixel, a imagem digital pode ser catalogada como Imagem em RGB, Imagem em Escala de Cinza, Imagem Binária, entre outras.

3.2.1 Imagem em RGB

Em uma imagem em RGB, cada *pixel* tem três canais de cores conhecidos como RGB *(Red, Green e Blue)*, a cada canal pode ser atribuído um valor de 0 a 255.

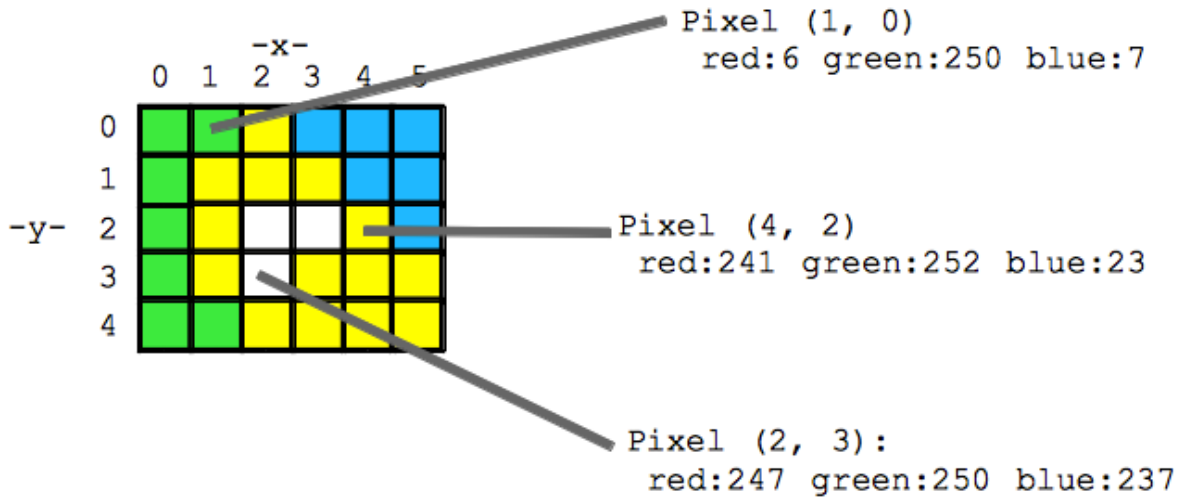
Figura 7 – Modelo de Representação RGB



Fonte: Ferlixwangg

Para formar cores além do vermelho, verde e azul, necessitamos de intensidades diferentes para cada canal, como representado na decomposição dos *pixels* em uma matriz bidimensional *(x,y)* na figura 8.

Figura 8 – Matriz de Pixel RGB



Fonte: Editado de Stanford.edu

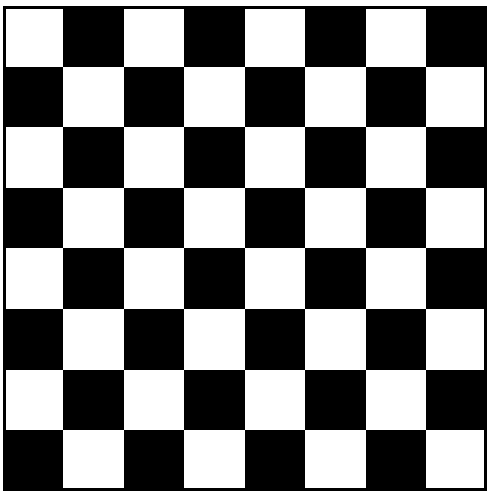
3.2.2 Imagem Binaria

Uma imagem Binária, é uma representação de imagem no qual cada pixel possui apenas dois valores possíveis, podendo ser representado por preto e branco. Tendo como

referência a figura 9, podemos verificar que os *pixels* da figura são alternados entre branco e preto, se organizamos os *pixels* em uma sequência iniciando no índice $n = 1$, temos:

- pixel branco para $2n - 1$
- pixel preto para $2n$

Figura 9 – Imagem Binaria



Fonte: Autor

3.2.3 Imagem em Escala de Cinza

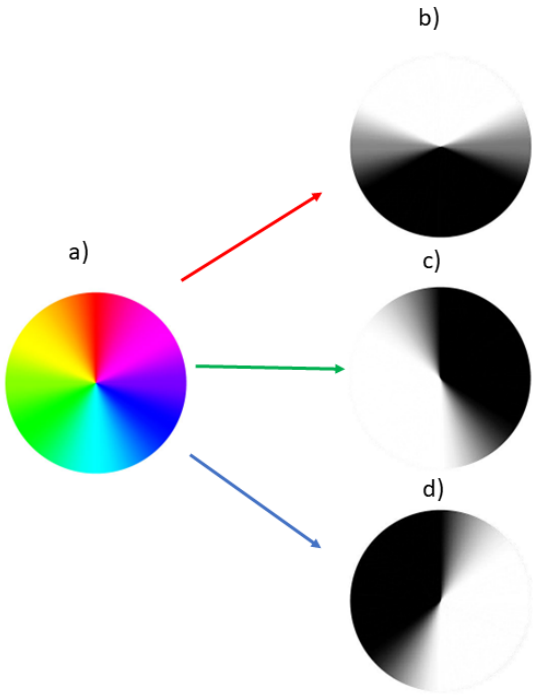
Em imagens digitais em escala de cinza o valor do *pixel* varia em uma escala de 0 a 255, onde 0 é representado pela cor preta e 255 sendo um *pixel* branco, e os valores entre 0 e 255 sendo uma variação em escala de cinza.

Figura 10 – Níveis de Intensidade de Cinza

Fonte: Utilização de Técnicas de *Data Augmentation* em Imagens: Teoria e Prática

Para converter uma imagem em RGB em uma imagem em escala de cinza, podem ser usadas algumas abordagens, já que a escala de cinza varia entre 0 e 255, sendo a mesma variação de intensidade em uma escala RGB, pode-se usar como referência a intensidade de uma das três cores primárias, como vemos na figura a abaixo:

Figura 11 – Método de Conversão de RGB para Escala de Cinza



Fonte: Autor

Entretanto o resultado da imagem pode ser muito variado, dependendo de qual referencial de intensidade de *pixel* usar, ao analisar a figura 11, temos 3 resultados de uma mesma imagem a). sendo a figura 11 b) a conversão em escala de cinza usando como referencial a intensidade da cor vermelha na Figura 11 a). a Figura 11 c) o uso de referencial da intensidade da cor verde e a figura 11 d) o uso da intensidade da cor azul.

Com tudo, uma abordagem mais inteligente pode ser o uso das três cores primárias do RGB na participação na convenção dos *pixels* para escala de cinza. Com o uso das três cores primárias temos duas abordagens. A primeira consiste em uma média simples, usando como intensidade do *pixel* o resultado inteiro.

$$Intensidade = (R + G + B)/3$$

sendo:

R: a intensidade da componente vermelha do *pixel*;

G: a intensidade da componente verde do *pixel*;

B: a intensidade da componente azul do *pixel*.

Onde como resultado temos como exemplo a figura 12 b).

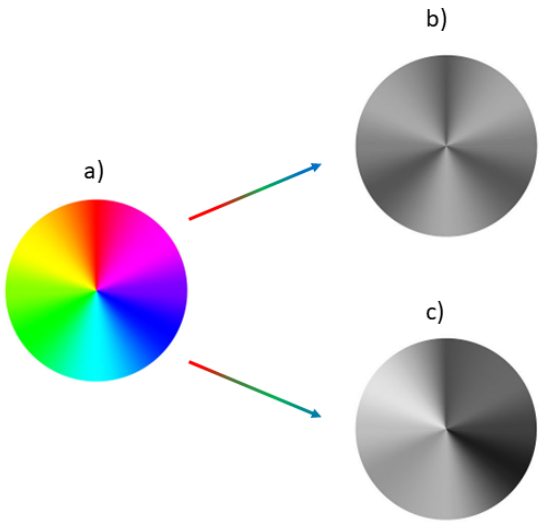
Porém, a forma com que o olho humano capta essa intensidade de iluminação é diferente para cada cor primária levando em consideração vários fatores, porém para o modelo adotamos uma abordagem que mesmo não sendo a mesma empregado pelo olho humano, é uma abordagem que fornece pesos razoáveis para a conversão, empregando alguns valores de intensidade para cada canal de cor. [1] [2]

- 59% de intensidade da componente de cor verde
- 30% de intensidade da componente de cor vermelha
- 11% de intensidade da componente de cor azul

Então para o resultado visto na figura 12 b) temos:

$Intensidade = 0{,}30R + 0{,}59G + 0{,}11B$

Figura 12 – Método de Conversão de RGB para Escala de Cinza por Média



Fonte: Autor

[1] O olho humano interpreta as intensidades das cores de modo que depende de diversos fatores, onde levam em consideração desde de iluminação, comprimento de onda, até fatores gama, como uma forma de facilitar essa transformação de imagens em cores para imagens em escala de cinza foram empregados valores que são comumente aplicado a sistemas de vídeo.

[2] Charles Poynton, *The magnitude of nonconstant luminance errors in Charles Poynton, A Technical Introduction to Digital Video*. New York: John Wiley & Sons, 1996.

3.3 Conceitos Básicos de Vídeo

Segundo o (Oxford Languages, 2022) um vídeo pode ser descrito como, uma técnica de reprodução eletrônica de imagens em movimento; conjunto de dispositivos que reproduzem a imagem transmitida.

3.3.1 Configurações de Vídeo

Além de quais aparelhos são usados para capturar um vídeo e quais dispositivos serão usados para reproduzi-los, vários fatores podem definir a qualidade de um arquivo de vídeo, como:

- resolução: é a altura e largura de um vídeo em *pixels*, uma resolução maior, pode proporcionar uma imagem em maior qualidade, entretanto, duplicar a resolução de uma vídeo aumenta o tamanho do arquivo de vídeo em quatro vezes.
- proporção: relação entre altura e a largura de um quadro, algumas das proporções usadas são, 4:3, 16:9 e 21:9.
- taxa de bits: é a quantidade de dados em *bits*, que são codificados para formar um segundo de vídeo, medida em geralmente em kilobits por segundo (Kbps), quanto maior a taxa de *bits*, maior é qualidade de vídeo, devido a quantidade de informação de cada segundo.
- taxa de quadros : é a quantidade de imagens contidas em um único segundo de um vídeo, popularmente conhecida como *Frames* por Segundo (FPS), quanto maior o FPS, maior é a fluidez da movimentação de um vídeo.
- codec: é o algoritmo de compressão e descompressão de mídia de vídeo, dependendo do algoritmo, pode haver perda de informação no processo, possibilitando a exibição ou transferência de mídia em meios com taxa de transferência menores, entretanto, há também algoritmos com baixa ou perda nula.

3.3.2 Formato de Vídeo

Dependendo dos fatores descritos na seção 3.2.1, temos diversos formatos de vídeo, cada formato contendo suas peculiaridades e sendo melhor adaptado para o tipo de transmissão de vídeo, dispositivo de reprodução ou finalidade.

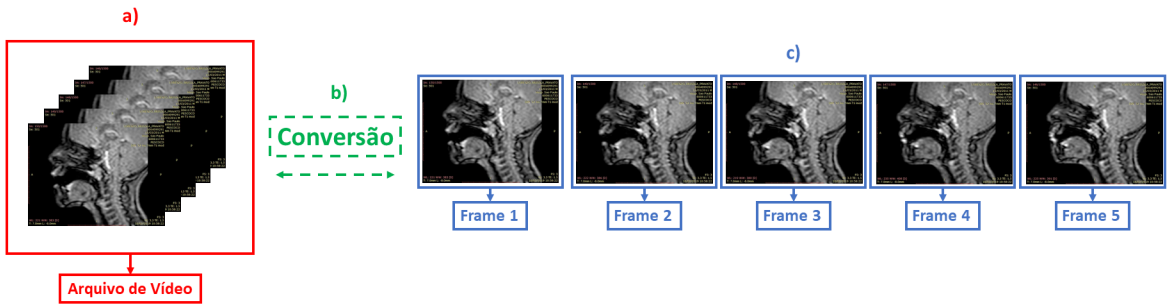
Fomato	Codec	Resolução	Taxa de Bits
AVI	MPEG1	768*576	1.8M
	MPEG2 MP@HL	1920*1080	40M
	MPEG-4 SP@HL 3.0	1920*1080	40M
	MPEG-4 ASP@HL 4.0	1920*1080	40M
MP4	MPEG-4 SP@HL 3.0	1920*1080	40M
	MPEG-4 ASP@HL 4.0	1920*1080	40M
MKV	MPEG-4 SP@HL 3.0	1920*1080	40M
	MPEG-4 ASP@HL 4.0	1920*1080	40M

Tabela 5 – Características de Formatos de Vídeo

3.3.3 Conversão de Vídeo em Imagem

Sendo um vídeo um conjunto de imagens sequenciais que pode ou não está acompanhado de efeitos sonoros, podemos representar um arquivo de vídeo como vários arquivos separados em formato de imagem.

Figura 13 – Modelo de Conversão de Vídeo em Imagens



Fonte: Autor

A figura 13 demonstra em um modelo simplificado a conversão de um arquivo de vídeo para vários arquivos de imagens independentes, onde, a) um arquivo único de vídeo é inserido em um *software* ou *framework* de conversão b) o *software* converte o arquivo de vídeo que, c) que como resultado gera *frames* independentes.

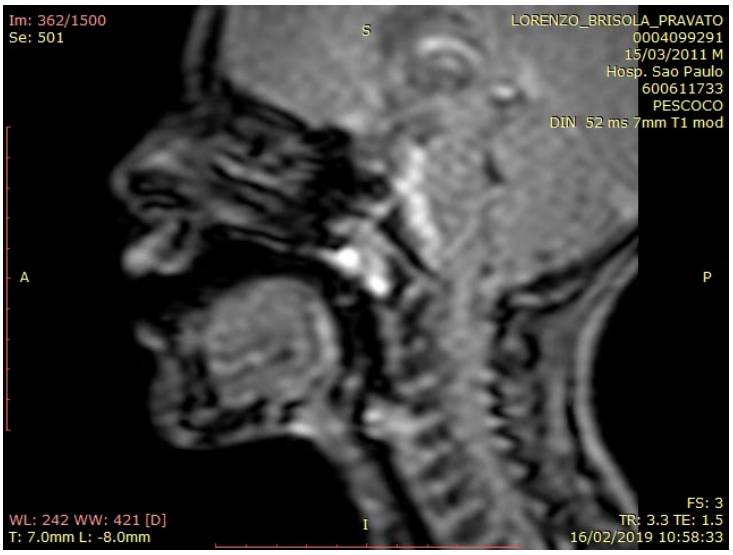
3.4 Segmentação de Imagem

3.4.1 Segmentação Tradicional

A segmentação de imagem pode ser compreendida como a divisão de uma imagem em diversas regiões de modo a classificar uma imagem digital *pixel* a *pixel*, essa abordagem

amplia a análise de uma imagem, tendo o intuito de simplificar uma imagem complexa e detectar regiões de interesse. Entre as diversas técnicas de segmentação uma das mais clássicas é chamada de *Thresholding*, nesse método o algoritmo aplicado segmenta a imagem usado como parâmetro uma constante de intensidade c que pode assumir valores entre $(0 \leq c \geq 255)$. Após a definição da constante de intensidade o algoritmo analisa cada *pixel* da imagem e caso o *pixel* analisado possua intensidade menor ou igual a definida pela constante, então será convertido para preto, se possuir intensidade maior é convertido para branco. Possuindo como resultado uma imagem binária.

Figura 14 – Imagem Antes da Segmentação



Fonte: *A Dataset Of Word Sequences Through Mri* (2019)

Figura 15 – Imagem Depois da Segmentação

Fonte: *Adaptado de A Dataset Of Word Sequences Through Mri*(2019)

A imagem acima é segmentada usando como constante de intensidade 45, os *pixels* com intensidade abaixo ou igual a 45 foram categorizados como preto, e os *pixels* com intensidade maior que 45 foram categorizados como branco. A Figura 15 torna-se uma imagem simplificada em *pixels* pretos e brancos, em comparação com a Figura 14, a imagem perde bastante informações, e regiões que podem ser de interesse em algumas análises podem não serem detectadas, entretanto, ainda pode ser compreendida como uma imagem do trato vocal.

3.4.2 Segmentação Semântica e Segmentação de Instâncias

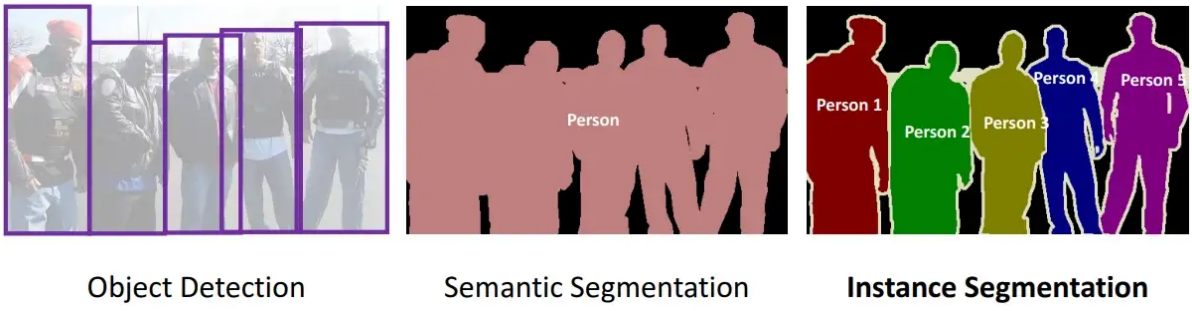
Com os avanços da tecnologia a detecção de objeto obteve mais áreas de atuação e novos modos de implementações junto a segmentação de imagem.

Detecção de Objetos : como resultado são criadas caixas delimitadoras, onde localizam um ou vários objetos dentro de uma imagem.

Segmentação Semântica : a classificação é feita a nível de pixel, tendo como resultado uma imagem onde os objetos são relacionados a classe.

Segmentação de Instâncias : além de cada pixel ser classificado na imagem, são criados instâncias separadas para cada objeto, onde objetos semelhantes são relacionados a uma mesma classe.

Figura 16 – Comparação entre: detecção de objetos, segmentação semântica e segmentação de instâncias



Fonte: *Single Stage Instance Segmentation — A Review*

3.5 Aprendizado de máquina e Aprendizado Profundo

O aprendizado de máquina e aprendizado profundo podem ser compreendidos como camadas de uma única área, no aprendizado de máquina *(Machine Learning)*, a abordagem é filtrar certos dados e encontrar padrões, podendo ser divida em 3 campos:

- Aprendizado supervisionado: nesse campo o algoritmo recebe os dados de entrada e as variáveis de respostas ou classes de resposta, e o objetivo do algoritmo é aprender como chegar nas saídas corretas, analisando os dados de entradas e respostas que possui.
- Aprendizado não supervisionado: possui como desafio encontrar a saída desejada através das entradas, mas nesse caso não recebe os dados de resposta, sendo em sua grande maioria associados a problemas mais complexos.
- Aprendizado por reforço: emprega recompensas nas ações de um determinado agente, treinando o agente para escolher a melhor resposta para uma ação levando em consideração experiências realizadas.

O aprendizado profundo *(Deep Learning)* se concentra em funções cerebrais artificiais, que simulam os neurônios reais que conhecemos, os conjuntos de algoritmos usados no aprendizado profundo trabalham como redes neurais que analisam os dados recebidos continuamente com o intuito de compreender esses dados e entender seus padrões para prever a melhor resposta. Além disso, possui o diferencial de analisar grandes volumes de dados, diferente do aprendizado de máquina que esses dados são reduzidos.

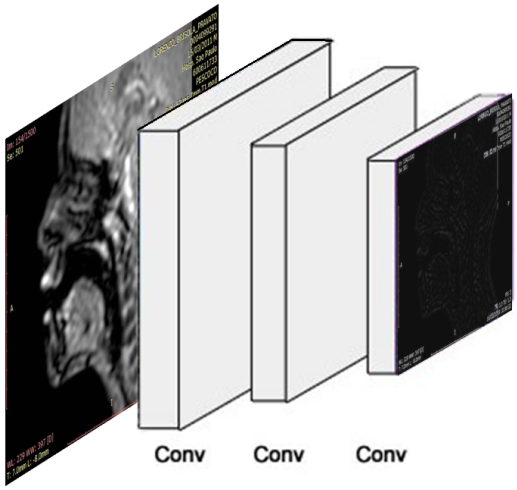
3.6 *Mask R-CNN* como método de segmentação

A técnica *Mask R-CNN* foi implementada em 2017 por cientistas de dados e pesquisadores do FAIR - *Facebook AI Research*, sendo uma extensão da arquitetura de detecção de imagem, *Faster R-CNN*.

3.6.1 Comprometes da Arquitetura *Mask R-CNN*: *Backbone*

A rede *backbone* ilustrada na Figura 17 – (Extrator de Características - *Backbone*) é rede neural localizada nas etapas iniciais da arquitetura, responsável por extrair as características de um objeto. As camadas convolucionais iniciais dessa rede neural são usadas para extração em baixo nível como contornos e bordas, enquanto as camadas convolucionais posteriores extraem características específicas dos objetos em alto nível.

Figura 17 – Extrator de Características - *Backbone*



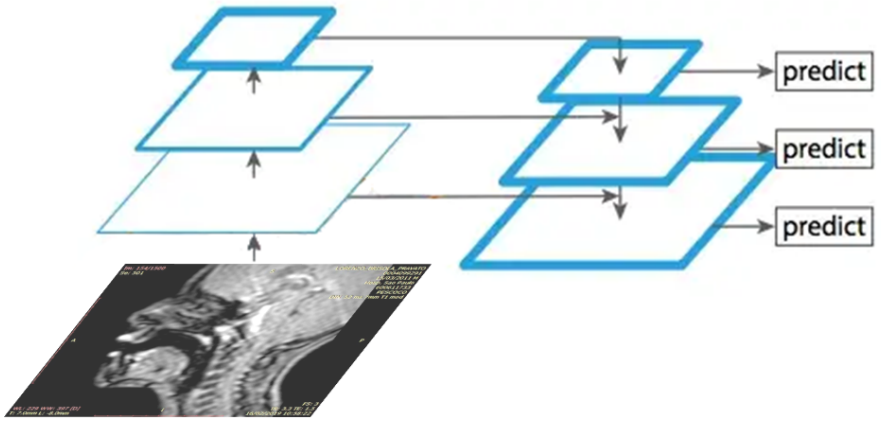
Fonte: *Adaptado de Object Detection Papers: Mask R-CNN*

Ao percorrer a rede *backbone* a imagem de entrada é convertida de 1024x1024px x3(RGB) para um mapa de características com o formato 32x32x2048. Sendo o módulo de entrada para as etapas posteriores.

3.6.2 Comprometes da Arquitetura *Mask - RCNN: Feature Pyramid Network* (FPN)

Como uma forma de melhorar o extrator de características a introdução do FPN é uma implementação na *Mask-RCNN*, sendo uma melhor representação dos objetos em múltiplas escalas. representado na Figura 18 - FPN, como um duas pirâmides lado a lado, de modo que, a primeira pirâmides extrai os características de alto nível dos dados de entrada e envia para as camadas inferiores da segunda pirâmide, com isso, fazendo com que todos os níveis possuam acesso a recursos de níveis inferiores e superiores.

Figura 18 – *Feature Pyramid Network* - FPN

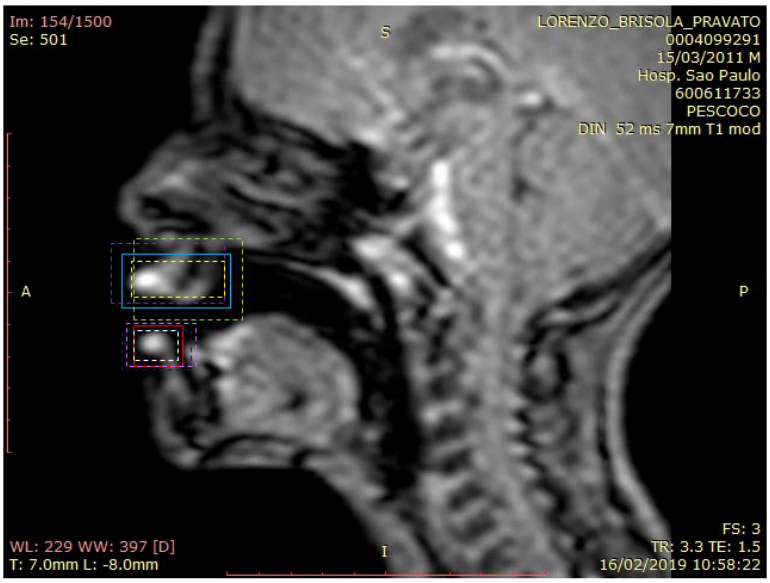


Fonte: Adaptado de *Object Detection Papers: Mask R-CNN*

3.6.3 Comprometes da Arquitetura *Mask - RCNN: Region Proposal Network* (RPN)

Nessa etapa, uma janela deslizante chamada de âncora percorre toda a imagem, com o intuito de determinar regiões que possam obter o objeto em questão a ser detectado, representado na Figura 19, no entanto, o RPN pode gerar em uma única imagem cerca de 200 mil âncoras de diversas proporções e tamanhos que podem se sobrepor, para varrer o máximo possível de possibilidades e no final da etapa conseguir escolher a melhor âncora entre as geradas.

Figura 19 – *Region Proposal Network* - RPN



Fonte: Adaptado de *Object Detection Papers: Mask R-CNN*

O *Region Proposal Network* produz duas saídas para cada âncora, sendo elas:

Classe Âncora : detecta se a classe é de *foreground* (FG) ou *background* (BG), caso seja uma classe FG a âncora indica um objeto;

Refinamento da caixa delimitadora : no caso de uma âncora FG, pode não estar corretamente centralizada sobre o objeto o RPN estima um delta para o refinamento da âncora e seu melhor ajuste ao objeto;

Utilizando as previsões RPN, são selecionadas um conjunto de âncoras que possuem possíveis objetos, essas âncoras são refinadas em tamanho, e caso, hajam múltiplas âncoras que se sobreponham semelhantemente, a âncora com maior pontuação BG é mantida e as demais semelhantes são descartadas. Posteriormente são geradas propostas finais sendo chamadas de *Regions of Interest*(ROIs).

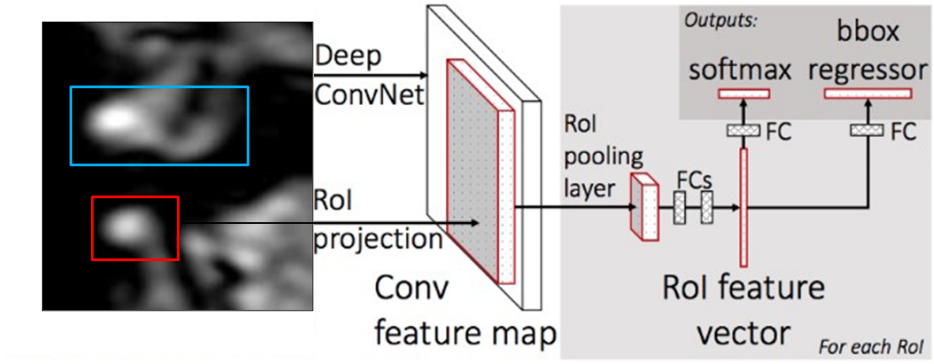
3.6.4 Comprometes da Arquitetura *Mask - RCNN: region of interest Classifier & Bounding Box Regressor*

Após receber como entrada as ROIs geradas pela RPN, a etapa ilustrada na Figura 20 é executada e gera duas saídas para cada uma das ROIs:

Classe : possuindo uma rede neural mais profunda que retorna uma classe específica do objeto além de possuir a capacidade de distinguir o objeto de *background*;

Refinamento da caixa delimitadora : agindo semelhante ao refinamento que já ocorre na etapa RPN, como o intuito de refinar ainda mais progressivamente a caixa delimitadora do objeto;

Figura 20 – Etapa (ROI) & *Bounding Box Regressor*

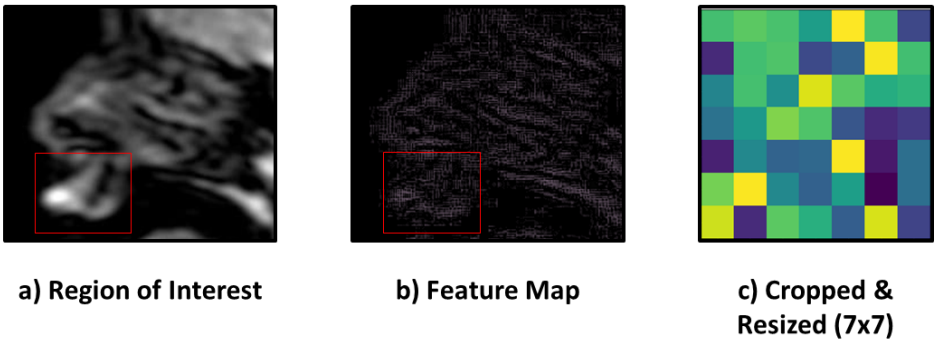


Fonte: Adaptado de *Object Detection Papers: Mask R-CNN*

3.6.5 Comprometes da Arquitetura Mask - RCNN: ROI Pooling

Na arquitetura *Mask-RCNN*, os classificadores possuem dificuldades ao manipular imagens com variações de resoluções, em geral necessitando de imagens com resoluções fixas. Então, já que na etapa de geração de ROIs, pode-se obter ROIs de diferentes tamanhos, se faz necessário o uso da técnica *ROI Pooling*.

Figura 21 – Tecnica ROI *Pooling*



Fonte: Adaptado de *Object Detection Papers: Mask R-CNN*

Region Of Interest : podendo ser visto na Figura 21 (a), a imagem possui uma ROI sendo um lábio.

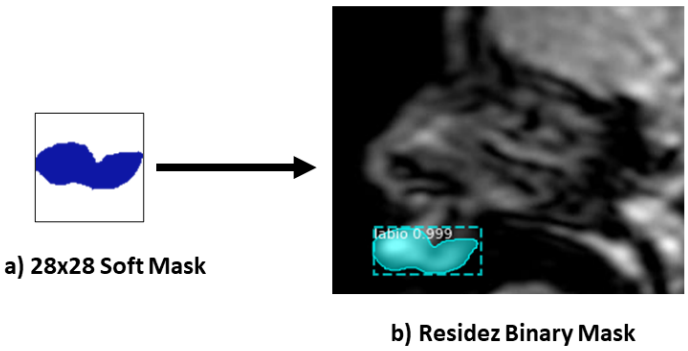
Feature Map : podendo ser visto na Figura 21 (b), o *Feature Map* que possui as características da imagem.

Cropped & Resized : já na Figura 21 (c), a técnica *ROI Pooling* extrai o ROI do *Feature Map* e redimensiona para um tamanho fixo, no exemplo sendo usando um tamanho 7x7.

3.6.6 Comprometes da Arquitetura *Mask - RCNN: Segmentation Masks*

A *Segmentation Mask* é gerada através de uma rede neural convolucional que representa as regiões positivais do *ROI Classifier* por uma mascara de segmentação para cada objeto detectado na imagem.

Figura 22 – Segmentation Masks



Fonte: Adaptado de Object Detection Papers: Mask R-CNN

As máscaras como visto na Figura 22 (a), possuem um tamanho 28x28, entretanto, são representadas por números flutuantes, por tanto, possuem mais detalhes que uma

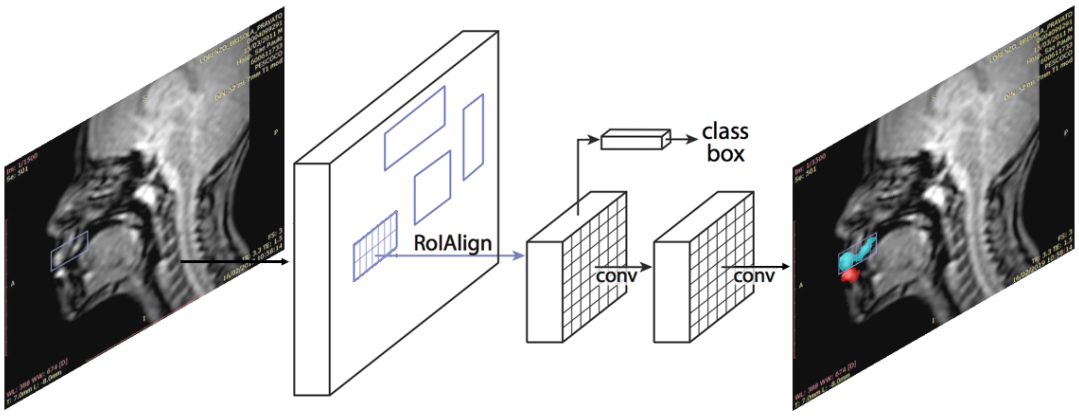
máscara binária. Essa máscara contém informações que indicam quais *pixels* referem-se ao objeto detectado e quais pertencem ao fundo da imagem.

Para produzir as máscaras de segmentação, a *mask r-cnn* utiliza uma sub-rede neural convolucional que recebe uma proposta de caixa delimitadora gerada pela camada RPN junto a uma ROI, para então aplicar uma série de camadas de convolução para extrair características da região específica da imagem com o intuito de gerar através disso uma máscara.

Importante destacar que a *Segmentation Mask* é treinada simultaneamente com as outras camadas do modelo, durante a etapa de treinamento da rede neural, o modelo recebe um conjunto de dados de treinamento e validação já rotulados que consistem em imagens com suas máscaras de segmentação reais. Durante a etapa de validação o modelo avalia as *Segmentation Masks* gerados pelo rede neural com as máscaras reais para avaliar a performance do modelo.

Em um modo geral, a arquitetura *Mask R-CNN* pode ser resumida em três principais etapas.

Figura 23 – *Mask R-CNN*



Fonte: Editada de Pinho e Pontes(2008, P.8)

Etapa 1 : gera um mapa da imagem após aplicar um extrator de *features*;

Etapa 2 : treinar uma rede neural cujo o objetivo é gerar uma proposta de região do objeto;

Etapa 3 : gera uma rede para classificar, localizar o objeto e cria uma mapa de segmentação

3.7 *Data Augmentation*

A obtenção de resultado aceitáveis a partir do *deep learning*, dependem de diversos fatores, um algoritmo otimizado, linguagem de programação atual e adequada, disponibili-

dade de recursos computacionais, entretendo além disso, principalmente quando se trata de um grande fluxo de variáveis e deseja um maior índice de acurácia, se faz necessidade de um *dataset* com grande volume e variedade de dados.

Uma rede neural artificial em grande parte é eficaz quando é usada para detectar ou analisar dados que já fizeram parte de seu treinamento, porém, o grande desafio dos algoritmos de *deep learning* são a análise de dados não treinados, os chamados dados de teste, para aumentar a eficiência e a generalização de um algoritmo de *deep learning*, podemos aplicar técnicas de aumento de dados, proporcionando ao algoritmo o treinamento com dados diversificados, podendo também gerar dados para serem usado na fase de testes para verificar os resultados diversos do modelo em dados não previamente treinados e fora de um ambiente previsto. Um dos meios de ampliar e gerar novos dados a partir de dados já coletados é por meio de técnicas de *data augmentation*

As técnicas *data augmentation* podem ser aplicadas em diversos tipos de dados, entretanto os dados usados na implementação desse trabalho, são exclusivamente *frames*, por tanto, as técnicas discutidas serão em razão desses tipos de dados. Sendo elas a aplicação de filtros como:

- Ruído RGB
- Inversão de cores
- *Gaussian Blur*
- Canais de Cores

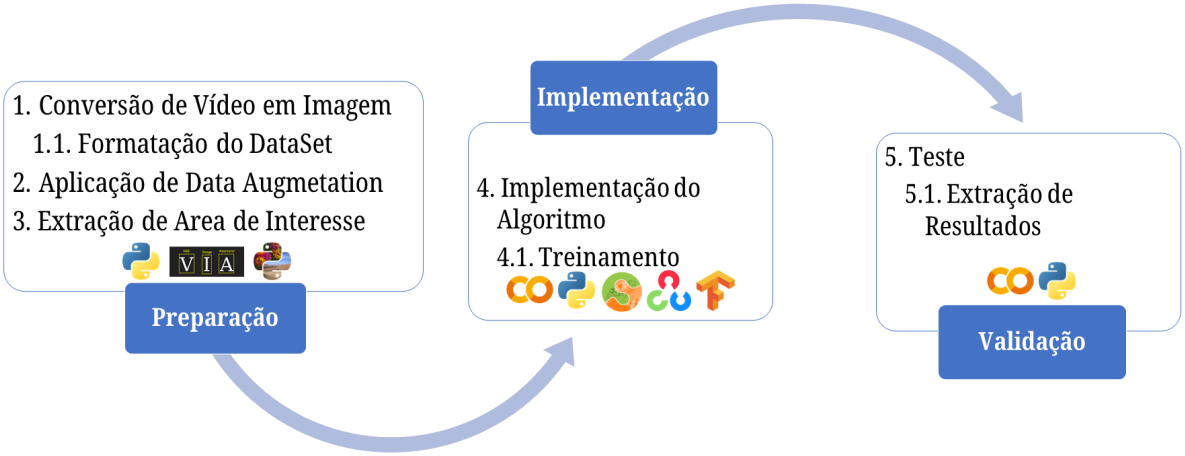
4 METODOLOGIA

A seguir os métodos, processos e recursos a serem seguidos e foram adotados para a execução do projeto proposto.

4.1 Etapas do Projeto

Para a obtenção do modelo de segmentação de lábios em imagens de ressonância magnética, a execução pode ser catalogada em três partes principais, sendo, Preparação, Implementação e Validação, dispostas na figura abaixo:

Figura 24 – Etapas do Projeto



Fonte: Autor

As etapas foram aplicadas em modelo cascata, ou seja, uma após a outra, sendo a etapa inicial a preparação, a etapa responsável pela organização prévia dos dados que serão usados nas demais etapas, sendo subdivida da seguinte forma:

1 Conversão de Vídeo em Imagem

1.1 Formatação do *Dataset*

2 Aplicação de *Data Augmetation*

3 Extração de Área de Interesse

4.1.1 Etapa de Preparação: 1. Conversão de Vídeo em Imagem

Os dados iniciais para a concepção do projeto, são provenientes de um conjunto de dados em video que estão a disposição publica na plataforma de dados científicos e

tecnológicos IEEE DataPort, mantida pela IEEE *(Institute of Electrical and Electronics Engineers)*, os dados fazem parte do *A Dataset Of Word Sequences Through Mri.*

Para serem inseridos no projeto, os dados de video precisaram passar por uma trateamento de dados, com o intuito de converter um video em especifico para um conjunto de imagens.

Descrição	Video
Resolução	640x480
Formato	wmv
Duração	150 segundos
Taxa de Quadros	10FPS

Tabela 6 – descrição das características dos dados de video

A extração de *frames* de um video é relativamente simples, tendo em vista que na tabela 6 - onde descreve as características dos dados de video, nota-se que, o video possui 150 segundos com uma taxa de 10 *frames* por segundo, então, é correto afirmar que levando em consideração isso, usando o vídeo em uma extrator de *frames*, o resultado final será um conjunto de 1500 imagens.

4.1.2 Etapa de Preparação: 1.1 Formatação do DataSet

Inicialmente, para a formação do dataset, o conjunto de 1500 imagens é disposta da seguinte forma:

$$DataSet = Imagem_{train} + Imagem_{val} + Imagem_{test} \tag{7}$$

Junto ao conjunto de dados teste, uma novo conjunto foi atribuído, com imagens de 5 pacientes, formando um conjunto de 100 imagens para teste, com o intuito de analisar o modelo e suas previsões em um conjunto de teste mais variado, com pacientes cujo modelo não recebeu dados na fase de treinamento. Proporcionando então mais um conjunto de imagens de teste.

sendo: $Imagem_{train}$: um conjunto de 1050 usados na fase de treinamento do modelo.

$Imagem_{val}$: um conjunto de 350 imagens usadas na fase de validação do modelo.

$Imagem_{teste_1}$: um conjunto de 100 imagens usadas na fase de teste do modelo.

$Imagem_{teste_2}$: um conjunto de 100 imagens de diferentes pacientes usadas na fase de teste do modelo.

4.1.3 Etapa de Preparação: 2. Aplicação de *Data Augmetation*

Com o intuito de testar o modelo em variáveis dados, esta etapa consiste na aplicação de métodos *Data Augmetation* para ampliar os dados. A ampliação de dados adotado no projeto consiste na aplicação de filtros, onde de maneira geral, um filtro pode ser compreendido como uma transformação em escala de pixel, que consiste na modificação de valores de uma imagem, onde muitos dos casos são usando um mecanismo similar a convolução.

A convolução pode ser descrita como o processo de filtragem que ocorre devido ao deslocamento de uma máscara sobre a imagem, onde este deslocamento ocorre em um conjunto de pixels a serem transformados matematicamente, onde o valor do pixel central da máscara é modificado através de artifícios matemáticos levando em consideração os pixels envoltos pela máscara.

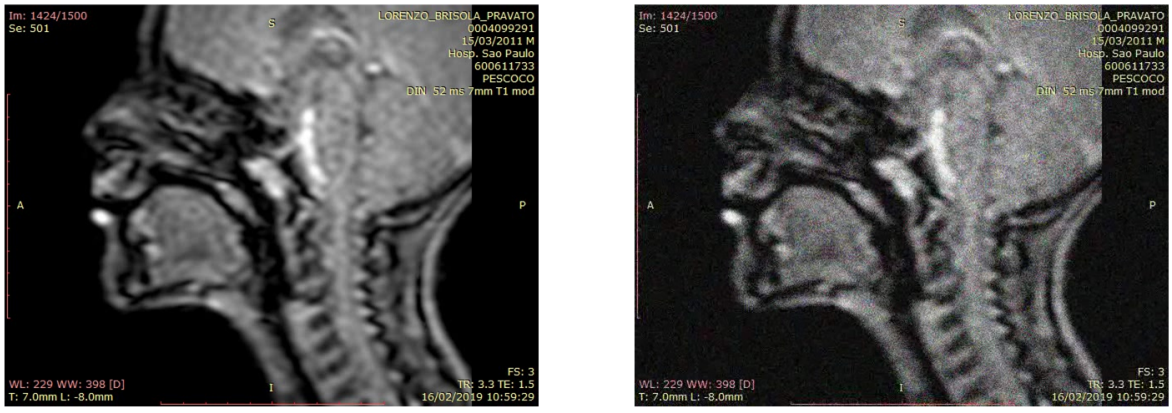
4.1.4 Etapa de Preparação: 2. Aplicação de *Data Augmetation* - Aplicação de Filtros

Para a implementação dos filtros, se fez necessária o uso da biblioteca *Pillow* em conjunto com as bibliotecas nativas do interpretador *Python*. *Pillow* é uma biblioteca que adiciona ao interpretador *Python* recursos de Processamento de Imagem Digital, como aplicação de filtros e carregamento de arquivos de imagens.

4.1.4.1 Filtro Ruido RGB

O filtro ruido RGB, consiste na adição de um valor aleatório dentro de uma faixa pré definida nos canais RGB de uma imagem, simulando a presença de ruído na imagem. Como visto na figura abaixo:

Figura 25 – Comparação entre Imagem Original × Imagem Com Ruido RGB



Fonte: Autor

Uma representação matematicamente simplificada desse filtro pode ser vista como:

$$I(r, g, b) = I(r, g, b) + \epsilon \tag{8}$$

sendo:

$I(r, g, b)$: o valor de cada canal de um pixel na posição (x,y);

ϵ: variável aleatória representando a quantidade de ruido a ser adicionado nos canais da componente (x,y).

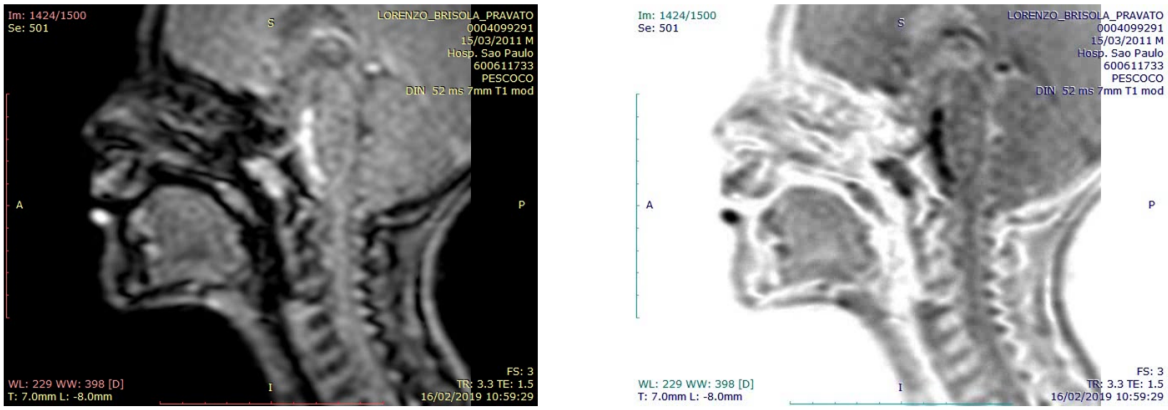
O grau de crescimento de dado usando o filtro ruido RGB é de $2x$ sendo:

x: a quantidade de dados usados como entrada para o uso do filtro.

4.1.4.2 Filtro Inversão de Cores

O filtro de inversão de cores consiste em inverter os valores dos canais RGB de cada pixel de uma imagem, o resultado da aplicação desse filtro e uma imagem em cores opostas a imagem original, sendo possível analisar em um exemplo na figura abaixo:

Figura 26 – Comparação Imagem Original × Imagem após Invenção de Cores



Fonte: Autor

Matematicamente podemos demostrar este filtro como:

$$I'(r, g, b) = L - I(r, g, b) \tag{9}$$

sendo:

$I(r, g, b)$: o valor de cada canal de um pixel na posição (x,y);

$'I(r, g, b)$: o valor resultante do pixel após a conversão de cores;

L: o nível máximo de intensidade para o canal de cor.

O grau de crescimento de dados usando o filtro Inversão de Cores é de $2x$ sendo:

x: a quantidade de dados usados como entrada para o uso do filtro.

4.1.4.3 Filtro *Gaussian blur*

O filtro *Gaussian blur*, em tradução livre Desfoque Gaussiano, consiste na passagem de um *kernel* gaussiano em cada pixel da imagem, um *kernel* em processamento de imagem digital é uma matriz de pesos utilizada para realizar uma operação matemática local, onde o valor da matriz é multiplicado pelo pixel em questão, que é posteriormente inserido como novo valor do pixel a ser modificado.

de maneira simples o filtro *GaussianBlur* aplica uma curva normal para determinar os pesos do *kernel*,

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{10}$$

sendo:

(x,y): são as coordenadas do pixel;

σ: é o desvio padrão da distribuição Gaussiana.

A matriz obtida como resultado da distribuição gaussiana é usada como *kernel* na operação de convolução sobre a imagem original, gerando uma imagem suavizada, sendo uma técnica bastante usada para suavização de imagem e remoção de ruídos.

No filtro *GaussianBlur* implementado com a biblioteca Pillow existe um parâmetro *radius* que consiste em delimitar o tamanho do *kernel* usado. Na implementação desse filtro para o projeto, foram usados *radius* de 7 a 10, com o objetivo de analisar os resultados em um grande espaço de desfoque de imagem.

Na figura abaixo temos uma imagem do *dataset* original do conjunto de imagens para teste (a), e uma imagem após a aplicação do filtro *GaussianBlur* com um parâmetro de $radius = 7$.

Figura 27 – Comparação Imagem Original × Imagem após Filtro Gaussianblur

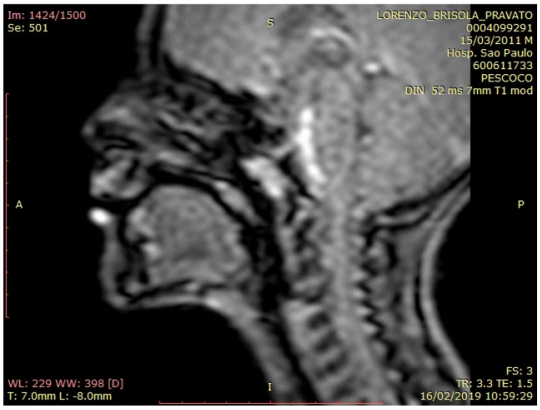


Imagem Original

Imagem a pois Filtro Gaussianblur

Fonte: Autor

O grau de crescimento de dados usando o filtro *GaussianBlur* é de $(1 + r)x$ sendo:

r: a variedade de *radius* usado como argumento na função;

x: a quantidade de dados usados como entrada para o uso do filtro.

4.1.4.4 Filtro de Extração de Canais RGB

Como já mencionado no tópico 3.2.1, uma imagem digital possui três canais de cores, correspondentes a Vermelho, Verde e Azul, a técnica de extração de canais RGB, consiste na anulação de dois canais de cores dos três existentes, gerando assim, três imagens de saída, cada uma respectivamente possuindo apenas os pixels correspondentes de um canal dos distintos canais de cores. representados na figura abaixo:

Figura 28 – Comparação Imagem Original × Extração de Canais RGB

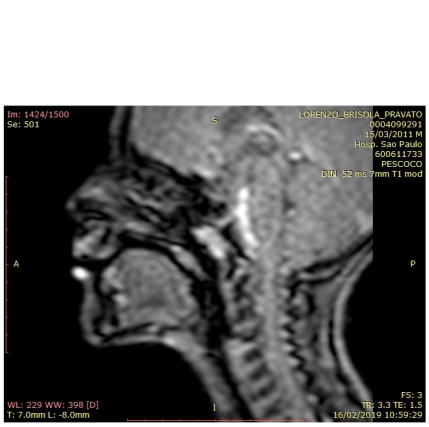
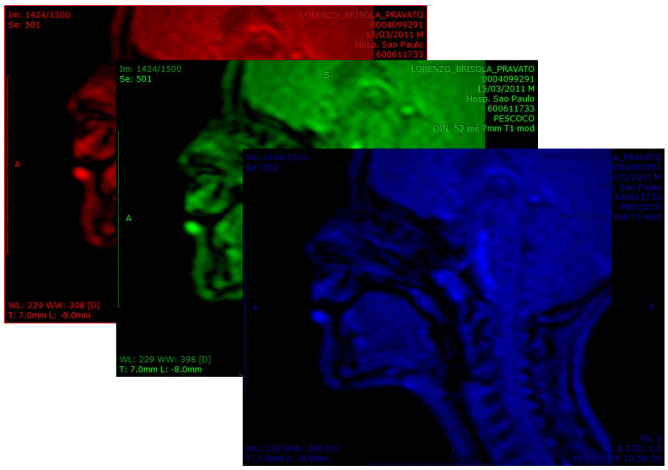


Imagem Original



Conjunto de Imagens a pois Extração de Canais

Fonte: Autor

Para expressar matematicamente esse método, temos três funções matemáticas, uma para cada canal em que o método ira manter os pixels. Sendo:

Para a imagem resultante com intensidade apenas de tons vermelhos:

$$R'(r,g,b) = R(r,0,0) \tag{11}$$

Para a imagem resultante com intensidade apenas de tons verdes:

$$G'(r,g,b) = G(0,g,0) \tag{12}$$

Para a imagem resultante com intensidade apenas de tons azuis:

$$B'(r,g,b) = B(0,0,b) \tag{13}$$

sendo:

$R'(r,g,b)$: é o valor dos pixels na imagem resultante de intensidade vermelha;

$R(r,0,0)$: corresponde ao anulamento dos canais verde e azul;

$G'(r,g,b)$: é o valor dos pixels na imagem resultante de intensidade verde;

$G(0,g,0)$: corresponde ao anulamento dos canais vermelho e azul;

$B'(r,g,b)$: é o valor dos pixels na imagem resultante de intensidade azul;

$B(0,0,b)$: corresponde ao anulamento dos canais vermelho e verde.

O grau de crescimento de dado usando o filtro extração de canais RGB é de $4x$ sendo:

x: a quantidade de dados usados como entrada para o uso do filtro.

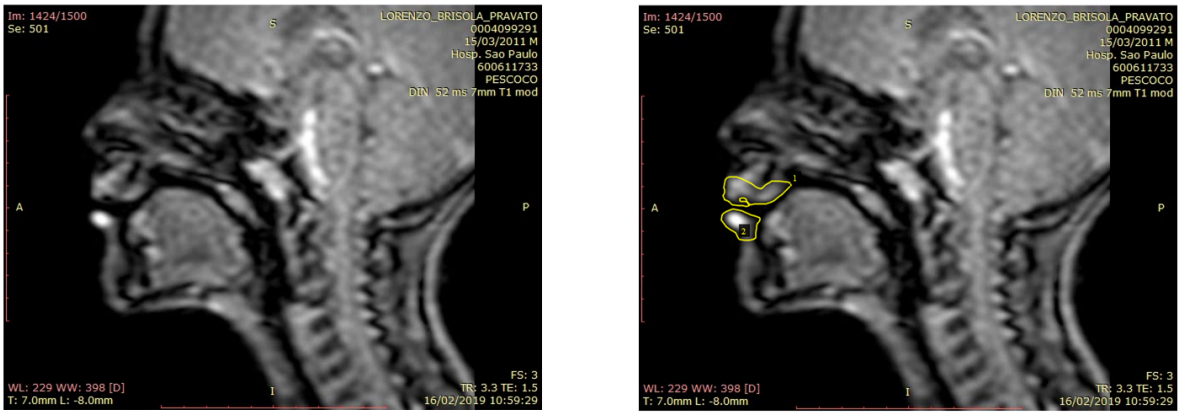
4.1.5 Etapa de Preparação: 3. Extração de Área de Interesse

A arquitetura *Mask R-CNN* necessita de uma previa marcação mas régios de interesse nas imagens de treinamento e validação do dataset, para isso é essencial o uso de alguma ferramente de anotação.

4.1.5.1 Vgg *Image Annotator* - VIA

A ferramenta de código aberto VIA foi desenvolvida pelo grupo VGG da Universidade de *Oxford* sendo um anotador manual de imagens através de formas geométricas como retângulos, círculos e polígonos. Após as anotações a ferramenta gera como resultado um arquivo json, com as chaves únicas de identificação das imagens é as coordenadas de cada ponto das formas geométricas. Na figura abaixo temos um exemplo de uma imagem antes e depois de uma anotação:

Figura 29 – Comparação Imagem Original × Imagem Anotada



Fonte: Autor

As imagens anotadas pela ferramenta serão usadas na etapa de treinamento do modelo, tanto para auxiliar o modelo no treinamento como para validar se o treinamento está gerando resultados dentro do esperado. Esse tipo de abordagem é conhecido como

aprendizagem supervisionada. Pois a rede neural tera acesso as áreas de interesse delimitadas pelo especialista, onde a rede neural pode comparar as imagens segmentadas pelo modelo com as anotações reais, assim podendo verificar o desempenho do algoritmo.

4.1.6 Etapa de Implementação: 4. Implementação do Algoritmo

A implementação do algoritmo foi executada usando a plataforma *google Colaboratory*, um ambiente que não necessita de configuração prévia e seu uso é recomendado em projetos de ciência de dados e *machine learning*. O uso do *google Colaboratory* tem como um dos melhores atrativos a possibilidade do acesso gratuito a GPUs e fácil compartilhamento. O uso de GPUs permite que um usuário com baixo recurso computacional seja capaz de treinar modelos robustos sem estressar seu próprio ambiente local de execução.

4.1.6.1 Configuração do Algorítimo

Para a implementação do algoritimo, foram definidas alguns parâmetros, que podem ser vistos abaixo:

NAME : é a variável usada para identificar a configuração da rede neural, que para esse modelo foi definida como lábio;

IMAGES_PER_GPU :é a definição da qualidade de imagens a serem enviadas para a GPU por vez, que para evitar grandes gastos computacionais é geralmente definido como 1, no entanto, devido ao uso da plataforma *google colab* e os recursos disponíveis pela plataforma, a quantidade de imagens foi definida como 2;

NUM_CLASSES : o modelo possui duas classes, sendo, uma classe correspondente ao fundo da imagem e outra correspondente aos lábios, por tanto, o número de classes foi definido como 2;

EPOCH é a quantidade de vezes em que o conjunto de dados de treinamento e validação será apresentado a rede neural na etapa de treinamento, para o modelo implementado foi usado um total de 25 épocas;

STEPS_PER_EPOCH é o número de passos por épocas de treinamento, indicando a quantidade de exemplos de treinamento processadas por cada época de treinamento, sendo sua quantidade estabelecida por padrão como 100;

DETECTION_MIN_CONFIDENCE delimita a confiabilidade do modelo para definir se sua previsão foi correta, em casos em que a probabilidade de confiança do objeto é menor que o mínimo definido, o modelo considera a previsão como incorreta, para detecções com confiabilidade igual ou acima do mínimo definido, o modelo avalia o a amostra prevista como correta, para o modelo implementado a confiabilidade foi definida em 0.9

BACKBONE o extrator de características do modelo, que para a rede neural do projeto foi delimitada como a *ResNET-101*, sendo ela, uma rede neural profunda contendo 101 camadas e 5 blocos de convolução.

4.1.7 Etapa de Implementação: 4.1 Treinamento

A etapa de treinamento de uma rede neural é uma das principais etapas na concepção de um modelo, o fundamento básico dessa etapa é o ajuste dos pesos e os parâmetros da rede neural, sendo os pesos em uma rede neural valores numéricos associados aos neurônios artificiais do modelo, que caso sejam ajustados na maneira correta pela rede neural possibilitam que o modelo realize previsões de dados corretamente.

4.1.7.1 Transferência de Aprendizagem

A técnica de transferência de aprendizagem consiste na transferência de pesos pré-treinados como um ponto de partida para o treinamento de pesos de um novo modelo, assim, acelerando o processo de aprendizagem. Para o modelo em questão, a técnica foi aplicada usando o conjunto de dados *Common Objects in Context* - COCO, sendo um conjunto de dados de referência para a detecção e segmentação de imagens. Nesse caso, a aplicação da técnica consistiu em utilizar a transferência de aprendizagem para a extração das características de baixo nível dos dados de treinamento, como bordas e contornos. Desse modo, reduzindo bastante o tempo de treinamento do modelo.

4.1.7.2 Recursos Para Treinamento

Devido às grandes quantidades de cálculos e operações executadas na etapa de treinamento, que podem chegar a bilhões de operações matemáticas, dependendo da complexidade da rede neural, o treinamento de uma rede neural exige grandes recursos de *hardware*, no entanto o *google colab* disponibiliza ótimos recursos para a implementação de modelos de rede neural. Como:

Hardware	Descrição
Memoria de Armazenamento	78.2 Gb
Memoria RAM	12.7 Gb
GPU - Tesla K80	24 Gb

Tabela 7 – *Hardwars* Disponíveis Para Treinamento

4.1.7.3 Tempo de Treinamento

O tempo de treinamento de uma rede neural depende de diversos fatores, em redes neurais complexas e com um conjuntos de dados numerosos, o treinamento pode durar

dias, semanas ou até meses. No entanto, otimizações e técnicas podem ser usadas para reduzir o tempo de treinamento.

Devido os recursos disponíveis, apresentados na Tabela 7 - (*Hardwars* Disponıveis Para Treinamento) e as técnicas aplicadas como a transferência de aprendizagem, o tempo de treinamento do modelo foram de 51 minutos.

4.1.8 Etapa de Validação: 5. Teste

A fase de testes é uma parte essencial para a validação de um modelo, o objetivo dos testes é avaliar a capacidade de generalização e habilidade de previsões corretas. O tipo de teste adotado para avaliar o modelo é chamado de teste de divisão de dados, essa técnica consiste na divisão dos dados em dois conjuntos de dados, um conjunto de treinamento e outro de teste. Enquanto o conjunto de treinamento é usado para ajustar os pesos e parâmetros do modelo, o conjunto de teste é utilizado para avaliar o modelo. Para melhorar a capacidade de avaliar o modelo, além da divisão entre conjunto de treinamento e teste, o conjunto de teste possui subconjuntos de teste, para avaliar os diferentes resultados do modelo em uma variedade maior de dados.

4.1.9 Etapa de Validação: 5.1 Extração de Resultados

Para extrair os resultados dos testes, é necessário entender alguns conceitos básicos, como a matriz de confusão:

		Predição	
		positivo	Negativo
Real	Positivo	Verdairo Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Tabela 8 – Matriz de Confusão

Onde:

Predição: Positivo são previsões onde o modelo prevê classifica uma amostra como positiva, ou seja, em casos onde o modelo classifica uma um objeto como lábio;

Predição: Negativo : são previsões onde o modelo prevê uma amostra como negativa, ou seja, classifica como uma previsão incorreta, em casos onde o modelo não classifica um objeto como lábio;

Real: Positivo é a real classe das amostras positivas, ou seja uma imagem que possui um lábio dentro das condições definidas pelo especialista;

Real: Negativo é a real classe das amostras negativas, ou seja, uma imagem que não possui um lábio dentro das condições definidas pelo especialista;

Verdadeiro Positivo (VP) amostras que foram corretamente classificadas como positivas pelo modelo, ou seja, objeto que era realmente um lábio é foi classificado como lábio pelo modelo;

Verdadeiro Negativo (VN) amostras que foram corretamente classificadas como negativas pelo modelo, ou seja, objeto que não era um lábio é foi classificado como uma amostra que não possuía lábio pelo modelo;

Falso Positivo (FP) amostras que foram erroneamente classificadas como positivas pelo modelo, ou seja, um objeto que não era um lábio é foi classificado como lábio pelo modelo;

Falso Negativo (FN) amostras que foram erroneamente classificadas como negativas pelo modelo, objeto que possuía um lábio e foi classificado como um objeto que não era um lábio pelo modelo.

4.1.9.1 Métricas de Avaliação de Desempenho

As métricas de avaliação de desempenho são usadas para analisar a qualidade das previsões do modelo, ou seja, o quão assertivo é o modelo. Existem várias métricas de avaliação, onde uma ou mais métricas são empregadas, dependendo do aspecto a ser avaliado, como avaliar o quão preciso é um modelo para analisar corretamente uma classe, ou analisar quantos falsos positivos ou falsos negativos um modelo previu como corretos.

Para avaliar o modelo foram usadas as métricas de precisão, acurácia e revocação onde o objetivo da precisão é verificar entre as amostras que o modelo classificou como positiva, quais estão realmente corretas. Sendo calculada como:

$$P = \frac{VP}{VP + FP} \tag{14}$$

Onde, a precisão é calculada pela quantidade de Verdadeiro Positivo (VP), divido pela soma de Verdadeiro Positivo e Falso Positivo.

Já a acurácia avalia a proporção de amostras classificadas corretamente de maneira geral, levando em consideração todas as amostras, sendo elas classificadas como verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo. Sendo calculada como:

$$A = \frac{VP + VN}{VP + VN + FP + FN} \tag{15}$$

Onde, acurácia é obtida pela soma de Verdadeiro Positivo (VP) e Verdadeiro Negativo (VN), dividido pela soma de Verdadeiro Positivo (VP), Verdadeiro Negativo (VN), Falso Positivo (FP) e Falso Negativo (FN).

A revocação se faz importante para a identificação de quantas das amostras positivas foram corretamente classificadas como verdadeiro positivo. Sendo calculada como o número de amostras Verdadeiro Positivo, sobre a soma de Verdadeiro Positivo e Falso Negativo:

$$R = \frac{VP}{VP + FN} \tag{16}$$

5 RESULTADOS

Esse capítulo aborda os resultados óbitos pelos conjuntos de teste do modelo.

5.1 Resultados do Modelo

A análise de resultados se dá sobre dois conjuntos de teste, sendo nomeados como teste 01 e teste 02. Sendo:

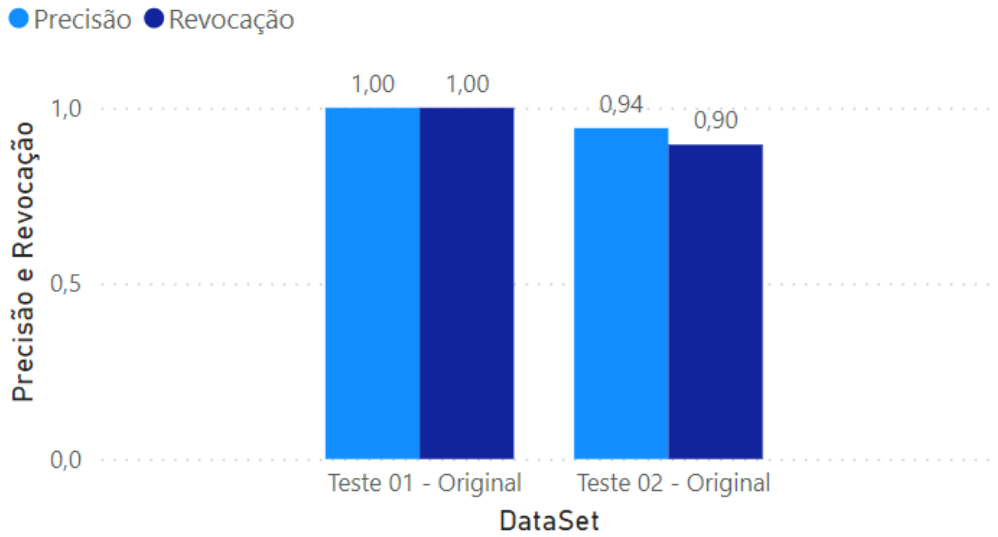
Teste 01 consiste em um conjunto de teste com 100 imagens não examinadas pelo modelo na etapa de treinamento, porém, sendo um conjunto de imagens composta pelo mesmo indivíduo usado na fase de treinamento, formando uma base de dados que simula um ambiente de imagens ideais para o teste do modelo;

Teste 02 consiste em um conjunto de 100 imagens amostradas de 5 indivíduos diferentes, constituindo um conjunto de treinamento em um ambiente de teste real, para analisar o desempenho do modelo.

5.1.1 Resultados do Modelo: Precisão e Revocação do Conjunto de Teste 01 em Imagens Originais

Pelo fato da precisão ser diretamente impactada pelas amostras Verdadeiro Positivo e Falso Positivo, em casos em que o modelo possa vir detectar Verdadeiro Negativo e Falso Negativo por maiores que possam ser os números de amostras assim analisadas a precisão não sofre impacto, podemos assim assumir apenas pela precisão que um modelo venha a ser um ótimo modelo, mesmo que sua taxa de FN seja elevada, então para melhor analisar o modelo, a métrica de revocação também foi aplicada como um meio de comparar precisão e revocação ao ponto de se obter mais informações sobre o desempenho do modelo.

Figura 30 – Precisão por Revocação do conjunto de dados teste 01 e teste 02 em imagens original



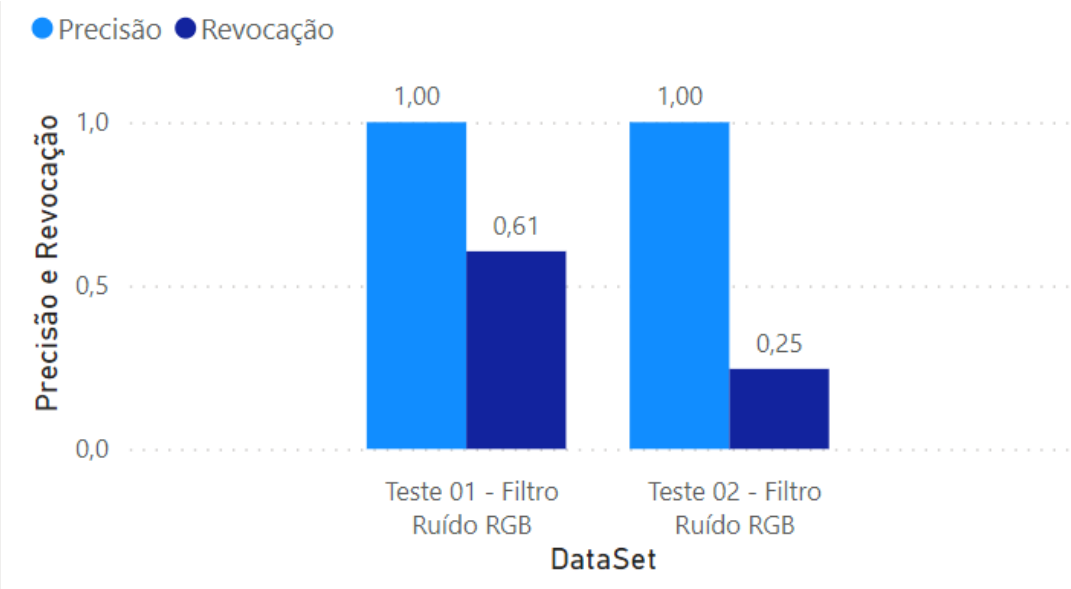
Fonte: Autor

Na figura - 30, ao analisar a precisão e revocação do modelo nos dados referentes ao teste 01, pode-se assumir que todas as amostras foram Verdadeiro Positivo foram classificadas de maneira correta pelo modelo. Ainda sobre a figura - 30, podemos observar uma diminuição nas duas métricas nos dados do teste 02, tanto precisão quanto revocação, isso se dá pela classificação de amostras Falso Positivo e Falso Negativo pelo modelo nesse conjunto de dados, no entanto sendo uma pequena parcela das amostras, sendo 21 amostras analisadas como Falso Positivo e 11 amostras classificadas como Falso Negativo, dentre as 200 amostras de lábios analisadas pelo modelo nesse conjunto.

5.1.2 Resultados do Modelo: Precisão e Revocação do Conjunto de Teste 01 e Teste 02 em Imagens com Ruído RGB

Na figura abaixo, pode-se notar que em relação à precisão, os dois modelos possuem precisão máxima de 1,00. No entanto, quando analisamos a revocação, temos um melhor resultado no conjunto de teste 01, possuindo uma revocação de 0,61 contra uma revocação de 0,25 no conjunto de dados de teste 02.

Figura 31 – Precisão por Revocação do conjunto de dados teste 01 e teste 02 em imagens com adição de Ruído RGB

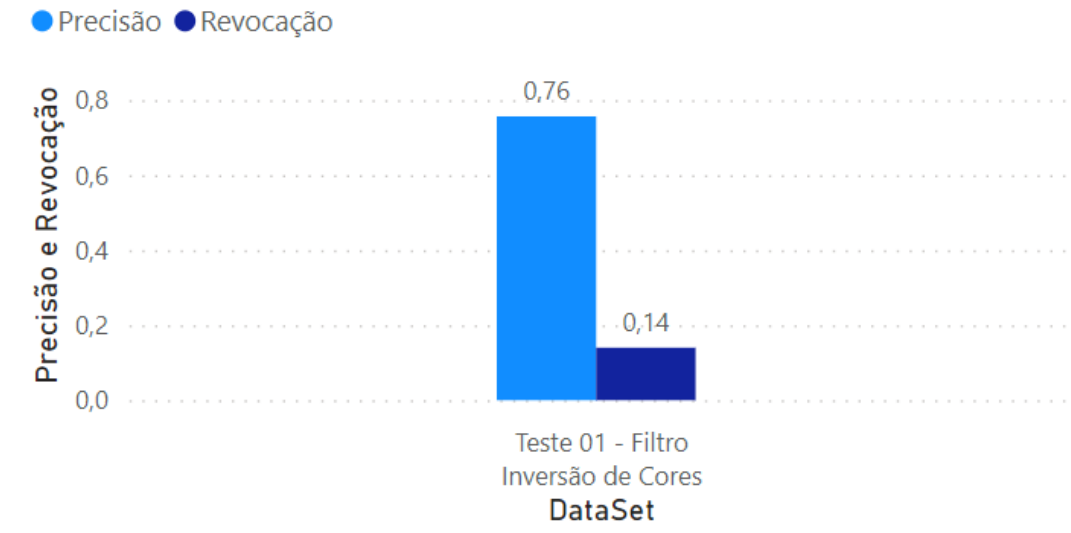


Fonte: Autor

5.1.3 Resultados do Modelo: Precisão e Revocação do Conjunto de Teste 01 em Imagens com Filtro de Inversão de Cores

O filtro de Inversão de cores promoveu um grande impacto na revocação dos dados, fazendo com que apenas 14% das amostras fossem detectadas, no entendo teve um impacto menor em relação a classificação de Falso Positivo, produzindo uma precisão de 0,76, nos dados do conjunto de Teste 01.

Figura 32 – Precisão por Revocação do conjunto de dados teste 01 em Imagens com Filtro de Inversão de Cores



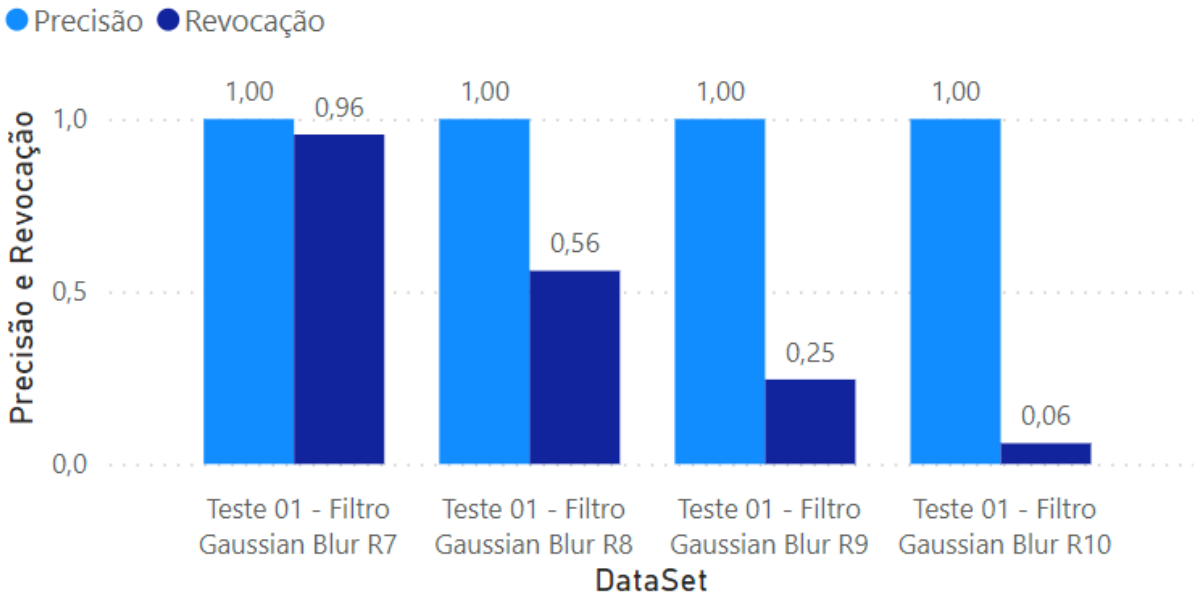
Fonte: Autor

O conjunto de Dados de Teste 02 não gerou dados relevantes para serem analisados pois das 200 amostras de lábios analisadas pelo modelo, nenhuma foi classificada corretamente, assim não gerando resultados relevantes para construção do gráfico[3]

5.1.4 Resultados do Modelo: Precisão e Revocação do Conjunto de Teste 01 em Imagens com Filtro *Gaussian Blur*

O filtro *gaussian Blur* proporciona um impacto negativamente gradativo nos conjuntos de dados onde, quanto maior é o raio de modificação do filtro, maior é o impacto na revocação das amostras, isso pode ser notado tanto na figura - 33, que demonstra esse resultado no conjunto de Teste 01, quanto no conjunto de Teste 02, visto na figura 34. No entanto esse impacto na revocação dos conjuntos é maior no conjunto de teste 02, embora pouco impacto é notado na precisão do modelo causado pelo filtro, sendo notado apenas em um dos conjuntos de imagens do teste 02.

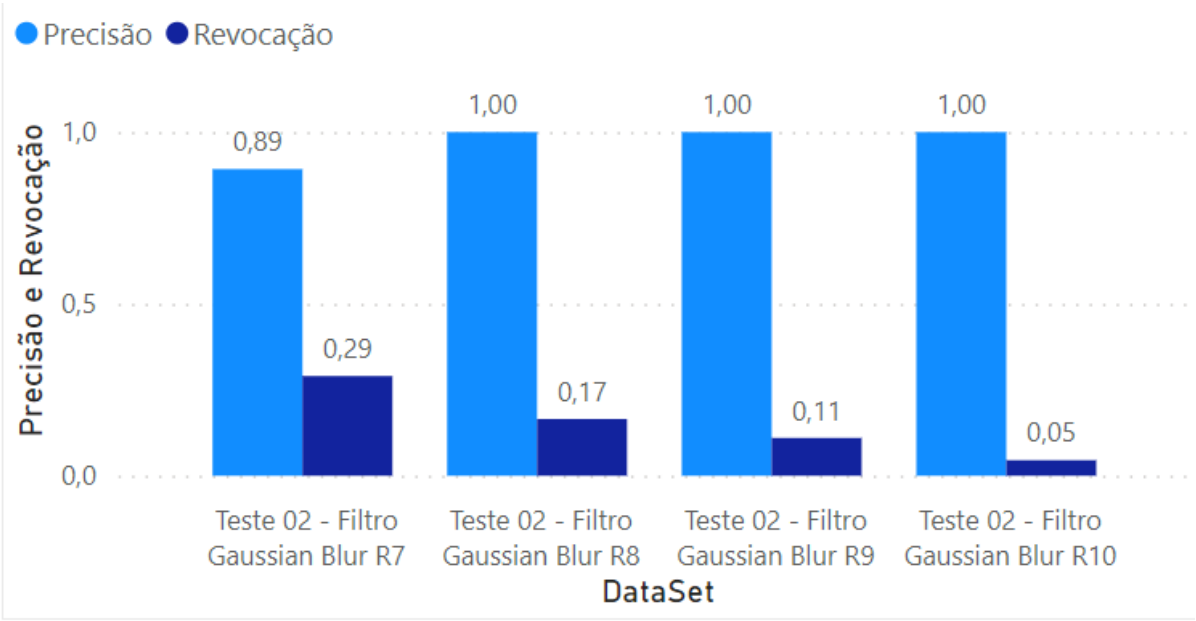
Figura 33 – Precisão por Revocação do conjunto de dados teste 01 em Imagens com Filtro *Gaussian Blur*



Fonte: Autor

[3]Devido aos cálculos usados para avaliação do desempenho serem compostas por somatórias e divisões, para o cálculo da precisão em casos onde não possui um divisor maior que 0, o cálculo se tornar uma indeterminação matemática, impossibilitando a extração de resultados, já para o cálculo de revocação temos uma resultado de 0,0

Figura 34 – Precisão por Revocação do conjunto de dados teste 02 em Imagens com Filtro *Gaussian Blur*



Fonte: Autor

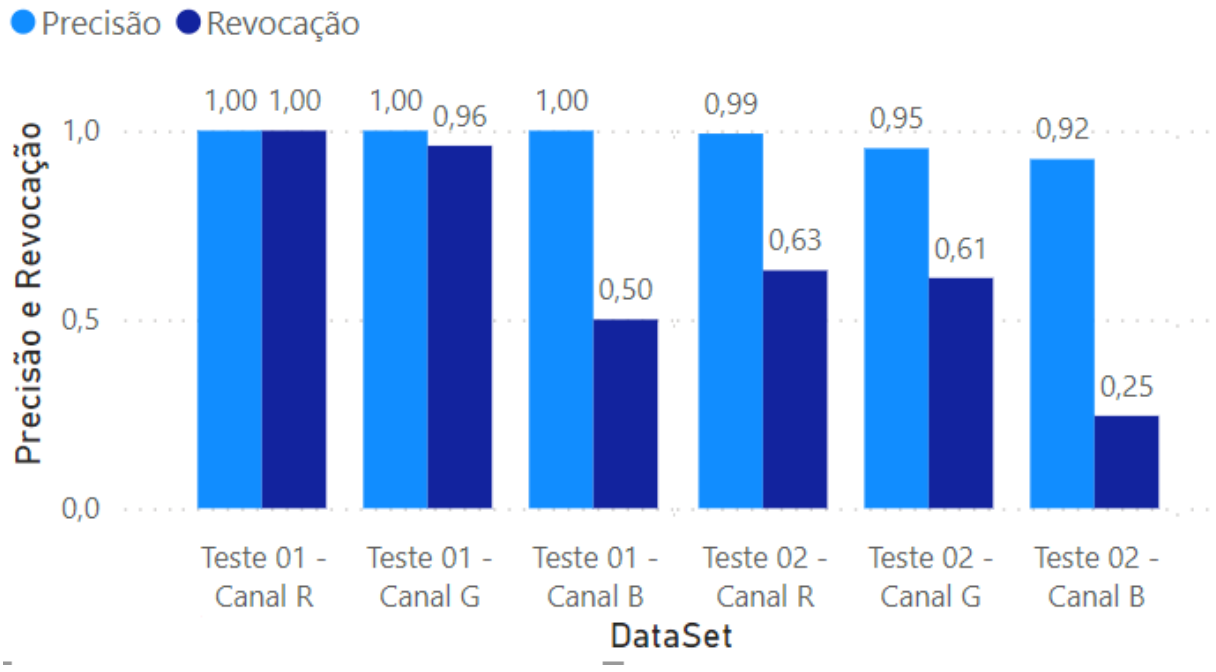
Como já citado o filtro *Gaussian Blur* possui um parâmetro *radius* que manipula o tamanho do *kernel* usado na aplicação do filtro, tanto na Figura 33, quando na Figura 34, as legendas das barras possuem um indicador R que indica o parâmetro *radius* usado[4]

5.1.5 Resultados do Modelo: Precisão e Revocação do Conjunto de Teste 01 e Teste 02 em Imagens após extração RGB

Após a extração RGB, dispomos de três subconjuntos para cada conjunto de teste, onde ao analisar cada componente RGB distintamente, é notável que nos dois conjuntos de testes, o subconjunto que mais sofreu alteração em sua revocação é o subconjunto de imagens apenas com componentes em tons de azul, possuindo uma revocação de 0,50 no conjunto de teste 01 e 0,25 no conjunto de teste 02, em seguida temos as imagens com componentes verde com uma revocação 0,96 para o conjunto de teste 01 e 0,61 para o conjunto de teste 02, e 1,00 de revocação para componentes em vermelho do conjunto de teste 01 e 0,63 para o conjunto de teste 02. No entanto as variações de precisão são baixas no conjunto de teste 02, sendo de 0,99 para componentes em tons de vermelho, 0,95 para componentes em tons de verde e 0,92 para componentes em tons de azul. Se comportando de maneira constante em 1,00 para os subconjuntos de teste 01.

[4]Nas figuras 33 e 34, temos os *radius* de 7 a 10, pois os *radius* anteriores a esse não geraram impactos significativos para a geração das métricas de desempenho, devido aos menores graus de desfoque de imagens não causarem tanta variação na classificação feita pelo modelos nos dois conjunto de testes

Figura 35 – Precisão por Revocação do conjunto de Teste 01 e Teste 02 em Imagens após extração RGB

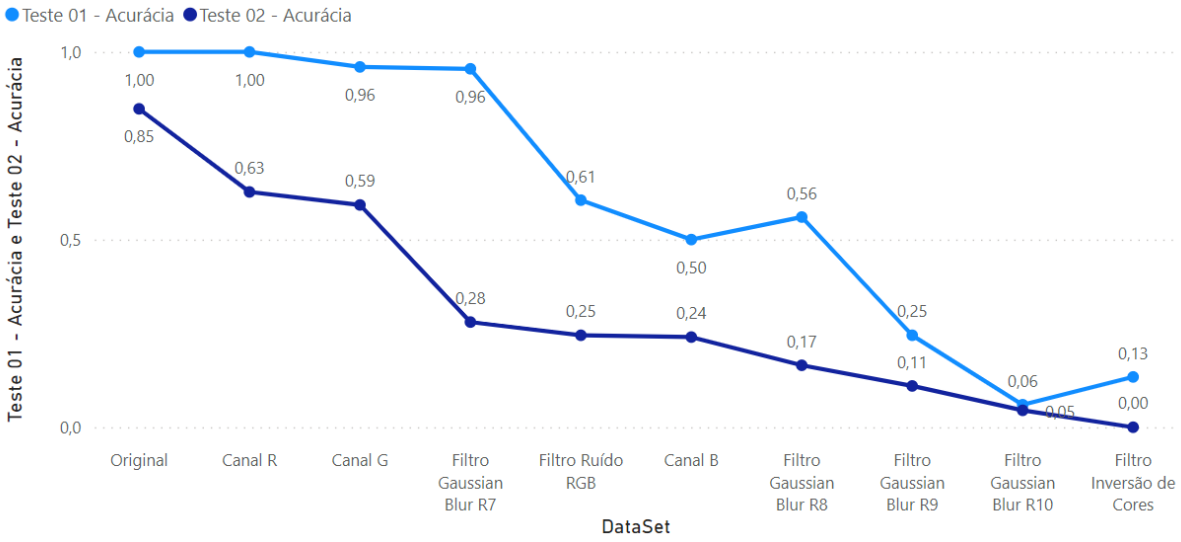


Fonte: Autor

5.1.6 Resultados do Modelo: Acurácia dos Conjuntos de Teste 01 e Teste 02

As amostras com melhor acurácia em ambos os conjuntos, são compostas pelas imagens originais sem adição de filtro ou modificação do conjunto de teste 1 e teste 2, dispondo respectivamente de acurácia 1,00 e 0,85. No entanto os impactos variam quando considerado as menores acurácias, sendo para o conjunto de teste 01 sua menor acurácia vista no subconjunto de imagens *Graussian Blur R10* com acurácia de 0,06, e para o conjunto de teste 02 podendo ser observado no subconjunto de dados de imagem com filtro de inversão de cores, possuindo 0,00 de acurácia.

Figura 36 – Acurácia dos Conjuntos de Teste 01 e Teste 02



Fonte: Autor

A figura 36 é uma comparação entre a acurácia do conjunto de teste 01 e conjunto de teste 02, onde se inicia na maior acurácia obtida pelos dois até as acurácias menores.[5]

[5]Para a organização do gráfico foi utilizado como referência predominante as métricas do conjunto de teste 02, com o intuído de melhor ordenar os subconjuntos avaliados

6 CONSIDERAÇÕES FINAIS

É notável que as tecnologias vêm causando impactos em diversas áreas, com o uso de *deep learning* atualmente já é possível encontrar mecanismo e tecnologias que auxiliam tarefas que antes eram em sua totalidade manual, possibilitando a exploração de novos cenários como diagnósticos médicos com o auxílio de *deep learning*, análise de imagens médicas, auxílio em cirurgias invasivas, entre outros.

Porém, é necessário entender os desafios que esse tipo de abordagem enfrenta atualmente, os resultados obtidos por *deep learning* são gerados em meio a uma ”caixa preta”, onde nem sempre é possível afirmar com tanta precisão como uma rede neural chegou a determinado resultado, e isso pode ser visto com desconfiança por alguns especialistas na área, outro ponto a ser discutido é o baixo volume de dados de imagens de ressonância magnética dispostas em meio público, tendo em vista que para modelos complexos, é necessário um amplo conjunto de dados para treinar e testar o modelo. A dificuldade de encontrar dados confiáveis e em grande volume, e um grande desafio para implementações na área. No entanto, solucionar problemas através de tecnologia é a base da computação, e foi o que proporcionou implementações de arquiteturas de *deep learning*.

A arquitetura *Mask R-CNN* obteve resultados promissores na segmentação de lábios em imagens de ressonância magnética, obtendo uma precisão acima de 94% em seus conjuntos de dados sem a adição de filtros e acurácia acima de 84% , demonstrando resultados que encorajam seu uso em implementações de segmentação de imagens em ambientes para auxiliar profissionais em análises de MRI.

Além dos resultados promissores do modelo, a arquitetura se destaca pela possibilidade de diminuir o custo computacional de sua execução através do uso de abordagens como as que foram executadas nesse projeto como transferência de aprendizagem e computação em nuvem, proporcionando sua execução em computadores pessoais que não necessariamente necessitam de um ambiente de execução de grande porte.

Os resultados adquiridos nesse experimento, proporcionam parâmetros otimistas para implementações futuras, visando a continuidade do projeto, pode-se aplicar os conhecimentos teóricos e práticos adquiridos pelo experimento, na implementação de um modelo de segmentação em conjuntos de vídeo de ressonância magnética, possibilitando criar um modelo que análise arquivos em vídeo junto com a adição de novas classes, como a implementação da segmentação do trato vocal como um todo.

REFERÊNCIAS

GONZALEZ, Rafael. WOODS, Richard. **Processamento de Imagens Digitais**. p. 1-2.

RONCERO, Gomes Valeriana. **Um estudo de segmentação de imagens baseado em um método de computação evolucionária**. Disponível em <http://www.pee.ufrj.br/index.php/pt/producao-academica/dissertacoes-de-mestrado/2005-1/2005062702-2005062702/file>.Acesso em: 2 Set. 2022.

TURING, Alan M. **Computing machinery and intelligence**. **Mind** 59.

Data Science Academy. **Segmentação de Imagens Médicas com Deep Learning**.2021. Disponível em: <https://blog.dsacademy.com.br/segmentacao-de-imagens-medicas-com-deep-learning/>. Acesso em: 2 Set. 2022.

WOLFEWICZ, Arne. **Deep Learning vs. Machine Learning – What's The Difference?**. Levity. Disponível em: <https://levity.ai/blog/difference-machine-learning-deep-learning>. Acesso em: 5 Set. 2022.

JANIESCH, Christian. ZSCHECH, Patrick. HEINRICH, Kai. **Machine learning and deep learning**. 2021. Springer. Disponível em: <https://link.springer.com/article/10.1007/s12525-021-00475-2>. Acesso em: 5 Set. 2022.

ERYILDIRIM, Abdulkadir. BERGER, Marie-Odile. **A guided approach for automatic segmentation and modeling of the vocal tract in mri images**. 2021. Disponível em: <https://ieeexplore. ieee.org/abstract/document/7074020>. Acesso em: 13 Set. 2022.

SULEYMAN, Mehmet. DANDIL, Emre. **Automatic detection of multiple sclerosis lesions using Mask R-CNN on magnetic resonance scans**. 2021 .Disponível em: <https://doi.org/10 .1049/iet-ipr.2020.1128>. Acesso em: 13 Set. 2022

ZHUGE, Ying. NING, Holly. MATHEN, Peter. CHENG, Jason. KRAUZE, Andra. CAMPHAUSEN, Kevin. MILLER, Robert. **Automated glioma grading on conventional MRI images using deep convolutional neural networks**. 2020. Disponível em: <https://aapm.onlinelibrary. wiley.com/doi/full/10.1002/mp.14168>. Acesso em: 13 Set. 2022.

apud ZHANG **Automatic Detection and Segmentation of Breast Cancer on MRI Using Mask R-CNN Trained on Non-Fat-Sat Images and Tested on Fat-Sat Images** 2020.Disponível em: <https://pubmed.ncbi.nlm.nih.gov/33317911/>.Acesso em: 13 Set. 2022.

apud MASOOD. **A Novel Deep Learning Method for Recognition and Classification of Brain Tumors from MRI Images**. 2021. Disponível em: <https://www.mdpi.com/2075-4418/11/5/744>. Acesso em: 13 Set. 2022.

FERREIRA, Fernanda. Nacif, Marcelo. **Manual de Técnicas em Ressonância Magnética**. 2011. Rubio Ltda. p. 17-36.

FERRARINI, Maria. IWASAKI, Masao. **Imagem por ressonância magnética: princípios básicos**. Disponível em: <https://www.scielo.br/j/cr/a/mmPL6rMp5vmPCRpmYH84Kbm/?lang=pt>. Acesso em: 1 Out. 2022.

DOUGLAS, Herculys. **RM COLUNA VERTEBRAL** Teresina, 2015. p. 15-19. Disponível em: <https://www.slideshare.net/herculys/rm-coluna-vertebral-54889367>. Acesso em: 5 Out. 2022.

Vugman, N.V. Herbst, M.H. **Introdução à ressonância paramagnética eletrônica de onda contínua. Aplicações ao estudo de complexos de metais de transição**. 1. ed. Rio de Janeiro: Auremn, 2005.

VIEIRA, Rafael. **Ressonância Magnética (RM): Princípios básicos**. KENHUB. Disponível em: <https:// www.kenhub.com/pt/library/ensino/rm-principios-basicos>. Acesso em: 11 Out. 2022.

OLIVEIRA, Sumaia. **Revisão bibliográfica dos princípios de ressonância magnética nuclear**. Universidade de Brasília, Facudade de Tecnologia. Disponível em: <https://bdm.unb.br/bitstream/10483/890/1/2008_SumaiaCelledeOliveira.pdf>. Acesso em: 11 Out. 2022.

DOUGLAS, Herculys. **II Jornada de Radiologia do Hospital Getúlio Vargas, Estudo de Imagens por Ressonância Magnética**. Teresina 2015, P. 17.

C.A, Bertulani. **Teoria Cinética dos Gases**. 2021. Disponível em:<https://www.if.ufrj.br/teaching/fis2/teoria_cinetica/teoria_cinetica.html>. Acesso em: 4 Nov. 2022.

MAZZOLA, Alessandro. **Ressonância magnética: princípios de formação da imagem e aplicações em imagem funcional**. Disponível em: <https://www.rbfm.org.br/rbfm/article/view/51/v3n1p117>. Acesso em: 18 Nov. 2022.

DOUGLAS, Herculys. **Estudo de Imagens por Ressonância Magnética** Teresina, 2015. p. 21. Disponível em: <https://www.slideshare.net/herculys/ressonncia-magnticaatualizao>. Acesso em: 18 Nov. 2022.

Sem autor. **Conceitos básicos de vídeo e terminologia**. 2021. Adobe. Disponível em: <https://helpx.adobe.com /br/experience-manager/scene7/kb/evideo/video-general/basic-video-concepts-terminology. html>. Acesso em: 20 Nov. 2022.

SHORTEN, Connor. KHOSHGOFTAAR, Taghi. **A survey on Image Data Augmentation for Deep Learning**. Springer Open. Disponível em: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0>. Acesso em: 8 Jan. 2023.

Sem autor. **Hisense TV Supported Video Formats**. 2019. Multipelife. Disponível em: <http://www.multipelife.com/hisense-tv-supported-formats.html>. Acesso em: 21 Nov. 2022.

Sem Autor. **Image-1 Introduction to Digital Images**. Stanford. Disponível em: <https://web.stanford.edu/class/cs101/image-1-introduction.html. Acesso em: 01 Dez. 2022.

CLARO, Maía. VOGADO, Luiz. SANTOS, Justino, Veras, Rodrigo. **Utilização de Técnicas de Data Augmentation em Imagens: Teoria e Prática**. Manuscrito.

GOMES, Haroldo. **A DataSet of word sequences through MRI**. **IEEE Dataport**, February 5, 2020. doi: https://dx.doi.org/10.21227/7jan-t860.

HE, Kaiming. GKIOXARI, Georgia. DOLLÁR, Piotr, GIRSHICK, Ross **Mask R-CNN**. Cornell University. 2017. Disponível em: <https://arxiv.org/abs/1703.06870>. Acesso em: 9 Dez. 2022.

CASS, Stephen. **Top Programming Languages 2021**. IEEE Spectrum, 2021. Disponível em: <https://spectrum.ieee.org/top-programming-languages-2021>. Acesso em: 13 Dez. 2022.

Python. **Python Software Foundarion**. 2021. Disponível em: <https://www.python.org/>. Acesso em: 14 Dez. 2022.

google colaboratory. Disponível em: <https://colab.research.google.com/>. Acesso em: Acesso em: 14 Dez. 2022.

JESUS, Edison. COSTA JR, Roberto. **A Utilização de Filtros Gaussianos na Análise de Imagens Digitais**. Disponível em: <https://proceedings.sbmac.org.br/sbmac/article/view/477/483>. Acesso em: 10 Fev. 2023.

Pillow **ImageFilter Module**. Disponível em: <https://pillow.readthedocs.io/en/stable/reference/ImageFilter.html>. Acesso em: 10 Fev. 2023.

GWOSDEK, Pascal. GREWENIG, Sven. BRUHN, Andrés. WEICKERT, Joachim. **Theoretical Foundations of Gaussian Convolution by Extended Box Filtering**. Disponível em: <https://www.mia.uni-saarland.de/Publications/gwosdek-ssvm11.pdf>. Acesso em: 12 Fev. 2023.

VIA Annotator. **VGG Image Annotator**. Disponível em: <https://gitlab.com/vgg/via>. Acesso em: 12 Fev. 2023.

VAKILI, Meysam. GHAMSARI, Mohammad. REZAEI, Masoumeh. **Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification**. Disponível em: <https://arxiv.org/abs/2001.09636>. Acesso em: 23 Fev. 2023.