

# Software para estimativa da idade gestacional com imagem de ultrassom

Igor Alexandre de Lima  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
Rua Emídio dos Santos, S/N, Barbalho  
Salvador, Bahia  
igoralwork@gmail.com

Antonio Carlos Santos Souza  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
Rua Emídio dos Santos, S/N, Barbalho  
Salvador, Bahia  
antoniocarlos@ifba.edu.br

## RESUMO

Na atualidade, um método relativamente barato, não invasivo e popular para estimação do tempo de gestação é a ultrassonografia, também conhecida por ecografia ou ultrassom, um exame de imagem diagnóstico que serve para visualizar em tempo real qualquer órgão ou tecido do corpo. Um obstáculo na ecografia é a identificação de determinadas regiões do feto, que por vezes são confundidas por análises equivocadas. Em razão disso, nesta monografia encontra-se uma medida de resolução ao problema de detecção por meio do emprego de redes neurais convolucionais, uma classe de rede neural artificial que trabalha no processamento e análise de imagens digitais, para detecção da circunferência cefálica fetal na plataforma mobile. Ademais, a solução utiliza métodos suplementares na etapa de pré processamento para aperfeiçoar a qualidade da imagem, reprimindo o ruído causado por sombras ou fetos fora do paradigma.

## Palavras-chave

Imagens de ultrassom, anatomia fetal, obstetrícia, exame de imagem, processamento

## 1. INTRODUÇÃO

Os avanços tecnológicos progressivamente agregam contribuições essenciais para a medicina atual, possibilitando um melhor cuidado contínuo à saúde dado a evolução na assistência prestada. Nesse contexto, a realização de exames preventivos se destacam como um recurso imprescindível para a realização do diagnóstico uma vez que propiciam ao médico a obtenção de maior conhecimento laboratorial e anatomopatológico<sup>1</sup> do paciente.

Para o diagnóstico, o exame de imagem é um dos recursos mais utilizados onde, ao todo, são encontrados 8 procedimentos distintos que podem ser classificados em dois grupos: os que utilizam de radiação ionizante<sup>2</sup>, como radiografia, fluoroscopia, angiografia e tomografia computadorizada, e os que utilizam de radiação não-ionizantes, como a ressonância magnética e a ultrassonografia.

<sup>1</sup>Consiste na análise macro e microscópica de fragmentos de tecido, órgãos ou parte de órgãos do paciente, o qual é submetido a um procedimento cirúrgico, seja ambulatorial ou não.

<sup>2</sup>Radiação ionizante é a radiação que possui energia suficiente para ionizar átomos e moléculas, ou seja, é capaz de arrancar um elétron de um átomo ou molécula.

A ultrassonografia, também conhecida por ecografia ou ultrassom, é um procedimento de diagnóstico clínico que usa de ondas sonoras de alta frequência para gerar imagens representativas de tecidos moles no interior do corpo de um indivíduo [34], permitindo visualizar em tempo real qualquer órgão ou tecido do corpo humano por meio do manuseio do transdutor, um aparelho responsável por emitir as ondas sonoras de alta frequência, inaudível para os ouvidos humanos, na localização em que é diretamente posicionado e gravar os ecos à medida que as ondas sonoras retornam para determinar o tamanho, a forma e a consistência dos tecidos moles e órgãos.

Esse método diagnóstico é uma ferramenta amplamente utilizada para a investigação de doenças cardíacas e dores abdominais, além de ser utilizada em sessões de fisioterapias para o tratamento da inflamação das articulações e dor lombar, em virtude da sua capacidade de estimular a cascata inflamatória e diminuir a dor, o inchaço e os espasmos musculares.

A ultrassom também tem aplicabilidade na realização de detecção de males em recém-nascidos e de exames preventivos em gestantes [34], onde o segundo tem como enfoque a monitoração da saúde do feto uma vez que é um diagnóstico prático não ionizante que, conseqüentemente, oferece baixo risco às mulheres grávidas e aos fetos devido a ausência de efeitos colaterais.

No ramo de obstetrícia ela desempenha um papel considerável, pois além de ser não invasiva ou minimamente invasiva [19], isto é, não sofre invasão do organismo por meio de instrumentos cirúrgicos que adentram o corpo por pele, cavidade ou abertura anatômica, consegue capturar tecidos moles<sup>3</sup>, não apresenta efeitos nocivos significativos dentro do uso diagnóstico medicinal, possui baixo custo e maior acessibilidade comparada com alternativas.

A aplicação de processamento de imagens em ultrassonografias a fim de preparar as mesmas para análises morfológicas é um passo importante no estudo das partes do corpo fetal tendo como objetivo melhorar a qualidade e contraste da imagem facilitando assim a tarefa de estabelecer formatos e tamanhos e, portanto, ampliar a acurácia na determinação da idade do feto e de seu desenvolvimento.

<sup>3</sup>Grupo de tecidos localizados entre a epiderme e as vísceras, sem levar em consideração a parte óssea.



**Figura 1: Ultrassonografia Transvaginal Obstétrica [7]**

A despeito de sua popularidade, a ultrassonografia possui algumas imperfeições, pois, em adição ao caráter subjetivo do médico responsável, a ultrassonografia sofre perturbações como exibição de ecos não estruturais, remoção de ecos estruturais reais, deslocamento e distorção de ecos, e inserção de ruídos do tipo speckle que degradam a imagem ocultando as estruturas finas [42], dificultando a detecção de regiões do feto como por exemplo a do abdômem [31]. Também se infligem danos devido a ruídos externos tais como pela movimentação respiratória e pela anatomia da gordura corporal [25], além de ser considerada a pequena dimensão do feto.

Não obstante, acerca da ultrassonografia se encontra a técnica de Imagem de Ressonância Magnética (MRI), utilizada principalmente para examinações mais profundas, quando a ultrassom não resulta em imagens claras, quando há a necessidade de retratar em imagem outras partes do corpo, comumente concernindo problemas musculoesqueléticos [33]. Outro método utilizado no âmbito médico para a concepção de imagens é a tomografia computadorizada (CT) que, devido ao uso de radiação por raio X, a torna uma alternativa bem menos estimada se comparada às duas anteriores.

Durante os últimos anos, houve um crescimento significativo na área de Deep Learning (traduzido como aprendizagem profunda) [40], com ênfase na subárea de Redes Neurais Convolucionais (CNN do inglês Convolutional Neural Network ou simplesmente ConvNet). Essa expansão pôde ser vista estender-se na análise de imagens fetais de ultrassom como fonte compacta de informações para jovens pesquisadores, além de convir como um documento de visão geral de referência para aqueles que já trabalham em campo.

Um obstáculo na ecografia é a identificação de determinadas regiões do feto, que por vezes são confundidas por análises equivocadas ou errôneas. Em razão disso, e considerando que partos prematuros são a principal causa de morte neonatal, além de fontes de sequelas vitalícias decorrentes de órgãos e sistema imunológico imaturos [27], nesta monografia encontra-se uma medida de resolução ao problema de detecção por meio do emprego de Redes Neurais Convolucionais, uma classe de rede neural artificial que trabalha no processamento e análise de imagens digitais. Além disso, a solução utiliza de métodos suplementares para aperfeiçoar a qualidade da imagem, reprimindo o ruído causado por sombras ou fetos localizados fora do paradigma.

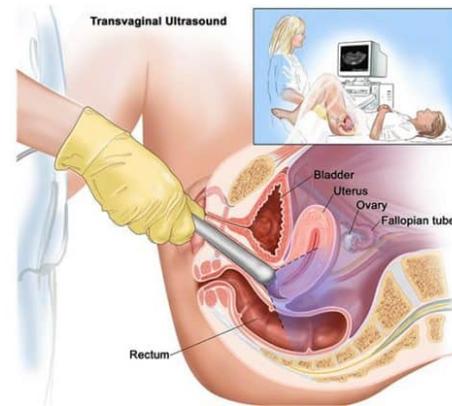
## 2. REFERENCIAL BIBLIOGRÁFICO

Essa seção apresenta as principais referências usadas para compreensão do problema e desenvolvimento da solução proposta. Ela está organizada da seguinte maneira: A subseção 2.1 retrata brevemente os diferentes tipos de exames de ultrassom, já a subseção 2.2 traz uma explanação sobre redes neurais convolucionais, por sua vez a subseção 2.3 aborda a anatomia fetal e, por último, a subseção 2.4 apresenta o conceito de tratamento de imagens digitais.

### 2.1 Exames de ultrassom

De forma sucinta, o exame de ultrassom convencional possibilita a obtenção de uma imagem em duas dimensões, podendo ser divididas em tipos específicos. Entre os mais populares se encontram o ecodoppler, o ecocardiograma transesofágico, a elastografia e a transvaginal [10], onde o último exame é próprio para detectar a gravidez.

Na ultrassom Transvaginal, o paciente é despidido da cintura para baixo onde, após ser situado em posição ginecológica, isso é, em posição dorsal; a seção dos pés da mesa cirúrgica é abaixada completamente, e os membros inferiores são elevados e abduzidos em peneiras metálicas, para haver exposição da região perineal para acesso aos órgãos pélvicos e genitais [3]. O ultrassonografista cobre o transdutor com um preservativo e gel lubrificante para facilitação da passagem do equipamento antes de inseri-lo na vagina onde deve permanecer de 10 a 15 minutos no canal, podendo ser movimentado para gerar melhores imagens. Uma vez que o transdutor esteja no posicionado, ele produzirá imagens a partir das ondas sonoras que ressaltam os órgãos internos de volta a ele e que são diretamente enviadas para um monitor para avaliação do profissional.



**Figura 2: Realização da Ultrassonografia Transvaginal no paciente [4]**

No âmbito da obstetrícia, são realizados diferentes tipos de ultrassom ao longo dos meses de desenvolvimento do bebê, onde o obstetra realiza o acompanhamento durante todas as etapas, determinando dados como o estado, sexo, peso e altura do feto para avaliação do seu estado de saúde. Também deve ser considerado que nem todo exame de ultrassonografia obstétrica deve ser feito por todas as gestantes, sendo recomendado caso a caso.

O primeiro exame de ultrassom utilizado é o Ultrassom Obstétrico Endovaginal [12], que é recomendado para ser realizado entre a 8ª e a 12ª semana da gestação. Este exame é de demasiada importância devido a confirmar o tempo de vida do feto (idade gestacional) e seu tamanho, além

de verificar a quantidade de embriões para determinar se é gestação múltipla. É a partir dele que é determinada as condutas médicas até o parto.

A Ultrassonografia Obstétrica com Translucência Nucal [8] é realizada no primeiro trimestre para avaliar ou confirmar a idade gestacional através da medição da cabeça às nádegas do feto, para conferir as características da placenta, identificar a presença do osso nasal e para aferir o comprimento (em milímetros) da nuca do bebê. Esse tipo de ultrassom é muito importante para rastrear o risco de o bebê ter doenças genéticas ou alterações estruturais em seu organismo, dado que elas geralmente causam o acúmulo de líquido nesta região. Esse exame é capaz de detectar, por exemplo, a Síndrome de Down e é realizado normalmente via abdominal, mas pode ser complementado via transvaginal, se necessário.

A Ultrassonografia Obstétrica Morfológica [19] é realizada, geralmente, no segundo trimestre da gravidez e é especificamente conduzida com o objetivo de avaliar o crescimento e desenvolvimento fetal quanto à sua formação e anatomia, detalhando seus órgãos. Neste exame podem ser detectadas doenças decorrentes de anomalias estruturais como hidrocefalia, anencefalia e descobertos defeitos no fechamento da coluna, ossos encurtados ou faltando, malformações de rins, paralisia cerebral, Síndrome de Down e hérnias [5].

No terceiro trimestre da gravidez ocorre o procedimento denominado de ultrassom da reta final da gravidez que assiste o acompanhamento da taxa de crescimento do bebê, a sua vitalidade e posição dentro do útero e, ademais, a quantidade de líquido amniótico presente. Nessa fase, as consultas com o obstetra são mais focadas na preparação para o parto e no acompanhamento do bebê uma vez que estruturas mais importantes já estão todas formadas e agora é só uma questão de tempo para o nascimento.

Na atualidade, o diagnóstico auxiliado por computador, denominado de *computer-aided diagnosis* (CAD), recebe cada vez mais atenção na área de medicina, onde a tecnologia, apesar de não substituir os profissionais da área, fornece maior acurácia no julgamento dos diagnósticos e proporciona auxílio a campos médicos escassos de especialistas [20]. Em especial, a arquitetura *Convolutional neural network* (CNN) vêm ganhando popularidade na metodologia de processamento de imagem devido a sua capacidade de detecção automática sem qualquer supervisão humana.

Na área de reconhecimento de imagens, a aplicação de filtros em imagens é uma das principais ferramentas utilizadas para refinar a qualidade da imagem, recuperando a nitidez de contornos e estruturas presentes na mesma. Nesse contexto, a imagem é uma representação bidimensional de um objeto composta pelos eixos x e y, no qual diversas funções podem ser aplicadas, tal como o filtro Gaussiano [22] que tem como finalidade suavizar imagens através de um cálculo aplicado a todos os pixels da imagem.

Aprofundando-se para a detecção de objetos na imagem, a transformada de watershed é uma das mais proeminentes ferramentas de segmentação morfológica na área de topologia [28], capaz de interpretar segmentos como superfícies, cada qual com um pixel correspondente a uma posição e os níveis de cinza determinam as altitudes. Outra ferramenta usada em operações morfológicas são as transformadas top-hat e bottom-hat.

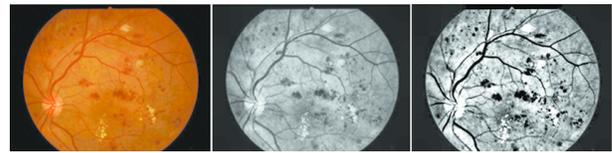


Figura 3: Imagem original e transformadas top-hat e bottom-hat [47]

Ambas técnicas são utilizadas para remoção de objetos e detalhes na imagem com base em um elemento estruturante, definido como um conjunto de coordenadas de pixel, o qual pode assumir formatos poligonais. Elas são aplicadas em operações de abertura (erosão da imagem seguida de uma dilatação) ou de fechamento (dilatação seguida de erosão) que não se encaixam nos objetos a serem removidos. A diferença entre elas é em relação a luminosidade da imagem, enquanto a transformada de top-hat atua sobre objetos claros de fundo escuro, a de bottom-hat age sobre objetos escuros com fundo claro [23], dando às operações as denominações complementares de white-hat e bottom-hat, respectivamente.

Após a aplicação das transformadas é necessária a detecção de bordas das estruturas de interesse nas imagens por meio de algoritmos. A detecção depende de uma imagem uniformizada a partir de funções gaussianas unidimensionais, as quais sofrem uma diferenciação. Para localizar as bordas é necessário calcular o valor de intensidade para cada ponto onde aqueles de máxima intensidade e mínima intensidade são omitidos. O algoritmo Canny, baseado nos critérios de detecção e localização, encontra o limiar da borda incluindo os pixels com valores acima do limite e excluindo aqueles que estiverem abaixo [36], funcionando também como um suavizador de ruídos [22].

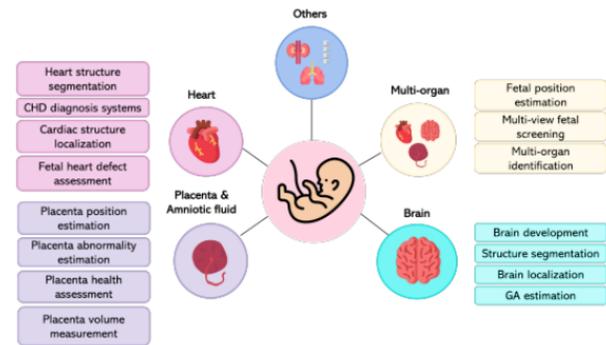


Figura 4: Visão geral dos elementos genéricos na análise do feto [21]

## 2.2 Redes neurais convolucionais

O reconhecimento de imagem é um clássico problema de classificação, e as CNN possuem um histórico de alta acurácia para resolução desse empecilho. CNN são uma classe especial de rede neural artificial, do tipo feed-forward<sup>4</sup>. Ins-

<sup>4</sup>São redes neurais onde a saída de uma camada é usada como entrada para a próxima camada. Isso significa que não há loops na rede – as informações sempre são alimentadas para a frente, nunca são enviadas de volta.

piradas em processos biológicos, as CNN vêm sendo aplicadas, principalmente, no processamento e análise inteligente de imagens digitais identificando rostos, indivíduos e outros elementos presentes em uma imagem.

As redes neurais artificiais (RNAs) são métodos de *Deep Learning* que tem como propósito simular o sistema nervoso de organismos inteligentes, treinando-os para adquirir conhecimento através da experiência mediante o reconhecimento de padrões específicos. Análogo ao elemento mais básico do cérebro humano, a unidade de processamento é chamada de neurônio artificial, capaz de emular as funções básicas do neurônio biológico como uma espécie de portal binário, composto por três elementos básicos: os sinais de entrada, a função de soma e a função de ativação.

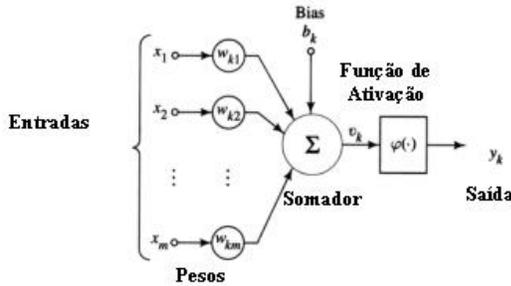


Figura 5: Modelo de Neurônio Artificial [30]

Um neurônio artificial possui um ou mais sinais de entrada, camada no qual são introduzidas informações da unidade de processamento anterior a ela. Os sinais de entrada, além de serem possíveis saídas de outros neurônios, podem ser combinações de entrada externas, *i.e.* *inputs* do programador, e do *bias* (do inglês, tendência), um parâmetro adicional na Rede Neural utilizado para ajustar a saída junto da soma ponderada das entradas para o neurônio. A cada sinal de entrada é associado um peso (*weight*), uma unidade positiva ou negativa em conformidade com as análises de treinamento, que é aplicado ao valor de entrada antes dele ser transportado para outras camadas.

A função de ativação, também denominada de camada intermediária, escondida ou oculta, é a camada onde são realizados os processamentos através da soma ponderada dos valores de entrada podendo, em modelos básicos, substituir a função de soma das entradas do neurônio. Por último, a função de saída é uma função complexa onde o resultado final é concluído e apresentado para o próximo neurônio da rede.

Como uma subclasse das RNAs, as CNNs possuem uma arquitetura semelhante, diferindo principalmente no número (maior) de camadas intermediárias e, conseqüentemente, de operações realizadas. Um modelo simples de CNN é composto de três camadas principais: camada convolucional, *Pooling* e camada totalmente conectada.

### 2.2.1 Camada Convolucional

A camada convolucional é a primeira camada responsável por extrair recursos de alto nível dos dados de entrada, mapeando seu conteúdo através de operações da matriz de uma imagem e de filtros [41].

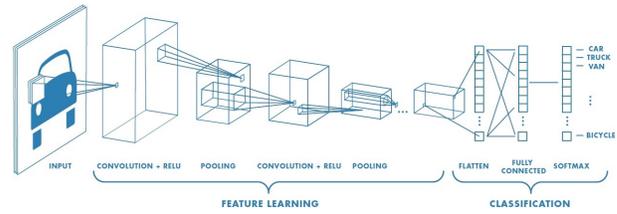


Figura 6: Arquitetura de uma CNN com muitas camadas convolucionais [41]

A convolução realizada entre esses dois dados resulta no mapa de características, também chamado de mapa de ativação dado que é um mapeamento correspondente à ativação de diferentes partes da imagem assim como é um mapeamento de um determinado tipo de característica encontrado na imagem, uma matriz condensada que pode retratar operações distintas como detecção de bordas, desfoque e nitidez por meio da aplicação de filtros [41].

### 2.2.2 Pooling

A camada *Pooling* (agrupamento) é responsável, como indicado em seu nome, por receber a saída do bloco de informações extraídas e mapeadas na camada de convolução, o mapa de características, e, caso o bloco compreenda de um tamanho espacial extenso, condensar as dimensões dos dados aplicando camadas de agrupamento independentemente em cada fração da entrada. Operações posteriores são executadas em cima desses dados mais robustos, substituindo os gerados pela camada convolucional, com o intuito de subtrair a quantidade de parâmetros a serem aprendidos, conseqüentemente reduzindo a quantidade de computação e a variância de pequenas alterações.

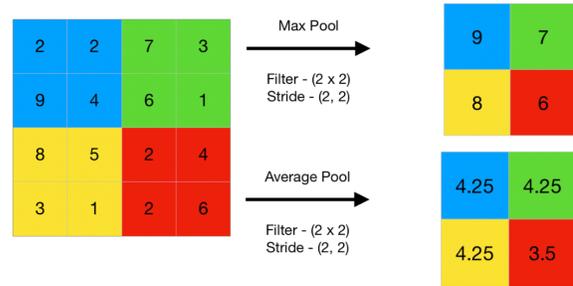


Figura 7: Operações de *Max Pooling* e *Average Pooling* [29]

Detendo de diferentes tipos, a camada de *pooling* tem subdivisões como *Max Pooling* que tem como resolução trabalhar no elemento mais predominante da região do mapa de recursos coberto pelo filtro, questionando a rede se um elemento específico pode ser encontrado em qualquer lugar de uma região da imagem, *Average Pooling*, que fornece a média de recursos presentes no mapa, e, por último e menos popular, o *Global Pooling*, que atua em cima de todos os neurônios do mapa de características.

### 2.2.3 Camada totalmente conectada

Conectando duas camadas diferentes e consistindo de pe-

tos e tendências junto com os neurônios, é nesta camada que a tarefa de classificação é realizada, onde os dados anteriormente extraídos, agora em forma de um vetor gerado a partir do achatamento do mapa de recursos, são submetidos a cálculos definidos na função de ativação. É frequentemente localizada no final de cada arquitetura CNN, onde cada neurônio está conectado a todos os neurônios da camada anterior.

### 2.2.4 Arquiteturas de CNN

A primeira aplicação bem-sucedida de uma CNN foi desenvolvida em 1998 [32], com sete camadas compostas de convoluções e totalmente conectada. Desde então as CNNs ficaram cada vez mais profundas e complexas, como Alex-Net em 2012, que, apesar de ter apenas oito camadas (cinco convoluções e três totalmente conectadas), apresenta sessenta milhões de parâmetros, e a GoogleNet com vinte e duas camadas e quatro milhões de parâmetros.

O uso de redes neurais convolucionais na detecção de estruturas fetais em imagens de ultrassom já foi proposto anteriormente onde, coadjuvante com mapas de saliência, obteve acurácia relativamente alta nas áreas da cabeça, fêmur e coluna com as classificações de 93.7%, 82.6% e 71.6% de precisão, respectivamente, conseqüentes do alto contraste entre os ossos e tecidos dessas regiões. Não obstante, outras regiões, como os braços, pernas e abdômen, tiveram como resultado um baixo índice de precisão, da qual as duas primeiras compartilham de características em comum que dificultam sua identificação: ossos compridos cercados por tecidos, e a última é decorrente de sua similar composição com o coração [31]. O estudo sinaliza que o método utilizado poderia ser aprimorado pela realização de uma abordagem hierárquica detendo de classificadores capazes de diferenciar planos anatômicos muito semelhantes.

## 2.3 Anatomia fetal

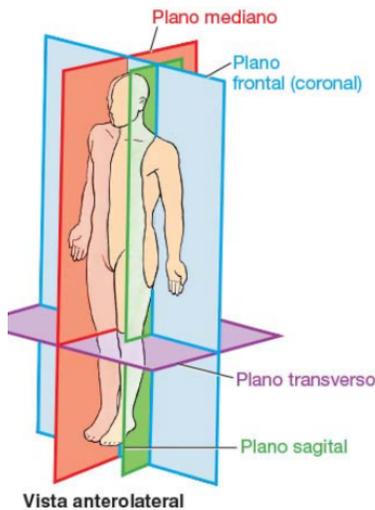


Figura 8: Planos anatômicos principais [38]

Na anatomia, planos anatômicos são planos imaginários que cruzam o corpo, criando vários cortes ou fatias de vários órgãos e estruturas, sendo que existem quatro planos principais: mediano (médio-sagital), sagital, frontal (coronal) e transversal (axial).

- Plano mediano (médio-sagital) é o plano vertical que atravessa longitudinalmente o centro do corpo, dividindo-o em metades direita e esquerda.
- Planos sagitais são planos verticais que atravessam o corpo, paralelos ao plano mediano.
- Planos frontais (coronais) são planos verticais que passam através do corpo em ângulos retos ao plano mediano, dividindo-o em partes anterior (frontal) e posterior (dorsal)
- Planos transversais ou horizontais são planos que atravessam o corpo em ângulos retos aos planos mediano e frontal. Um plano transversal divide o corpo em partes superior e inferior

Uma das formas da diagnose da progressão da saúde do feto, além do estudo do sangue materno, testes de rastreamento, amniocentese, amniocentese precoce, cordocentese, fetoscopia, entre outros, é realizada por meio da taxonomia de específicos planos anatômicos durante o diagnóstico ultra-sonográfico. Dentro dessa classificação, alguns planos possuem maior ênfase, se sobressaindo na identificação de qualquer desarmonia no desenvolvimento do feto, a figura 9 ostenta o abdômen, o fêmur, o colo materno, as quatro câmaras cardíacas e o cérebro.

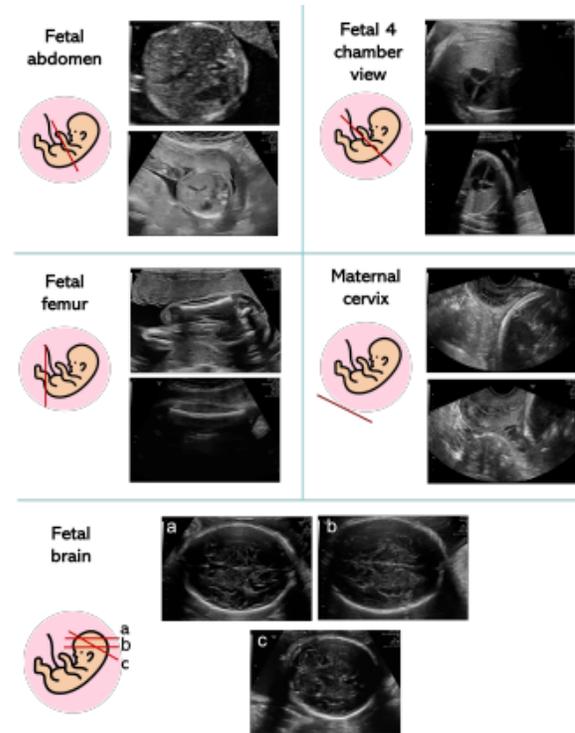


Figura 9: Amostras visuais dos planos fetais mais comuns utilizados em diagnósticos [21]

## 2.4 Tratamento de imagens digitais

Atualmente, há um foco crescente em técnicas para melhorar a qualidade e o conteúdo da informação das imagens de ultrassom e muitas dessas técnicas empregam pré-processamento digital de sinais de eco coerentes antes mesmo

da geração da imagem. No entanto, nem todos possuem acesso às instalações ou equipamentos de ultrassonografia para acompanhamento do pré-processamento, levando-os a desenvolver técnicas ou utilizar de procedimentos já existentes para aprimorar o conteúdo das imagens. Esta seção faz abordagem aos métodos pertinentes ao desenvolvimento da solução elucidando suas justificativas.

Derivado do termo "philtru" do latim, filtro é entendido como um processo de seleção, onde é detida a passagem de algo e é permitida a passagem de tudo que não é filtrado. Na área da química, a filtração é um processo físico de separação de misturas heterogêneas nos três estados físicos: sólido, líquido ou gasoso. Na computação, pode ser um programa ou seções de código destinadas a examinar solicitações distintas a partir de determinados critérios de qualificação e processá-las de acordo. No campo de imagens digitais, seu principal papel é salientar aspectos estipulados por dados de entrada e reduzir ruídos introduzidos nos processos de digitalização, compressão, quantização, transmissão, entre outros.

O processo de filtragem de imagens digitais enquadra dois tipos de técnicas, a correção e a remoção. A primeira visa realçar elementos e características desejáveis na imagem, enquanto o segundo intenta remover os desfavoráveis. Para essa finalidade, é incorporado o operador linear convolução, definido como uma operação matemática de somatório do produto entre duas funções, a matriz da imagem a ser tratada e a matriz de tratamento chamado de kernel, ao longo da região em que elas se sobrepõem, em razão do deslocamento existente entre elas.

$$(f * g)(x) = h(x) = \int_{-\infty}^{\infty} f(u) \cdot g(x - u) du$$

**Figura 10: Convolução contínua**

A fórmula contínua, representada por 10, é dada pela integral deste produto, onde as matrizes são representadas por  $f$  e  $g$ , dado que a convolução do conjunto é  $f * g$ , desde que as funções sejam integráveis no intervalo. A função resultante  $h$  sujeita-se ao valor de deslocamento correspondente a  $u$  e a variável independente  $x$ <sup>5</sup>.

$$(f * g)(k) = h(k) = \sum_{j=0}^k f(j) \cdot g(k - j)$$

**Figura 11: Convolução discreta**

Partindo para a área de computação, é utilizada a convolução discreta, visto em 11, onde  $f$  e  $g$  representam sequências de tamanho e  $k$  é o elemento do conjunto do qual a fórmula computa o resultado.

$$(f * g)(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j)g(x - i, y - j)$$

**Figura 12: Convolução em processamento de imagens**

A convolução bidimensional, dada pela expressão 12, é utilizada na área de processamento de imagens para Inteli-

<sup>5</sup>Variável que representa uma grandeza que está sendo manipulada em um experimento

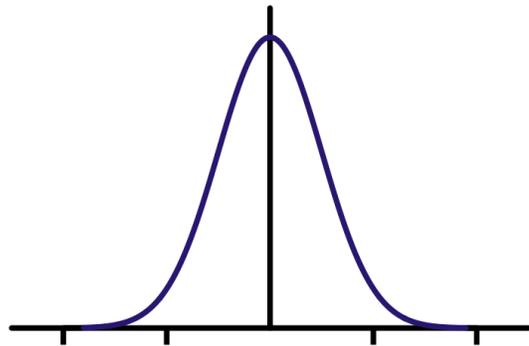
gência Artificial devido às imagens possuírem duas dimensões: altura e largura.

### 2.4.1 Filtro Gaussiano

A filtragem gaussiana é um processo de suavização ou desfoque que envolve a imagem com uma função gaussiana que pode ser caracterizada pela seguinte fórmula matemática em sua forma isotrópica:

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

Onde o coeficiente  $x$  representa a distância da origem no eixo horizontal,  $y$  é a distância da origem no eixo vertical e  $\sigma$  (sigma minúsculo) é o desvio padrão da distribuição gaussiana. A imagem abaixo ilustra um filtro gaussiano típico, no qual sua média de distribuição é igual a zero (centrada no eixo  $x = 0$ ).



**Figura 13: Representação unidimensional da distribuição Gaussiana com média zero**

### 2.4.2 Filtro Bilateral

Proposto por Tomasi e Manduchi como uma alternativa às abordagens, o filtro bilateral é um filtro de suavização não linear, que lida com a preservação de bordas e a redução de ruído nas imagens. O método combina níveis de cinza ou cores com base em sua proximidade geométrica e em sua similaridade fotométrica, preferindo valores próximos a valores distantes tanto no domínio quanto no alcance.

Além de apresentar bons resultados na filtragem, possivelmente melhores que das abordagens iterativas, recentes estudos mostram que ela pode ser utilizada na supressão de ruído gaussiano. Entretanto, possui dificuldade na remoção do ruído speckle [35] que é um tipo de ruído multiplicativo e segue a distribuição de Rayleigh.

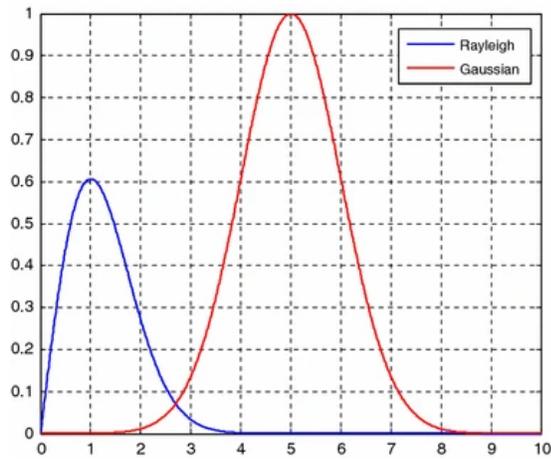


Figura 14: Tabela da distribuição gaussiana do ruído e a distribuição Rayleigh do Speckle [35]

Mencionado anteriormente, o ruído Speckle é um fenômeno de interferência que ocorre quando duas ou mais ondas que viajam para a sonda, a partir dos dispersores, interferem entre si. Por outro lado, a distribuição de Rayleigh é um termo na teoria da probabilidade utilizado para descrever o nível do sinal recebido (afetado por um canal com desvanecimento plano) em função da variação temporal. Na área de engenharia, ele é utilizado para mensuração do tempo de vida de um objeto.

### 2.4.3 Filtro de nitidez

Em inglês, *unsharp mask* (máscara de nitidez), é uma técnica originalmente implementada na câmera escura<sup>6</sup> utilizada para aguçar uma imagem, enfatizando a textura e os detalhes contidos na mesma, consistindo em uma das atividades fundamentais no pós-processamento de imagens digitais.

O funcionamento dela consiste em utilizar uma versão ligeiramente desfocada da imagem original, para isso, muitos softwares utilizam do filtro gaussiano. Uma vez com esse processo efetivado, a imagem alterada é comparada com a original, subtraindo as diferenças a partir de uma configuração especificada pelo usuário para determinar a presença de bordas, efetivamente criando um filtro passa-altas<sup>7</sup>.

<sup>6</sup> Aparelho óptico baseado no princípio de mesmo nome, consistindo de uma caixa com um orifício em uma de suas faces para permitir a entrada de luz que irá formar uma imagem invertida na superfície interna oposta.

<sup>7</sup> Filtro que permite a passagem das frequências altas com facilidade, porém atenua (ou reduz) a amplitude das frequências abaixo de frequência de corte.

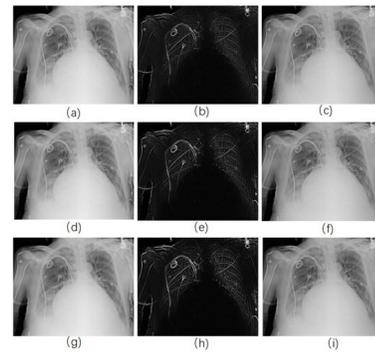


Figura 15: Realce de radiografia de tórax, onde os elementos a, d, g foram suavizadas pelo uso de kernel gaussiano, os itens b, e, h foram submetidos ao filtro de nitidez correspondentes aos três primeiros itens e os três últimos, c, f, i são as imagens aprimoradas correspondentes [18]

Geralmente, o mascaramento de nitidez requer compreensão de três configurações definidas pelo usuário: Quantidade, Raio e Limiar.

- Quantidade trata a magnitude de cada sobre-sinal, isso é, a ocorrência de um sinal ou função que excede seu alvo. Através da sua avaliação é definida a quantidade de contraste adicionado nas bordas. É geralmente medida como porcentagem.
- Raio afeta o tamanho de reforçamento das bordas, simplesmente um raio menor aumenta a largura da aresta em menor escala. Raio e quantidade interagem um com outro de modo inversamente proporcional. Como se trata de imagens, o raio é definido, por exemplo, em pixels.
- Por último, Limiar define a alteração mínima de brilho que será reforçada ou a distância que os valores tonais adjacentes devem possuir antes que o filtro aja em cima da imagem. Essa variável pode ser utilizada para afiar bordas mais pronunciadas, deixando as bordas mais sutis intocadas, consequentemente se tornando excepcionalmente eficaz para evitar amplificação de ruídos, que torna áreas lisas em trechos manchados e desfocados.

## 3. TRABALHOS RELACIONADOS

Nesta seção serão apresentados alguns trabalhos acadêmicos que tratam de temas similares ao estudo proposto, utilizam técnicas computacionais na solução dos problemas apresentados e ferramentas que podem alcançar áreas distintas de aplicabilidade. Para isso, foram abordados três sistemas desenvolvidos com diferentes técnicas e ferramentas, mas de propósitos semelhantes.

### 3.0.1 Detection of Fetal Anatomies from Ultrasound Images using a Constrained Probabilistic Boosting Tree

O trabalho *Detection of Fetal Anatomies from Ultrasound Images using a Constrained Probabilistic Boosting Tree* de Gustavo Carneiro, Bogdan Georgescu, Sara Good e Dorin Comaniciu, propõe um método novo para detecção de estruturas anatômicas em imagens de ultrassom que trouxe

diversas complicações devido a metodologia da detecção. O sistema implementa o mesmo algoritmo básico de detecção que é utilizado em larga escala em soluções antigas da qual os autores informam, em tradução livre, contar com a codificação explícita de conhecimento prévio e formulação do problema como um tarefa de agrupamento resolvida por meio de agrupamento ou variacional abordagens, mas que é diferenciado especialmente por ser treinado em imagens de um grande banco de dados de estruturas anatômicas fetais registradas por especialistas em imagens de ultrassom e por ser desenvolvido por meio da aprendizagem de árvore de decisão.

Árvore de decisão, ou *Tree Classifier* em inglês, é uma técnica de modelagem preditiva bastante utilizada em aprendizado de máquina, onde é utilizada de uma árvore de decisão como um modelo preditivo para observar uma variável e realizar conclusões sobre o valor da mesma, representadas, respectivamente, pelos ramos e folhas da árvore.

A solução foi especificamente desenvolvida em duas partes em virtude da facilitação do treinamento e dos procedimentos de teste. A primeira parte foi originada dos trabalhos sobre *marginal space learning* (MSL), uma abordagem genérica para detecção automática de estruturas anatômicas 3D em muitas modalidades de imagens médicas que reduz o número de hipóteses de teste em cerca de seis ordens de magnitude, e sobre sequential sampling, onde o autor utiliza do método de Monte Carlo para propagar sequência de distribuições de probabilidade por um período contínuo.

### 3.0.2 Plane Identification In Fetal Ultrasound Images Using Saliency Maps And Convolutional Neural Networks

Uma monografia mais recente, como o *Plane Identification In Fetal Ultrasound Images Using Saliency Maps And Convolutional Neural Networks* de 2016, combina CNN com a utilização de mapas de saliência que, na visão computacional, tem como principal funcionalidade refletir o grau de importância de um pixel para o sistema visual humano. A metodologia deste trabalho envolveu o treinamento de duas CNN para melhor aprendizagem da distinção de características em imagens ultrassom.

O trabalho utilizou de um dataset com 7568 imagens de ultrassom de fetos adquiridos de 185 fetos no *Nepean Center for Perinatal Care*, Universidade de Sydney, Austrália, e foram classificadas manualmente em 16 planos (13 estruturas anatômicas, 3 snapshots não estruturais) com assistência dos rótulos de visualização atribuídos pelo ultrassonografista. Desses 16, apenas 13 planos anatômicos foram avaliados, como pode ser visto na figura 16, onde o método de classificação empregado no trabalho é comparado com um método de última geração para classificação de imagens de planos faciais que usa do algoritmo SIFT (Scale Invariant Feature Transform) com assistência da representação vetorial de Fisher.

Anatomical Plane	Our Method	SIFT+Fisher [11]
Abdomen	21.6	0.7
Arm (including Hands)	17.9	25.5
Blood Vessel	68.9	52.6
Face	65.4	7.4
Femur	82.6	4.7
Foot	16.7	8.3
Genitals	23.8	0
Head	93.7	29.1
Heart	38.7	9.3
Kidney	61.3	3.2
Leg	10.9	0
Spine	71.6	22.7
Umbilical Cord	45.2	0

Figura 16: Resultados das avaliações dos planos anatômicos [31]

Primeiramente, foram gerados os mapas de saliência baseados nas imagens de ultrassom obtidas, onde foi-se identificado os principais pontos de interesse para que as duas CNN pudessem ser trabalhadas cada qual com sua finalidade, uma delas para mapas de saliência e outra voltadas para imagens de ultrassom. As CNNs são então usadas como extratores de recursos; as características dos dois CNNs são fundidas e usadas para classificação multiclasse.

A figura 17 ilustra este processo, no qual a caixa cinza representa a etapa de afinamento para minimizar a perda empírica. Uma outra particularidade presente na caixa cinza é a utilização da arquitetura AlexNet, uma arquitetura de rede neural convolucional, projetada por Alex Krizhevsky em colaboração com Ilya Sutskever e Geoffrey Hinton. Ela é uma CNN originalmente escrita com CUDA (*Compute Unified Device Architecture*) para rodar com suporte a GPU. Sua implementação no projeto foi devido a necessidade de contornar um problema que surge em CNN complexas, onde um elevado número de parâmetros exigem treinamento com grandes conjuntos de dados, podendo atingir a casa dos milhões.

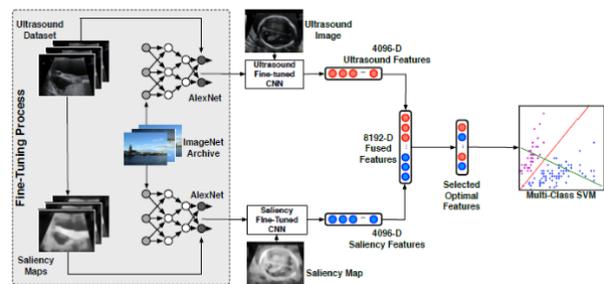


Figura 17: Visão geral da metodologia aplicada [31]

Como visto anteriormente, essa metodologia obteve maior acurácia do que a base de referência da qual foi comparada, originada do artigo Automatic Recognition of Fetal Facial Standard Plane in Ultrasound Image via Fisher Vector. O trabalho também é o primeiro na literatura a tratar com as medidas do comprimento do útero, às vezes medido como um parâmetro biométrico fetal adicional em uma ultrassonografia do segundo trimestre para identificar riscos de aneuploidia, restrição de crescimento intrauterino e displasia de

membro curto, e o comprimento da garupa da coroa (CRL), que pode detectar se a gravidez é um aborto silencioso, que contrário ao aborto espontâneo não apresenta sintomas como dor abdominal similar à da cólica e leve sangramento, em conformidade com o tamanho da CRL e da ausência de batimentos cardíacos.

### 3.0.3 *Interactive automatic fetal head measurements from ultrasound images using multimedia computer technology*

Contextualizando a história da aplicação de tecnologias computacionais no ramo da medicina, em 1997 foi submetido a versão final de uma aplicação que detectava automaticamente os contornos internos e externos do crânio de um feto em imagens de ultrassom.

Desenvolvendo em um *Sun SparcStation 20/71*, uma estação de trabalho descontinuada introduzida em março de 1994 com base na CPU *SuperSPARC* ou *hyperSPARC*, o algoritmo elaborado usou do modelo de contorno ativo, também chamado de *snakes*, um *framework* de visão computacional, capaz de delinear os contornos de objeto em imagens 2D possivelmente com ruídos, portanto que a forma aproximada do contorno seja conhecida. Essa técnica continua sendo utilizada em trabalhos atuais no processamento de imagens médicas, como nas áreas de tomografia computadorizada [39], pesquisar outras, etc.

A ferramenta *snakes* trabalha gerando curvas computadorizadas que se movem dentro de imagens para encontrar os limites do objeto, sendo definida como uma spline<sup>8</sup> que enfatiza minimizar a função de energia de uma força externa e interna. Como dito anteriormente, a ferramenta *snakes* consegue trabalhar ignorando informações de contornos ausentes, estabelecendo contornos ilusórios

## 4. SOLUÇÃO DESENVOLVIDA

### 4.1 Tecnologias utilizadas

No desenvolvimento do software foram aplicadas diversas tecnologias durante as diferentes etapas da construção do mesmo, cada qual selecionada mediante critérios que se estendem de praticidade a facilidade de uso a eficácia dos resultados. Para o pré processamento foi feito uso da ferramenta PyCharm [9] para construção de aplicações na linguagem de programação Python. O treinamento, contudo, foi realizado na plataforma Google Colab [1] com apoio do software Roboflow [14], onde as bibliotecas essenciais de processamento e análise de imagens digitais foram adotadas. Por fim, o desenvolvimento da aplicação mobile foi realizada na plataforma Visual Studio Code [37].

Durante o desenvolvimento deste trabalho foram favorecidas, quando possível, a aplicação de tecnologias de código aberto. Abaixo segue uma breve descrição de cada uma dessas tecnologias.

#### 4.1.1 PyCharm

A solução foi desenvolvida na linguagem de programação Python por intermédio da IDE multiplataforma PyCharm. Devido a praticidade, o PyCharm permite que você execute seus testes independente da estruturação, seja em um arquivo de teste, uma única classe de teste, um método ou

<sup>8</sup>Curva definida matematicamente por dois ou mais pontos de controle, chamados de nós.

todos os testes em uma pasta, além de gerar uma configuração de depuração especial em cada execução do script de teste que pode ser editada posteriormente, tornando o versátil para diversas situações. A versão da IDE utilizada foi a 2022.1, a mais recente, na edição Community.

#### 4.1.2 Git

Para versionamento das aplicações de pré processamento foi utilizada a plataforma Git, que é um projeto open-source de constante manutenção que trabalha como sistema de controle de versões distribuído, sendo usado principalmente no desenvolvimento de softwares por mérito de sua enorme flexibilidade e performance no versionamento de códigos.

#### 4.1.3 Roboflow

O treinamento do modelo foi realizado pela aplicação Roboflow, um *framework* de visão computacional para melhor coleta de dados para pré-processamento e técnicas de treinamento de modelos para redes neurais. A aplicação dispõe de *datasets* públicos, porém, devido à especificação do trabalho, não foi identificado um modelo compatível, necessitando construí-lo do início.

#### 4.1.4 Google Colab

O Colaboratory, chamado de "Colab", é um ambiente de notebooks Jupyter elaborado pelo Google Research que permite escrever e executar códigos Python como um serviço de nuvem gratuito hospedado pelo próprio Google que dispõe de acesso gratuito a GPUs e fácil compartilhamento, desprezando de pré configuração.

#### 4.1.5 Visual Studio Code

Desenvolvida pela Microsoft, o Visual Studio Code é um editor de código-fonte disponível nos sistemas operacionais Windows, Linux e macOS. O editor oferece suporte para operações de desenvolvimento como depuração, execução de tarefas e controle de versão, além de fácil de ser satisfatoriamente intuitivo devido a fácil personalização fornecida pelo mapeamento de atalhos contribuídos pela comunidade. O VS Code vem com suporte integrado para JavaScript, TypeScript e Node.js, possuindo extensões para outras linguagens como Java, Python e PHP.

### 4.2 Configuração do Ambiente

Para a preparação do ambiente de desenvolvimento foram requeridas a importação de várias bibliotecas de computação. Na seleção da configuração, do mesmo modo que no emprego das tecnologias, foram priorizados programas que possuem o código fonte disponibilizado abertamente na área de visão computacional. Essa seção descreve brevemente sobre as bibliotecas principais do projeto.

#### 4.2.1 Tensorflow

TensorFlow é uma biblioteca de código aberto para computação numérica e *machine learning* em larga escala criada pela equipe do Google Brain, inicialmente desenvolvida para uso interno em pesquisa e produção do Google, sendo disponibilizada para o público geral sob a Licença Apache 2.0 em 2015. O TensorFlow é uma estrutura altamente flexível e versátil multi-plataforma, podendo ser executado nos sistemas operacionais Windows, MacOS ou Linux, além de suportar execução em CPUs, GPUs e também TPUs (circuitos integrados específicos desenvolvidos pelo Google para acelerar o aprendizado de máquina).

### 4.2.2 Keras

É uma biblioteca poderosa e fácil de usar para desenvolver e avaliar modelos de *deep learning*, executada na plataforma de *machine learning* TensorFlow. Sua principal característica é comportar, modelar e treinar modelos de redes neurais com poucas linhas de código, dado que foi desenvolvida com foco em permitir rápida experimentação. Essa biblioteca foi utilizada somente durante o treinamento do modelo na plataforma Google Colab, inclusive participando da geração de gráficos e tabelas para análise de dados.

### 4.2.3 OpenCV

OpenCV (*Open Source Computer Vision Library*), é a principal biblioteca multiplataforma para o desenvolvimento de aplicativos na área de visão computacional, desenvolvida como uma proposta da Intel Research de melhorar aplicações de uso intensivo de processamento. A biblioteca NumPy é bastante utilizada durante o processamento das imagens em OpenCV, que converte matrizes OpenCV de e para matrizes Numpy em suas operações, estruturadas com uma sintaxe no estilo MATLAB.

### 4.2.4 Matplotlib

Biblioteca de software para criação de gráficos e visualizações de dados de modo estático, animado e interativo. Ela é uma alternativa viável de código aberto e plataforma cruzada ao MATLAB (Matrix Laboratory), uma plataforma de programação licenciada projetada para engenheiros e cientistas de dados. Fornece, a partir de kits de ferramentas de GUI (Graphical user interface) de uso geral, como Tkinter, wxPython, Qt ou GTK, uma API orientada a objetos para plotagem gráfica.

### 4.2.5 YOLOv5

YOLOv5, anagrama de *You only look once* (do inglês, "Você só olha uma vez") é uma família de arquiteturas e modelos de detecção de objetos pré-treinados no conjunto de dados COCO. É um algoritmo de código aberto oriundo da pesquisa do Ultralytics sobre aplicações futuras de tecnologias de visão artificial, considerado mais veloz do que o YOLO4.

## 4.3 Metodologia

A solução foi desenvolvida com o apoio de *dataset* de imagens 2D, nos formatos PNG (Portable Network Graphics), de ultrassonografia de fetos disponibilizados online, contabilizando, ao todo, 1334 imagens.

Nas fases preliminares, o *dataset* foi dividido e foram utilizadas 999 imagens para treinamento, cada qual com tamanho de 800 por 540 pixels com um tamanho de pixel variando de 0,052 a 0,326 mm. É observado que, dentro desse *dataset*, algumas imagens possuem aparência similares devido a serem produzidas durante o mesmo exame ectoscópico, com um número de participantes não divulgado, dado que esses dados foram tratados de forma anônima de acordo com os princípios da Declaração de Helsinque<sup>9</sup>. Esse *dataset* faz parte de um desafio denominado HC18 [45] desenvolvido com objetivo de projetar um algoritmo para medir automaticamente a circunferência da cabeça fetal (*Head Circumference* ou HC) dada uma imagem de ultrassom 2D, medida que também possibilita a estimativa da idade gestacional.

<sup>9</sup>Declaração de princípios éticos para fornecer orientações aos médicos e outros participantes em pesquisas clínicas envolvendo seres humanos.

rence ou HC) dada uma imagem de ultrassom 2D, medida que também possibilita a estimativa da idade gestacional.

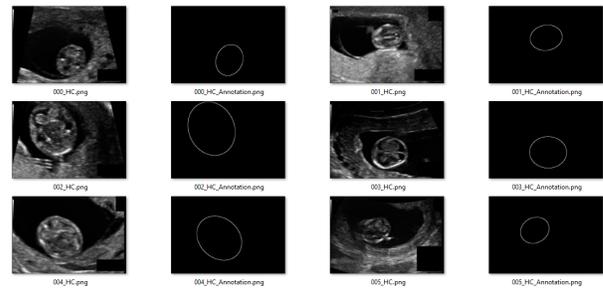


Figura 18: Dataset do desafio de medição automática da circunferência da cabeça fetal

Acompanhados neste dataset se encontram anotações manuais para cada HC realizadas por um ultrassonografista, também disponíveis em imagens PNG, e dois arquivos em formato CSV (Comma Separated Values, em português, Valores Separados por Vírgula) contendo o tamanho da circunferência cefálica e sua representação de tamanho de pixel, ambas em milímetro, correspondentes a cada imagem da coleção de teste.

Dado a obtenção do *dataset*, o projeto deslocou-se para o desenvolvimento da rede neural convolucional, começando com o pré processamento de dados das imagens a partir de uma simples aplicação composta do filtro Gaussiano e dos operadores morfológicos (filtros bilateral e de nitidez) a fim de aprimorar a qualidade das imagens. Os filtros foram utilizados apenas nas imagens encontradas na pasta de treinamento, nomeada *training\_set*, composta de, ao total, 1998 itens, da qual metade são imagens de ultrassom.

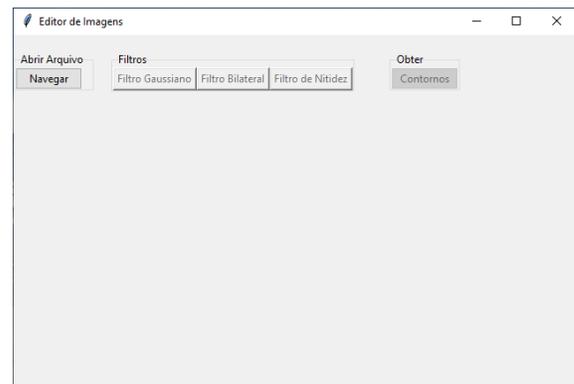


Figura 19: Software de aplicação dos filtros

Para o cálculo dos filtros foi utilizada a biblioteca OpenCV do Python, uma biblioteca *open-source* (portanto que obedeça a licença BSD<sup>10</sup>), em razão da facilidade de acesso a algoritmos *state of the art*, onde as imagens são convertidas em arrays multidimensionais, simplificando sua manipulação. Dentro dos algoritmos disponibilizados pela biblioteca, três foram relevantes ao pré processamento: *GaussianBlur()*, *bilateralFilter()* e *filter2D()*.

<sup>10</sup>Licença inicialmente utilizada nos sistemas operacionais do tipo Berkeley Software Distribution que demanda o reconhecimento dos autores do código base.

*gaussianBlur()*: função que desfoca uma imagem usando um filtro gaussiano. O filtro gaussiano remove ruído estatístico cuja função densidade de probabilidade (FDP) é igual a da distribuição normal. Neste algoritmo, em vez de uma *box filter*, um filtro passa-baixa que suaviza a imagem fazendo com que cada pixel de saída seja a média dos vizinhos, é utilizado do kernel gaussiano fornecido pela própria função.

*bilateralFilter()*: função que aplica um filtro bilateral não iterativo para suavização com preservação de arestas como descrito em CITAÇÃO. O algoritmo reduz os ruídos indesejados de forma eficiente, cuidando para não remover as bordas, porém é muito lento em comparação com a maioria dos outros filtros.

*filter2D()*: função que aplica um filtro linear arbitrário a uma imagem a partir da utilização de um kernel (geralmente uma matriz 2d), da qual a convolução, a nível de pixel, produz uma imagem de saída. Neste trabalho ela foi responsável por atuar como um filtro de nitidez, podendo inclusive, gerar filtros de suavização, desfoque e detecção de borda em uma imagem.

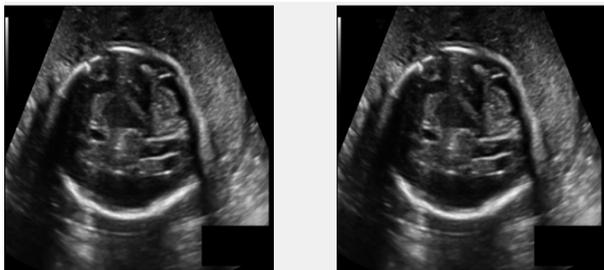


Figura 20: Comparação da imagem antes e após aplicação do filtro Gaussiano

A anotação de dados, a rotulagem ativa do conjuntos de dados de treinamento de modelo de aprendizado de máquina, foi realizada por uma aplicação construída na IDE PyCharm a partir das anotações manuais já existentes na pasta, responsáveis por destacar as regiões de interesse a serem detectadas pela CNN. A figura 21 ilustra a configuração das imagens com suas anotações correspondentes.

A aplicação teve como dado de entrada apenas a imagem anotada manualmente no *dataset*, empregando a função *findContours()* da biblioteca OpenCV para detecção dos pontos da linha de contorno. Esse algoritmo recebe três argumentos: a imagem de origem, o modo de recuperação de contorno e o método de aproximação de contorno, definidos na aplicação como, respectivamente, a imagem anotada, o parâmetro *cv2.RETR\_EXTERNAL*, método que recupera apenas os contornos externos extremos e *cv2.CHAIN\_APPROX\_SIMPLE*, método que devolve apenas os pontos finais que são necessários para desenhar a linha de contorno.

Ademais, foi utilizada a aplicação Roboflow no pré-processamento, redimensionando as imagens do dataset para o tamanho 416x416, considerada uma boa prática dado que é um dos padrões de tamanho de imagem com qual o YOLOv5 opera favoravelmente. A aplicação Roboflow inclusive auxiliou no preparo das anotações, por ser uma ferramenta que simplifica o processo de preparação e treinamento de dados, permitindo converter as anotações em formatos universais. Para o desenvolvimento dessa rede neural foi utilizada a anotação no formato *YOLOv5 PyTorch*, um formato de anota-

ção semelhante ao *YOLO Darknet TXT*, mas com a adição de um arquivo YAML (acrônimo recursivo de *YAML Ain't Markup Language*, em português, *YAML não é linguagem de marcação*) contendo a configuração do modelo e os valores de classe suportada pelo algoritmo YOLOv5.

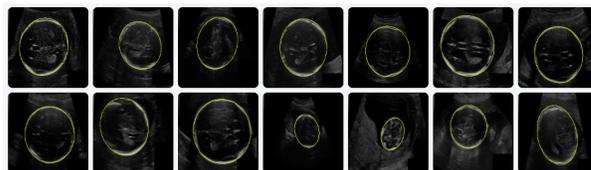


Figura 21: Imagens de ultrassom com o ponto de interesse destacado, geradas a partir do roboflow

O próximo passo, o treinamento próprio da CNN, foi realizado na plataforma Google Colab que, durante o período de desenvolvimento, dispôs da seguinte configuração de sistema: CPU Intel(R) Xeon(R) @ 2.20GHz, GPU Tesla T4 e memória GDDR6.



Figura 22: Importação do dataset no ambiente de treinamento

Uma vez com o *dataset* importado localmente, processo visto na figura 22, a rede foi treinada com diversas épocas (epochs) variando entre 100 e 400 buscando melhor otimizar o treinamento. O treinamento se restringiu a valores relativamente baixos de épocas quanto a quantidade de imagens consumidas devido a versão gratuita do Google Colab funcionar com um limite de uso dinâmico do GPU, onde os valores exatos não são documentados, em parte porque eles podem variar rapidamente.

```
python train.py --img 416 --batch 200 --epochs 400 --data (dataset.location)/data.yaml --weights yolov5s.pt --cache
```

Abaixo são descritos os principais parâmetros utilizados no treinamento executado pela linha de código acima, que aciona o início da aprendizagem:

- **batch**: batch, ou lote, define o número de amostras que serão propagadas pela rede.
- **data**: localização do conjunto de dados salvos.
- **weights**: especifica o caminho dos pesos para começar a transferir o aprendizado.
- **cache**: imagens de cache para treinamento mais rápido

A rede foi treinada inicialmente utilizando 100 *batches*, reduzindo para apenas 16 *batches* nos testes finais, uma quantidade baixa considerando o total de imagens. Isso ocorreu em consequência da memória limitada disponibilizada pelo

Google Colab. Com o treinamento da rede utilizando de menos amostras, o procedimento geral de treinamento requer menos memória, permitindo flexibilidade na quantidade de testes.

#### 4.4 Rede neural convolucional

Os testes foram divididos em dois segmentos: testes dos *softwares* de pré processamento (concentrando a aplicação dos filtros e obtenção dos contornos) e testes produzidos durante o treinamento do modelo da CNN. A figura 23 ilustra a aplicação dos filtros em uma das imagens de treinamento.



Figura 23: Aplicação de filtros no pré processamento. Da esquerda a direita: imagem original, imagem com filtro gaussiano, imagem com filtro bilateral e imagem com filtro de nitidez

O treinamento utilizou somente as imagens localizadas no diretório *training\_set* do desafio HC18, em consequência das imagens encontradas no diretório *test\_set* não possuem *ground truth*, a imagem de referência fornecida por observação direta e evidência empírica que representa o resultado ideal da segmentação, totalizando 998 itens. Esse conjunto de imagens foi então dividido no seguinte fracionamento: 70% imagens para treinamento, 20% para validação e 10% para realização de testes. Na figura 24 podem ser visto os resultados gerados automaticamente pela biblioteca tensorboard com a configuração de 100 épocas e 16 lotes.

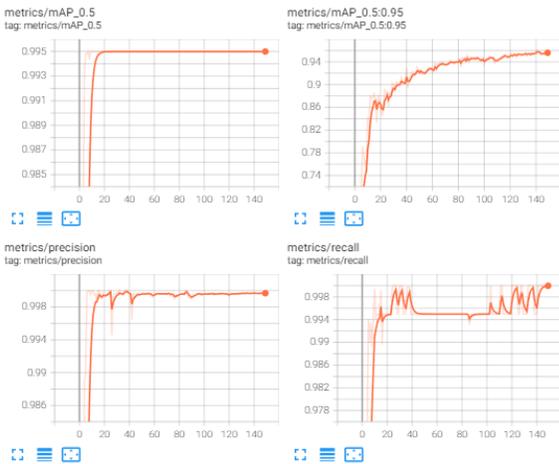


Figura 24: Métricas do modelo treinado com 100 épocas

Notou-se uma desvantagem da utilização do baixo número de *batches* em relação a precisão da estimativa do gradiente. Na figura 25 pode ser visto que a direção do gradiente de 16 *batches* flutua muito mais em comparação com a direção do gradiente de 100 *batches*.

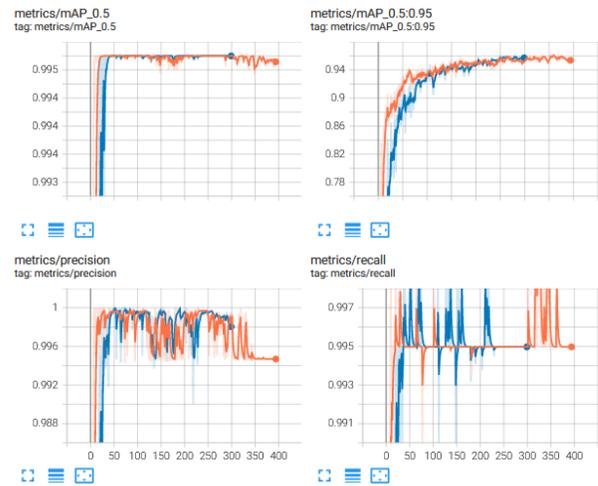


Figura 25: Métricas de modelos treinados com 300 épocas, onde o de linha laranja utilizou 16 lotes enquanto o azul usou 100 lotes.

Além do limite de GPU mencionado anteriormente, o treinamento teve interrupção prematura em 496 épocas, pois nenhuma melhora foi observada nas últimas 100 épocas, atingindo os melhores resultados observados na época 395, como percebido na figura 26.

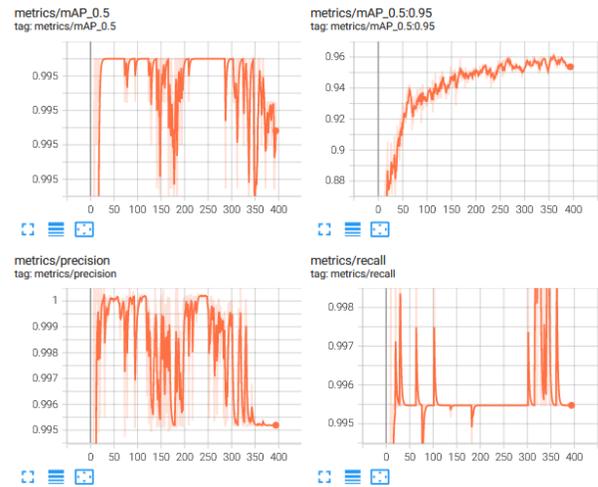


Figura 26: Métricas do modelo treinado com 395 épocas

## 5. APLICATIVO DE DETECÇÃO

Com o treinamento da rede neural realizado, foi obtido um arquivo *.PT*, um arquivo gerado pela biblioteca *PyTorch* que representa um modelo de aprendizado de máquina, contendo algoritmos capazes de executar tarefas automaticamente, como identificação de imagem e processamento de linguagem natural.

Para manipulação desse modelo foi desenvolvida uma aplicação mobile utilizando do *framework Flask* como servidor *back-end* para efetuar a detecção do perímetro cefálico em imagens de ultrassom. A aplicação em *Flask* foi então responsável por receber imagens nos formatos PNG e JPEG

(*Joint Photographics Experts Group*), inferir os pontos de interesse na imagem a partir do modelo treinado e devolver a mesma e os dados pertinentes à avaliação, como medidas aproximadas do perímetro e dados da precisão da detecção.

A seguir, foi desenvolvida uma aplicação em React Native para consumir o *back-end*, cuja responsabilidade se manteve ao envio das imagens de ultrassom, figura 27 ilustra um segmento da aplicação.

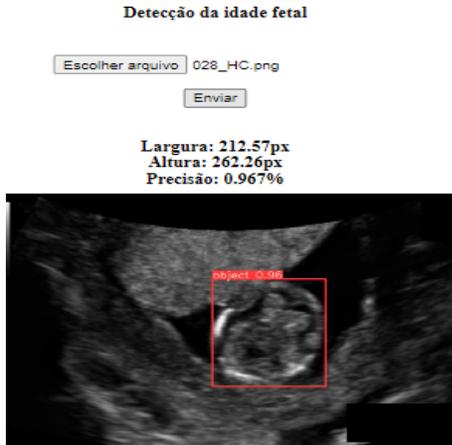


Figura 27: Detecção do perímetro cefálico na aplicação desenvolvida em React Native

## 6. CONCLUSÃO

Como observado anteriormente, a ultrassonografia tem um papel importante no acompanhamento do desenvolvimento do feto, tanto para a saúde do mesmo quanto da gestante. O emprego de técnicas computacionais para o tratamento de imagem busca auxiliar o processo a fim de evitar que o carácter subjetivo da avaliação comprometa o procedimento.

Com a expansão da inteligência artificial e aprendizagem de máquina na área de medicina, além da elaboração de projetos abertos ao público, como o HC18, surgem gradativamente pesquisas mais extensas na área e técnicas otimizadas para melhor obtenção de resultados. Esta monografia apresenta uma alternativa a soluções mais custosas, tanto como pela perspectiva financeira quanto pelo ângulo de processamento, empregando o YOLOv5 para construção de uma arquitetura de rede personalizada.

Foi observado que a baixa quantidade de imagens, devido em parte à ausência das *ground truth* correspondentes a imagens de validação e por consistir exclusivamente de imagens do perímetro cefálico fetal, do *dataset* também influenciou no número de lotes utilizado, contudo o limite da alocação de GPU por usuário pelo ambiente impediu a realização de testes com valores mais elevados.

Para mitigação de determinados problemas encontrados poderiam ser aplicados cálculos de pós processamento nas imagens, a fim de aprimorar a qualidade para usos futuros. Do mesmo modo, o arquivo *.PT* gerado no final do algoritmo funciona como um modelo pré-treinado, podendo ser utilizado em futuras aprendizagens.

## 7. TRABALHOS FUTUROS

Neste trabalho foi utilizada de uma configuração base de pré processamento do próprio Roboflow, desconsiderando ferramentas de augmentação, uma opção para otimizar a detecção. Também foi evitado redimensionar as imagens do dataset para tamanhos fora do padrão do YOLOv5, o que poderia gerar resultados imprevistos, sejam eles negativos ou não, tornando a uma possível via para otimização do treinamento.

Devido a própria natureza da arquitetura de redes neurais, que é um algoritmo faminto por dados, exigindo numerosos exemplos para tomar decisões informadas, o modelo treinado pode ser trabalhado para obter melhor acurácia empregando de metodologias como aprendizado ativo, uma estratégia que identifica proativamente subconjuntos de dados de treinamento para promover o desempenho.

Em trabalhos futuros, pode ser trabalhada a otimização do código para aproximação da execução em tempo real, além de efetuar testes na etapa de treinamento para discernição da otimização derivada da aplicação dos filtros de imagem no pré processamento, comparando, inclusive, segmentações com pré processamento e sem processamento.

## 8. NOMENCLATURA

CNN: convolutional neural network  
 CRL: crown rump length  
 CT: computed tomography  
 DL: deep learning  
 RNA: rede neural artificial  
 US: ultrassom  
 GPU: graphics processing unit  
 CPU: central processing unit  
 MRI: magnetic resonance imaging  
 MSL: marginal space learning  
 SIFT: scale invariant feature transform  
 CUDA: compute unified device architecture:  
 JPEG: joint photographic experts group  
 GUI: graphical user interface

## 9. REFERÊNCIAS

- [1] Colaboratory - frequently asked questions.
- [2] Sharpening: Unsharp mask. cambridge in colour.
- [3] Transvaginal ultrasound.
- [4] Pdq screening and prevention editorial board. ovarian, fallopian tube, and primary peritoneal cancer screening (pdq®): Patient version., 2002.
- [5] Mar 2013.
- [6] Capítulo 9 - a arquitetura das redes neurais, Mar 2018.
- [7] Exames clínica de ultrassom dr. marcelo duarte pelotas, 2018.
- [8] O que É translucência nucal? entenda para que serve o exame, Sep 2019.
- [9] Quick documentation, Mar 2019.
- [10] Ultrassom: conheça os principais tipos e saiba por que esse exame é tão importante, Ago 2019.
- [11] Rede neural convolucional, Dec 2020.
- [12] Ultrassonografia obstétrica de 1º trimestre – imfep - instituto da mulher e de medicina fetal em curitiba, Jul 2021.
- [13] Exames de imagem: 8 procedimentos e para que servem, Mar 2022.
- [14] Roboflow - overview, Apr 2022.

- [15] What are the implications of a short fetal humerus?nbsp;, Feb 2022.
- [16] G. Alves. Entendendo redes convolucionais (cnns), Dec 2018.
- [17] R. Bracewell. *The Fourier Transform and Its Application*, volume 34. 01 2000.
- [18] S. Chen and Y. Cai. Enhancement of chest radiograph in emergency intensive care unit by means of reverse anisotropic diffusion-based unsharp masking model, Apr 2019.
- [19] M. Cristina Chammas. Entenda o que É e quais são os tipos de ultrassom.
- [20] B. I. Edwards, N. H. O. Khougali, and A. D. Cheok. Trends in computer-aided diagnosis using deep learning techniques: A review of recent studies on algorithm development.
- [21] M. C. Fiorentino, F. P. Villani, M. D. Cosmo, E. Frontoni, and S. Moccia. A review on deep-learning algorithms for fetal ultrasound-image analysis, Jan 2022.
- [22] M. Giovane, V. Do, P. Aluir, and D. Poz. O processo de detecção de bordas de canny: Fundamentos, algoritmos e avaliação experimental. 12 2020.
- [23] R. Gonzalez and R. Woods. Processamento digital de imagens. *Pearson Educación*, 12 2009.
- [24] R. Gonzalez and R. Woods. Tratamiento digital de imagenes. *SERBIULA (sistema Librum 2.0)*, 12 2020.
- [25] P. Hiremath, P. T. Akkasaligar, and S. Badiger. Speckle noise reduction in medical ultrasound images. 2013.
- [26] J. Jang, P. Yejin, B. Kim, S. M. Lee, J.-Y. Kwon, and J. Seo. Automatic estimation of fetal abdominal circumference from ultrasound images. *IEEE Journal of Biomedical and Health Informatics*, PP:1–1, 11 2017.
- [27] H. Johnson, S. Cousens, J. Perin, S. Scott, J. Lawn, I. Rudan, H. Campbell, R. Cibulskis, M. Li, and R. Black. Global, regional, and national causes of child mortality: An updated systematic analysis for 2010 with time trends since 2000. *Lancet*, 379:2151–61, 05 2012.
- [28] W. Khan. Image segmentation techniques: A survey. *Journal of Image and Graphics*, 1:166–170, 01 2014.
- [29] S. Khosla. Cnn: Introduction to pooling layer, Jul 2021.
- [30] D. Klerfors, Nov 1998.
- [31] A. Kumar, P. Sridar, A. Quinton, R. Krishna kumar, D. D. F. Feng, R. Nanan, and J. Kim. Plane identification in fetal ultrasound images using saliency maps and convolutional neural networks. pages 791–794, 04 2016.
- [32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition, Nov 1998.
- [33] P. H. Lento and S. Primack. Advances and utility of diagnostic ultrasound in musculoskeletal medicine, Mar 2008.
- [34] T. Lewis. What is a medical ultrasound?, Oct 2014.
- [35] H. Li, J. Wu, A. Miao, P. Yu, J. Chen, and Y. Zhang. Rayleigh-maximum-likelihood bilateral filter for ultrasound image enhancement - biomedical engineering online, Apr 2017.
- [36] R. Maia, H. Wagner, and A. Von Wangenheim. Uma ferramenta de visualização e manipulação de imagens radiológicas no padrão dicom 3.0. 01 2003.
- [37] Microsoft. Documentation for visual studio code, Nov 2021.
- [38] R. Paresque. Introdução ao estudo da anatomia.
- [39] S. D. Pathak, V. Chalana, and Y. Kim. Interactive automatic fetal head measurements from ultrasound images using multimedia computer technology. *Ultrasound in Medicine Biology*, 23(5):665–673, 1997.
- [40] Z. M. Research. Outlook on the global deep learning market size, share amp; growth 2022 - 2028: Estimated to achieve a revenue of \$80769.6 million with growing at a cagr 38.3% report by zion market research, May 2022.
- [41] S. Saha. A comprehensive guide to convolutional neural networks - the eli5 way, Dec 2018.
- [42] Z. Sobhaninia, S. Rafei, A. Emami, N. Karimi, K. Najarian, S. Samavi, and S. Soroushmehr. Fetal ultrasound image segmentation for measuring biometric parameters using multi-task deep learning. volume 2019, pages 6545–6548, 07 2019.
- [43] M. Tan, R. Pang, and Q. Le. Efficientdet: Scalable and efficient object detection. pages 10778–10787, 06 2020.
- [44] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou. Fixing the train-test resolution discrepancy: Fixefficientnet, 2020.
- [45] T. L. A. van den Heuvel, D. d. Bruijn, C. L. d. Korte, and B. v. Ginneken. Automated measurement of fetal head circumference using 2d ultrasound images, 2018.
- [46] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Scaled-YOLOv4: Scaling cross stage partial network. *arXiv preprint arXiv:2011.08036*, 2020.
- [47] H.-Q. Wu, X. Zhang, X. Geng, J. Dong, and G. Zhou. Computer aided quantification for retinal lesions in patients with moderate and severe non-proliferative diabetic retinopathy: A retrospective cohort study. *BMC ophthalmology*, 14:126, 10 2014.

## 10. ANEXOS

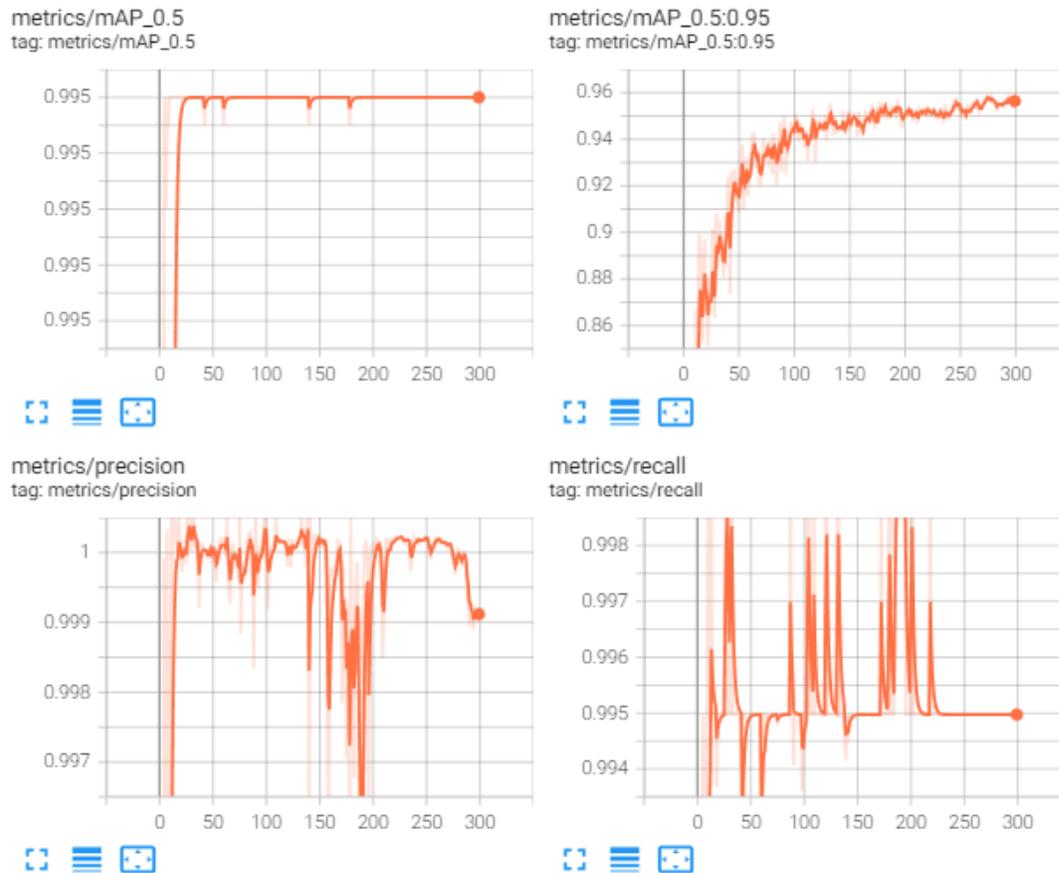
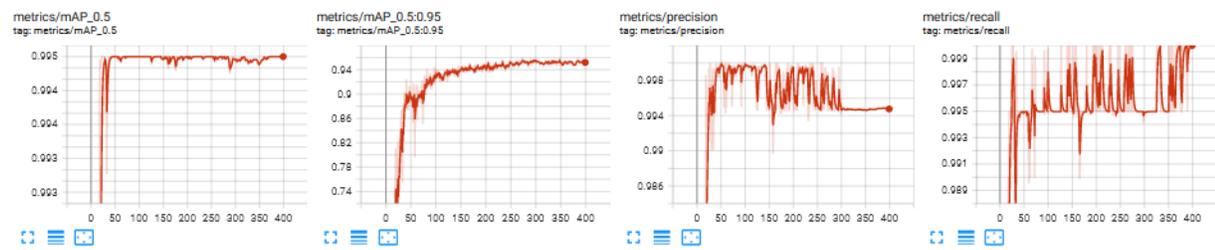


Figura 28: Métricas do modelo treinado com 300 épocas e 16 lotes geradas a partir do pacote tensorboard



train

3 ^

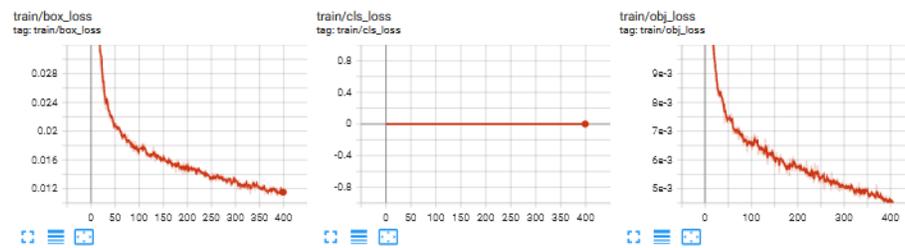


Figura 29: Métricas do modelo treinado com 400 épocas e 100 lotes geradas a partir do pacote tensorboard

